In [7]:

```
import pandas as pd
import os

# Define the directories for raw and processed data
raw_data_dir = r"C:\Users\matth\OneDrive\Documents\data_science_projec
processed_data_dir = r"C:\Users\matth\OneDrive\Documents\data_science_
os.makedirs(processed_data_dir, exist_ok=True)

# List of seasons
seasons = ["2018-2019", "2019-2020", "2020-2021", "2021-2022", "2022-2

def clean_match_data(season):
    print(f"Cleaning match data for {season}")

    # Construct the file path for the raw data
    raw_file_path = os.path.join(raw_data_dir, f"{season}.csv")

    # Load the CSV file, specifying the 'score' column as a string
    try:
        df = pd.read_csv(raw_file_path, encoding='utf-8', dtype={'scor
    except FileNotFoundError:
        print(f"  File not found for {season}. Skipping.")
        return

    # Remove Unnecessary Columns
    columns_to_drop = ['match_report', 'notes', 'gameweek']
    df = df.drop(columns=columns_to_drop, errors='ignore')

    # Remove Blank Rows
    df = df.dropna(how='all')

    # Fix Score Formatting (Unicode replacement) and add prefix/spaces
    df['score'] = df['score'].str.replace(u'\u2013', ' - ', regex=Fals

    # Convert Attendance to Numeric
    if 'attendance' in df.columns:
        df['attendance'] = df['attendance'].str.replace(',', '', regex
        df['attendance'] = pd.to_numeric(df['attendance'], errors='coe

    return df

# Clean data for all seasons and concatenate
all_seasons_data = []
for season in seasons:
    cleaned_data = clean_match_data(season)
    if cleaned_data is not None:
        cleaned_data['season'] = season
        all_seasons_data.append(cleaned_data)

# Concatenate all seasons' data into a single DataFrame
if all_seasons_data:
    combined_df = pd.concat(all_seasons_data, ignore_index=True)

    # Save the combined data to a single CSV file
    combined_file_path = os.path.join(processed_data_dir, "all_seasons
    combined_df.to_csv(combined_file_path, index=False)
    print(f"Combined data saved to {combined_file_path}")
else:
    print("No data was cleaned.")
print("Match data cleaning and combining complete.")
```

```
Cleaning match data for 2018-2019
Cleaning match data for 2019-2020
Cleaning match data for 2020-2021
Cleaning match data for 2021-2022
Cleaning match data for 2022-2023
Cleaning match data for 2023-2024
Combined data saved to C:\Users\matth\OneDrive\Documents\data_science
_project\premier-league-home-advantage\data\processed_data\all_season
s_match_data.csv
Match data cleaning and combining complete.
```

In [8]:

```
import pandas as pd
import os

# Define the directories for raw and processed data
raw_data_dir = r"C:\Users\matth\OneDrive\Documents\data_science_projec
processed_data_dir = r"C:\Users\matth\OneDrive\Documents\data_science_
os.makedirs(processed_data_dir, exist_ok=True)

# List of seasons
seasons = ["2018-2019", "2019-2020", "2020-2021", "2021-2022", "2022-2

def clean_team_stats_data(season):
    print(f"Cleaning team stats data for {season}")

    # Construct the file path for the raw data
    raw_file_path = os.path.join(raw_data_dir, f"{season}_home_away_st

    # Load the CSV file
    try:
        df = pd.read_csv(raw_file_path, encoding='utf-8')
    except FileNotFoundError:
        print(f"  File not found for {season}. Skipping.")
        return None  # Return None if file not found

    # Drop unnecessary columns
    columns_to_drop = [
        'home_goals_for', 'home_goals_against', 'home_points',
        'away_goals_for', 'away_goals_against', 'away_points'
    ]
    df = df.drop(columns=columns_to_drop, errors='ignore')

    return df

# Clean and save data for each season
for season in seasons:
    cleaned_data = clean_team_stats_data(season)
    if cleaned_data is not None:
        # Construct the file path for the cleaned data
        cleaned_file_path = os.path.join(processed_data_dir, f"clean_{
        cleaned_data.to_csv(cleaned_file_path, index=False)
        print(f"Cleaned data for {season} saved to {cleaned_file_path}
    else:
        print(f"No data was cleaned for {season}.")

print("Team stats data cleaning complete.")
```

```
Cleaning team stats data for 2018-2019
Cleaned data for 2018-2019 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2018-2019_home_away_stats.csv
Cleaning team stats data for 2019-2020
Cleaned data for 2019-2020 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2019-2020_home_away_stats.csv
Cleaning team stats data for 2020-2021
Cleaned data for 2020-2021 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2020-2021_home_away_stats.csv
Cleaning team stats data for 2021-2022
Cleaned data for 2021-2022 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2021-2022_home_away_stats.csv
Cleaning team stats data for 2022-2023
Cleaned data for 2022-2023 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2022-2023_home_away_stats.csv
Cleaning team stats data for 2023-2024
Cleaned data for 2023-2024 saved to C:\Users\matth\OneDrive\Documents
\data_science_project\premier-league-home-advantage\data\processed_da
ta\clean_2023-2024_home_away_stats.csv
Team stats data cleaning complete.
```

In [10]:

```python
# Load the cleaned match data CSV file
file_path = r"C:\Users\matth\OneDrive\Documents\data_science_project\p
df = pd.read_csv(file_path)

# Replace empty values in the 'attendance' column with 0
df['attendance'] = df['attendance'].fillna(0)

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

In [13]:

```
# Define the directory where the processed (cleaned) data is stored
processed_data_dir = r"C:\Users\matth\OneDrive\Documents\data_science_

# List of seasons to process
seasons = ["2018-2019", "2019-2020", "2020-2021", "2021-2022", "2022-2

# Create an empty list to hold the DataFrame for each season
all_dataframes = []

print("Starting to load and combine team stats data...")

# Loop through each season
for season in seasons:
    print(f" Processing season: {season}")

    # Construct the file path for the cleaned data file for the curren
    cleaned_file_path = os.path.join(processed_data_dir, f"clean_{seas

    # Load the CSV file for the season
    try:
        df_season = pd.read_csv(cleaned_file_path, encoding='utf-8')

        # Add a new column 'season' containing the season identifier
        df_season['season'] = season
        print(f"  Successfully loaded and added season column for {sea

        # Append the DataFrame to our list
        all_dataframes.append(df_season)

    except FileNotFoundError:
        # Print a message if the file for a season is not found and co
        print(f"  File not found for {season}: {cleaned_file_path}. Sk
    except Exception as e:
        # Print other potential errors during file loading
        print(f"  An error occurred loading {season}: {e}. Skipping.")


# Check if we have successfully loaded any DataFrames
if all_dataframes:
    print("\nCombining all seasonal dataframes...")
    # Concatenate all the DataFrames in the list into a single DataFra
    all_seasons_df = pd.concat(all_dataframes, ignore_index=True)

    # Define the path for the final combined CSV file
    output_file_path = os.path.join(processed_data_dir, "all_seasons_t

    # Save the combined DataFrame to a new CSV file
    try:
        all_seasons_df.to_csv(output_file_path, index=False, encoding=
        print(f"\nCombined team stats data saved successfully to: {out
        print(f"  Total rows in combined file: {len(all_seasons_df)}")
    except Exception as e:
        print(f"\nAn error occurred while saving the combined file: {e

else:
    # Print a message if no dataframes were loaded (e.g., all files we
    print("\nNo dataframes were loaded. Cannot combine or save.")
print("\nTeam stats data combination process complete.")
```

```
Starting to load and combine team stats data...
 Processing season: 2018-2019
  Successfully loaded and added season column for 2018-2019.
 Processing season: 2019-2020
  Successfully loaded and added season column for 2019-2020.
 Processing season: 2020-2021
  Successfully loaded and added season column for 2020-2021.
 Processing season: 2021-2022
  Successfully loaded and added season column for 2021-2022.
 Processing season: 2022-2023
  Successfully loaded and added season column for 2022-2023.
 Processing season: 2023-2024
  Successfully loaded and added season column for 2023-2024.

Combining all seasonal dataframes...

Combined team stats data saved successfully to: C:\Users\matth\OneDri
ve\Documents\data_science_project\premier-league-home-advantage\data
\processed_data\all_seasons_team_data.csv
  Total rows in combined file: 120

Team stats data combination process complete.
```

In [ ]: