Instructions manual for

Author: Ilias Samathrakis

Contents

1 Introduction

Mourinho plays the catenaccio of the modern years. Guardiola's Barcelona was better than Heynckes's Bayern Munich. Robert Lewandowski is a 'killer' striker. These are just a few examples of statements that can make football fans chatting or even arguing depending on their personal feelings, taste and emotions for hours. An interesting question though is whether there is an unbiased way of assessing these statements, providing evidence regarding their validity based on facts. The answer to the latter question is surprisingly 'yes'.

Football analytics refers to the use of statistical analysis and advanced metrics to gain insights and make informed decisions in the world of football. It involves the collection, processing, and interpretation of various types of data, such as match statistics, player and team performance metrics. Particular examples of performance metrics include, among others the expected assists (xA), the expected threat (xT), passes per defensive action (PPDA) and progressive carries.

The most popular metric though is undoubtedly the expected goals (xG) which intends to measure the probability of a shot resulting in a goal. For the evaluation of xG, several parameters are taken into account, the most popular of which are the distance and the angle from the goal, the part of the body used to kick the ball, the position of the defenders and others.

This repository contains **myfootballanalytics**, a user-friendly in-house developed python-based tool to analyse and visualise football data obtained from understat.com in order to assess the performance of football clubs and players and extract meaningful insights.

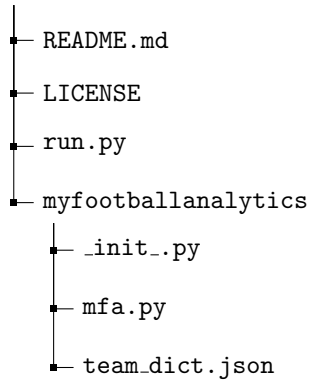
In a nutshell, by using **myfootballanalytics**, you are able to:

- Analyse the performance of your favourite team:
  - at a specific match
  - during a specific season
- Compare this year's performance of your favourite team with
  - last year's
  - last to last year's
  - any other year's
- Compare players' performance to answer the following questions:
  - who is the best 'killer'?
  - who should perform better?
- Compare teams' performance to answer the following questions:
  - which team has the best attacking performance ever?
  - which team has the best defensive performance ever?
  - which team won the league while being less than average in attack?
  - has your favourite team been improved this year compared to last year or not?

... and many others.

2 Description

The current version of the code includes files according to the following tree:



A short and comprehensive description of each of the files is presented:

- **run.py**
  Sample file to run the code
- **LICENSE**
  File containing the license of the code
- **Instruction_manual.pdf**
  Detailed instructions
- **setup.py**
  File containing the information about the distribution
- **myfootballanalytics**
  Directory containing the code.
- **teams_dict.json**
  File containing the teams involved with their corresponding league. The code will automatically notify in case a new entry is required.
- **mfa.py**
  File containing all classes

3 Installation

You can get the code either from the pip package manager or from the github repository. Detailed instructions for both ways follow.

3.1 Package manager pip

The preferred way is to install it via pip. In order to do so, type the command

```
$ pip install myfootballanalytics
```

3.2 Github

Alternatively, you can download the code from the github repository. Installation requires Python3 and pip. Follow the instructions to download it.

1. Download the repository
2. Paste it at a convenient directory
3. Install the required external packages by typing the following commands

```
$ pip install beautifulsoup4
$ pip install requests
$ pip install tqdm
$ pip install mplsoccer
```

4. Modify run.py
5. Run run.py from command line by using

```
python run.py
```

   or directly from your python IDE

Note: run.py should be in the same directory as myfootballanalytics directory. Otherwise, the import statements have to be modified accordingly.

4 Capabilities

The current version of the code supports five functionalities which are enabled by combinations of parameters found as tags within the run.py file. Detailed description follows:

Parameters

- parallel
  Type: boolean.
  Description: Enables parallel execution with the maximum CPU available.
  Available values: True or False
  Default value: No
- save_dir_path
  Type: string.
  Description: The path Football_Data directory is saved.
  Available paths: All valid directories.
  Default value: current working directory
- leagues
  Type: list of strings.
  Description: List containing the chosen leagues to perform analysis.
  Available leagues: Ligue1, PremierLeague, Bundesliga, LaLiga, SerieA
  Default value: No

- seasons
  Type: list of strings.
  Description: List containing the chosen seasons to perform analysis.
  Available seasons: All seasons from 2014-2015.
  Default value: No

- home_team
  Type: string.
  Description: Home team of a match.
  Available home teams: All teams included in the file teams_dict.json
  Default value: No

- away_team
  Type: string.
  Description: Away team of a match.
  Available away teams: All teams included in the file teams_dict.json
  Default value: No

- season
  Type: string.
  Description: A single season.
  Available seasons: All seasons from 2014-2015.
  Default value: No

- league
  Type: string.
  Description: A single league.
  Available leagues: Ligue1, PremierLeague, Bundesliga, LaLiga, SerieA
  Default value: No

- team
  Type: string.
  Description: A single team.
  Available teams: All teams included in the file teams_dict.json

- save_csv_file_leagues
  Type boolean.
  Description: Saves a csv file containing league's stats under the save_dir_path directory.
  Available values: True or False
  Default value: False

- save_csv_file_players
  Type: boolean.
  Description: Saves a csv file containing players' stats under the save_dir_path directory.
  Available values: True or False
  Default value: False

Functionalities

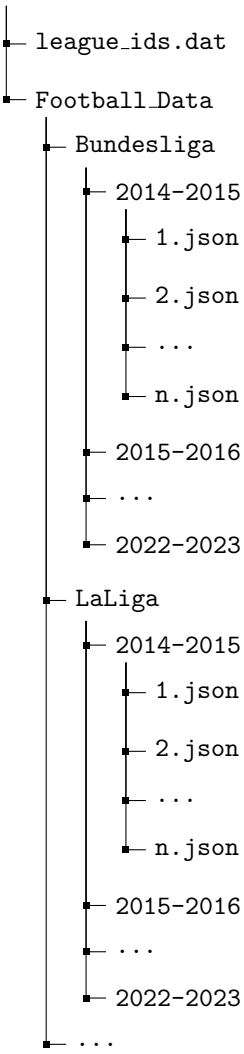
- update_id_file
  Football Data for this repository are extracted from understat.com website. Understat associates each match with a unique id. Setting this tag True, enables the creation of the file league_ids.dat, that contains the ids of each match in the following format:

```
"""
Format of file league_ids.dat

SerieA: 2014-2015:  4780 4781 4782 4779 ...
SerieA: 2015-2016:  552 551 553 554 ...
   .
   .
   .
SerieA: 2022-2023: 18582 18583 18584 18585 ...
Bundesliga: 2014-2015: 5160 5161 5159 5162 ...
   .
   .
   .
"""
```

Required parameters: parallel, save_dir_path

- get_data
  Reads the file league_ids.dat and extracts data from understat.com website for the specified leagues and seasons. The data are stored under the directory Football_Data located within the path specified by the tag save_dir_name, following the tree



Required parameters: leagues, seasons, save_dir_path, save_csv_file_leagues

- analyze_match
  Setting the tag True plots the positions of each shot for the selected match and further provides a table with statistics

  Required parameters: home_team, away_team, season, save_dir_path

- analyze_league
  Setting the tag True
  - plots the average xG-for vs the average xG-against figure of all teams participated in the selected leagues and seasons
  - plots the average xG difference of all teams in the selected leagues and seasons

  - plots the median xG difference of all teams in the selected leagues and seasons
  - displays a table with the best shot performance of the players participated in the selected leagues and seasons

  Required parameters: league, seasons, save_dir_path, save_csv_file_players

- analyze_team
  Setting the tag True
  - plots the histogram of average chances created and conceded of the selected team for every selected seasons
  - plots the average xG-for vs the average xG-against figure for a selected team over the selected seasons
  - plots the average chances created and conceded of the selected team over the selected seasons
  - plots the average shot distance (for and against) of the selected team over the selected seasons

  Required parameters: team, seasons, save_dir_path

The following table summarizes the required parameters of each functionality

Functionalities	Required parameters
update_id_file	parallel save_dir_path
get_data	leagues seasons save_dir_path save_csv_file_leagues
analyze_league	home_team away_team season save_dir_path
analyze_league	league seasons save_dir_path save_csv_file_players
analyze_team	teams seasons save_dir_path

5 How to use

A sample running file called run.py is provided

```
import myfootballanalytics as mfa ## Imports the package

update_id_file = False   ## Updates file containing match ids from https://understat.com/
get_data =       False   ## Scrapes data from https://understat.com/ and saves them in the defined
    directory
analyze_match =  True    ## Plots the position of all shots of a given match and provides a table
    with information
analyze_league = False   ## Plots xG figure for the teams of the selected league
analyze_team =   False   ## Plots xG figure for the selected team over the selected seasons
```

```
save_csv_file_leagues = True   ## Saves league's data to csv file
save_csv_file_players = True   ## Saves players' data to csv file
parallel = True   ## Activates multiprocessing
save_dir_path = ""   ## Saving directory. Default: current working directory. Empty string activates
    the default setting.
leagues = ["PremierLeague","Ligue1","Bundesliga","LaLiga","SerieA"]   ## Available leagues: Ligue1,
    PremierLeague, Bundesliga, LaLiga, SerieA
seasons = ["2014-2015","2015-2016","2016-2017","2017-2018","2018-2019","2019-2020","2020-2021","
    2021-2022","2022-2023"]   ## Available seasons: 2014-2022
home_team = "FC Cologne"   ## Available teams: all teams
away_team = "Bayern Munich"   ## Available teams: all teams
season = "2022-2023"   ## Available season: all seasons
league = "Bundesliga"   ## Choose league to analyze. Available leagues: Ligue1, PremierLeague,
    Bundesliga, LaLiga, SerieA
team = "Arsenal"   ## Available teams: all teams

if update_id_file:
    ids = mfa.FindIDs(parallel,save_dir_path)

if get_data:
    data = mfa.DataUpdater(leagues,seasons,save_dir_path,save_csv_file_leagues)

if analyze_match:
    md = mfa.MatchDataLoader(home_team,away_team,season,save_dir_path)
    match_data = md.load_match_data()

    match = mfa.MatchAnalyzer(match_data)
    table = match.analyze_match()
    print(table)

if analyze_league:
    league_data = mfa.LeagueAnalyzer(league,seasons,save_dir_path,save_csv_file_players)

if analyze_team:
    td = mfa.TeamDataLoader(seasons,team,save_dir_path)
    team_data = td.load_team_data()

    team_analysis = mfa.TeamAnalyzer(team_data,seasons,team)
    team_analysis.analyze_team()
```

The package **myfootballanalytics** contains the following classes and methods

```
#The package myfootballanalytics contains the following classes and public methods (without their
    dependencies and their attributes)

FindIDs()
#       Public method
                findids()
DataUpdater()
#       Public method
                update_data()
MatchDataLoader()
#       Public method
                load_match_data()
MatchAnalyzer()
#       Public method
                analyze_match()
LeagueAnalyzer()
#       Public method
                analyze_league()
TeamDataLoader()
#       Public method
```

```
                load_team_data()
TeamAnalyzer()
#       Public method
                analyze_team()
```

Use the following command to run the code from the command line

```
python run.py
```

or alternatively run from your python IDE.

6 The file teams_dict.json

In order to associate each team with a league, the file teams_dict.json is used. It contains the following dictionary

```
{
    "Arsenal": "PremierLeague",
    "West Ham": "PremierLeague",
    "Liverpool": "PremierLeague",
    "Stoke": "PremierLeague",
    "Manchester United": "PremierLeague",
    "Everton": "PremierLeague",
    "Tottenham": "PremierLeague",
                  .
                  .
                  .
}
```

In case a team does not exist in the dictionary, the code will notify about its absence. Add the missing team with its corresponding league. The available leagues are "Ligue1", "PremierLeague","LaLiga","SerieA","Bundesliga"

7 Examples

- Question: How can I download all data?

  Answer: Run run.py with the following input parameters

  ```
  update_id_file = True
  get_data = True
  analyze_match = False
  analyze_league = False
  analyze_team = False

  save_dir_path = ""
  leagues = ["Ligue1","PremierLeague","Bundesliga","LaLiga","SerieA"]
  seasons = ["2014-2015","2015-2016","2016-2017","2017-2018","2018-2019","2019-2020","2020-2021",
      "2021-2022","2022-2023"]
  ```

- Question: How can I collect new data within the season?

  Answer: Run run.py with the following input parameters

  ```
  update_id_file = True
  get_data = True
  analyze_match = False
  analyze_league = False
  analyze_team = False
  ```

```
save_dir_path = ""
leagues = ["Ligue1","PremierLeague","Bundesliga","LaLiga","SerieA"]
seasons = ["2022-2023"]
```

- Question: How can I analyze the match between Manchester United and Manchester City in the season 2018-2019?

  Answer: Run run.py with the following input parameters

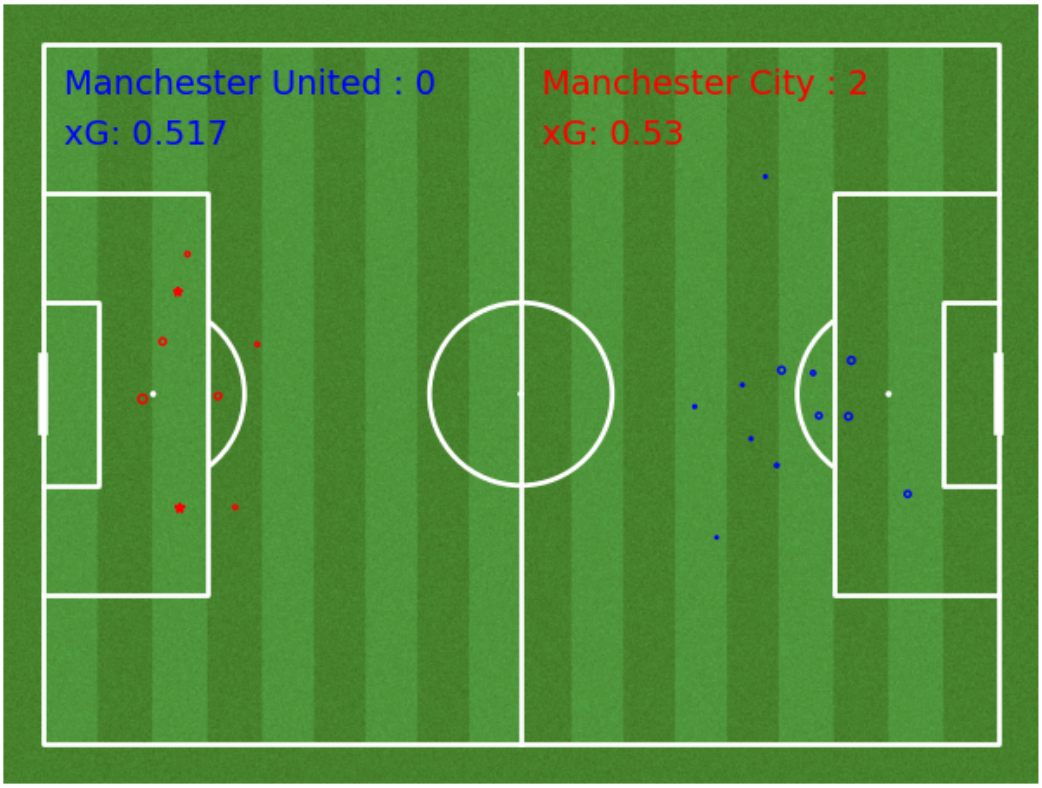
  Input:

```
update_id_file = False
get_data = False
analyze_match = True
analyze_league = False
analyze_team = False

save_dir_path = ""
home_team = "Manchester United"
away_team = "Manchester City"
season = "2018-2019"
```

  Output:

	Manchester United	Manchester City
Total xG	0.517	0.530
xG per chance	0.043	0.066
Number of chances	12	8
Number of big chances	0	0
Chances within the box	3	5



where goals are illustrated with a filled star symbol

- Question: How can I analyze SerieA for seasons 2015-2016 and 2021-2022?

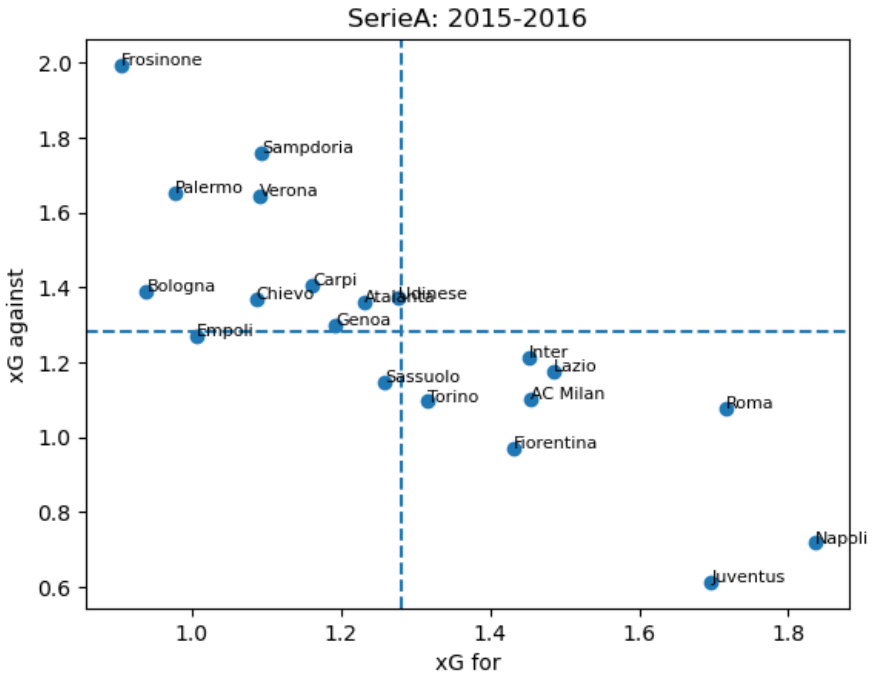
  Answer: Run run.py with the following input parameters

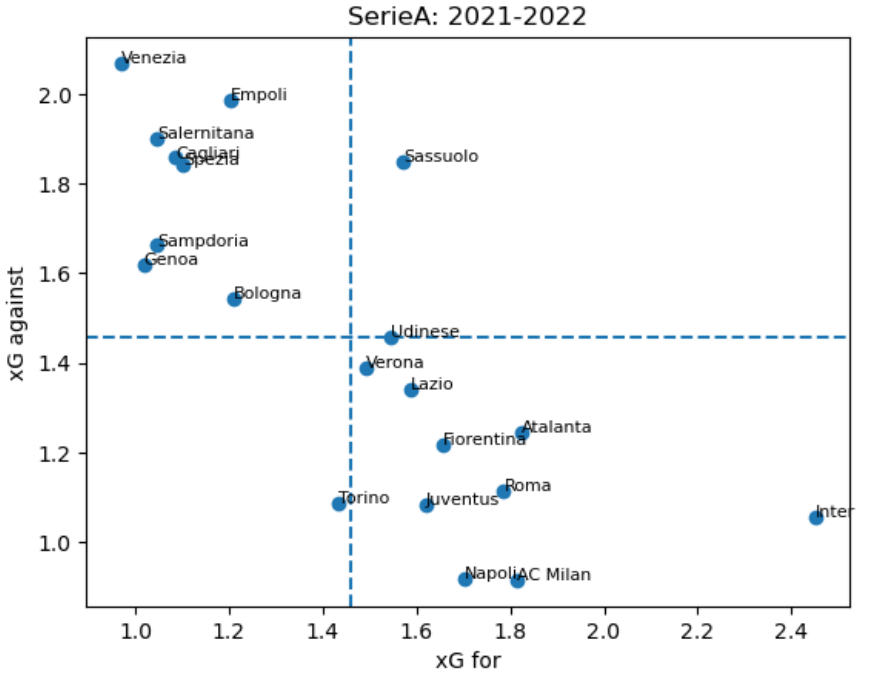
  Input:

```
update_id_file = False
get_data = False
analyze_match = False
analyze_league = True
analyze_team = False

save_dir_path = ""
league = "SerieA"
seasons = ["2015-2016","2021-2022"]
```

  Output:





- Question: How can I analyze the performance of Paris Saint Germain during the seasons 2014-2022?

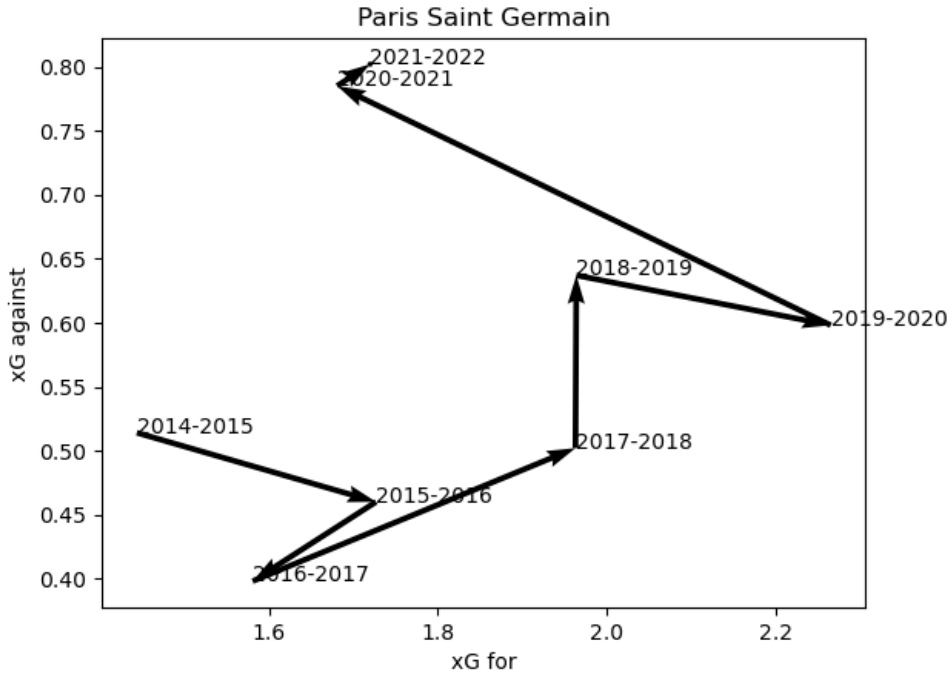
  Answer: Run run.py with the following input parameters

  Input:

```
update_id_file = False
get_data = False
analyze_match = False
analyze_league = False
analyze_team = True

save_dir_path = ""
team = "Paris Saint Germain"
seasons = ["2014-2015","2015-2016","2016-2017","2017-2018","2018-2019","2019-2020","2020-2021",
    "2021-2022"]
```

Output:



- Question: How can I analyze LaLiga's teams for the season 2020-2021?

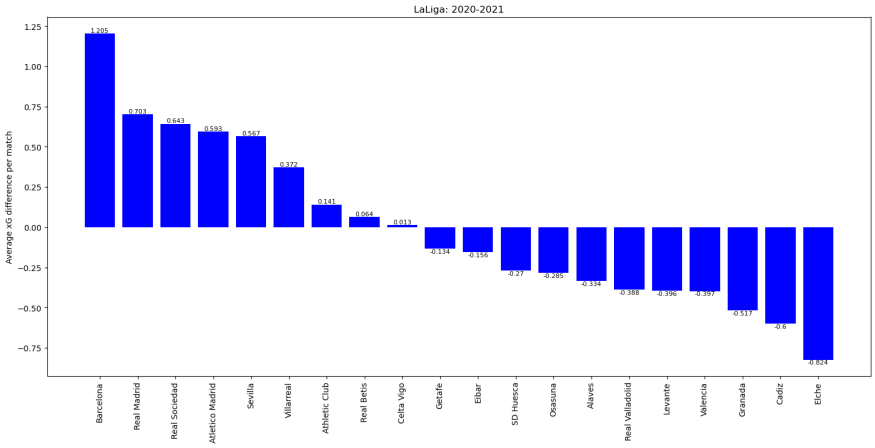
  Answer: Run run.py with the following input parameters

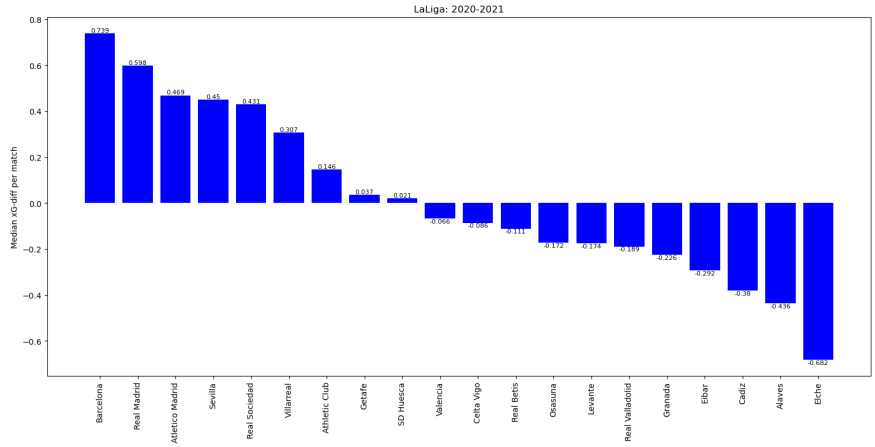
  Input:

```
update_id_file = False
get_data = False
analyze_match = False
analyze_league = True
analyze_team = False

save_dir_path = ""
league = "LaLiga"
seasons = ["2020-2021"]
```

  Output:





8 Frequently Asked Questions

- I have accidentally deleted Football_Data directory. Now what?

  No worries! Run run.py with the following input parameters

```
update_id_file = True
get_data = True
analyze_match = False
analyze_league = False
analyze_team = False

save_dir_path = ""
leagues = ["Ligue1","PremierLeague","Bundesliga","LaLiga","SerieA"]
seasons = ["2014-2015","2015-2016","2016-2017","2017-2018","2018-2019","2019-2020","2020-2021",
    "2021-2022","2022-2023"]
```

- I have accidentally deleted league_ids.dat file. Now what?

  No worries! Run run.py with the following input parameters

```
update_id_file = True
get_data = True
analyze_match = False
analyze_league = False
analyze_team = False

save_dir_path = ""
leagues = ["Ligue1","PremierLeague","Bundesliga","LaLiga","SerieA"]
seasons = ["2014-2015","2015-2016","2016-2017","2017-2018","2018-2019","2019-2020","2020-2021",
    "2021-2022","2022-2023"]
```

- I have accidentally deleted teams_dict.json file. Now what?

  You should download it again from the repository. The code does not run without this file