In [7]:
```python
# Define the list of seasons
seasons = ["2018-2019", "2019-2020", "2020-2021", "2021-2022", "2022-2023"
base_url = "https://fbref.com/en/comps/9/{season}/{season}-Premier-League-

# Print the URLs for verification
for season in seasons:
    url = base_url.format(season=season)
    print(url)  # Print each URL to confirm it matches the expected format
```

```
https://fbref.com/en/comps/9/2018-2019/2018-2019-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2018-2019/2018-2019-Premier-League-Stats)
https://fbref.com/en/comps/9/2019-2020/2019-2020-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2019-2020/2019-2020-Premier-League-Stats)
https://fbref.com/en/comps/9/2020-2021/2020-2021-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2020-2021/2020-2021-Premier-League-Stats)
https://fbref.com/en/comps/9/2021-2022/2021-2022-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2021-2022/2021-2022-Premier-League-Stats)
https://fbref.com/en/comps/9/2022-2023/2022-2023-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2022-2023/2022-2023-Premier-League-Stats)
https://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Stats (ht
tps://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Stats)
```

In [8]:
```python
import requests

# Define the URL for the test season (2023-2024)
url = "https://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Sta

# Fetch the HTML content
response = requests.get(url)

# Print the first 500 characters of the HTML to verify
print(response.text[:500])  # Print the first 500 characters of the HTML
```

```


<!DOCTYPE html>
<html data-version="klecko-" data-root="/home/fb/deploy/www/base" lang="e
n" class="no-js" >
<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=2.0" />
    <link rel="dns-prefetch" href="https://cdn.ssref.net/req/202504030" /
>
<script>
/* https://docs.osano.com/hc/en-us/articles/22469433444372-Google-Consent
-Mode-v2 (https://docs.osano.com/hc/en-us/articles/22469433444372-Google-
Consent-Mode-v2)  */
  window.dataLayer = w
```

In [9]:
```
from bs4 import BeautifulSoup

# Parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Construct the table ID dynamically based on the test season
table_id = "results2023-202491_home_away"  # Replace with the correct ID f

# Locate the Home/Away Table using its ID
table = soup.find('table', {'id': table_id})

if table:
    print("Table found!")  # Confirm the table was located
    print(table.prettify()[:1000])  # Print the first 1000 characters of t
else:
    print("Table not found.")  # Notify if the table could not be found
```

```
Table found!
<table class="stats_table sortable min_width force_mobilize" data-cols-to
-freeze=",2" id="results2023-202491_home_away">
 <caption>
  Premier League Table
 </caption>
 <colgroup>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
  <col/>
 </colgroup>
 <thead>
  <tr class="over_header">
   <th aria-label="" class="over_header center" colspan="2" data-stat="">
   </th>
   <th aria-label="" class="over_header center group_start" colspan="13"
data-stat="header_home">
    Home
   </th>
   <th aria-label="" class="over_header center group_start" colspan="13"
data-stat="header_away">
    Away
   </th>
  </tr>
  <tr>
   <th aria-label="Rank" class="poptip sort_default_asc center" data-stat
="rank" data-tip="&lt;strong&gt;Rank&lt;/strong&gt;&lt;br&gt;Squad finish
in competition&lt;br&gt;Finish within the l
```

In [18]:
```python
# Initialize an empty list for headers
headers = []

# Extract headers from <th> elements
for th in table.find_all('th'):
    if 'data-stat' in th.attrs:
        headers.append(th['data-stat'])  # Append the 'data-stat' attribut

# Print the extracted headers for verification
print("Headers:", headers)  # Print the headers to confirm they match the
```

```
Headers: ['', 'header_home', 'header_away', 'rank', 'team', 'home_games',
'home_wins', 'home_ties', 'home_losses', 'home_goals_for', 'home_goals_ag
ainst', 'home_goal_diff', 'home_points', 'home_points_avg', 'home_xg_fo
r', 'home_xg_against', 'home_xg_diff', 'home_xg_diff_per90', 'away_game
s', 'away_wins', 'away_ties', 'away_losses', 'away_goals_for', 'away_goal
s_against', 'away_goal_diff', 'away_points', 'away_points_avg', 'away_xg_
for', 'away_xg_against', 'away_xg_diff', 'away_xg_diff_per90', 'rank', 'r
ank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'ra
nk', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'ran
k']
```

In [17]:
```python
# Initialize an empty list for rows
rows = []

# Extract rows from <tr> elements
for row in table.find_all('tr'):
    cells = [cell.text.strip() for cell in row.find_all('td')]  # Extract
    if cells:  # Skip empty rows
        rows.append(cells)  # Append non-empty rows to the rows list

# Print the first few rows for verification
print("First few rows:")  # Print a message before displaying rows
for row in rows[:5]:
    print(row)  # Print the first 5 rows to confirm the data is correct
```

```
First few rows:
['Manchester City', '19', '14', '5', '0', '51', '16', '+35', '47', '2.4
7', '40.7', '14.0', '+26.7', '+1.41', '19', '14', '2', '3', '45', '18',
'+27', '44', '2.32', '39.8', '21.6', '+18.2', '+0.96']
['Arsenal', '19', '15', '2', '2', '48', '16', '+32', '47', '2.47', '43.
5', '13.5', '+30.0', '+1.58', '19', '13', '3', '3', '43', '13', '+30', '4
2', '2.21', '32.6', '14.5', '+18.2', '+0.96']
['Liverpool', '19', '15', '3', '1', '49', '17', '+32', '48', '2.53', '54.
7', '17.6', '+37.1', '+1.95', '19', '9', '7', '3', '37', '24', '+13', '3
4', '1.79', '33.0', '28.1', '+4.9', '+0.26']
['Aston Villa', '19', '12', '4', '3', '48', '28', '+20', '40', '2.11', '3
9.0', '26.3', '+12.7', '+0.67', '19', '8', '4', '7', '28', '33', '-5', '2
8', '1.47', '24.3', '33.6', '-9.3', '-0.49']
['Tottenham', '19', '13', '0', '6', '38', '27', '+11', '39', '2.05', '39.
2', '29.3', '+9.9', '+0.52', '19', '7', '6', '6', '36', '34', '+2', '27',
'1.42', '28.9', '34.1', '-5.2', '-0.27']
```

In [19]:
```python
# Print headers and rows for debugging
print("Headers:", headers)
print("First few rows:")
for row in rows[:5]:
    print(row)

# Keep only the relevant headers (columns 5 to 31)
headers = headers[4:31]  # Python uses 0-based indexing, so 4:31 gives col

# Verify the number of headers matches the number of columns in rows
print(f"Number of headers: {len(headers)}")
print(f"Number of columns in rows: {len(rows[0])}")
```

```
Headers: ['', 'header_home', 'header_away', 'rank', 'team', 'home_games',
'home_wins', 'home_ties', 'home_losses', 'home_goals_for', 'home_goals_ag
ainst', 'home_goal_diff', 'home_points', 'home_points_avg', 'home_xg_fo
r', 'home_xg_against', 'home_xg_diff', 'home_xg_diff_per90', 'away_game
s', 'away_wins', 'away_ties', 'away_losses', 'away_goals_for', 'away_goal
s_against', 'away_goal_diff', 'away_points', 'away_points_avg', 'away_xg_
for', 'away_xg_against', 'away_xg_diff', 'away_xg_diff_per90', 'rank', 'r
ank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'ra
nk', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'ran
k']
First few rows:
['Manchester City', '19', '14', '5', '0', '51', '16', '+35', '47', '2.4
7', '40.7', '14.0', '+26.7', '+1.41', '19', '14', '2', '3', '45', '18',
'+27', '44', '2.32', '39.8', '21.6', '+18.2', '+0.96']
['Arsenal', '19', '15', '2', '2', '48', '16', '+32', '47', '2.47', '43.
5', '13.5', '+30.0', '+1.58', '19', '13', '3', '3', '43', '13', '+30', '4
2', '2.21', '32.6', '14.5', '+18.2', '+0.96']
['Liverpool', '19', '15', '3', '1', '49', '17', '+32', '48', '2.53', '54.
7', '17.6', '+37.1', '+1.95', '19', '9', '7', '3', '37', '24', '+13', '3
4', '1.79', '33.0', '28.1', '+4.9', '+0.26']
['Aston Villa', '19', '12', '4', '3', '48', '28', '+20', '40', '2.11', '3
9.0', '26.3', '+12.7', '+0.67', '19', '8', '4', '7', '28', '33', '-5', '2
8', '1.47', '24.3', '33.6', '-9.3', '-0.49']
['Tottenham', '19', '13', '0', '6', '38', '27', '+11', '39', '2.05', '39.
2', '29.3', '+9.9', '+0.52', '19', '7', '6', '6', '36', '34', '+2', '27',
'1.42', '28.9', '34.1', '-5.2', '-0.27']
Number of headers: 27
Number of columns in rows: 27
```

In [20]:
```
import pandas as pd

# Create a DataFrame with the aligned headers and rows
df = pd.DataFrame(rows, columns=headers)

# Print the first few rows of the DataFrame to verify alignment
print("DataFrame preview:")
print(df.head())
```

```
DataFrame preview:
              team home_games home_wins home_ties home_losses home_goals_
for  \
0  Manchester City         19        14         5           0           
51   
1          Arsenal         19        15         2           2           
48   
2        Liverpool         19        15         3           1           
49   
3      Aston Villa         19        12         4           3           
48   
4        Tottenham         19        13         0           6           
38   

  home_goals_against home_goal_diff home_points home_points_avg  ...  \
0                 16            +35          47            2.47  ...   
1                 16            +32          47            2.47  ...   
2                 17            +32          48            2.53  ...   
3                 28            +20          40            2.11  ...   
4                 27            +11          39            2.05  ...   

  away_losses away_goals_for away_goals_against away_goal_diff away_point
s  \
0           3             45                 18            +27           4
4   
1           3             43                 13            +30           4
2   
2           3             37                 24            +13           3
4   
3           7             28                 33             -5           2
8   
4           6             36                 34             +2           2
7   

  away_points_avg away_xg_for away_xg_against away_xg_diff away_xg_diff_p
er90  
0            2.32        39.8            21.6        +18.2              +
0.96  
1            2.21        32.6            14.5        +18.2              +
0.96  
2            1.79        33.0            28.1         +4.9              +
0.26  
3            1.47        24.3            33.6         -9.3              -
0.49  
4            1.42        28.9            34.1         -5.2              -
0.27  

[5 rows x 27 columns]
```

In [21]:

```python
import os

# Define the path to save the scraped data
save_dir = r"C:\Users\matth\OneDrive\Documents\data_science_project\premie
os.makedirs(save_dir, exist_ok=True)  # Ensure the directory exists

# Define the file path for the CSV file
file_path = os.path.join(save_dir, "2023-2024_home_away_stats.csv")

# Save the DataFrame to a CSV file
df.to_csv(file_path, index=False)

# Print confirmation message
print(f"Data saved to {file_path}")  # Confirm the file has been saved
```

```
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2023-2024_home_away_stats.csv
```

In [24]:
```
import time

# Define the list of seasons and base URL
seasons = ["2018-2019", "2019-2020", "2020-2021", "2021-2022", "2022-2023"
base_url = "https://fbref.com/en/comps/9/{season}/{season}-Premier-League-

# Define the directory to save the scraped data
save_dir = r"C:\Users\matth\OneDrive\Documents\data_science_project\premie
os.makedirs(save_dir, exist_ok=True)  # Ensure the directory exists

# Function to scrape data for a single season
def scrape_season(season):
    print(f"Scraping data for season: {season}")

    # Construct the URL for the current season
    url = base_url.format(season=season)

    # Fetch the HTML content
    response = requests.get(url)
    if response.status_code != 200:
        print(f"Failed to fetch data for season {season}. Status code: {re
        return

    # Parse the HTML content
    soup = BeautifulSoup(response.content, 'html.parser')

    # Locate the Home/Away Table
    table_id = f"results{season}91_home_away"
    table = soup.find('table', {'id': table_id})

    if not table:
        print(f"Table not found for season {season}. Skipping...")
        return

    # Extract headers
    headers = []
    for th in table.find_all('th'):
        if 'data-stat' in th.attrs:
            headers.append(th['data-stat'])

    # Keep only the relevant headers (columns 5 to 31)
    headers = headers[4:31]

    # Extract rows
    rows = []
    for row in table.find_all('tr'):
        cells = [cell.text.strip() for cell in row.find_all('td')]
        if cells:  # Skip empty rows
            rows.append(cells)

    # Create a DataFrame
    df = pd.DataFrame(rows, columns=headers)

    # Define the file path for the CSV file
    file_path = os.path.join(save_dir, f"{season}_home_away_stats.csv")

    # Save the DataFrame to a CSV file
    df.to_csv(file_path, index=False)
    print(f"Data saved to {file_path}")

    # Add a delay to avoid overloading the server
    time.sleep(3)

# Loop through all seasons and scrape their data
for season in seasons:
    scrape_season(season)
```

```
Scraping data for season: 2018-2019
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2018-2019_home_away_stats.csv
Scraping data for season: 2019-2020
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2019-2020_home_away_stats.csv
Scraping data for season: 2020-2021
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2020-2021_home_away_stats.csv
Scraping data for season: 2021-2022
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2021-2022_home_away_stats.csv
Scraping data for season: 2022-2023
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2022-2023_home_away_stats.csv
Scraping data for season: 2023-2024
Data saved to C:\Users\matth\OneDrive\Documents\data_science_project\prem
ier-league-home-advantage\data\raw_data\2023-2024_home_away_stats.csv
```

In [ ]: