Most exciting European soccer league for fans

Guenter Wallnoefer

2019-02-23

Dataset

For this analysis, we used the soccer dataset that is available at https://www.kaggle.com/hugomathien/soccer

The dataset contains data related to

- European soccer leagues
- Teams in these leagues
- Players of these teams
- Matches from these teams in the seasons between 2008/09 and 2015/16

Motivation

From the perspective of a soccer fan, soccer is most exciting when a lot of equally strong teams compete to win a league. Leagues with one superior team that wins the league easily with a huge difference in points to the runner-ups are not very attractive for soccer fans.

Therefore, **the goal of this analysis is to find out which European top soccer league is most exciting and attractive for soccer fans based on the soccer dataset**.

From the perspective of an investor, this question might be also interesting. The most exciting league might also create the highest profits for investors because it attracts the highest number of fans.

Research Question(s)

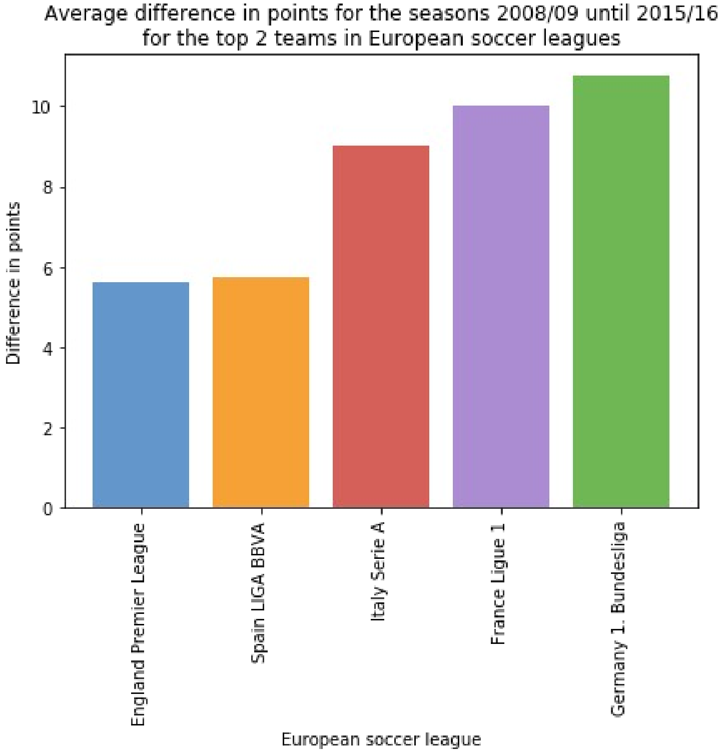
In order to determine the most exciting European soccer league, we define the following research question.

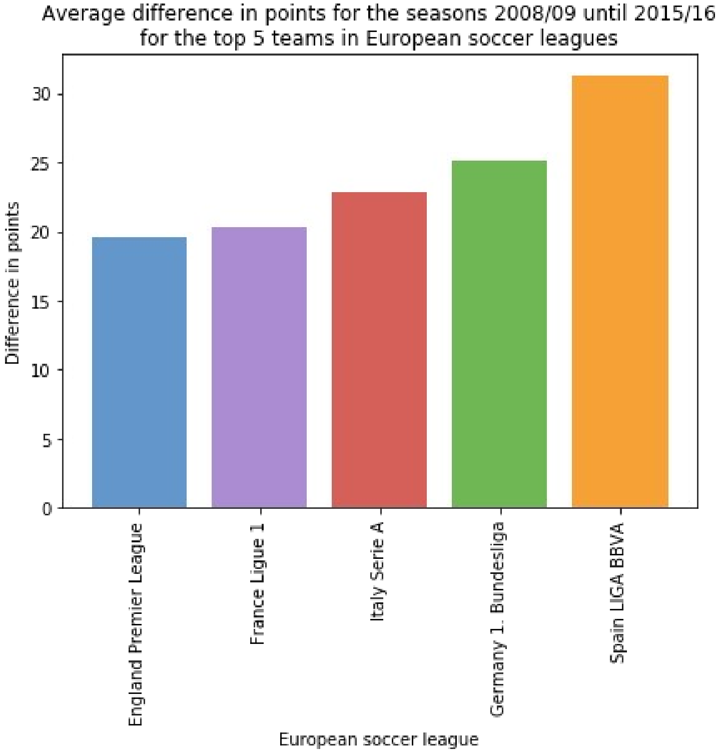
Research question: Which European top soccer league is characterized by the smallest difference in points of their top 2 teams and top 5 teams at the end of the season?

Basic assumptions:

- the smallest difference in points of the top 2 and top 5 teams at the end of the season is an indicator for the level of excitement for fans.
- European top soccer leagues include England Premier League, France Ligue 1, Germany 1. Bundesliga, Italy Serie A and Spain LIGA BBVA

Findings – Top 2 and Top 5 Teams

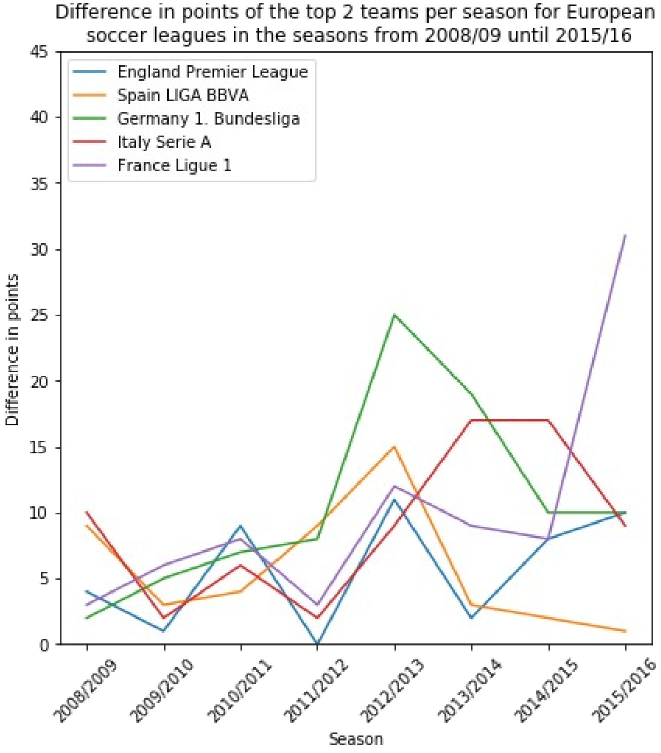




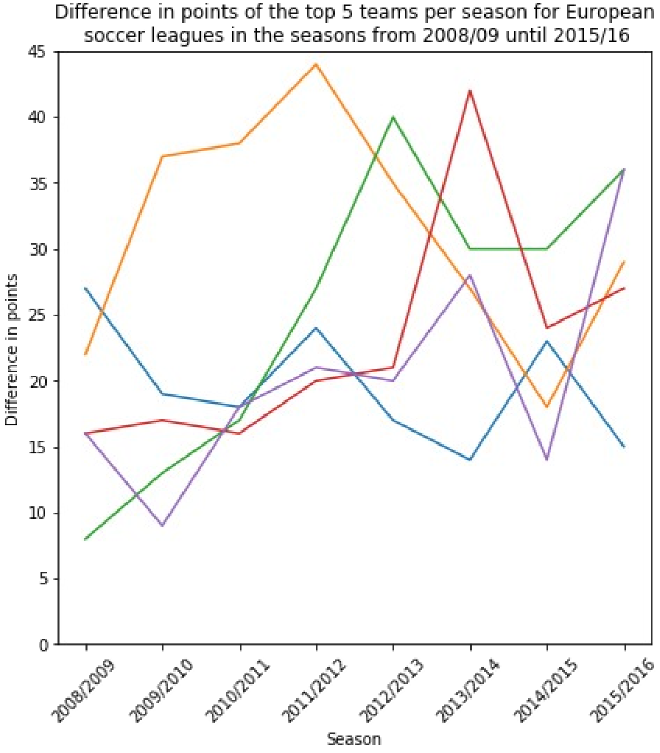
Findings - Top 2 and Top 5 Teams

- English Premier League has the lowest difference in points of the top 2 teams and of the top 5 teams, i.e. the opponents are closest to each other at the end of the season.
- German Bundesliga is characterized by a high difference of the top 2 and by a high difference of the top 5 teams.
- Italian Serie A shows average differences considering the top 2 and top 5 teams.
- Spanish Liga BBVA is characterized by only a little difference between the top 2 teams, but by a high difference between the top 5 teams.
- French Ligue 1 shows an inverse pattern compared to Spanish Liga BBVA. While the top 2 teams seem to be far apart, the top 5 teams are close in terms of difference of points.

Findings – Top 2 and Top 5 Teams over seasons



Difference in points of the top 2 teams per season for European soccer leagues in the seasons from 2008/09 until 2015/16



Difference in points of the top 5 teams per season for European soccer leagues in the seasons from 2008/09 until 2015/16

Findings - Top 2 and Top 5 Teams over seasons

- The difference between the top 2 teams of English Premier League was never more than 11 points.
- German Bundesliga and French Ligue 1 show remarkable differences for the top 2 teams for one seasons which may be considered as outliers (31 points for French ligue in season 2015/2016).
- While the difference between the top 2 teams seems to be quite constant for all leagues until season 2011/2012, only in Spain and England the difference stays constant over the entire period of our dataset. In Germany, Italy and France we may identify a trend for the top team to become more dominant since season 2012/2013.
- In England, the difference in points for the top 5 teams stay quite constant. In Germany, Italy and France, we observe a trend for a higher differences starting with season 2012/2013. However, in Spain we observe a contrary trend for the 5 top teams which seem to become closer to each other.

Answer to Research Question and Conclusions

Answer to Research question: English Premier league is characterized by the smallest difference in points of the top 2 teams and top 5 teams at the end of the season on average.

Therefore, we conclude that English Premier league can be considered to be the most exciting and attractive soccer league for fans and investors. Not only is it characterized by the least difference of points between the top 2 teams at the end of the seasons. Also, the top 5 teams seem to be able to compete more close with each others compared to other European soccer leagues.

References

[1]“Bundesliga,” Bundesliga. [Online]. Available: https://en.wikipedia.org/wiki/Bundesliga. [Accessed: 22-Feb-2019].

[2]“European Soccer Database,” European Soccer Database. [Online]. Available: https://www.kaggle.com/hugomathien/soccer. [Accessed: 15-Feb-2019].

[3]“La Liga,” La Liga. [Online]. Available: https://en.wikipedia.org/wiki/La_Liga. [Accessed: 22-Feb-2019].

[4]“Ligue 1,” Ligue 1. [Online]. Available: https://en.wikipedia.org/wiki/Ligue_1. [Accessed: 22-Feb-2019].

[5]“Most Powerful European Football (Soccer) Leagues,” Most Powerful European Football (Soccer) Leagues. [Online]. Available: https://www.thetoptens.com/powerful-european-football-leagues/. [Accessed: 22-Feb-2019].

[6]“Premier League,” Premier League. [Online]. Available: https://en.wikipedia.org/wiki/Premier_League. [Accessed: 22-Feb-2019].

[7]“Serie A,” Serie A. [Online]. Available: https://en.wikipedia.org/wiki/Serie_A. [Accessed: 22-Feb-2019].

Mini Project - UCSD - Python for Data Science

February 23, 2019

1 Mini Project

In this mini project, we focus on an analysis of the soccer dataset obtained from https://www.kaggle.com/hugomathien/soccer.

1.1 Dataset exploration

As a first step, we read the soccer dataset from the sqlite database and explore which tables and related columns are available for analysis.

Import libraries We import all needed libraries for exploration.

```
In [1]: import sqlite3
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import matplotlib.cm as cm
```

Create connection to database Next, we create a connection to the database. **IMPORTANT NOTE: If you run this code, please make sure to adapt the path to the dataset**

```
In [2]: path_2_dataset = '/home/gonte/LargeDatasets/soccer/'
        cnx = sqlite3.connect(path_2_dataset + 'database.sqlite')
```

List all available tables and columns We list all available tables in the database.

```
In [3]: df_tables = pd.read_sql_query("SELECT name, tbl_name FROM sqlite_master " +
                                      "WHERE type='table' AND name != 'sqlite_sequence';",
                                      cnx)
        df_tables
```

```
Out[3]:                  name           tbl_name
        0  Player_Attributes  Player_Attributes
        1             Player             Player
        2              Match              Match
        3             League             League
        4            Country            Country
```

```
5               Team               Team
6    Team_Attributes    Team_Attributes
```

Besides the player attributes, we find other tables containing match, league, country and team data. We have a look into all of these tables to get a grasp of the available columns.

```
In [4]: tbls = {}
        for table_name in df_tables.tbl_name:
            tbls[table_name] = pd.read_sql_query("SELECT * FROM " + table_name +
                                                 " LIMIT 5;", cnx)
```

We start by examining the tables that contain players data. In these tables, we find data about the age (birthday), height, weight, as well as ratings regarding play style, accuracy or penalties. (*gk_ prefix means goal keeper*)

```
In [5]: tbls["Player"]

Out[5]:    id  player_api_id         player_name  player_fifa_api_id  \
        0   1         505942  Aaron Appindangoye              218353
        1   2         155782     Aaron Cresswell              189615
        2   3         162549         Aaron Doran              186170
        3   4          30572       Aaron Galindo              140161
        4   5          23780        Aaron Hughes               17725

                      birthday  height  weight
        0  1992-02-29 00:00:00  182.88     187
        1  1989-12-15 00:00:00  170.18     146
        2  1991-05-13 00:00:00  170.18     163
        3  1982-05-08 00:00:00  182.88     198
        4  1979-11-08 00:00:00  182.88     154
```

```
In [6]: tbls["Player_Attributes"].columns

Out[6]: Index(['id', 'player_fifa_api_id', 'player_api_id', 'date', 'overall_rating',
               'potential', 'preferred_foot', 'attacking_work_rate',
               'defensive_work_rate', 'crossing', 'finishing', 'heading_accuracy',
               'short_passing', 'volleys', 'dribbling', 'curve', 'free_kick_accuracy',
               'long_passing', 'ball_control', 'acceleration', 'sprint_speed',
               'agility', 'reactions', 'balance', 'shot_power', 'jumping', 'stamina',
               'strength', 'long_shots', 'aggression', 'interceptions', 'positioning',
               'vision', 'penalties', 'marking', 'standing_tackle', 'sliding_tackle',
               'gk_diving', 'gk_handling', 'gk_kicking', 'gk_positioning',
               'gk_reflexes'],
              dtype='object')
```

```
In [7]: tbls["Player_Attributes"]

Out[7]:    id  player_fifa_api_id  player_api_id                 date  overall_rating  \
        0   1              218353         505942  2016-02-18 00:00:00              67
```

```
1   2              218353         505942  2015-11-19 00:00:00              67
2   3              218353         505942  2015-09-21 00:00:00              62
3   4              218353         505942  2015-03-20 00:00:00              61
4   5              218353         505942  2007-02-22 00:00:00              61

   potential preferred_foot attacking_work_rate defensive_work_rate  crossing  \
0         71          right              medium              medium        49
1         71          right              medium              medium        49
2         66          right              medium              medium        49
3         65          right              medium              medium        48
4         65          right              medium              medium        48

      ...       vision  penalties  marking  standing_tackle  sliding_tackle  \
0     ...           54         48       65               69              69
1     ...           54         48       65               69              69
2     ...           54         48       65               66              69
3     ...           53         47       62               63              66
4     ...           53         47       62               63              66

   gk_diving  gk_handling  gk_kicking  gk_positioning  gk_reflexes
0          6           11          10               8            8
1          6           11          10               8            8
2          6           11          10               8            8
3          5           10           9               7            7
4          5           10           9               7            7

[5 rows x 42 columns]
```

Then, we have a look into the team related tables. Also in this table, we find various characteristics and ratings of the teams.

```
In [8]: tbls["Team"]

Out[8]:    id  team_api_id  team_fifa_api_id     team_long_name team_short_name
        0   1         9987               673           KRC Genk             GEN
        1   2         9993               675        Beerschot AC             BAC
        2   3        10000             15005   SV Zulte-Waregem             ZUL
        3   4         9994              2007   Sporting Lokeren             LOK
        4   5         9984              1750  KSV Cercle Brugge             CEB

In [9]: tbls["Team_Attributes"].columns

Out[9]: Index(['id', 'team_fifa_api_id', 'team_api_id', 'date', 'buildUpPlaySpeed',
               'buildUpPlaySpeedClass', 'buildUpPlayDribbling',
               'buildUpPlayDribblingClass', 'buildUpPlayPassing',
               'buildUpPlayPassingClass', 'buildUpPlayPositioningClass',
               'chanceCreationPassing', 'chanceCreationPassingClass',
               'chanceCreationCrossing', 'chanceCreationCrossingClass',
               'chanceCreationShooting', 'chanceCreationShootingClass',
               'chanceCreationPositioningClass', 'defencePressure',
               'defencePressureClass', 'defenceAggression', 'defenceAggressionClass',
               'defenceTeamWidth', 'defenceTeamWidthClass',
               'defenceDefenderLineClass'],
              dtype='object')
```

```
In [10]: tbls["Team_Attributes"]
```

```
Out[10]:    id  team_fifa_api_id  team_api_id                 date  buildUpPlaySpeed  \
         0   1               434         9930  2010-02-22 00:00:00                60
         1   2               434         9930  2014-09-19 00:00:00                52
         2   3               434         9930  2015-09-10 00:00:00                47
         3   4                77         8485  2010-02-22 00:00:00                70
         4   5                77         8485  2011-02-22 00:00:00                47

           buildUpPlaySpeedClass  buildUpPlayDribbling buildUpPlayDribblingClass  \
         0              Balanced                   NaN                    Little
         1              Balanced                  48.0                    Normal
         2              Balanced                  41.0                    Normal
         3                  Fast                   NaN                    Little
         4              Balanced                   NaN                    Little

            buildUpPlayPassing buildUpPlayPassingClass           ...             \
         0                  50                   Mixed           ...
         1                  56                   Mixed           ...
         2                  54                   Mixed           ...
         3                  70                    Long           ...
         4                  52                   Mixed           ...

           chanceCreationShooting  chanceCreationShootingClass  \
         0                     55                       Normal
         1                     64                       Normal
         2                     64                       Normal
         3                     70                         Lots
         4                     52                       Normal

           chanceCreationPositioningClass  defencePressure defencePressureClass  \
         0                      Organised               50               Medium
         1                      Organised               47               Medium
         2                      Organised               47               Medium
         3                      Organised               60               Medium
         4                      Organised               47               Medium

            defenceAggression defenceAggressionClass defenceTeamWidth  \
         0                 55                  Press               45
         1                 44                  Press               54
         2                 44                  Press               54
         3                 70                 Double               70
```

```
4                  47                    Press                52

   defenceTeamWidthClass defenceDefenderLineClass
0                 Normal                    Cover
1                 Normal                    Cover
2                 Normal                    Cover
3                   Wide                    Cover
4                 Normal                    Cover

[5 rows x 25 columns]
```

The matches table holds data about the league, season, stage, the specific players that participated in the matches, about odds from various betting brands and the goals scored in the matches.

```
In [11]: tbls["Match"].columns

Out[11]: Index(['id', 'country_id', 'league_id', 'season', 'stage', 'date',
                'match_api_id', 'home_team_api_id', 'away_team_api_id',
                'home_team_goal',
                ...
                'SJA', 'VCH', 'VCD', 'VCA', 'GBH', 'GBD', 'GBA', 'BSH', 'BSD', 'BSA'],
               dtype='object', length=115)

In [12]: tbls["Match"]

Out[12]:    id  country_id  league_id     season  stage                 date  \
         0   1           1          1  2008/2009      1  2008-08-17 00:00:00
         1   2           1          1  2008/2009      1  2008-08-16 00:00:00
         2   3           1          1  2008/2009      1  2008-08-16 00:00:00
         3   4           1          1  2008/2009      1  2008-08-17 00:00:00
         4   5           1          1  2008/2009      1  2008-08-16 00:00:00

            match_api_id  home_team_api_id  away_team_api_id  home_team_goal  ...   \
         0        492473              9987              9993               1  ...
         1        492474             10000              9994               0  ...
         2        492475              9984              8635               0  ...
         3        492476              9991              9998               5  ...
         4        492477              7947              9985               1  ...

             SJA   VCH   VCD   VCA   GBH   GBD   GBA   BSH   BSD   BSA
         0  4.00  1.65  3.40  4.50  1.78  3.25  4.00  1.73  3.40  4.20
         1  3.80  2.00  3.25  3.25  1.85  3.25  3.75  1.91  3.25  3.60
         2  2.50  2.35  3.25  2.65  2.50  3.20  2.50  2.30  3.20  2.75
         3  7.50  1.45  3.75  6.50  1.50  3.75  5.50  1.44  3.75  6.50
         4  1.73  4.50  3.40  1.65  4.50  3.50  1.65  4.75  3.30  1.67

         [5 rows x 115 columns]
```

Finally, the country and league table contain only the name column.

```
In [13]: tbls["Country"]

Out[13]:       id     name
         0      1  Belgium
         1   1729  England
         2   4769   France
         3   7809  Germany
         4  10257    Italy

In [14]: tbls["League"]

Out[14]:       id  country_id                      name
         0      1           1    Belgium Jupiler League
         1   1729        1729    England Premier League
         2   4769        4769            France Ligue 1
         3   7809        7809     Germany 1. Bundesliga
         4  10257       10257             Italy Serie A
```

1.2 Research question

Personal motivation As a soccer fan, I'm personally primarily interested in an exciting soccer season that is characterized by various strong teams that are able to compete with each other on a similar level. I don't like boring seasons in which teams win their leagues with a huge difference in points, i.e. the winner is fixed long before the season ends. Therefore, I would be interested to find out which league is characterized by the most excitement for fans and the least variance in performance by the top teams.

Definition of research question and variables In order to determine the most exciting European soccer league for fans, we come up with the research question below.

Research question: ***Which European top soccer league is characterized by the smallest difference in points of their top 2 teams and top 5 teams at the end of the season?***

For our purposes, we claim that

- the smallest difference in points of the top 2 and top 5 teams at the end of the season is an indicator for the level of excitement for fans.
- European top soccer leagues include England Premier League, France Ligue 1, Germany 1. Bundesliga, Italy Serie A and Spain LIGA BBVA

Of course, this could be a point for discussion. :-)

1.3 Solution approach

From the metadata investigation above, we see that we have data about matches from various leagues and seasons. As a consequence, we can determine the final standings for each season by calculating the gained points per match and aggregate them per season and league. Then, we can determine the difference and variance of the top 5 teams in the final standings and compare them per season and country.

1.4 Data preparation

Get the data from the database For our analysis, we need the match, team and league tables. We read these tables into dataframes. We filter the matches table to only include matches from the specified 5 top leagues. In addition, we only select the columns that we are interested in, so that we don't have to drop them fro the dataframe afterwards.

```
In [15]: matches = pd.read_sql_query("SELECT league_id, season, stage, home_team_api_id, " +
                                     "away_team_api_id, home_team_goal, away_team_goal " +
                                     "FROM match WHERE league_id IN (1729, 7809, 4769,  " +
                                     "10257, 21518)", cnx)
         leagues = pd.read_sql_query("SELECT * FROM league", cnx)
         teams = pd.read_sql_query("SELECT * FROM team", cnx)
```

The resulting dataset contains 14,585 rows.

```
In [16]: matches.shape

Out[16]: (14585, 7)
```

Data transformation First, we join the datasets to a single dataframe for our analysis.

```
In [17]: df = matches.merge(leagues, left_on="league_id", right_on="id")
```

Next, we drop the unneeded columns.

```
In [18]: df.drop(["league_id", "id", "country_id"], axis=1, inplace=True)
```

And we rename the column "name" to something more meaningful.

```
In [19]: df.rename(columns = { 'name': 'league'}, inplace=True)

In [20]: df.head()

Out[20]:       season  stage  home_team_api_id  away_team_api_id  home_team_goal  \
         0  2008/2009      1             10260             10261               1
         1  2008/2009      1              9825              8659               1
         2  2008/2009      1              8472              8650               0
         3  2008/2009      1              8654              8528               2
         4  2008/2009      1             10252              8456               4

            away_team_goal                  league
         0               1  England Premier League
         1               0  England Premier League
         2               1  England Premier League
         3               1  England Premier League
         4               2  England Premier League
```

As a next step, we calculate the points for each math that are counted towards the league standings based on the goals scored. For each win, a team gains 3 points. And for a draw, a team gets 1 point.

```
In [21]: def determine_hometeam_points(r):
             if r["home_team_goal"] > r["away_team_goal"]:
                 return 3
             elif r["home_team_goal"] == r["away_team_goal"]:
                 return 1
             else:
                 return 0
```

```
In [22]: df["home_team_points"] = df.apply(determine_hometeam_points, axis = 1)
```

```
In [23]: def determine_awayteam_points(r):
             if r["home_team_goal"] < r["away_team_goal"]:
                 return 3
             elif r["home_team_goal"] == r["away_team_goal"]:
                 return 1
             else:
                 return 0
```

```
In [24]: df["away_team_points"] = df.apply(determine_awayteam_points, axis = 1)
```

In order to aggregate the table in the correct way, we have to find a way to "unpivot" the two columns for the points for the two teams. We achieve this by creating a copy of the dataframe. For the original dataframe, we create new columns with the points for the home team. For the copy of the dataframe, we take the data of the away team. Then, we drop unused columns and append the copy to the original dataframe to create a final dataframe for analysis.

```
In [25]: # copy
         df2 = df.copy()

         # add new columns with data from the home team for the first dataframe
         df["team_id"] = df["home_team_api_id"]
         df["points"] = df["home_team_points"]

         # we do the same for the away teams
         df2["team_id"] = df2["away_team_api_id"]
         df2["points"] = df2["away_team_points"]

         # then we merge the two dataframes
         df = pd.concat([df, df2])
```

We check two see if we now have the correct number of rows.

```
In [26]: df.shape
```

```
Out[26]: (29170, 11)
```

Then, we join the team table to have the team names included into our dataframe.

```
In [27]: df = df.merge(teams, left_on="team_id", right_on="team_api_id")
In [28]: df.drop(["home_team_api_id", "away_team_api_id", "home_team_goal",
                  "away_team_goal", "home_team_points", "away_team_points",
                  "team_api_id", "id", "team_fifa_api_id",
                  "team_short_name", "team_id"], axis=1, inplace=True)
```

We check the structure of the resulting dataframe.

```
In [29]: df.head()
```

```
Out[29]:       season  stage                 league  points     team_long_name
         0  2008/2009      1  England Premier League       1  Manchester United
         1  2008/2009     10  England Premier League       3  Manchester United
         2  2008/2009     11  England Premier League       3  Manchester United
         3  2008/2009     13  England Premier League       3  Manchester United
         4  2008/2009     16  England Premier League       3  Manchester United
```

As a next step we group the data by by season, league and team and sum up the points.

```
In [30]: standings = pd.DataFrame(df.groupby(["league", "season",
                                              "team_long_name"]).points.sum())
```

We reset the index of the resulting dataframe to get rid of the MultiIndex and have normal columns instead.

```
In [31]: standings.reset_index(inplace=True)
```

We sort the dataframe to get the standings in order.

```
In [32]: standings = standings.sort_values(by="points",
             ascending=False).sort_values(by=["league", "season"])
```

Check if data is correct To check our transformation procedure, we compare the results with the real results by picking two samples. First, we check the English Premier League season of 2012/13 with the data from compare to https://en.wikipedia.org/wiki/2012%E2%80%9313_Premier_League#League_table.

```
In [33]: standings[ (standings.league == "England Premier League") &
                    (standings.season == "2012/2013")]
```

```
Out[33]:                   league     season        team_long_name  points
         87  England Premier League  2012/2013     Manchester United      89
         86  England Premier League  2012/2013       Manchester City      78
         82  England Premier League  2012/2013               Chelsea      75
         80  England Premier League  2012/2013               Arsenal      73
         96  England Premier League  2012/2013     Tottenham Hotspur      72
         83  England Premier League  2012/2013               Everton      63
         85  England Premier League  2012/2013             Liverpool      61
         97  England Premier League  2012/2013  West Bromwich Albion      49
         98  England Premier League  2012/2013       West Ham United      46
```

```
95  England Premier League  2012/2013          Swansea City      46
89  England Premier League  2012/2013          Norwich City      44
84  England Premier League  2012/2013                Fulham      43
93  England Premier League  2012/2013            Stoke City      42
81  England Premier League  2012/2013           Aston Villa      41
88  England Premier League  2012/2013      Newcastle United      41
92  England Premier League  2012/2013           Southampton      41
94  England Premier League  2012/2013            Sunderland      39
99  England Premier League  2012/2013        Wigan Athletic      36
91  England Premier League  2012/2013               Reading      28
90  England Premier League  2012/2013   Queens Park Rangers      25
```

Then, we check the data with the data from https://en.wikipedia.org/wiki/2011%E2%80%9312_Bundesliga#

```
In [34]: standings[ (standings.league == "Germany 1. Bundesliga") &
                    (standings.season == "2011/2012")]
```

```
Out[34]:                     league     season             team_long_name  points
         379  Germany 1. Bundesliga  2011/2012          Borussia Dortmund      81
         382  Germany 1. Bundesliga  2011/2012           FC Bayern Munich      73
         383  Germany 1. Bundesliga  2011/2012             FC Schalke 04      64
         380  Germany 1. Bundesliga  2011/2012  Borussia Mönchengladbach      60
         378  Germany 1. Bundesliga  2011/2012        Bayer 04 Leverkusen      54
         390  Germany 1. Bundesliga  2011/2012              VfB Stuttgart      53
         385  Germany 1. Bundesliga  2011/2012                Hannover 96      48
         391  Germany 1. Bundesliga  2011/2012              VfL Wolfsburg      44
         388  Germany 1. Bundesliga  2011/2012          SV Werder Bremen      42
         376  Germany 1. Bundesliga  2011/2012             1. FC Nürnberg      42
         389  Germany 1. Bundesliga  2011/2012        TSG 1899 Hoffenheim      41
         387  Germany 1. Bundesliga  2011/2012                SC Freiburg      40
         377  Germany 1. Bundesliga  2011/2012            1. FSV Mainz 05      39
         381  Germany 1. Bundesliga  2011/2012                FC Augsburg      38
         384  Germany 1. Bundesliga  2011/2012               Hamburger SV      36
         386  Germany 1. Bundesliga  2011/2012          Hertha BSC Berlin      31
         375  Germany 1. Bundesliga  2011/2012                1. FC Köln      30
         374  Germany 1. Bundesliga  2011/2012       1. FC Kaiserslautern      23
```

Both checks succeeded. The data in our transormed dataframe seems to be correct.

1.5 Results of explorations

Determine Top 5 teams and difference in points After we prepared the dataframe, we determine the top 2 and top 5 teams per season and league.

```
In [35]: top2 = pd.DataFrame(standings.groupby(["league", "season"]).head(2))
         top5 = pd.DataFrame(standings.groupby(["league", "season"]).head(5))
```

First, we create the difference in points between the top two teams by subtracting the values.

```
In [36]: top2_diff = np.array(top2.iloc[0:-1, 3]) - np.array(top2.iloc[1:, 3] )
         top2.loc[:,"diff"] = np.append(top2_diff, 0)
```

We ignore every second line and re-assign the dataframe.

```
In [37]: top2 = top2.loc[ ::2, :]
```

Then, we repeat the procedure for the top 5 teams.

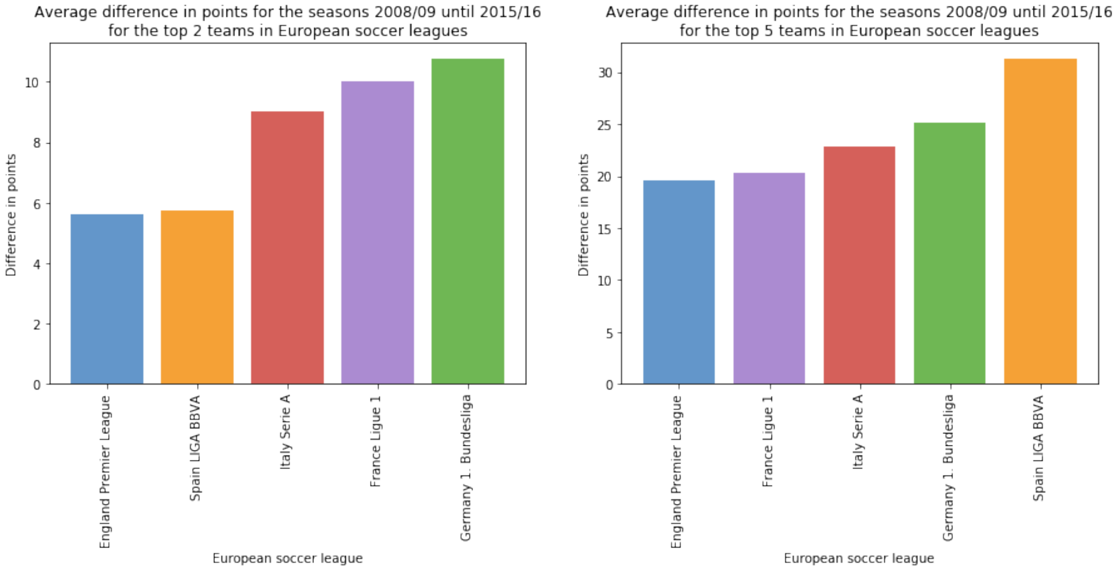
```
In [38]: top5_diff = np.array(top5.iloc[0:-4, 3]) - np.array(top5.iloc[4:, 3] )
         top5.loc[:,"diff"] = np.append(top5_diff, np.zeros(4))
         top5 = top5.loc[ ::5, :]
```

Average difference per league First, we have a look at the average difference between the top teams grouped by league while ignoring the season.

```
In [39]: top2_league = pd.DataFrame(top2.groupby("league").diff.mean().
                                    sort_values(ascending=True))
         top2_league.reset_index(inplace=True)
         top2_league

Out[39]:                    league    diff
         0  England Premier League   5.625
         1         Spain LIGA BBVA   5.750
         2           Italy Serie A   9.000
         3          France Ligue 1  10.000
         4   Germany 1. Bundesliga  10.750
```

```
In [40]: top5_league = pd.DataFrame(top5.groupby("league").diff.mean().
                                    sort_values(ascending=True))
         top5_league.reset_index(inplace=True)
         top5_league

Out[40]:                    league    diff
         0  England Premier League  19.625
         1          France Ligue 1  20.250
         2           Italy Serie A  22.875
         3   Germany 1. Bundesliga  25.125
         4         Spain LIGA BBVA  31.250
```

```
In [41]: %matplotlib inline

         fig, ax = plt.subplots(1, 2, figsize=(15, 5))

         ax[1].bar(top5_league["league"], top5_league["diff"],
                   color=["#6396ca", "#ab8bd1", "#d5605a", "#6fb754", "#f5a136"])
         ax[1].set_ylabel("Difference in points")
         ax[1].set_xlabel("European soccer league")
         ax[1].set_title("Average difference in points for the seasons 2008/09" +
                         " until 2015/16\nfor the top 5 teams in European soccer leagues")
         ax[1].set_xticklabels(top5_league["league"], rotation=90)

         ax[0].bar(top2_league["league"], top2_league["diff"],
                   color=["#6396ca", "#f5a136", "#d5605a", "#ab8bd1", "#6fb754"])
         ax[0].set_ylabel("Difference in points")
         ax[0].set_xlabel("European soccer league")
         ax[0].set_title("Average difference in points for the seasons 2008/09 " +
                         "until 2015/16\nfor the top 2 teams in European soccer leagues")
         ax[0].set_xticklabels(top2_league["league"], rotation=90)

         plt.show()
```



The figures show some interesting facts.

- English Premier League has the lowest difference in points between the top 2 and top 5 teams, i.e. the oponents are closest to each other at the end of the season.
- German Bundesliga is characterized by a high difference between the top 2 and top 5 teams.
- Italian Serie A shows average differences considering the top 2 and top 5 teams.
- Spanish Liga BBVA is characterized by a little difference between the top 2 teams, but by a high difference between the top 5 teams.
- French Ligue 1 shows an inversed pattern compared to Spanish Liga BBVA. While the top 2 teams seem to be far apart, the top 5 teams are close in terms of difference of points.

Average difference per league and season

```
In [42]: top2_league_season = pd.DataFrame(top2.groupby(["league", "season"]).diff.mean())
         top2_league_season.reset_index(inplace=True)
```

```
In [43]: top2_league_season_t = pd.DataFrame({
             "England Premier League": list(top2_league_season.loc[
                 top2_league_season.league == "England Premier League", "diff" ]),
             "Spain LIGA BBVA": list(top2_league_season.loc[
                 top2_league_season.league == "Spain LIGA BBVA", "diff" ]),
             "Germany 1. Bundesliga": list(top2_league_season.loc[
                 top2_league_season.league == "Germany 1. Bundesliga", "diff" ]),
             "Italy Serie A": list(top2_league_season.loc[
                 top2_league_season.league == "Italy Serie A", "diff" ]),
             "France Ligue 1": list(top2_league_season.loc[
                 top2_league_season.league == "France Ligue 1", "diff" ])
         }, index = np.unique(top2_league_season.loc[
             top2_league_season.league == "England Premier League", "season" ]))
         top2_league_season_t
```

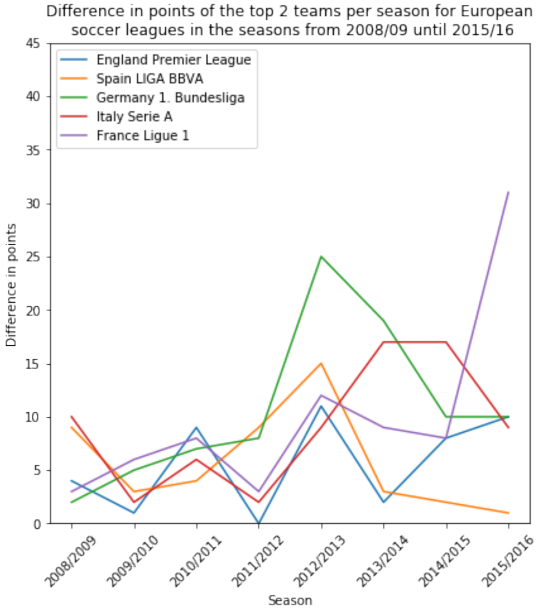
```
Out[43]:            England Premier League  Spain LIGA BBVA  Germany 1. Bundesliga  \
         2008/2009                       4                9                      2
         2009/2010                       1                3                      5
         2010/2011                       9                4                      7
         2011/2012                       0                9                      8
         2012/2013                      11               15                     25
         2013/2014                       2                3                     19
         2014/2015                       8                2                     10
         2015/2016                      10                1                     10

                    Italy Serie A  France Ligue 1
         2008/2009             10               3
         2009/2010              2               6
         2010/2011              6               8
         2011/2012              2               3
         2012/2013              9              12
         2013/2014             17               9
         2014/2015             17               8
         2015/2016              9              31
```

```
In [44]: top5_league_season = pd.DataFrame(top5.groupby(["league", "season"]).diff.mean())
         top5_league_season.reset_index(inplace=True)
```

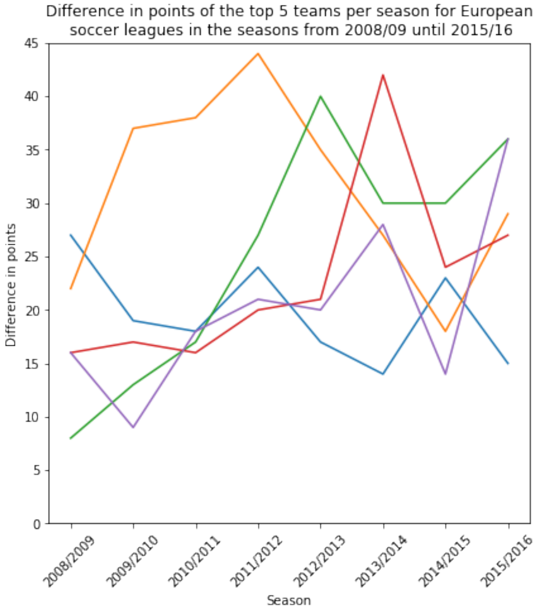
```
In [45]: top5_league_season_t = pd.DataFrame({
             "England Premier League": list(top5_league_season.loc[
                 top5_league_season.league == "England Premier League", "diff" ]),
             "Spain LIGA BBVA": list(top5_league_season.loc[
                 top5_league_season.league == "Spain LIGA BBVA", "diff" ]),
             "Germany 1. Bundesliga": list(top5_league_season.loc[
                 top5_league_season.league == "Germany 1. Bundesliga", "diff" ]),
             "Italy Serie A": list(top5_league_season.loc[
                 top5_league_season.league == "Italy Serie A", "diff" ]),
             "France Ligue 1": list(top5_league_season.loc[
        top5_league_season.league == "France Ligue 1", "diff" ])
}, index = np.unique(top5_league_season.loc[
    top5_league_season.league == "England Premier League", "season" ]))
top5_league_season_t
```

```
Out[45]:            England Premier League  Spain LIGA BBVA  Germany 1. Bundesliga  \
         2008/2009                    27.0             22.0                    8.0
         2009/2010                    19.0             37.0                   13.0
         2010/2011                    18.0             38.0                   17.0
         2011/2012                    24.0             44.0                   27.0
         2012/2013                    17.0             35.0                   40.0
         2013/2014                    14.0             27.0                   30.0
         2014/2015                    23.0             18.0                   30.0
         2015/2016                    15.0             29.0                   36.0

                    Italy Serie A  France Ligue 1
         2008/2009           16.0            16.0
         2009/2010           17.0             9.0
         2010/2011           16.0            18.0
         2011/2012           20.0            21.0
         2012/2013           21.0            20.0
         2013/2014           42.0            28.0
         2014/2015           24.0            14.0
         2015/2016           27.0            36.0
```

```
In [46]: fig, ax = plt.subplots(1, 2,figsize=(15,7))

         ax[0].plot(top2_league_season_t)
         ax[0].set_ylim(0, 45)
         ax[0].set_ylabel("Difference in points")
         ax[0].set_xlabel("Season")
         ax[0].set_title("Difference in points of the top 2 teams per season " +
                         "for European\n soccer leagues in the seasons from " +
                         "2008/09 until 2015/16")
         ax[0].legend(top2_league_season_t.columns, loc ="upper left")
         ax[0].set_xticklabels(top2_league_season_t.index, rotation=45)

         ax[1].plot(top5_league_season_t)
         ax[1].set_ylim(0, 45)
         ax[1].set_ylabel("Difference in points")
         ax[1].set_xlabel("Season")
         ax[1].set_title("Difference in points of the top 5 teams per season " +
                         "for European\n soccer leagues in the seasons from " +
                         "2008/09 until 2015/16")
         # ax[1].legend(top5_league_season_t.columns, loc ="upper left")
         ax[1].set_xticklabels(top5_league_season_t.index, rotation=45)

         plt.show()
```



Difference in points of the top 2 teams per season for European soccer leagues in the seasons from 2008/09 until 2015/16



Difference in points of the top 5 teams per season for European soccer leagues in the seasons from 2008/09 until 2015/16

The comparison of the difference in points between different seasons allow us a more detailed inspection of the different leagues.

- The difference between the top 2 teams of English Premier League was never more than 11 points.
- German Bundesliga and French Ligue 1 show remarkable differences for the top 2 teams for one seasons which may be considered as outliers (31 points for French ligue in season 2015/2016).
- While the difference between the top 2 teams seems to be quite constant for all leagues until season 2011/2012, only in Spain and England the difference stays constant over the entire period of our dataset. In Germany, Italy and France we may identify a trend for the top team to become more dominant since season 2012/2013.
- In England, the difference in points for the top 5 teams stay quite constant. In Germany, Italy and France, we observe a trend for a higher difference starting with season 2012/2013. However, in Spain we observe a contrary trend for the 5 top teams which seem to become closer to each other.

1.6 Findings

Based on our dataset exploration, we may draw the following conclusions.

- English Premier league may be considered to be the most exciting soccer league for fans. Not only is it characterized by the least difference of points between the top 2 teams at the end of the seasons. Also, the top 5 teams seem to be able to compete more close with each others compared to other European soccer leagues. Therefore, English Premier league might be considered the most attractive league for soccer fans because not onl the top 2 teams are able to compete with each other on the same level, but also the top 5 teams.

- Spanish Liga BBVA is characterized by a constant small difference of points between the top 2 teams. On the contrary, Spanish liga BBVA show the widest spread in points for the top 5 teams. As a consequence, Spanish Liga BBVA might be considered to be exciting for fans of the two top teams, but not so much for ither fans.
- Italian Serie A may be classified as an average exciting league.
- French Ligue 1 may be also considered to be an average exciting league. However, the final standings of 2015/2016 show a significant outlier in the difference of points between the top 2 teams. However, runner-up teams in the top 5 seem to be quite close to each other.
- German Bundesliga may be classified as the least exciting soccer league in Europe.

We conclude that from the perspective of a fan it might be most exciting to focus on English Premier league, because the top teams seem to be on an equal level. This finding might also be interesting for investors who want to invest in the most attractive soccer league.

1.6.1 *Additional comment*

I found a website with a survey for the most powerful soccer league in Europe as voted by fans. https://www.thetoptens.com/powerful-european-football-leagues/

The results are the same as in our analysis except for German Bundesliga which was ranked Nr. 3 in this ranking.