PremiereLeagueAnalysis

June 26, 2023

1 Premiere League Analysis

In the following project, I'm doing an Exploratory Data Analysis to the 2022/2023 Premier League season. I have the data about all of the teams and their games (Opponent, Date, Result, Formation, Competition, Referee)

The goal of this project is to answer the following questions 1. Who has the most wins ? 2. Who has the most losses ? 3. Who has the most draws ? 4. Information about the formations (Most wins, losses, draws) 5. The championship course along the games

1.1 Data Import and Cleaning

Importing the modules

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings('ignore')
```

Importing the data

[2]:
```
import os

path = os.getcwd()
csv_files = []

for root, dirs, files in os.walk(path):
    for file in files:

        # Only getting the files with csv extension
        if(file.endswith(".csv")):
            csv_files.append(file)

csv_files
```

[2]:
```
['Brentford-Stats.csv',
 'Chelsea-Stats.csv',
```

```
 'Everton-Stats.csv',
 'Liverpool-Stats.csv',
 'Aston-Villa-Stats.csv',
 'Brighton-and-Hove-Albion-Stats.csv',
 'Nottingham-Forest-Stats.csv',
 'Tottenham-Hotspur-Stats.csv',
 'Crystal-Palace-Stats.csv',
 'Manchester-United-Stats.csv',
 'Fulham-Stats.csv',
 'Wolverhampton-Wanderers-Stats.csv',
 'Newcastle-United-Stats.csv',
 'Southampton-Stats.csv',
 'West-Ham-United-Stats.csv',
 'Leicester-City-Stats.csv',
 'Leeds-United-Stats.csv',
 'Manchester-City-Stats.csv',
 'Bournemouth-Stats.csv',
 'Arsenal-Stats.csv']
```

Creating a dataframe for each team

[3]:
```
dfs = []

for csv_file in csv_files:
    df = pd.read_csv(csv_file)
    dfs.append(df[df['Date'] != "Date"])
```

We're going to print an example of the dataframes

[4]:
```
dfs[2].head()
```

[4]:

	Date	Venue	Result	Formation	Comp	Opponent	\
0	2022-08-06	Home	L	5-4-1	Premier League	Chelsea	
1	2022-08-13	Away	L	3-4-3	Premier League	Aston Villa	
2	2022-08-20	Home	D	3-4-3	Premier League	Nott'ham Forest	
3	2022-08-23	Away	W	5-3-2	EFL Cup	Fleetwood Town	
4	2022-08-27	Away	D	3-4-3	Premier League	Brentford	

	Referee
0	Craig Pawson
1	Michael Oliver
2	Andre Marriner
3	Tom Reeves
4	John Brooks

Now, we're going to show the head for the first 3 teams

[5]:
```
for i in range(3):
    print(dfs[i].head(3))
```

```
         Date Venue Result Formation            Comp        Opponent  \
0  2022-08-07  Away      D     4-3-3  Premier League  Leicester City
1  2022-08-13  Home      W     5-3-2  Premier League  Manchester Utd
2  2022-08-20  Away      L     4-3-3  Premier League          Fulham

          Referee
0  Jarred Gillett
1  Stuart Attwell
2    Peter Bankes
         Date Venue Result Formation            Comp      Opponent  \
0  2022-08-06  Away      W     3-4-3  Premier League       Everton
1  2022-08-14  Home      D     3-5-2  Premier League     Tottenham
2  2022-08-21  Away      L     3-5-2  Premier League  Leeds United

          Referee
0    Craig Pawson
1  Anthony Taylor
2  Stuart Attwell
         Date Venue Result Formation            Comp         Opponent  \
0  2022-08-06  Home      L     5-4-1  Premier League          Chelsea
1  2022-08-13  Away      L     3-4-3  Premier League      Aston Villa
2  2022-08-20  Home      D     3-4-3  Premier League  Nott'ham Forest

          Referee
0    Craig Pawson
1  Michael Oliver
2  Andre Marriner
```

1.2 Exploratory Data Analysis

The teams with the most wins

[6]:
```
# Let's take the first team as an example

team = dfs[0]

team.head()
```

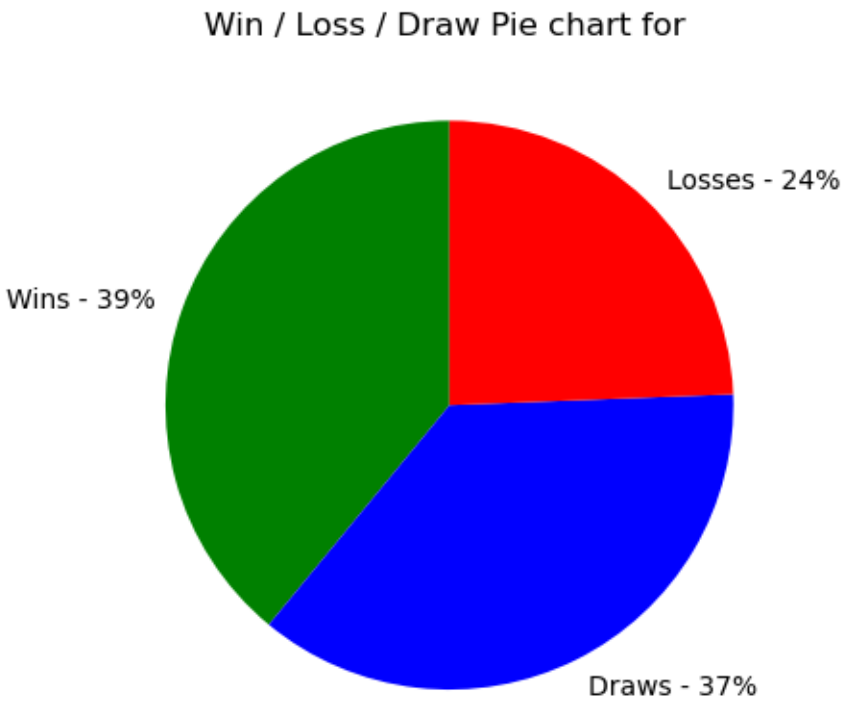
[6]:
```
         Date Venue Result Formation            Comp        Opponent  \
0  2022-08-07  Away      D     4-3-3  Premier League  Leicester City
1  2022-08-13  Home      W     5-3-2  Premier League  Manchester Utd
2  2022-08-20  Away      L     4-3-3  Premier League          Fulham
3  2022-08-23  Away      W   4-2-3-1         EFL Cup  Colchester Utd
4  2022-08-27  Home      D     4-3-3  Premier League         Everton

           Referee
0   Jarred Gillett
1   Stuart Attwell
```

```
2     Peter Bankes
3  James Linington
4      John Brooks
```

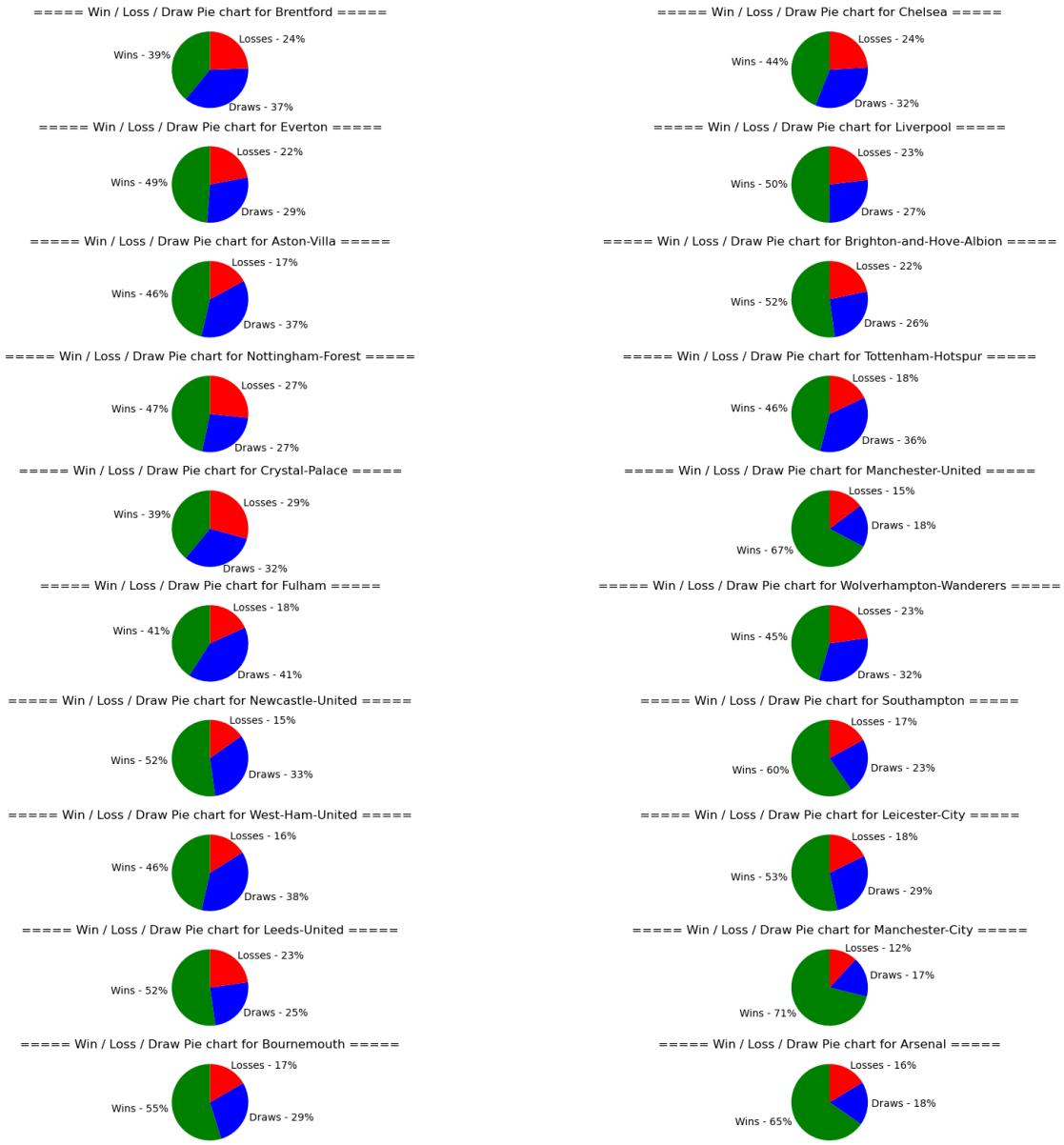
Let's go on by calculating the number of wins, losses and draws

[7]:
```
win_loss_draw_percentage = team['Result'].value_counts('W') * 100

labels = ["Wins",'Draws','Losses']

labels_with_percentage = [f'{label} - {round(percentage)}%' for␣
 ↪label,percentage in zip(labels,win_loss_draw_percentage)]

colors = ['Green','Blue','Red']

plt.
 ↪pie(win_loss_draw_percentage,labels=labels_with_percentage,colors=colors,startangle=90)
plt.title("Win / Loss / Draw Pie chart for ")
```

[7]:
```
Text(0.5, 1.0, 'Win / Loss / Draw Pie chart for ')
```



[8]:
```
# We're getting the names of the teams
part_to_delete = "-Stats.csv"

# Creating subplots
teams_numbers = len(csv_files)

fig, axes = plt.subplots(nrows = round(teams_numbers / 2),␣
 ↪ncols=2,figsize=(20,20))
for index,csv_file in enumerate(csv_files):
    df = pd.read_csv(csv_file)
    df = df[df['Date'] != "Date"]

    team = csv_file.replace(part_to_delete, "")

    win_loss_draw_percentage = df['Result'].value_counts('W') * 100

    labels = ["Wins",'Draws','Losses']

    labels_with_percentage = [f'{label} - {round(percentage)}%' for␣
 ↪label,percentage in zip(labels,win_loss_draw_percentage)]

    colors = ['Green','Blue','Red']

    row_index = index // 2
    col_index = index % 2

    axes[row_index, col_index].set_aspect('equal')  # Ensures a circular shape
    axes[row_index,col_index].
 ↪pie(win_loss_draw_percentage,labels=labels_with_percentage,colors=colors,startangle=90)
    axes[row_index,col_index].set_title(f"===== Win / Loss / Draw Pie chart for␣
 ↪{team} =====")
```



1.3 Top 5 teams in wins, losses and draws

1.3.1 Most Wins

[9]:
```
# Creating the wins, losses, draws dataframe
part_to_delete = "-Stats.csv"

games_df = pd.DataFrame(columns=['Team','Win Count','Draw Count','Loss Count'])

for index,csv_file in enumerate(csv_files):
    df = pd.read_csv(csv_file)
```

```
    df = df[df['Date'] != "Date"]

    team = csv_file.replace(part_to_delete, "")

    win_loss_draw_percentage = df[df['Comp'] == 'Premier League']['Result'].
 ↪value_counts()

    new_row = {'Team':team,
               'Win Count':win_loss_draw_percentage['W']
               ,'Draw Count':win_loss_draw_percentage['D'],
               'Loss Count':win_loss_draw_percentage['L']}

    games_df =  games_df.append(new_row,ignore_index=True)

games_df
```

[9]:

	Team	Win Count	Draw Count	Loss Count
0	Brentford	15	14	9
1	Chelsea	11	11	16
2	Everton	8	12	18
3	Liverpool	19	10	9
4	Aston-Villa	18	7	13
5	Brighton-and-Hove-Albion	18	8	12
6	Nottingham-Forest	9	11	18
7	Tottenham-Hotspur	18	6	14
8	Crystal-Palace	11	12	15
9	Manchester-United	23	6	9
10	Fulham	15	7	16
11	Wolverhampton-Wanderers	11	8	19
12	Newcastle-United	19	14	5
13	Southampton	6	7	25
14	West-Ham-United	11	7	20
15	Leicester-City	9	7	22
16	Leeds-United	7	10	21
17	Manchester-City	28	5	5
18	Bournemouth	11	6	21
19	Arsenal	26	6	6

1.3.2 Plot chart for the most wins in the Premier League and EFL

[10]:
```
# Sorting the dataframe according to wins

most_wins_df = games_df.sort_values(by='Win Count', ascending=False)
most_losses_df = games_df.sort_values(by='Loss Count', ascending=False)
most_draws_df = games_df.sort_values(by='Draw Count', ascending=False)
```

[11]:
```
most_wins_df
```

[11]:

	Team	Win Count	Draw Count	Loss Count
17	Manchester-City	28	5	5
19	Arsenal	26	6	6
9	Manchester-United	23	6	9
3	Liverpool	19	10	9
12	Newcastle-United	19	14	5
4	Aston-Villa	18	7	13
5	Brighton-and-Hove-Albion	18	8	12
7	Tottenham-Hotspur	18	6	14
0	Brentford	15	14	9
10	Fulham	15	7	16
8	Crystal-Palace	11	12	15
1	Chelsea	11	11	16
11	Wolverhampton-Wanderers	11	8	19
14	West-Ham-United	11	7	20
18	Bournemouth	11	6	21
6	Nottingham-Forest	9	11	18
15	Leicester-City	9	7	22
2	Everton	8	12	18
16	Leeds-United	7	10	21
13	Southampton	6	7	25

[12]:
```
most_draws_df
```

[12]:

	Team	Win Count	Draw Count	Loss Count
0	Brentford	15	14	9
12	Newcastle-United	19	14	5
2	Everton	8	12	18
8	Crystal-Palace	11	12	15
6	Nottingham-Forest	9	11	18
1	Chelsea	11	11	16
3	Liverpool	19	10	9
16	Leeds-United	7	10	21
5	Brighton-and-Hove-Albion	18	8	12
11	Wolverhampton-Wanderers	11	8	19
13	Southampton	6	7	25
15	Leicester-City	9	7	22
14	West-Ham-United	11	7	20
10	Fulham	15	7	16
4	Aston-Villa	18	7	13
9	Manchester-United	23	6	9
7	Tottenham-Hotspur	18	6	14
18	Bournemouth	11	6	21
19	Arsenal	26	6	6
17	Manchester-City	28	5	5

[13]:
```
most_losses_df
```

[13]:

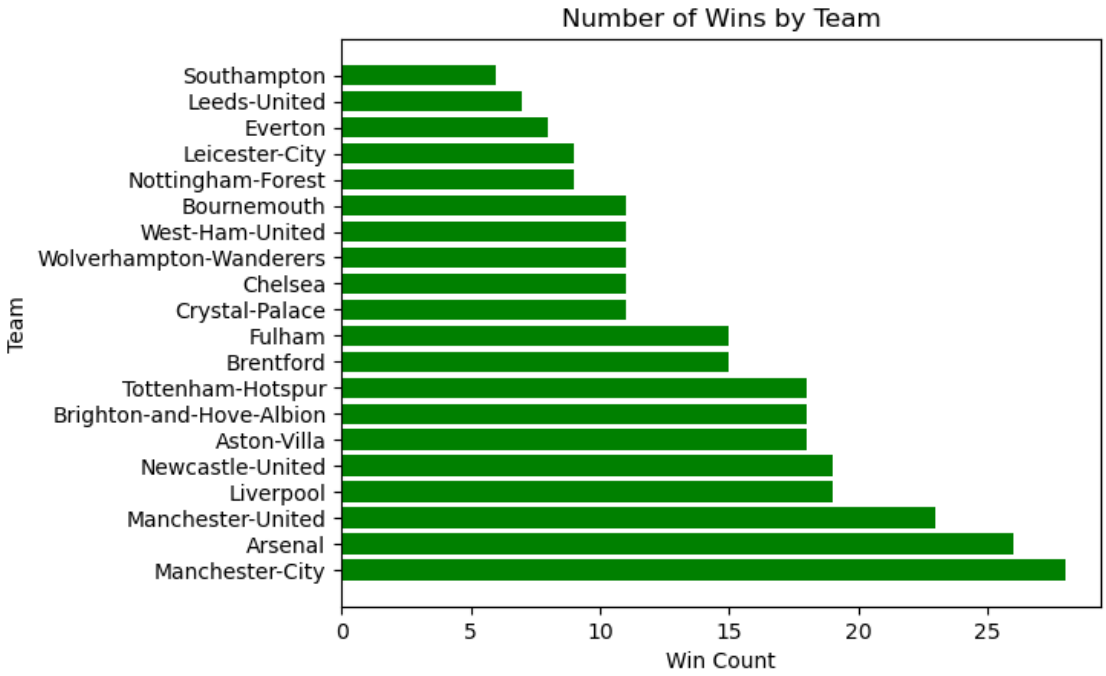
	Team	Win Count	Draw Count	Loss Count
13	Southampton	6	7	25
15	Leicester-City	9	7	22
18	Bournemouth	11	6	21
16	Leeds-United	7	10	21
14	West-Ham-United	11	7	20
11	Wolverhampton-Wanderers	11	8	19
2	Everton	8	12	18
6	Nottingham-Forest	9	11	18
10	Fulham	15	7	16
1	Chelsea	11	11	16
8	Crystal-Palace	11	12	15
7	Tottenham-Hotspur	18	6	14
4	Aston-Villa	18	7	13
5	Brighton-and-Hove-Albion	18	8	12
0	Brentford	15	14	9
9	Manchester-United	23	6	9
3	Liverpool	19	10	9
19	Arsenal	26	6	6
12	Newcastle-United	19	14	5
17	Manchester-City	28	5	5

Plotting the results in a bar chart

Most Wins

[14]:
```
# Create a horizontal bar plot
plt.barh(most_wins_df['Team'], most_wins_df['Win Count'], color='green')

# Set the tick labels and their positions
plt.yticks(most_wins_df['Team'], most_wins_df['Team'])

# Set labels and title
plt.xlabel('Win Count')
plt.ylabel('Team')
plt.title('Number of Wins by Team')
```

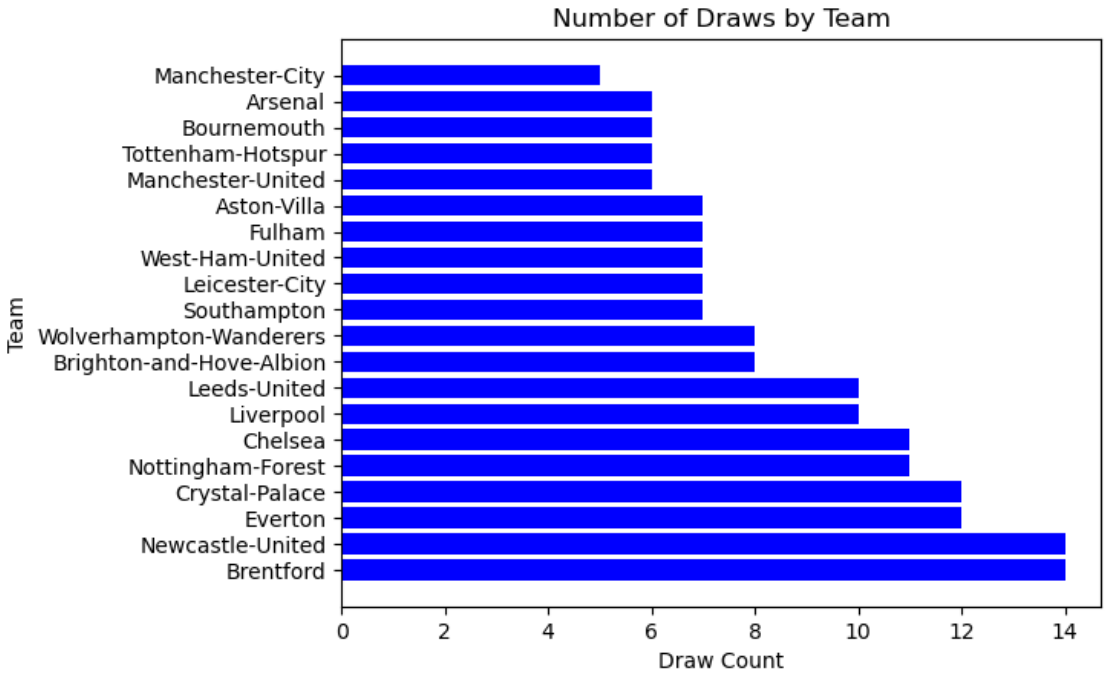
[14]:
```
Text(0.5, 1.0, 'Number of Wins by Team')
```



Most Draws

[15]:
```
# Create a horizontal bar plot
plt.barh(most_draws_df['Team'], most_draws_df['Draw Count'], color='blue')

# Set the tick labels and their positions
plt.yticks(most_draws_df['Team'], most_draws_df['Team'])

# Set labels and title
plt.xlabel('Draw Count')
plt.ylabel('Team')
plt.title('Number of Draws by Team')
```

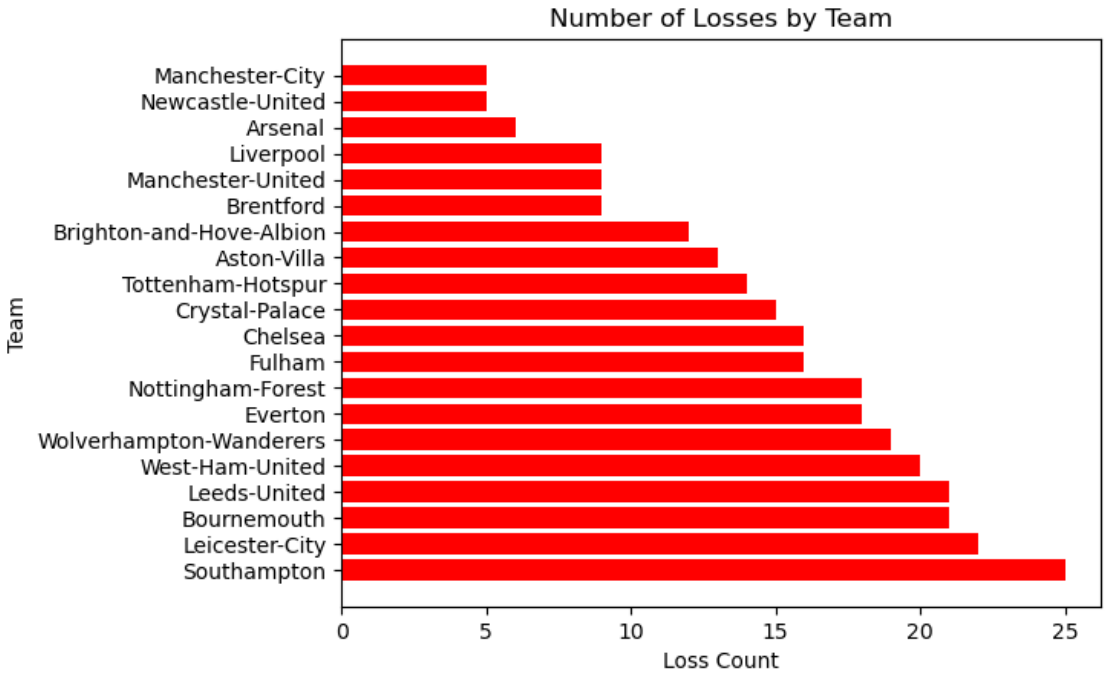
[15]:
```
Text(0.5, 1.0, 'Number of Draws by Team')
```



Most Losses

[16]:
```
# Create a horizontal bar plot
plt.barh(most_losses_df['Team'], most_losses_df['Loss Count'], color='Red')

# Set the tick labels and their positions
plt.yticks(most_losses_df['Team'], most_losses_df['Team'])

# Set labels and title
plt.xlabel('Loss Count')
plt.ylabel('Team')
plt.title('Number of Losses by Team')
```

[16]:
```
Text(0.5, 1.0, 'Number of Losses by Team')
```



1.4 Formations Analysis

[17]:
```
# Appending all of the dataframes to create one big csv containing all of the␣
↪games

all_games = pd.concat(dfs)

all_games
```

[17]:

	Date	Venue	Result	Formation	Comp	Opponent	Referee
0	2022-08-07	Away	D	4-3-3	Premier League	Leicester City	Jarred Gillett
1	2022-08-13	Home	W	5-3-2	Premier League	Manchester Utd	
2	2022-08-20	Away	L	4-3-3	Premier League	Fulham	
3	2022-08-23	Away	W	4-2-3-1	EFL Cup	Colchester Utd	
4	2022-08-27	Home	D	4-3-3	Premier League	Everton	
..	…	…	…	…	…	…	
45	2023-05-02	Home	W	4-3-3	Premier League	Chelsea	
46	2023-05-07	Away	W	4-3-3	Premier League	Newcastle Utd	
47	2023-05-14	Home	L	4-3-3	Premier League	Brighton	
48	2023-05-20	Away	L	4-3-3	Premier League	Nott'ham Forest	
49	2023-05-28	Home	W	4-3-3	Premier League	Wolves	

```
1    Stuart Attwell
2      Peter Bankes
3   James Linington
4       John Brooks
..               …
45     Robert Jones
46   Chris Kavanagh
47      Andy Madley
48   Anthony Taylor
49   Andre Marriner

[944 rows x 7 columns]
```

We group the dataframe by Formation

[18]:
```
formation_grouped_df = all_games.
 ↪pivot_table(index='Formation',columns='Result',aggfunc='size',fill_value=0)

formation_grouped_df = formation_grouped_df.sort_values(by='W',ascending=False)
```

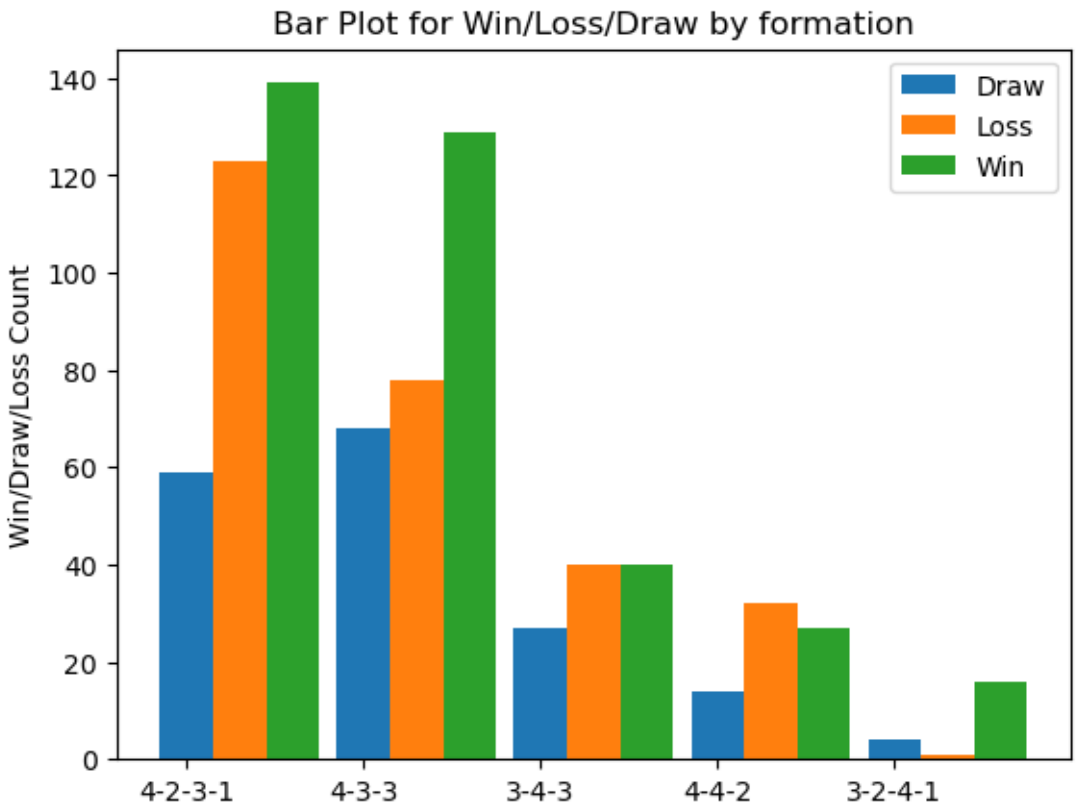
[19]:
```
# Taking the top 5 formations used in the Premier League

top_5_formations = formation_grouped_df.head()

top_5_formations
```

[19]:
```
Result      D    L    W
Formation
4-2-3-1    59  123  139
4-3-3      68   78  129
3-4-3      27   40   40
4-4-2      14   32   27
3-2-4-1     4    1   16
```

[20]:
```
# Extract the data from the pivot table
formations = top_5_formations.index
results = top_5_formations.columns
data = top_5_formations.values

# Getting the number of bars
num_bars = len(formations)
bar_positions = np.arange(num_bars)

bar_width = 0.3

for i, result in enumerate(results):
    plt.bar(bar_positions + (i * bar_width), data[:, i], width=bar_width,␣
 ↪label=result)
```

```
plt.xticks(bar_positions,formations)
plt.ylabel("Win/Draw/Loss Count")

plt.legend(['Draw','Loss','Win'])

plt.title("Bar Plot for Win/Loss/Draw by formation")
```

[20]: `Text(0.5, 1.0, 'Bar Plot for Win/Loss/Draw by formation')`



1.5 Championship race

In this section, we will study the championship race showcasing the rank and points for each time over the games

[21]:
```
# Calculating the final points for each team

games_df['Total Points'] = games_df['Win Count'] * 3 + games_df['Draw Count']

games_df.sort_values(by='Total Points',ascending=False)
```

[21]:

	Team	Win Count	Draw Count	Loss Count	Total Points
17	Manchester-City	28	5	5	89
19	Arsenal	26	6	6	84
9	Manchester-United	23	6	9	75
12	Newcastle-United	19	14	5	71
3	Liverpool	19	10	9	67
5	Brighton-and-Hove-Albion	18	8	12	62
4	Aston-Villa	18	7	13	61
7	Tottenham-Hotspur	18	6	14	60
0	Brentford	15	14	9	59
10	Fulham	15	7	16	52
8	Crystal-Palace	11	12	15	45
1	Chelsea	11	11	16	44
11	Wolverhampton-Wanderers	11	8	19	41
14	West-Ham-United	11	7	20	40
18	Bournemouth	11	6	21	39
6	Nottingham-Forest	9	11	18	38
2	Everton	8	12	18	36
15	Leicester-City	9	7	22	34
16	Leeds-United	7	10	21	31
13	Southampton	6	7	25	25

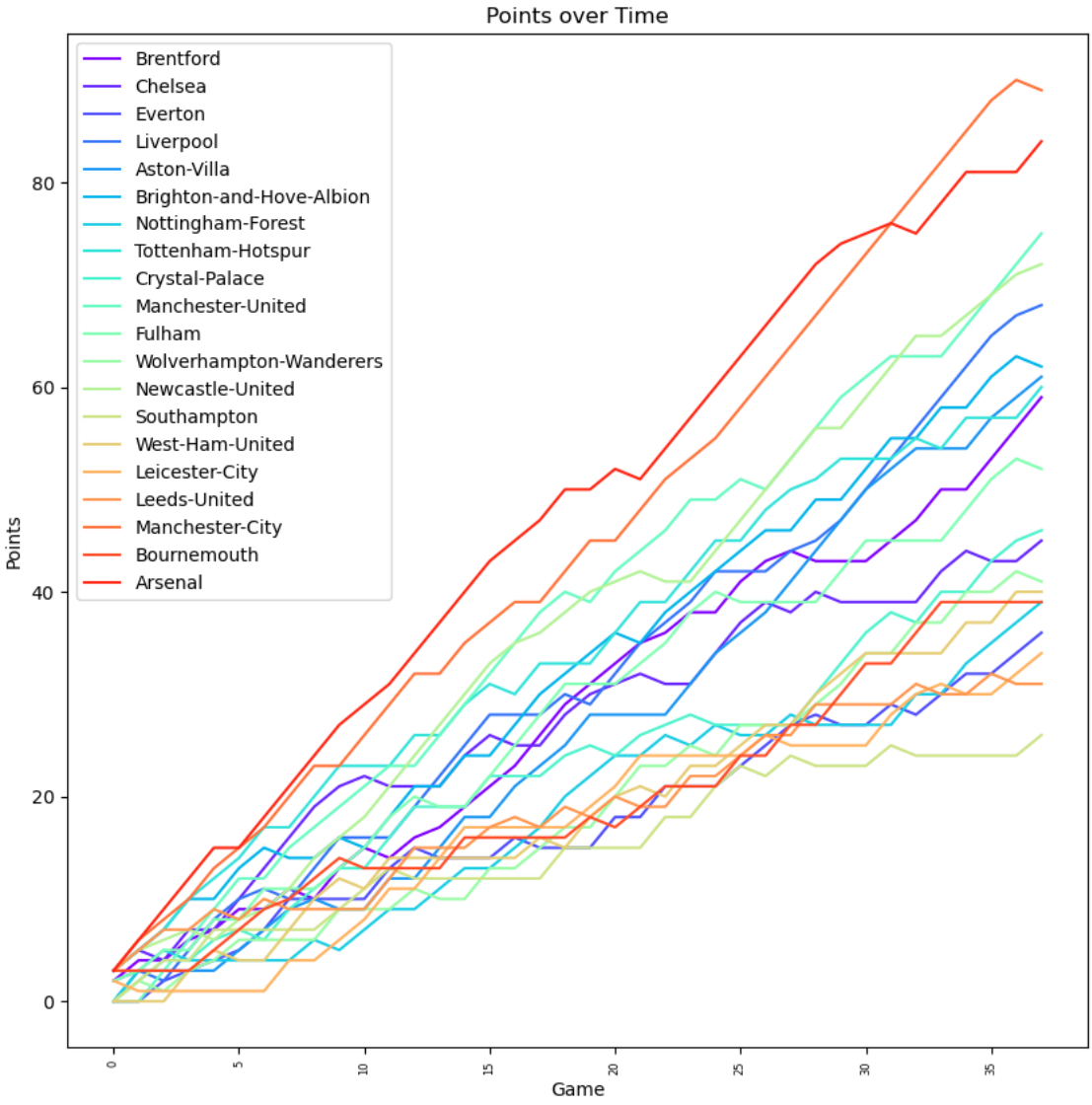
[22]:
```
part_to_delete = "-Stats.csv"

points_df = pd.DataFrame(columns=['Date','Points'])
points_array = []

for index, csv_file in enumerate(csv_files):
    df = pd.read_csv(csv_file)
    df = df[(df['Date'] != "Date") & (df['Comp'] == 'Premier League')]
    team = csv_file.replace(part_to_delete, "")

    total_points = 0  # Reset total_points for each CSV file

    for _, row in df.iterrows():
        if row['Result'] == 'W':
            # In case of Win
            total_points += 3
            new_row = {'Date': row['Date'], 'Points': total_points}
        elif row['Result'] == 'L':
            # In case of Loss
            new_row = {'Date': row['Date'], 'Points': total_points}
        elif row['Result'] == 'D':
            # In case of Draw
            total_points += 1
            new_row = {'Date': row['Date'], 'Points': total_points + 1}
```

```
        points_df = points_df.append(new_row, ignore_index=True)

    points_array.append(points_df.copy())  # Append a copy of points_df to␣
 ↪points_array
    points_df = points_df.iloc[0:0]  # Clear points_df for the next CSV file

points_array[9]
```

[22]:

	Date	Points
0	2022-08-07	0
1	2022-08-13	0
2	2022-08-22	3
3	2022-08-27	6
4	2022-09-01	9
5	2022-09-04	12
6	2022-10-02	12
7	2022-10-09	15
8	2022-10-16	17
9	2022-10-19	19
10	2022-10-22	21
11	2022-10-30	23
12	2022-11-06	23
13	2022-11-13	26
14	2022-12-27	29
15	2022-12-31	32
16	2023-01-03	35
17	2023-01-14	38
18	2023-01-18	40
19	2023-01-22	39
20	2023-02-04	42
21	2023-02-08	44
22	2023-02-12	46
23	2023-02-19	49
24	2023-03-05	49
25	2023-03-12	51
26	2023-04-02	50
27	2023-04-05	53
28	2023-04-08	56
29	2023-04-16	59
30	2023-04-27	61
31	2023-04-30	63
32	2023-05-04	63
33	2023-05-07	63
34	2023-05-13	66
35	2023-05-20	69
36	2023-05-25	72
37	2023-05-28	75

[23]:
```
import matplotlib.pyplot as plt
import matplotlib.cm as cm

teams = []
for index, csv_file in enumerate(csv_files):
    df = pd.read_csv(csv_file)
    df = df[(df['Date'] != "Date") & (df['Comp'] == 'Premier League')]
    team = csv_file.replace(part_to_delete, "")
    teams.append(team)

# Plotting all of the points for clubs along the year

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

# Generate a colormap based on the number of points_array
num_lines = len(points_array)
colors = cm.rainbow([i / num_lines for i in range(num_lines)])

for i, point_df in enumerate(points_array):
    ax.plot(range(38), point_df['Points'], color=colors[i])

# Set the rotation angle for x-axis labels
plt.xticks(rotation=90, fontsize=6)

# Add labels and title
plt.xlabel('Game')
plt.ylabel('Points')
plt.title('Points over Time')
plt.legend(teams)

plt.show()
```


Points over Time
Brentford
Chelsea
Everton
Liverpool
Aston-Villa
Brighton-and-Hove-Albion
Nottingham-Forest
Tottenham-Hotspur
Crystal-Palace
Manchester-United
Fulham
Wolverhampton-Wanderers
Newcastle-United
Southampton
West-Ham-United
Leicester-City
Leeds-United
Manchester-City
Bournemouth
Arsenal
Points
0
20
40
60
80
Game
0
5
10
15
20
25
30
35