investigate-a-soccer-dataset

May 30, 2021

1 Project: Investigate a Soccer Dataset

1.1 Table of Contents

Introduction

> **In this project i will investigate a soccer database, which include data for soccer matches, players, and teams from some leading European football countries during the 8 year-period, from 2008 to 2016.**
>
> The questions I asked:

- *Which team scores more goals on average per match in this period? What league are they from?*
- *In which league scores more goals on average per season in this period?*
- *What is the tendency of the goals per season in top 5 leagues in a certain period of time?*

> **Before starting investigations, i downloaded DB Browser for SQLite and installed dataset. Then i started researching it. In Data Wrangling section i will give you some information about certain dataset tables which i will need.**

1.1.1 Packages i need

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
```

Data Wrangling

1.1.2 General Properties

```
[2]: ##Ready tables from dataset
df_league = pd.read_csv('Soccer DataBase csv_file\League.csv')
df_match = pd.read_csv('Soccer DataBase csv_file\Match.csv')
df_team = pd.read_csv('Soccer DataBase csv_file\Team.csv')

##Tables which i prepared with SQL queries. The codes will soon
goals_tendency = pd.read_csv('number_of_goals_in_diff.
 ↪_leagues_in_a_certain_period_of_time.csv')
avg_league_goals = pd.read_csv('avg_goals_in_diff_leag_in_cert_period_of_time.
 ↪csv')
```

1.1.3 A brief overview of the data.

```
[3]: ##Show the first 5 rows of DataFrame(df)
df_league.head()
```

```
[3]:       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
```

```
[4]: df_league.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          11 non-null     int64
 1   country_id  11 non-null     int64
 2   name        11 non-null     object
dtypes: int64(2), object(1)
memory usage: 392.0+ bytes
```

```
[5]: ##Show the first 5 rows of DataFrame(df)
df_match.head()
```

```
[5]:    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
```

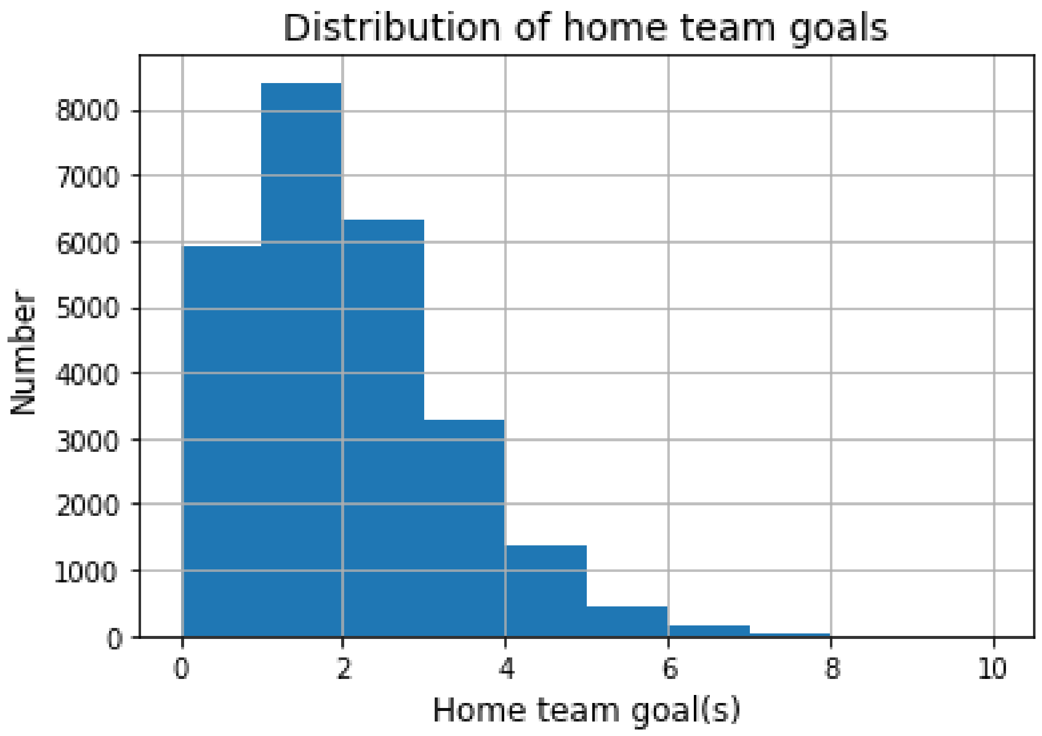
```
   match_api_id  home_team_api_id  away_team_api_id  home_team_goal  …  \
0        492473              9987              9993               1  …
1        492474             10000              9994               0  …
2        492475              9984              8635               0  …
3        492476              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]
```

[6]:
```
df_match.describe()
```

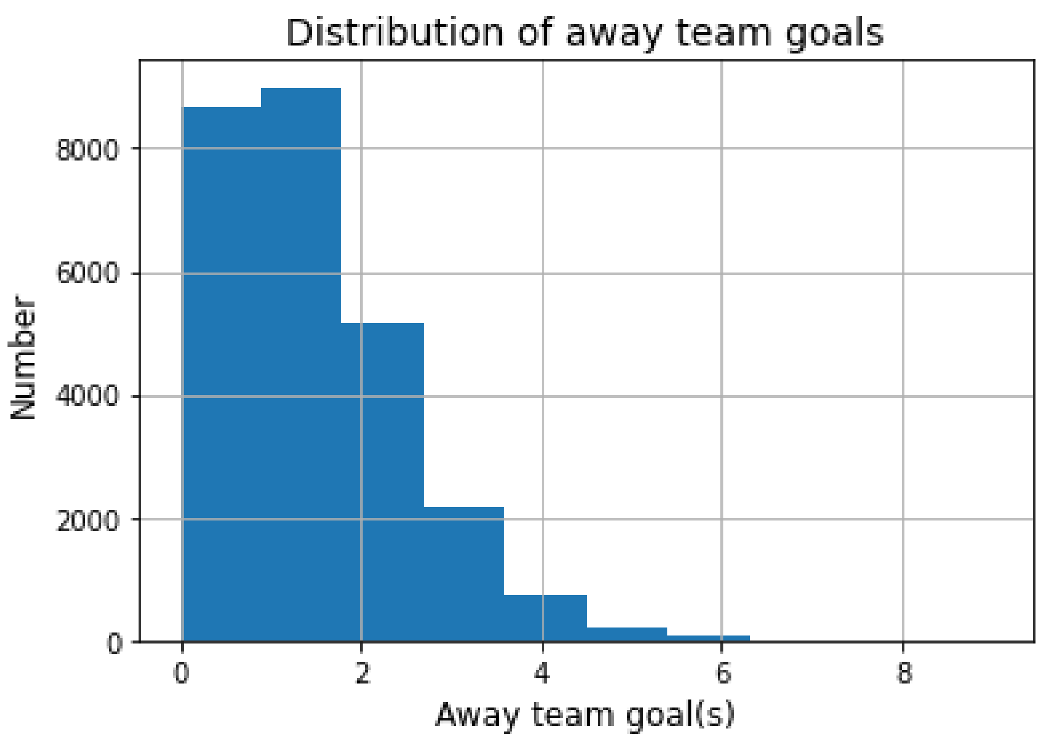
[6]:
```
                 id    country_id     league_id         stage  match_api_id  \
count  25979.000000  25979.000000  25979.000000  25979.000000  2.597900e+04
mean   12990.000000  11738.630317  11738.630317     18.242773  1.195429e+06
std     7499.635658   7553.936759   7553.936759     10.407354  4.946279e+05
min        1.000000      1.000000      1.000000      1.000000  4.831290e+05
25%     6495.500000   4769.000000   4769.000000      9.000000  7.684365e+05
50%    12990.000000  10257.000000  10257.000000     18.000000  1.147511e+06
75%    19484.500000  17642.000000  17642.000000     27.000000  1.709852e+06
max    25979.000000  24558.000000  24558.000000     38.000000  2.216672e+06

       home_team_api_id  away_team_api_id  home_team_goal  away_team_goal  \
count      25979.000000      25979.000000    25979.000000    25979.000000
mean        9984.371993       9984.475115        1.544594        1.160938
std        14087.453758      14087.445135        1.297158        1.142110
min         1601.000000       1601.000000        0.000000        0.000000
25%         8475.000000       8475.000000        1.000000        0.000000
50%         8697.000000       8697.000000        1.000000        1.000000
75%         9925.000000       9925.000000        2.000000        2.000000
max       274581.000000     274581.000000       10.000000        9.000000

       home_player_X1  …           SJA           VCH           VCD  \
count    24158.000000  …  17097.000000  22568.000000  22568.000000
mean         0.999586  …      4.622343      2.668107      3.899048
std          0.022284  …      3.632164      1.928753      1.248221
min          0.000000  …      1.100000      1.030000      1.620000
25%          1.000000  …      2.500000      1.700000      3.300000
50%          1.000000  …      3.500000      2.150000      3.500000
75%          1.000000  …      5.250000      2.800000      4.000000
max          2.000000  …     41.000000     36.000000     26.000000
```

```
                VCA           GBH           GBD           GBA           BSH  \
count  22568.000000  14162.000000  14162.000000  14162.000000  14161.000000
mean       4.840281      2.498764      3.648189      4.353097      2.497894
std        4.318338      1.489299      0.867440      3.010189      1.507793
min        1.080000      1.050000      1.450000      1.120000      1.040000
25%        2.550000      1.670000      3.200000      2.500000      1.670000
50%        3.500000      2.100000      3.300000      3.400000      2.100000
75%        5.400000      2.650000      3.750000      5.000000      2.620000
max       67.000000     21.000000     11.000000     34.000000     17.000000

                BSD           BSA
count  14161.000000  14161.000000
mean       3.660742      4.405663
std        0.868272      3.189814
min        1.330000      1.120000
25%        3.250000      2.500000
50%        3.400000      3.400000
75%        3.750000      5.000000
max       13.000000     34.000000

[8 rows x 105 columns]
```

[7]:
```
ax = df_match['home_team_goal'].hist()
ax.set_xlabel('Home team goal(s)', fontsize = 12)
ax.set_ylabel('Number', fontsize = 12)
ax.set_title('Distribution of home team goals', fontsize = 14);
```



[8]:
```
ax = df_match['away_team_goal'].hist()
ax.set_xlabel('Away team goal(s)', fontsize = 12)
ax.set_ylabel('Number', fontsize = 12)
ax.set_title('Distribution of away team goals', fontsize = 14);
```



[9]:
```
##Show the first 5 rows of DataFrame
df_team.head()
```

[9]:
```
   id  team_api_id  team_fifa_api_id     team_long_name team_short_name
0   1         9987             673.0           KRC Genk             GEN
1   2         9993             675.0       Beerschot AC             BAC
2   3        10000           15005.0   SV Zulte-Waregem             ZUL
3   4         9994            2007.0   Sporting Lokeren             LOK
4   5         9984            1750.0  KSV Cercle Brugge             CEB
```

[10]:
```
df_team.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   id                299 non-null    int64
 1   team_api_id       299 non-null    int64
 2   team_fifa_api_id  288 non-null    float64
 3   team_long_name    299 non-null    object
 4   team_short_name   299 non-null    object
dtypes: float64(1), int64(2), object(2)
memory usage: 11.8+ KB
```

```
[11]: df_team[df_team.team_fifa_api_id.isnull()]
```

```
[11]:         id  team_api_id  team_fifa_api_id                  team_long_name  \
      8        9         7947               NaN                    FCV Dender EH
      14      15         4049               NaN                           Tubize
      170  26561         6601               NaN                      FC Volendam
      204  34816       177361               NaN  Termalica Bruk-Bet Nieciecza
      208  35286         7992               NaN                         Trofense
      213  35291        10213               NaN                          Amadora
      223  36248         9765               NaN                      Portimonense
      225  36723         4064               NaN                         Feirense
      232  38789         6367               NaN                Uniao da Madeira
      233  38791       188163               NaN                          Tondela
      298  51606         7896               NaN                           Lugano

          team_short_name
      8               DEN
      14              TUB
      170             VOL
      204             TBN
      208             TRO
      213             AMA
      223             POR
      225             FEI
      232             MAD
      233             TON
      298             LUG
```

```
[12]: goals_tendency.head()
```

```
[12]:       season  stage  number_of_teams  total_goals                league_n  \
      0  2008/2009     38               20          942  England Premier League
      1  2009/2010     38               20         1053  England Premier League
      2  2010/2011     38               20         1063  England Premier League
      3  2011/2012     38               20         1066  England Premier League
      4  2012/2013     38               20         1063  England Premier League

        country_n
      0   England
      1   England
      2   England
      3   England
      4   England
```

```
[13]: goals_tendency.info()
```

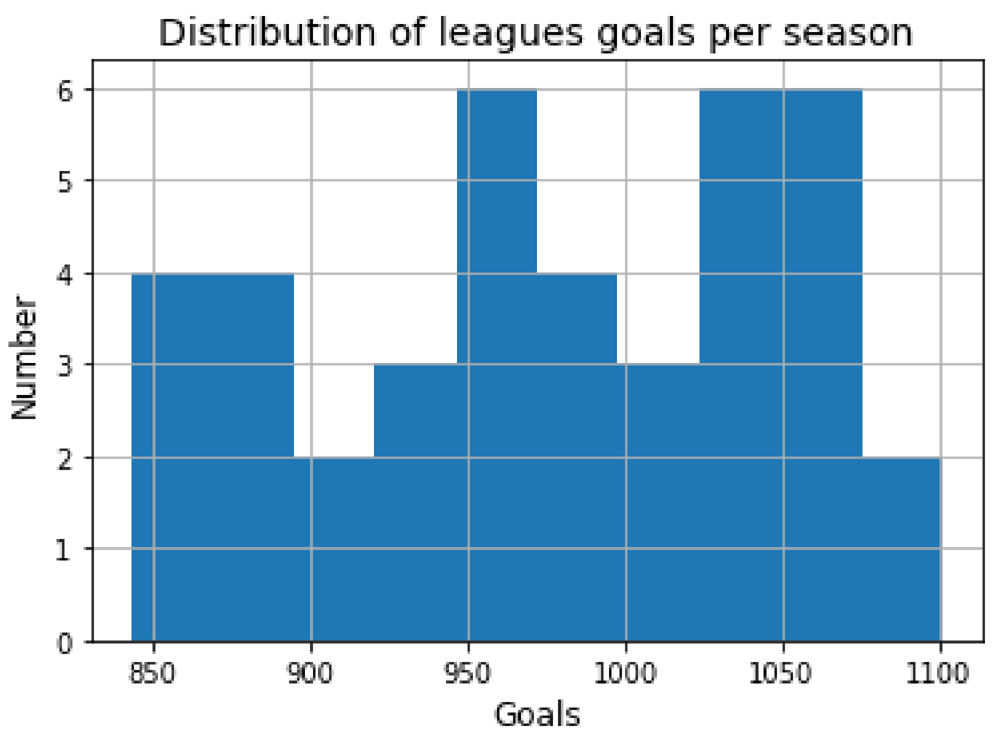
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
```

```
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   season           40 non-null     object
 1   stage            40 non-null     int64
 2   number_of_teams  40 non-null     int64
 3   total_goals      40 non-null     int64
 4   league_n         40 non-null     object
 5   country_n        40 non-null     object
dtypes: int64(3), object(3)
memory usage: 2.0+ KB
```

[14]:
```
goals_tendency.describe()
```

[14]:
```
           stage  number_of_teams  total_goals
count  40.000000        40.000000    40.000000
mean   37.200000        19.600000   976.925000
std     1.620383         0.810191    71.627021
min    34.000000        18.000000   843.000000
25%    38.000000        20.000000   922.750000
50%    38.000000        20.000000   977.000000
75%    38.000000        20.000000  1042.250000
max    38.000000        20.000000  1101.000000
```

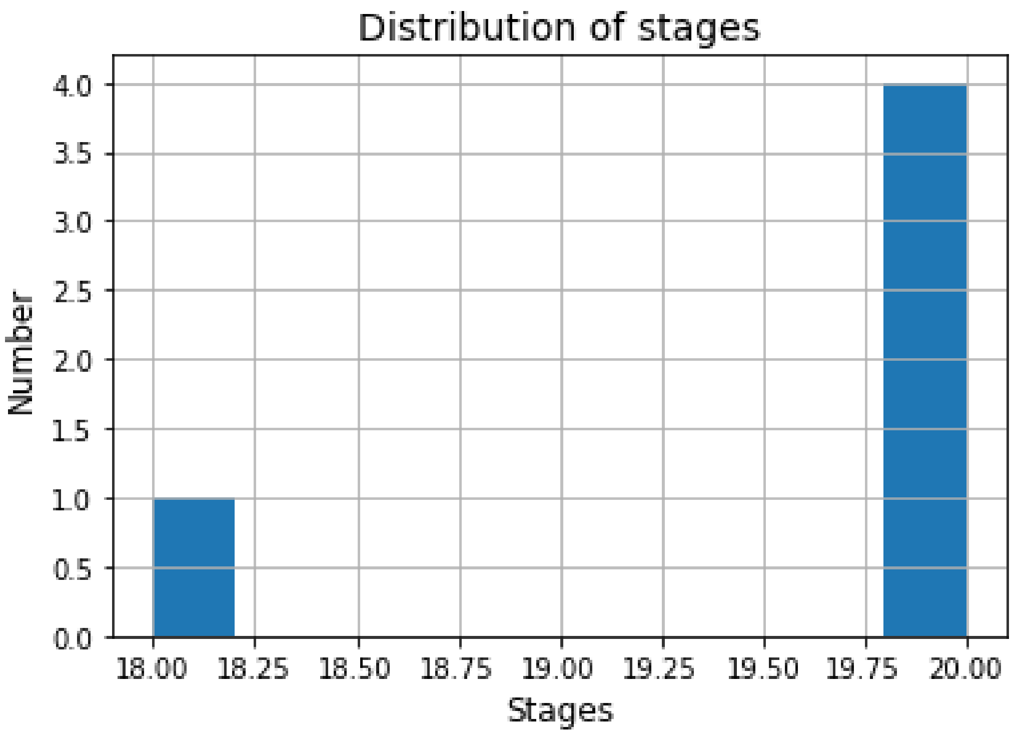
[15]:
```
ax = goals_tendency['total_goals'].hist()
ax.set_xlabel('Goals', fontsize = 12)
ax.set_ylabel('Number', fontsize = 12)
ax.set_title('Distribution of leagues goals per season', fontsize = 14);
```



[16]:
```
avg_league_goals
```

[16]:
```
              league_name  avg_goals_in_season  number_of_teams  total_stages
0  England Premier League             1030.000             20.0          38.0
1          France Ligue 1              928.375             20.0          38.0
2   Germany 1. Bundesliga              887.875             18.0          34.0
3           Italy Serie A              986.875             20.0          38.0
4          Spain LIGA BBVA            1051.500             20.0          38.0
```

[17]:
```
ax = avg_league_goals['number_of_teams'].hist()
ax.set_xlabel('Stages', fontsize = 12)
ax.set_ylabel('Number', fontsize = 12)
ax.set_title('Distribution of stages', fontsize = 14);
```



1.2 Data Cleaning

[18]:
```
df_team[df_team.team_fifa_api_id.isnull()]
```

[18]:
```
        id  team_api_id  team_fifa_api_id                team_long_name  \
8        9         7947               NaN                  FCV Dender EH
14      15         4049               NaN                         Tubize
170  26561         6601               NaN                    FC Volendam
204  34816       177361               NaN  Termalica Bruk-Bet Nieciecza
208  35286         7992               NaN                       Trofense
213  35291        10213               NaN                        Amadora
223  36248         9765               NaN                    Portimonense
225  36723         4064               NaN                       Feirense
232  38789         6367               NaN              Uniao da Madeira
233  38791       188163               NaN                        Tondela
298  51606         7896               NaN                         Lugano

    team_short_name
8               DEN
14              TUB
170             VOL
204             TBN
208             TRO
```

```
213               AMA
223               POR
225               FEI
232               MAD
233               TON
298               LUG
```

Generally data looks like clean, but in df above we have NaN values. We can drom them because we will not need data about this teams.

[19]:
```
##Dropping nulls from df_team table.
df_team.dropna(inplace = True)
```

[20]:
```
##Checking nulls
df_team[df_team.isnull()].count()
```

[20]:
```
id                  0
team_api_id         0
team_fifa_api_id    0
team_long_name      0
team_short_name     0
dtype: int64
```

Exploratory Data Analysis

1.2.1 Research Question 1 (Which team scores more goals on average per match in this period? What league are they from?)

[21]:
```
##For the first question i used only padnas tools
##First of all we will count the goals which were scored in home stadiums
##First we will delete all columns which will not be needed
df_match.drop(df_match.loc[:,'home_player_X1':], axis = 1, inplace = True)

##Then we group all teams with their 'home_team_api_id'
df_most_goals = df_match.groupby(['home_team_api_id'], as_index =␣
 ↪False)['league_id', 'home_team_goal', 'away_team_goal'].mean()

##Next step: we combine our current table with df_team df to add teams names
df_comb = df_team.merge(df_most_goals, left_on='team_api_id',␣
 ↪right_on='home_team_api_id', how='inner')

##Renaming the column for convenience
df_comb.rename(columns = {'team_api_id':'team_id'}, inplace = True)

##Droping unnenesarry columns
df_comb.drop(df_comb.columns[[0,2,5,8]], axis = 1, inplace = True)
```

```
<ipython-input-21-cd844c667205>:7: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
```

```
instead.
  df_most_goals = df_match.groupby(['home_team_api_id'], as_index =
False)['league_id', 'home_team_goal', 'away_team_goal'].mean()
```

[22]:
```
##Then we repeat this operations with goals which were scored in rival's stadium
df_most_goals_away = df_match.groupby(['away_team_api_id'], as_index =␣
 ↪False)['away_team_goal'].mean()

df_comb_away = df_team.merge(df_most_goals_away, left_on='team_api_id',␣
 ↪right_on='away_team_api_id', how='inner')

df_comb_away.rename(columns = {'team_api_id':'team_id'}, inplace = True)

df_comb_away.drop(df_comb_away.columns[[0,2,3,4,5]], axis = 1, inplace = True)
```

[23]:
```
##Combining two df above
df_comb_all = df_comb.merge(df_comb_away, left_on='team_id',␣
 ↪right_on='team_id', how='inner')
```

[24]:
```
##Adding new column with the sum of home and away goals
df_comb_all['team_goals'] = df_comb_all[['home_team_goal', 'away_team_goal']].
 ↪mean(axis = 1)

##Droping unnenesarry columns
df_comb_all.drop(['home_team_goal', 'away_team_goal'], axis = 1, inplace = True)
```

[25]:
```
##Merging with df_league for adding name of the league for each team
df_comb_all = df_comb_all.merge(df_league, left_on='league_id',␣
 ↪right_on='country_id', how='inner')

##Sorting df by 'team_goals' descending
df_comb_all = df_comb_all.sort_values(by='team_goals', ascending=False)

##Droping unnenesarry columns
df_comb_all.drop(['country_id', 'id'], axis = 1, inplace = True)
```

[26]:
```
##I have changed places of columns for convenience
df_comb_all = df_comb_all[['team_id', 'team_long_name', 'team_short_name',␣
 ↪'team_goals', 'name', 'league_id']]

##For the next investigations i chose only first 10 teams with most average␣
 ↪number of goals
top_10_teams = df_comb_all.head(10)
top_10_teams
```

[26]:
```
     team_id    team_long_name team_short_name  team_goals  \
248     8634      FC Barcelona             BAR    2.792763
```

```
246     8633    Real Madrid CF            REA    2.773026
92      9823  FC Bayern Munich            BMU    2.400735
163     8640               PSV            PSV    2.397059
159     8593              Ajax            AJA    2.378676
210     9772        SL Benfica            BEN    2.290323
232     9925            Celtic            CEL    2.286184
201     9773          FC Porto            POR    2.181452
277     9931          FC Basel            BAS    2.164336
225     8548           Rangers            RAN    2.131579

                         name  league_id
248            Spain LIGA BBVA    21518.0
246            Spain LIGA BBVA    21518.0
92       Germany 1. Bundesliga     7809.0
163     Netherlands Eredivisie    13274.0
159     Netherlands Eredivisie    13274.0
210  Portugal Liga ZON Sagres    17642.0
232    Scotland Premier League    19694.0
201  Portugal Liga ZON Sagres    17642.0
277  Switzerland Super League    24558.0
225    Scotland Premier League    19694.0
```

1.2.2 Below I will present two ways of visualization for the first question

[27]:
```
##I specifie some columns as index of df for the first visualization
top_teams_2 = top_10_teams.set_index(['team_long_name','name'])
```

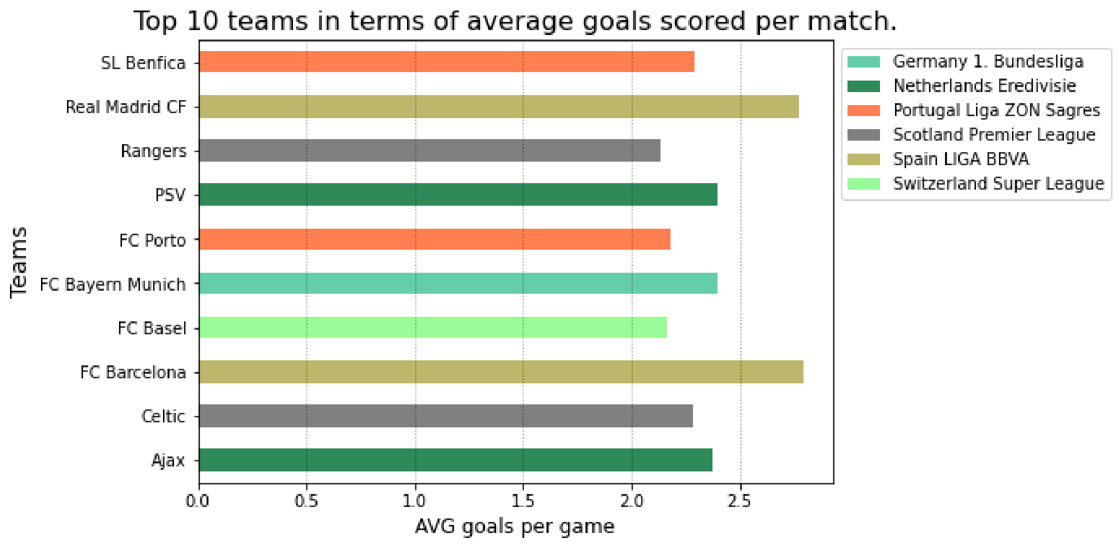
[28]:
```
top_teams_2
```

[28]:
```
                                           team_id team_short_name  \
team_long_name   name
FC Barcelona     Spain LIGA BBVA              8634             BAR
Real Madrid CF   Spain LIGA BBVA              8633             REA
FC Bayern Munich Germany 1. Bundesliga        9823             BMU
PSV              Netherlands Eredivisie       8640             PSV
Ajax             Netherlands Eredivisie       8593             AJA
SL Benfica       Portugal Liga ZON Sagres     9772             BEN
Celtic           Scotland Premier League      9925             CEL
FC Porto         Portugal Liga ZON Sagres     9773             POR
FC Basel         Switzerland Super League     9931             BAS
Rangers          Scotland Premier League      8548             RAN

                                           team_goals  league_id
team_long_name   name
FC Barcelona     Spain LIGA BBVA             2.792763    21518.0
Real Madrid CF   Spain LIGA BBVA             2.773026    21518.0
FC Bayern Munich Germany 1. Bundesliga       2.400735     7809.0
```

```
PSV                Netherlands Eredivisie      2.397059    13274.0
Ajax               Netherlands Eredivisie      2.378676    13274.0
SL Benfica         Portugal Liga ZON Sagres    2.290323    17642.0
Celtic             Scotland Premier League     2.286184    19694.0
FC Porto           Portugal Liga ZON Sagres    2.181452    17642.0
FC Basel           Switzerland Super League    2.164336    24558.0
Rangers            Scotland Premier League     2.131579    19694.0
```

1.2.3 Start of visualizations

[29]:
```
top_teams_2['team_goals'].unstack(1).plot(kind = 'barh',
                                               stacked = True,
                                               figsize = (7, 5),
                                               color = ['#66CDAA', '#2E8B57',␣
 ↪'#FF7F50', 'grey', '#BDB76B', '#98FB98'])

plt.legend(bbox_to_anchor = (1, 1))
plt.grid(axis = 'x', alpha = 0.4, linestyle=':', color = 'black');
plt.xlabel('AVG goals per game', fontsize = 12)
plt.ylabel('Teams', fontsize = 14)
plt.title('Top 10 teams in terms of average goals scored per match.', fontsize␣
 ↪= 16);
```



In the horizontal bar chart above depicted and grouped by league, top 10 teams in terms of average goals scored per match. We can see here that 2 Spanish grands are the most scoring teams. Followed by two Dutch and one German teams, who have scored the same number of goals on average. The lowest scoring team on this list is the Scottish Rangers

Below is the second method of visualization for the first question:

```
plt.figure(figsize=(7, 5))
plt.barh(top_10_teams.query('name == "Spain LIGA BBVA"')['team_long_name'], top_10_teams.query
plt.barh(top_10_teams.query('name == "Germany 1. Bundesliga"')['team_long_name'], top_10_teams
plt.barh(top_10_teams.query('name == "Netherlands Eredivisie"')['team_long_name'], top_10_teams
plt.barh(top_10_teams.query('name == "Portugal Liga ZON Sagres"')['team_long_name'], top_10_tea
plt.barh(top_10_teams.query('name == "Scotland Premier League"')['team_long_name'], top_10_tean
plt.barh(top_10_teams.query('name == "Switzerland Super League"')['team_long_name'], top_10_tea
plt.legend(bbox_to_anchor = (1, 1))
plt.xlabel('AVG goals per game', fontsize = 12)
plt.ylabel('Teams', fontsize = 14)
plt.title('Top 10 teams in terms of average goals scored per match.')
plt.grid(axis = 'x', alpha = 0.4, linestyle=':', color = 'black');
print('In the horizontal bar chart below depicted and grouped by league, top 10 teams in terms
```

1.2.4 Research Question 2 (In which league scores more goals on average per season in this period?)

For the second question i used only SQL queries:

```
SELECT t1.league_n as league_name, AVG(t1.total_goals) as avg_goals_in_season, AVG(t1.number_o
FROM (SELECT m.season, COUNT (DISTINCT m.stage) as stage, COUNT(DISTINCT m.home_team_api_id) a
      FROM Match m
      JOIN League l
      ON m.league_id = l.id
      JOIN Country c
      ON l.country_id = c.id
      WHERE country_n in ('Spain', 'Germany', 'France', 'Italy', 'England')
      GROUP BY country_n, league_n, m.season
      ORDER BY country_n, season) t1
GROUP BY t1.league_n
```

After this i saved results in CSV file. Then i read this CSV file and saved in avg_league_goals variable

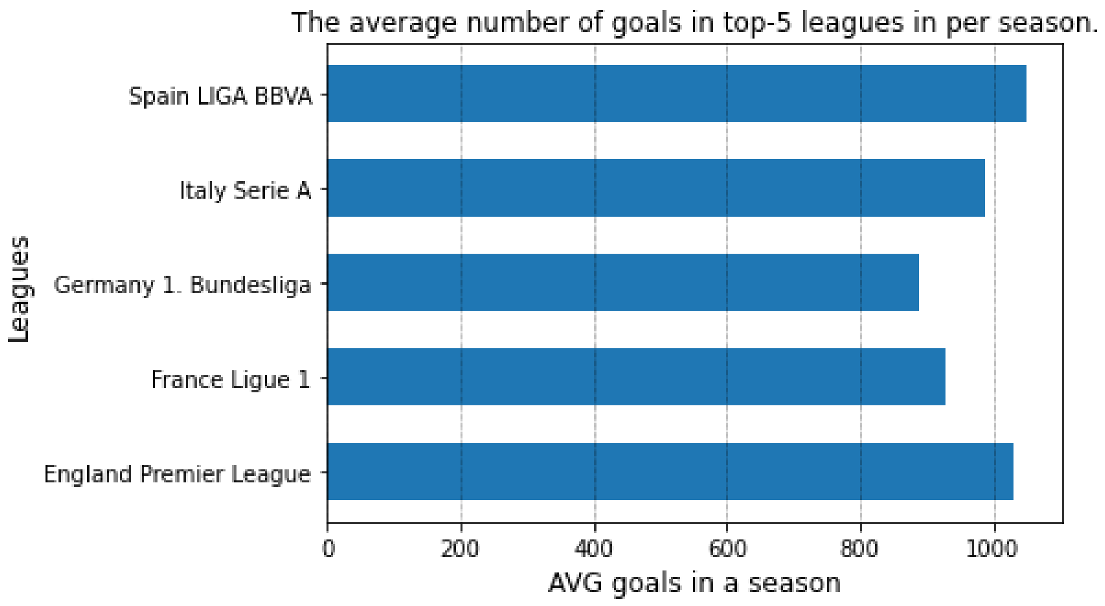
[30]:
```
##Renaming the column to understandable view
avg_league_goals.rename(columns = {'AVG(t1.total_goals)':
 ↪'avg_goals_in_season'}, inplace = True)
```

[31]:
```
avg_league_goals
```

[31]:
```
              league_name  avg_goals_in_season  number_of_teams  total_stages
0  England Premier League             1030.000             20.0          38.0
1          France Ligue 1              928.375             20.0          38.0
2   Germany 1. Bundesliga              887.875             18.0          34.0
3           Italy Serie A              986.875             20.0          38.0
4         Spain LIGA BBVA             1051.500             20.0          38.0
```

1.2.5 Start of visualization

```
[32]: plt.figure(figsize = (6,4))
plt.barh(avg_league_goals.league_name, avg_league_goals.avg_goals_in_season,␣
 ↪height = 0.6);
plt.xlabel('AVG goals in a season', fontsize = 12)
plt.ylabel('Leagues', fontsize = 12)
plt.title('The average number of goals in top-5 leagues in per season.')
plt.grid(axis = 'x', linestyle=':', linewidth=0.5, color='black');
```



The horizontal bar chart above shows us top-5 European leagues with the best indicator of average goals per season. The most scoring league is Spain LIGA BBVA. Then comes England Premier League, which is the only one with Spain LIGA BBVA to score more than 1000 goals per season on average. Followed by the Italy Serie A, France Ligue 1 and Germany 1. Bundesliga respectively.

1.2.6 Research Question 3 (What is the tendency of the goals per season in top 5 leagues in a certain period of time?)

For the third question i also used SQL queries:

```
SELECT m.season, COUNT (DISTINCT m.stage) as stage, COUNT(DISTINCT m.home_team_api_id) as numbe
FROM Match m
JOIN League l
ON m.league_id = l.id
JOIN Country c
ON l.country_id = c.id
WHERE country_n in ('Spain', 'Germany', 'France', 'Italy', 'England')
GROUP BY country_n, league_n, m.season
ORDER BY country_n, season
```

After this i saved results in CSV file. Then i read this CSV file and saved in goals_tendency variable

```
[33]: goals_tendency.head()
```

```
[33]:       season  stage  number_of_teams  total_goals                league_n  \
0  2008/2009     38               20          942  England Premier League
1  2009/2010     38               20         1053  England Premier League
2  2010/2011     38               20         1063  England Premier League
3  2011/2012     38               20         1066  England Premier League
4  2012/2013     38               20         1063  England Premier League

  country_n
0   England
1   England
2   England
3   England
4   England
```

```
[34]: ##Finding unique falues of league_n(league names)
goals_tendency.league_n.unique()
```

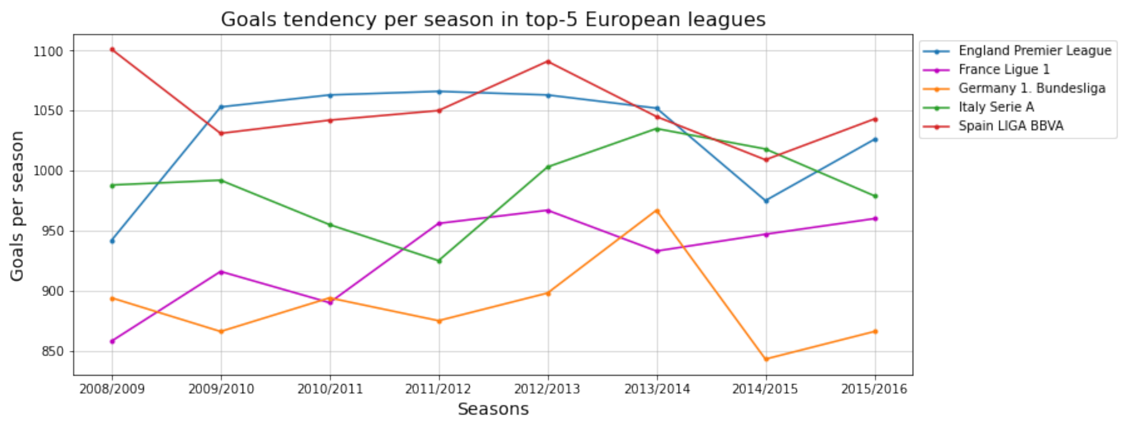
```
[34]: array(['England Premier League', 'France Ligue 1',
       'Germany 1. Bundesliga', 'Italy Serie A', 'Spain LIGA BBVA'],
      dtype=object)
```

```
[35]: ##Preparation for visualization

##Grouping seasons for x-axis
seasons = goals_tendency.groupby(['season'], as_index = False).sum()['season']
```

1.2.7 Start of visualization

```
[36]: ##Creating a function to simplify the code
def g_t(country):
    return goals_tendency.query('country_n =="'+country+'"')['total_goals']
```

```
[37]: plt.figure(figsize = (12,5))
plt.plot(seasons, g_t('England'), label = 'England Premier League', marker = '.
 ↪')
plt.plot(seasons, g_t('France'), label = 'France Ligue 1', marker = '.', color␣
 ↪= 'm' )
plt.plot(seasons, g_t('Germany'), label = 'Germany 1. Bundesliga', marker = '.')
plt.plot(seasons, g_t('Italy'), label = 'Italy Serie A', marker = '.' )
plt.plot(seasons, g_t('Spain'), label = 'Spain LIGA BBVA', marker = '.' )
plt.legend(bbox_to_anchor = (1, 1))
plt.xlabel('Seasons', fontsize = 14)
plt.ylabel('Goals per season', fontsize = 14)
plt.title('Goals tendency per season in top-5 European leagues', fontsize = 16)
plt.grid(alpha = 0.6);
```



The line graph above displays us goal tendency in top-5 European leagues per seasons during the all period of time(from 2008/2009 to 2015/2016 seasons). During this period the highest scoring leagues were: > England Premier League - 4 times Spain LIGA BBVA - 3 times > Italy Serie A - 1 time

Leagues with the lowest number of goals: > Germany 1. Bundesliga - 5 times France Ligue 1 - 3 times

Conclusions

1.2.8 Analysis flaws and data limitations

> df_match dataframe contains a lot of inaccurate (some columns of the dataframe contain absolutely incomprehensible information in the cells) and unnecessary columns, some of which contain many NaN values(which can affect the results of the analysis.). Most of the column names are not clear.
>
> In this DataBase, there is no statistics for players (goals, penalties, shot accuracy, passing, etc.) and teams for this period.

Conclusions for the first question: >After investigations and visualizations we can see that in this 8 year-period the largest number of goals in average scored *FC Barcelona* with **2.79 goals** per match. In second place is *Real Madrid CF* with almost the same number of goals(**2.77 goals**). it is important to note that both teams from the same country and league(*Spain LIGA BBVA*). These teams are located by a wide margin from the rest of the group of teams. If suddenly you want to watch spectacular match with a lot of goals, matches of *FC Barcelona* and *Real Madrid CF* would be a good option.

Conclusions for the second question: >As unsurprisingly the most scoring and spectacular league is *Spain LIGA BBVA*. As we saw above two most scoring teams precisely play in this league(*FC Barcelona, Real Madrid CF*). But what is most interesting is that the second the most scoring league is *England Premier League*. Despite the fact that no English team is in the top 10 scoring teams(only 13th(*Manchester City*)). The least scoring league is *Germany 1. Bundesliga*, but this is due to the fact that this league consist of only 18 teams and 34 stages, while in other leagues 20 teams and 38 stages.

Conclusions for the third question: >The most stable period goals per season was observed in *England Premier League* (after a sharp increase in the number of goals from **942 goals** to **1053 goals**) from 2009/2010 season to 2013/2014 season(in the area of **1060 goals**). Then in the next season there was a drop to abount **977 goals**. > >The biggest decline was observed in *Germany 1. Bundesliga* in 2014/2015 season and drop from **970 goals** to **840 goals!**(This is the worst result among these leagues) > >The best indicator of goals was observed in *Spain LIGA BBVA* in the beginning of the period and amounted **1100 goals**. The only time she got as close as possible to this result was after 4 years in 2012/2013 season(**1090 goals**) > >In the last season all leagues besides *Italy Serie A* had a rise