Investigate a dataset (European Soccer Database)

January 26, 2021

1 Project: Investigate a Dataset (European Soccer Database)

1.1 Table of Contents

## Introduction This soccer database comes from Kaggle and is well suited for data analysis and machine learning. It contains data for soccer matches,players, and teams from several European countries from 2008 to 2016. This dataset is quite extensive, and we encourage you to read more about it here.

note: I imported the data using sqlite then saved the needed output in csv files for easier analysis so I commented all the sql code after connecting and running the queries

Questions:

Q1: What teams improved the most over the time period?

Q2: Which players had the most penalties?

Q3: What team attributes lead to the most victories?

[136]:
```
import pandas as pd
import numpy as np
import sqlite3 as sql
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

## Data Wrangling

1.1.1 General Properties

[109]:

```
# connecting the data base through sqlite
db=sql.connect('database.sqlite')

# this query gets the team table joined with the team attributes table

# teams_query=""" select DISTINCT t.team_long_name team,ta.date date,ta.
 ↪buildUpPlaySpeed,ta.buildUpPlayDribbling,ta.buildUpPlayPassing from team t
#                    join Team_Attributes ta on ta.team_api_id=t.team_api_id
#                  """
# teams_df=pd.read_sql_query(teams_query,db)
# teams_df.to_csv('teams.csv',index=False)

teams_df=pd.read_csv('teams.csv')
teams_df.tail()
```

[109]:

```
                  team                 date  buildUpPlaySpeed  \
1445  SV Zulte-Waregem  2011-02-22 00:00:00                52
1446  SV Zulte-Waregem  2012-02-22 00:00:00                54
1447  SV Zulte-Waregem  2013-09-20 00:00:00                54
1448  SV Zulte-Waregem  2014-09-19 00:00:00                54
1449  SV Zulte-Waregem  2015-09-10 00:00:00                54

      buildUpPlayDribbling  buildUpPlayPassing
1445                   NaN                  52
1446                   NaN                  51
1447                   NaN                  51
1448                  42.0                  51
1449                  42.0                  51
```

1.1.2 Data Cleaning (Replacing the Null values)

[110]:

```
teams_df.fillna(teams_df.mean(),inplace=True)
teams_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1450 entries, 0 to 1449
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   team                  1450 non-null   object
 1   date                  1450 non-null   object
 2   buildUpPlaySpeed      1450 non-null   int64
 3   buildUpPlayDribbling  1450 non-null   float64
 4   buildUpPlayPassing    1450 non-null   int64
dtypes: float64(1), int64(2), object(2)
```

```
memory usage: 56.8+ KB
```

```
[111]: teams_df.head()
```

```
[111]:        team                 date  buildUpPlaySpeed  buildUpPlayDribbling  \
       0  FC Aarau  2010-02-22 00:00:00                60              48.59465
       1  FC Aarau  2014-09-19 00:00:00                52              48.00000
       2  FC Aarau  2015-09-10 00:00:00                47              41.00000
       3  Aberdeen  2010-02-22 00:00:00                70              48.59465
       4  Aberdeen  2011-02-22 00:00:00                47              48.59465

          buildUpPlayPassing
       0                  50
       1                  56
       2                  54
       3                  70
       4                  52
```

## Exploratory Data Analysis

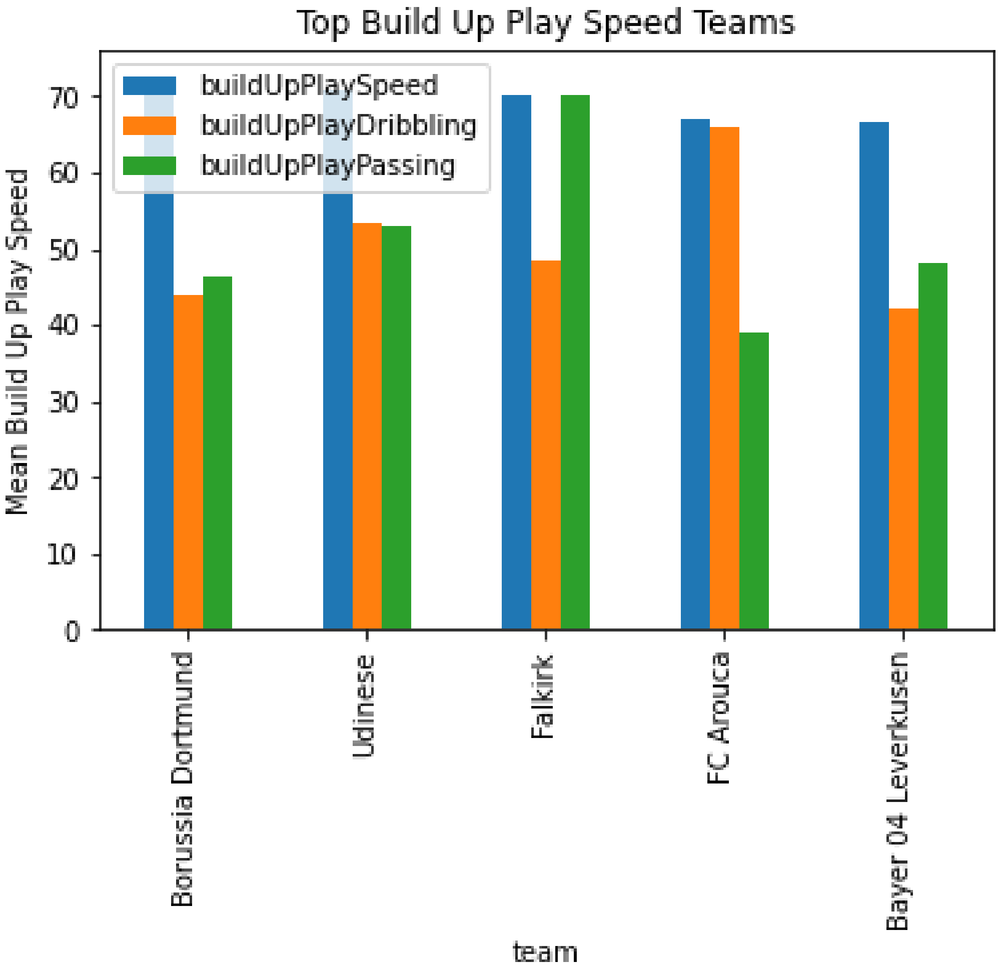
### Research Question 1: What teams improved the most over the time period?

- Getting the top 10 improved teams in terms of teams attributes over the period and analyzing by each attribute

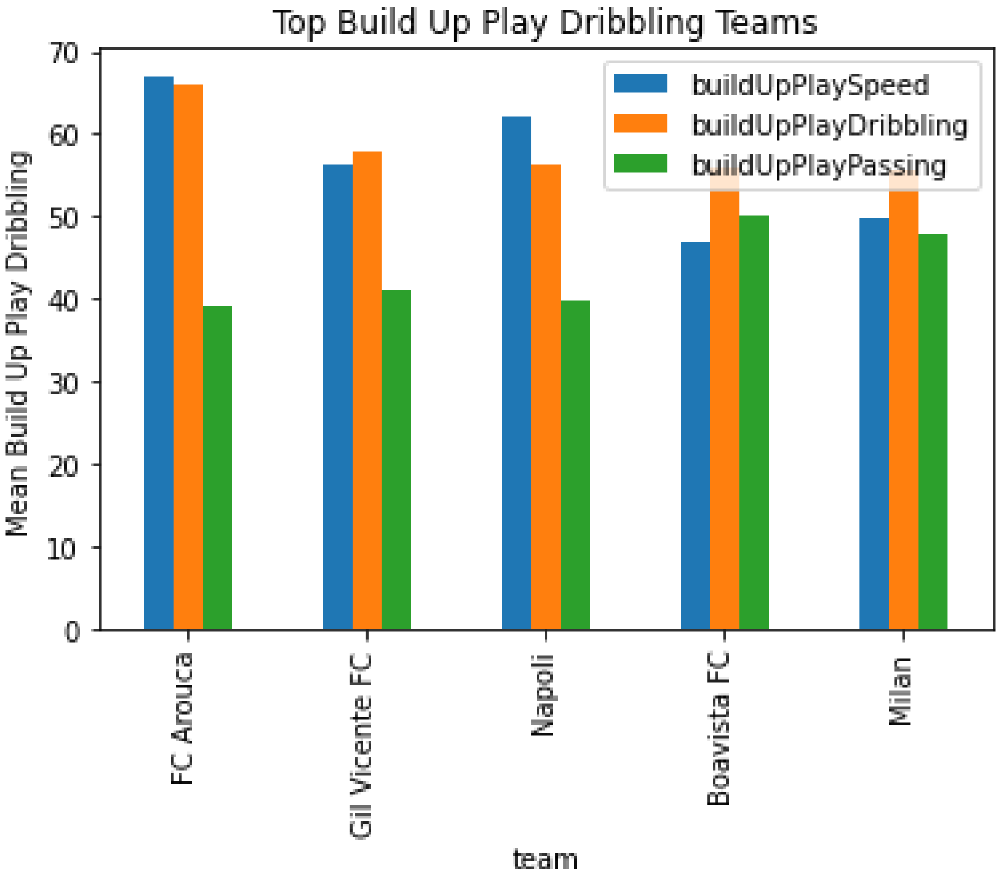
```
[112]: teams_df.groupby('team').mean().
        ↪sort_values(['buildUpPlaySpeed','buildUpPlayDribbling','buildUpPlayPassing'],ascending=Fals
        ↪head(5)
```

```
[112]:                      buildUpPlaySpeed  buildUpPlayDribbling  \
       team
       Borussia Dortmund           72.500000             44.063100
       Udinese                     71.000000             53.229767
       Falkirk                     70.000000             48.594650
       FC Arouca                   67.000000             66.000000
       Bayer 04 Leverkusen         66.833333             42.229767

                            buildUpPlayPassing
       team
       Borussia Dortmund                  46.5
       Udinese                            53.0
       Falkirk                            70.0
       FC Arouca                          39.0
       Bayer 04 Leverkusen                48.0
```

```
[123]: gr1=teams_df.groupby('team').mean().
        ↪sort_values(['buildUpPlaySpeed'],ascending=False).head(5)
       gr1.plot(kind='bar');
       plt.title('Top Build Up Play Speed Teams')
plt.ylabel('Mean Build Up Play Speed');
```

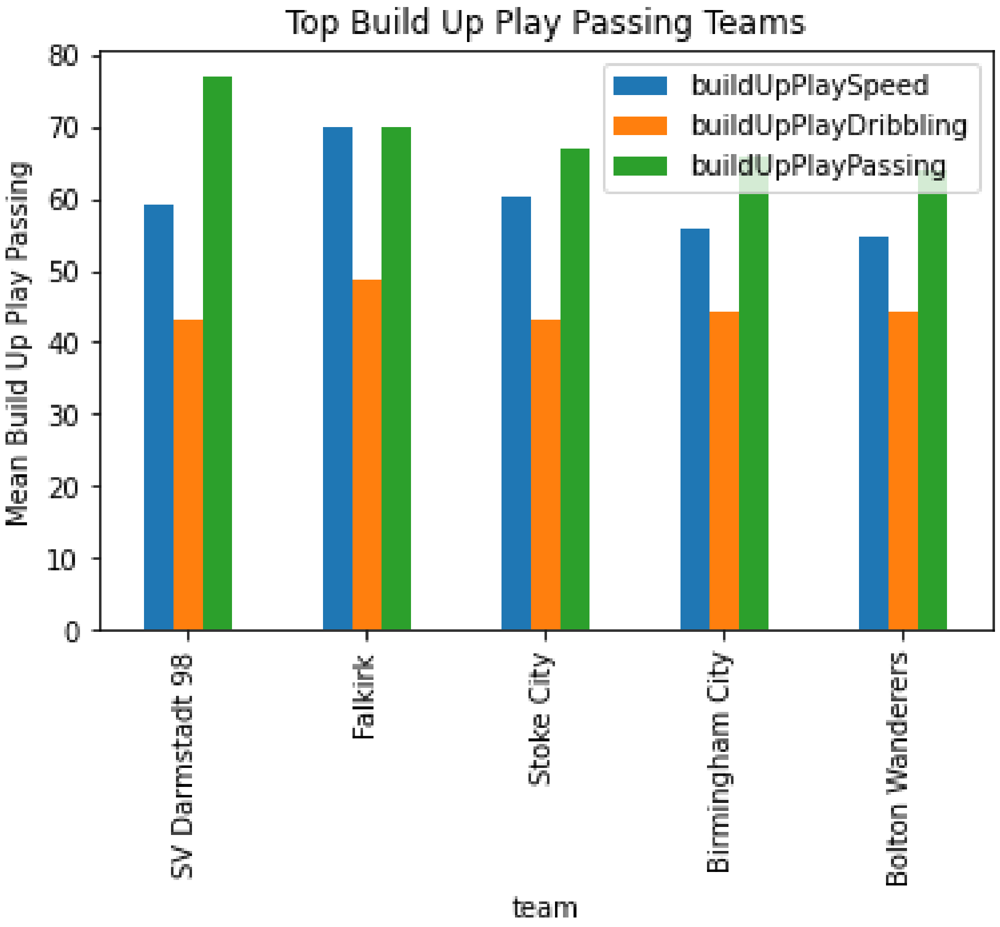


[124]:
```
gr2=teams_df.groupby('team').mean().
 ↪sort_values(['buildUpPlayDribbling'],ascending=False).head(5)
gr2.plot(kind='bar');
plt.title('Top Build Up Play Dribbling Teams')
plt.ylabel('Mean Build Up Play Dribbling');
```



[125]:

```
gr3=teams_df.groupby('team').mean().
 ↪sort_values(['buildUpPlayPassing'],ascending=False).head(5)
gr3.plot(kind='bar');
plt.title('Top Build Up Play Passing Teams')
plt.ylabel('Mean Build Up Play Passing');
```



These are the most improved teams in terms of attributes in the period for 2008 to 2016 also there are two common teams in more than one attribute result (Falkirk , FC Arouca)

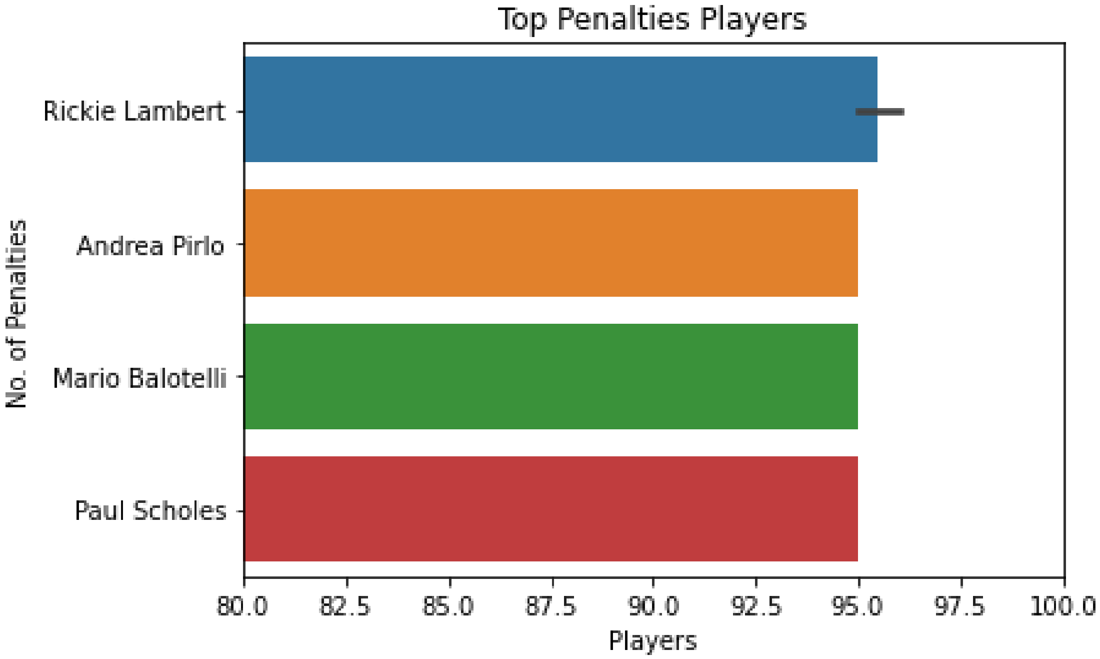
### Research Question 2: Which players had the most penalties?

[127]:

```
# this query gets the palyer and player attributes tables joined them gets the players with most penalties

# players_query='''
#                  select DISTINCT p.player_name,penalties from player p
#                  join player_attributes pa
#                  on pa.player_api_id=p.player_api_id
#                  order by 2 desc
#                  limit (5)
#              '''
# top_penalties_players=pd.read_sql_query(players_query,db)
# top_penalties_players=top_penalties_players.to_csv('players.csv',index=False)

top_penalties_players=pd.read_csv('players.csv')
top_penalties_players.head()
```

[127]:
```
        player_name  penalties
0    Rickie Lambert         96
1      Andrea Pirlo         95
2   Mario Balotelli         95
3      Paul Scholes         95
4    Rickie Lambert         95
```

[153]:
```
ax=sns.barplot(y="player_name", x="penalties",data=top_penalties_players)
ax.set(xlabel='Players', ylabel='No. of Penalties',title='Top Penalties␣
 ↪Players')
plt.xlim(80,100);
```



This query shows which players had the most penaltites and the most one is **Rickie Lambert**

### Research Question 3: What team attributes lead to the most victories?

[10]:
```
# this query joins country,league,match,team,team attributes tables together
# and shows the team attributes in every match for both home and away team then␣
 ↪write the output to winners.csv file
```

```
# winners_query=""" with sub as (
#                                  select DISTINCT c.name country,l.name␣
 ↪league,ht.team_long_name home_team,at.team_long_name␣
 ↪away_team,home_team_goal,away_team_goal,CASE when␣
 ↪home_team_goal>away_team_goal then ht.team_long_name when␣
 ↪home_team_goal<away_team_goal then at.team_long_name else 'draw' end as␣
 ↪Winner,m.date match_date,season,stage,h_ta.date team_atrribute_date,h_ta.
 ↪chanceCreationPassing,h_ta.chanceCreationCrossing,h_ta.
 ↪chanceCreationShooting,h_ta.defencePressure,h_ta.defenceAggression,h_ta.
 ↪defenceTeamWidth,a_ta.date team_atrribute_date,a_ta.
 ↪chanceCreationPassing,a_ta.chanceCreationCrossing,a_ta.
 ↪chanceCreationShooting,a_ta.defencePressure,a_ta.defenceAggression,a_ta.
 ↪defenceTeamWidth from match m
#                                  join country c on m.country_id=c.id
#                                  join League l on m.league_id=l.id
#                                  join team ht on ht.team_api_id=m.
 ↪home_team_api_id
#                                  join team at on at.team_api_id=m.
 ↪away_team_api_id
#                                  join Team_Attributes h_ta on h_ta.
 ↪team_api_id=m.home_team_api_id
#                                  join Team_Attributes a_ta on a_ta.
 ↪team_api_id=m.away_team_api_id
#                      )
#                      select * from sub
#                      where Winner!='draw'
#                  """
# winners_df=pd.read_sql_query(winners_query,db)
# winners_df.to_csv('winners.csv',index=False)

winners_df=pd.read_csv('winners.csv')
```

[11]:
```
winners_df.head(2)
```

[11]:
```
   country                  league         home_team       away_team  \
0  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
1  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht

   home_team_goal  away_team_goal          Winner           match_date  \
0               0               3  RSC Anderlecht  2008-08-16 00:00:00
1               0               3  RSC Anderlecht  2008-08-16 00:00:00

      season  stage  … defencePressure  defenceAggression  defenceTeamWidth  \
0  2008/2009      1  …              65                 60                70
1  2008/2009      1  …              65                 60                70
```

```
   team_atrribute_date:1  chanceCreationPassing:1  chanceCreationCrossing:1  \
0    2010-02-22 00:00:00                     70                       50
1    2011-02-22 00:00:00                     70                       50

   chanceCreationShooting:1 defencePressure:1  defenceAggression:1  \
0                       60                70                   50
1                       60                70                   50

   defenceTeamWidth:1
0                 70
1                 70

[2 rows x 24 columns]
```

[394]:

```
#creating a new column to indicate if the home team scored or no
home_scr = []

for g in winners_df.home_team_goal.tolist():
    if g > 0:
        home_scr.append('scored')
    else:
        home_scr.append('zero score')

winners_df['home_scr'] = np.array(home_scr)

winners_df.head()
```

[394]:

```
   country                 league          home_team       away_team  \
0  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
1  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
2  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
3  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
4  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht

   home_team_goal  away_team_goal          Winner           match_date  \
0               0               3  RSC Anderlecht  2008-08-16 00:00:00
1               0               3  RSC Anderlecht  2008-08-16 00:00:00
2               0               3  RSC Anderlecht  2008-08-16 00:00:00
3               0               3  RSC Anderlecht  2008-08-16 00:00:00
4               0               3  RSC Anderlecht  2008-08-16 00:00:00

      season  stage  … defenceAggression  defenceTeamWidth  \
0  2008/2009      1  …                60                70
1  2008/2009      1  …                60                70
2  2008/2009      1  …                60                70
3  2008/2009      1  …                60                70
4  2008/2009      1  …                60                70
```

```
   team_atrribute_date:1  chanceCreationPassing:1  chanceCreationCrossing:1  \
0    2010-02-22 00:00:00                       70                        50
1    2011-02-22 00:00:00                       70                        50
2    2012-02-22 00:00:00                       53                        57
3    2013-09-20 00:00:00                       68                        67
4    2014-09-19 00:00:00                       60                        53

   chanceCreationShooting:1  defencePressure:1 defenceAggression:1  \
0                        60                 70                  50
1                        60                 70                  50
2                        47                 45                  43
3                        47                 60                  43
4                        47                 60                  50

   defenceTeamWidth:1    home_scr
0                  70  zero score
1                  70  zero score
2                  52  zero score
3                  65  zero score
4                  65  zero score

[5 rows x 25 columns]
```

[13]:
```
# filtering the df to get only home team winners and dropping the away team␣
 ↪attributes
winners_df=pd.read_csv('winners.csv')
home_winners_df=winners_df.query('Winner == home_team')
home_winners_df.drop(home_winners_df.columns[17:],axis=1 , inplace=True)
home_winners_df.head(3)
```

[13]:
```
    country                   league home_team  away_team  home_team_goal  \
36  Belgium  Belgium Jupiler League  KAA Gent  RAEC Mons               5
37  Belgium  Belgium Jupiler League  KAA Gent  RAEC Mons               5
38  Belgium  Belgium Jupiler League  KAA Gent  RAEC Mons               5

    away_team_goal    Winner           match_date     season  stage  \
36               0  KAA Gent  2008-08-17 00:00:00  2008/2009      1
37               0  KAA Gent  2008-08-17 00:00:00  2008/2009      1
38               0  KAA Gent  2008-08-17 00:00:00  2008/2009      1

    team_atrribute_date  chanceCreationPassing  chanceCreationCrossing  \
36  2010-02-22 00:00:00                     60                      50
37  2010-02-22 00:00:00                     60                      50
38  2010-02-22 00:00:00                     60                      50

    chanceCreationShooting  defencePressure  defenceAggression  \
36                      60              45                50
37                      60              45                50
38                      60              45                50

    defenceTeamWidth
36                40
37                40
38                40
```

[384]:

```
group1=home_winners_df.groupby('Winner').mean()
group1.head()
```

[384]:

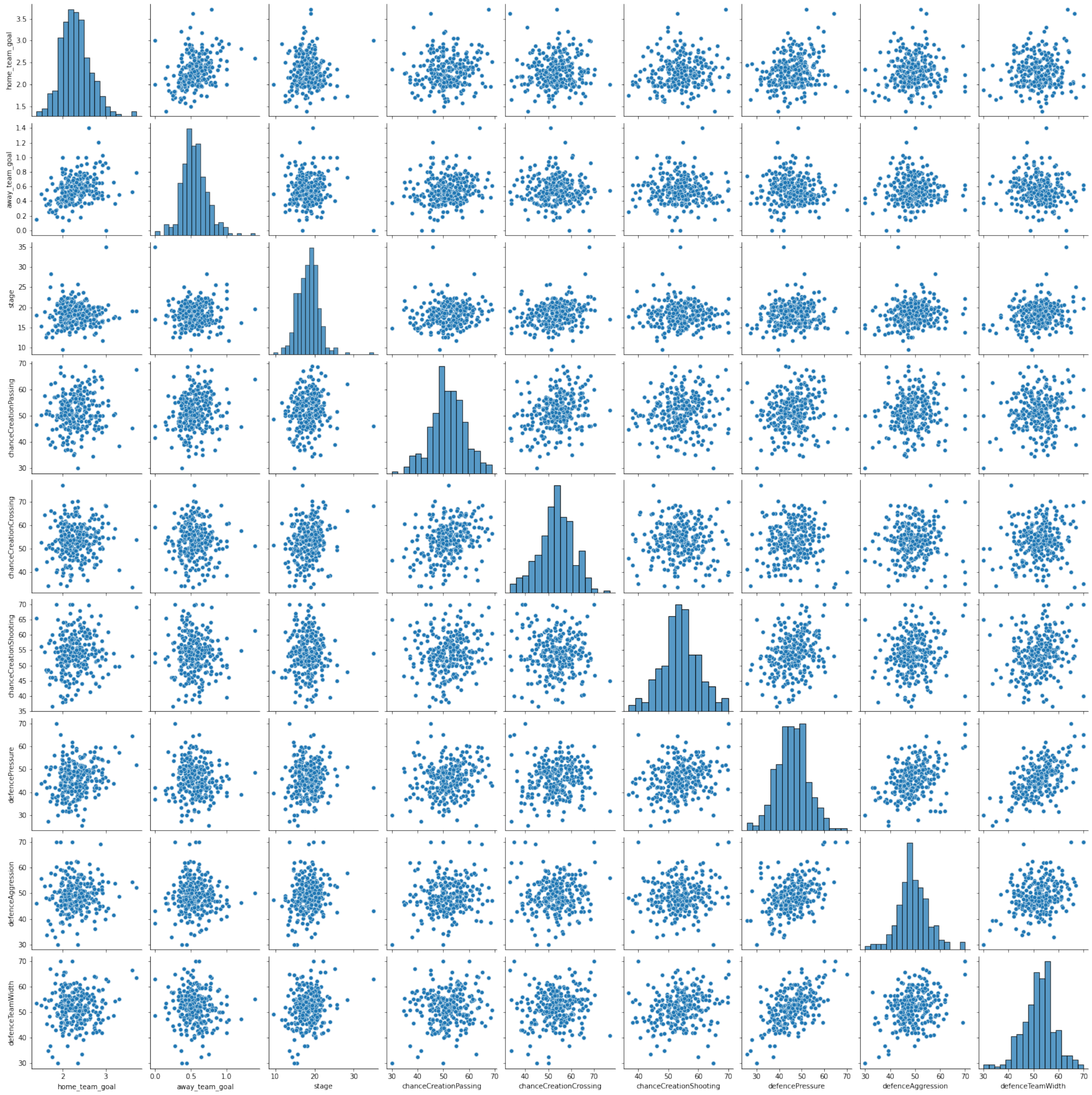
```
                      home_team_goal  away_team_goal      stage  \
Winner
1. FC Kaiserslautern        2.625000        0.500000  16.750000
1. FC Köln                  2.177419        0.521505  17.994624
1. FC Nürnberg              2.303371        0.528090  18.865169
1. FSV Mainz 05             2.310976        0.493902  16.439024
AC Ajaccio                  1.870968        0.569892  20.989247

                      chanceCreationPassing  chanceCreationCrossing  \
Winner
1. FC Kaiserslautern              47.166667               62.000000
1. FC Köln                        55.166667               41.666667
1. FC Nürnberg                    50.500000               53.000000
1. FSV Mainz 05                   53.000000               47.666667
AC Ajaccio                        50.333333               40.666667

                      chanceCreationShooting  defencePressure  \
Winner
1. FC Kaiserslautern               59.666667        46.833333
1. FC Köln                         59.000000        45.000000
1. FC Nürnberg                     59.166667        43.333333
1. FSV Mainz 05                    54.500000        52.500000
AC Ajaccio                         52.166667        37.833333

                      defenceAggression  defenceTeamWidth
Winner
1. FC Kaiserslautern          52.833333         55.166667
1. FC Köln                    51.166667         60.833333
1. FC Nürnberg                50.500000         44.000000
1. FSV Mainz 05               62.500000         49.666667
AC Ajaccio                    50.500000         48.666667
```

[ ]:

```
test=home_winners_df.mean()
```

[379]:

```
sns.pairplot(group1);
```



From the figures it's showing that the home teams most effective attributes for winning are: * Defence Pressure,
* Defence Aggression * Defence TeamWidth

As the positive correlation is clear for each one of them and of course the home team goal also has a positive correlation with the victory.

[16]:
```
# filtering the df to get only away team winners and dropping the home team attributes
winners_df=pd.read_csv('winners.csv')
away_winners_df=winners_df.query('Winner == away_team')
away_winners_df.drop(away_winners_df.columns[10:17],axis=1 , inplace=True)
away_winners_df.head(3)
```

[16]:

```
   country                 league          home_team       away_team  \
0  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
1  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht
2  Belgium  Belgium Jupiler League  KSV Cercle Brugge  RSC Anderlecht

   home_team_goal  away_team_goal          Winner           match_date  \
0               0               3  RSC Anderlecht  2008-08-16 00:00:00
1               0               3  RSC Anderlecht  2008-08-16 00:00:00
2               0               3  RSC Anderlecht  2008-08-16 00:00:00

      season  stage team_atrribute_date:1  chanceCreationPassing:1  \
0  2008/2009      1   2010-02-22 00:00:00                       70
1  2008/2009      1   2011-02-22 00:00:00                       70
2  2008/2009      1   2012-02-22 00:00:00                       53

   chanceCreationCrossing:1  chanceCreationShooting:1  defencePressure:1  \
0                        50                        60                 70
1                        50                        60                 70
2                        57                        47                 45

   defenceAggression:1  defenceTeamWidth:1
0                   50                  70
1                   50                  70
2                   43                  52
```

[380]:

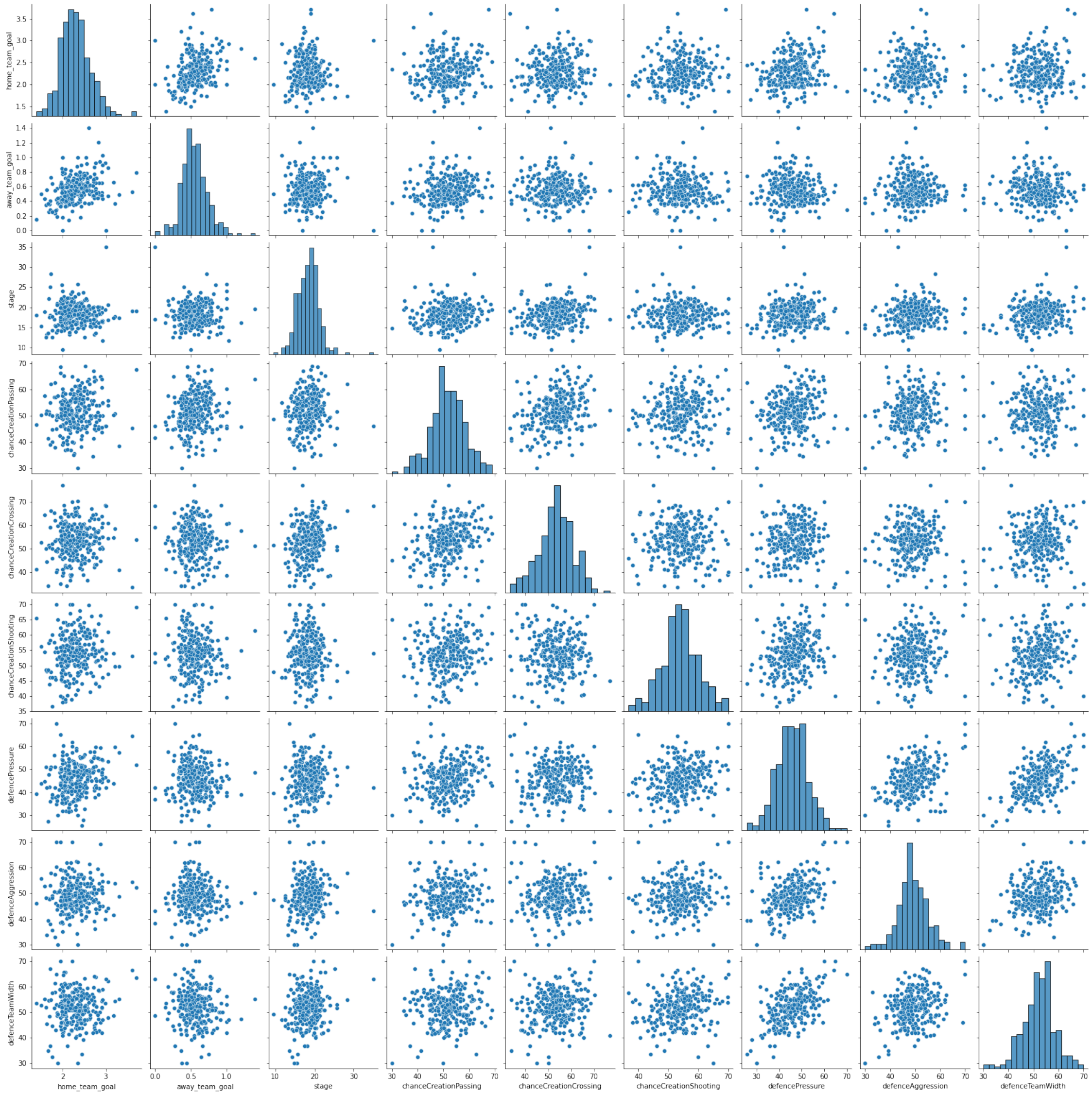
```
group2=away_winners_df.groupby('Winner').mean()
group2.head()
```

[380]:

```
                      home_team_goal  away_team_goal      stage  \
Winner
1. FC Kaiserslautern        0.888889        2.222222  21.333333
1. FC Köln                  0.722892        2.289157  15.734940
1. FC Nürnberg              0.842105        2.263158  18.894737
1. FSV Mainz 05             0.974093        2.440415  16.611399
AC Ajaccio                  0.833333        2.333333  20.166667

                      chanceCreationPassing:1  chanceCreationCrossing:1  \
Winner
1. FC Kaiserslautern                47.166667                 62.000000
1. FC Köln                          55.166667                 41.666667
1. FC Nürnberg                      50.500000                 53.000000
1. FSV Mainz 05                     53.000000                 47.666667
AC Ajaccio                          50.333333                 40.666667

                      chanceCreationShooting:1  defencePressure:1  \
Winner
1. FC Kaiserslautern                 59.666667          46.833333
```

```
1. FC Köln                           59.000000          45.000000
1. FC Nürnberg                       59.166667          43.333333
1. FSV Mainz 05                      54.500000          52.500000
AC Ajaccio                           52.166667          37.833333

                      defenceAggression:1  defenceTeamWidth:1
Winner
1. FC Kaiserslautern             52.833333           55.166667
1. FC Köln                       51.166667           60.833333
1. FC Nürnberg                   50.500000           44.000000
1. FSV Mainz 05                  62.500000           49.666667
AC Ajaccio                       50.500000           48.666667
```

[381]:
```
sns.pairplot(group2);
```



From the figures it's showing that the away teams most effective attributes for winning are: * Defence Pressure,
* Defence Aggression * Defence TeamWidth * Chance Creation Shooting

As the positive correlation is clear for each one of them and of course the away team goal also has a positive correlation with the victory.

## Conclusions

The results here showed the required answers for the questions we had by shaping the needed data frame the best fit the answer of each question. * At first we showed the most improved teams in terms of the attributes during the period from 2008 to 2016 * Secondly we framed the needed data that gave us the players who had most penalties * Finally we used the winning teams data frame and divided it into two data frames home team winners and away team winners so we can a clear relation between the victories and the team attributes in both conitions

Some Challanges and limitations: * Many columns in the Match table are empty it would make analysis more accurate if these values were recorded. * The DB has many unuseful records and is kind of intensive but sql helped in solving that issue. * As the DB has many tables after doing some joins the output csv files were big in size

1.1.3 Refrences I used

- Stackoverflow
- Github
- Pandas and Seaborn docs