CED19I028 - Sumit Kumar

ASBD Mid-Sem

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas as pd
import seaborn as sns
```

In [126…
```
df = pd.read_csv("brasil.csv")
df
```

Out[126]:

	ID	Data	Ano	Rodada	Equipe_mandante	Equipe_visitante	Gols_mandante	Gols_visitante	Mandante_UF	Visitante_UF	Vencedor	Arena	OBS
0	2003.01.0001	29/03/2003	2003	1	Guarani	Vasco	4	2	SP	RJ	Mandante	Brinco de Ouro	NaN
1	2003.01.0002	29/03/2003	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	Mandante	Arena da Baixada	NaN
2	2003.01.0003	30/03/2003	2003	1	Flamengo	Coritiba	1	1	RJ	PR	Empate	Maracanã	NaN
3	2003.01.0004	30/03/2003	2003	1	Goiás	Paysandu	2	2	GO	PA	Empate	Serra Dourada	NaN
4	2003.01.0005	30/03/2003	2003	1	Internacional	Ponte Preta	1	1	RS	SP	Empate	Beira-Rio	NaN
…	…	…	…	…	…	…	…	…	…	…	…	…	…
6881	2019.38.0376	08/12/2019	2019	38	Vasco	Chapecoense	1	1	RJ	SC	Empate	Maracanã	NaN
6882	2019.38.0377	08/12/2019	2019	38	Botafogo-RJ	Ceará	1	1	RJ	CE	Empate	Engenhão	NaN
6883	2019.38.0378	08/12/2019	2019	38	Avaí	Athletico-PR	0	0	SC	PR	Empate	Ressacada	NaN
6884	2019.38.0379	08/12/2019	2019	38	Goiás	Grêmio	3	2	GO	RS	Mandante	Serra Dourada	NaN
6885	2019.38.0380	08/12/2019	2019	38	CSA	São Paulo	1	2	AL	SP	Visitante	Rei Pelé	NaN

6886 rows × 13 columns

In [127…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6886 entries, 0 to 6885
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   ID                6886 non-null   object
 1   Data              6886 non-null   object
 2   Ano               6886 non-null   int64 
 3   Rodada            6886 non-null   int64 
 4   Equipe_mandante   6886 non-null   object
 5   Equipe_visitante  6886 non-null   object
 6   Gols_mandante     6886 non-null   int64 
 7   Gols_visitante    6886 non-null   int64 
 8   Mandante_UF       6886 non-null   object
 9   Visitante_UF      6886 non-null   object
 10  Vencedor          6886 non-null   object
 11  Arena             6886 non-null   object
 12  OBS               11 non-null     object
dtypes: int64(4), object(9)
memory usage: 699.5+ KB
```

In [ ]:

1. Rename column names and data to English

In [128…

```python
new_columns = {
    'Data': 'Match Date',
    'Ano': 'Season (Year)',
    'Rodada': 'Season round',
    'Equipe_mandante': 'Home team',
    'Equipe_visitante': 'Visitor team',
    'Gols_mandante': 'Home team Goals',
    'Gols_visitante': 'Visitor team Goals',
    'Mandante_UF': 'Home team Brazil\'s state',
    'Visitante_UF': 'Visitor team Brazil\'s state',
    'Vencedor': 'Winner',
    'OBS': 'Comments'
}

df.rename(columns = new_columns, inplace=True)
df
```

Out[128]:

	ID	Match Date	Season (Year)	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Winner	Arena	Comments
0	2003.01.0001	29/03/2003	2003	1	Guarani	Vasco	4	2	SP	RJ	Mandante	Brinco de Ouro	NaN
1	2003.01.0002	29/03/2003	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	Mandante	Arena da Baixada	NaN
2	2003.01.0003	30/03/2003	2003	1	Flamengo	Coritiba	1	1	RJ	PR	Empate	Maracanã	NaN
3	2003.01.0004	30/03/2003	2003	1	Goiás	Paysandu	2	2	GO	PA	Empate	Serra Dourada	NaN
4	2003.01.0005	30/03/2003	2003	1	Internacional	Ponte Preta	1	1	RS	SP	Empate	Beira-Rio	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6881	2019.38.0376	08/12/2019	2019	38	Vasco	Chapecoense	1	1	RJ	SC	Empate	Maracanã	NaN
6882	2019.38.0377	08/12/2019	2019	38	Botafogo-RJ	Ceará	1	1	RJ	CE	Empate	Engenhão	NaN
6883	2019.38.0378	08/12/2019	2019	38	Avaí	Athletico-PR	0	0	SC	PR	Empate	Ressacada	NaN
6884	2019.38.0379	08/12/2019	2019	38	Goiás	Grêmio	3	2	GO	RS	Mandante	Serra Dourada	NaN
6885	2019.38.0380	08/12/2019	2019	38	CSA	São Paulo	1	2	AL	SP	Visitante	Rei Pelé	NaN

6886 rows × 13 columns

In [ ]:

Change winners columns data into english..

In [129…

```
df['Winner'].value_counts()
```

Out[129]:

```
Mandante     3488
Empate       1795
Visitante    1603
Name: Winner, dtype: int64
```

In [130…

```
d = {
    'Mandante': 'home',
    'Empate': 'tie',
    'Visitante': 'visiting'
}

for each in d.keys():
    df.loc[df['Winner'] == each, 'Winner'] = d[each]

df
```

Out[130]:

	ID	Match Date	Season (Year)	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Winner	Arena	Comments
0	2003.01.0001	29/03/2003	2003	1	Guarani	Vasco	4	2	SP	RJ	home	Brinco de Ouro	NaN
1	2003.01.0002	29/03/2003	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	home	Arena da Baixada	NaN
2	2003.01.0003	30/03/2003	2003	1	Flamengo	Coritiba	1	1	RJ	PR	tie	Maracanã	NaN
3	2003.01.0004	30/03/2003	2003	1	Goiás	Paysandu	2	2	GO	PA	tie	Serra Dourada	NaN
4	2003.01.0005	30/03/2003	2003	1	Internacional	Ponte Preta	1	1	RS	SP	tie	Beira-Rio	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6881	2019.38.0376	08/12/2019	2019	38	Vasco	Chapecoense	1	1	RJ	SC	tie	Maracanã	NaN
6882	2019.38.0377	08/12/2019	2019	38	Botafogo-RJ	Ceará	1	1	RJ	CE	tie	Engenhão	NaN
6883	2019.38.0378	08/12/2019	2019	38	Avaí	Athletico-PR	0	0	SC	PR	tie	Ressacada	NaN
6884	2019.38.0379	08/12/2019	2019	38	Goiás	Grêmio	3	2	GO	RS	home	Serra Dourada	NaN
6885	2019.38.0380	08/12/2019	2019	38	CSA	São Paulo	1	2	AL	SP	visiting	Rei Pelé	NaN

6886 rows × 13 columns

2. Dropping Comments column, because it has dominant NaN value

In [131…

```
df.drop(['Comments'], axis=1, inplace=True)
df
```

Out[131]:

	ID	Match Date	Season (Year)	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Winner	Arena
0	2003.01.0001	29/03/2003	2003	1	Guarani	Vasco	4	2	SP	RJ	home	Brinco de Ouro
1	2003.01.0002	29/03/2003	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	home	Arena da Baixada
2	2003.01.0003	30/03/2003	2003	1	Flamengo	Coritiba	1	1	RJ	PR	tie	Maracanã
3	2003.01.0004	30/03/2003	2003	1	Goiás	Paysandu	2	2	GO	PA	tie	Serra Dourada
4	2003.01.0005	30/03/2003	2003	1	Internacional	Ponte Preta	1	1	RS	SP	tie	Beira-Rio
...	...	...	...	...	...	...	...	...	...	...	...	...
6881	2019.38.0376	08/12/2019	2019	38	Vasco	Chapecoense	1	1	RJ	SC	tie	Maracanã
6882	2019.38.0377	08/12/2019	2019	38	Botafogo-RJ	Ceará	1	1	RJ	CE	tie	Engenhão
6883	2019.38.0378	08/12/2019	2019	38	Avaí	Athletico-PR	0	0	SC	PR	tie	Ressacada
6884	2019.38.0379	08/12/2019	2019	38	Goiás	Grêmio	3	2	GO	RS	home	Serra Dourada
6885	2019.38.0380	08/12/2019	2019	38	CSA	São Paulo	1	2	AL	SP	visiting	Rei Pelé

6886 rows × 12 columns

3. Find duplicates and drop if exist..

In [132…
```
df[df.duplicated()]
```

Out[132]:

ID	Match Date	Season (Year)	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Winner	Arena

4. Find total number of matches in each year..

In [133…
```
df['ID'][df.duplicated()]
```

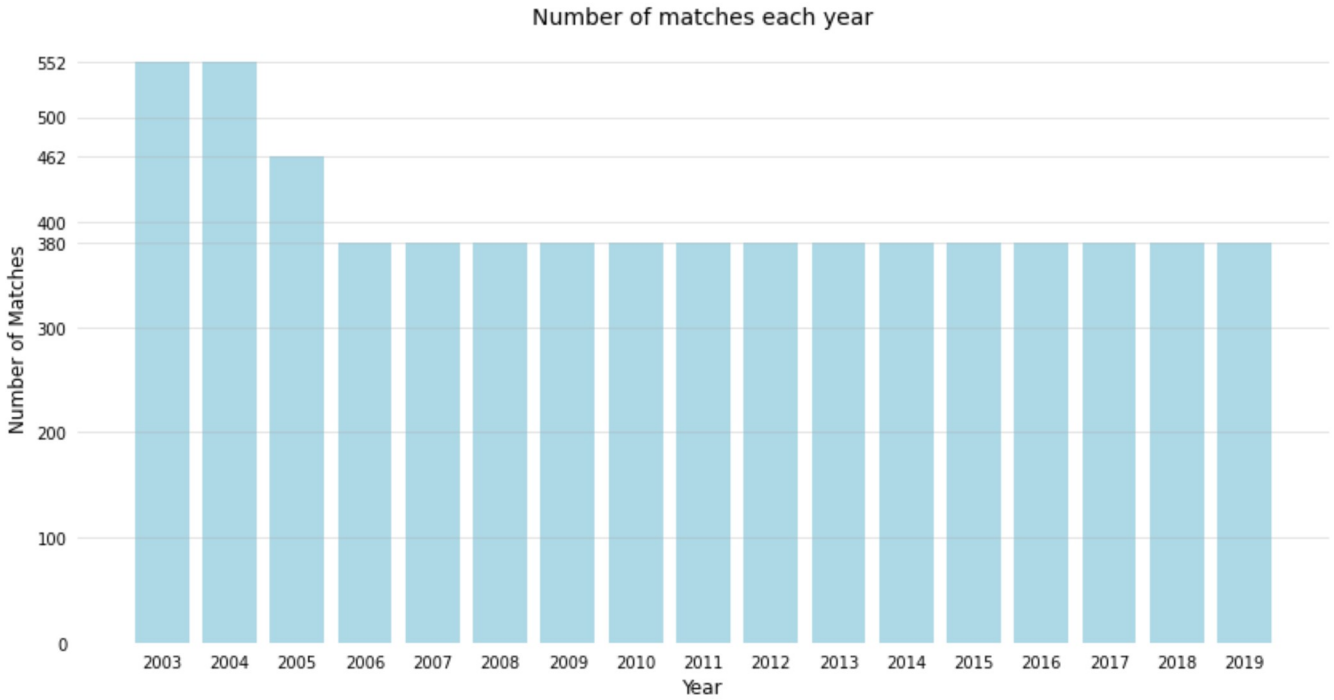
Out[133]:
```
Series([], Name: ID, dtype: object)
```

In [134…
```
bar_data = df.groupby(['Season (Year)']).count()['ID']
bar_data
```

Out[134]:
```
Season (Year)
2003    552
2004    552
2005    462
2006    380
2007    380
2008    380
2009    380
2010    380
2011    380
2012    380
2013    380
2014    380
2015    380
2016    380
2017    380
2018    380
2019    380
Name: ID, dtype: int64
```

In [136…

```python
def plotBar(x_axis, y_axis, title):
    plt.bar(x_axis, y_axis, color='lightblue')
    plt.xlabel("Year", size=12)
    plt.ylabel("Number of Matches", size=12)

    yticks = [0, 100, 200, 300, 380, 400, 462, 500, 552]
    plt.yticks(yticks)

    plt.grid(axis ='y',alpha=0.4)
    plt.tick_params(left = False, bottom=False)

    for spine in plt.gca().spines.values():
        spine.set_visible(False)

    plt.title(title, size=14)


plt.figure(figsize=(14,7))

x = [str(each) for each in bar_data.index.tolist()]
y = bar_data.values.tolist()
plotBar(x, y, "Number of matches each year")
```



In [ ]:

5. Number of teams in each state.

In [ ]:

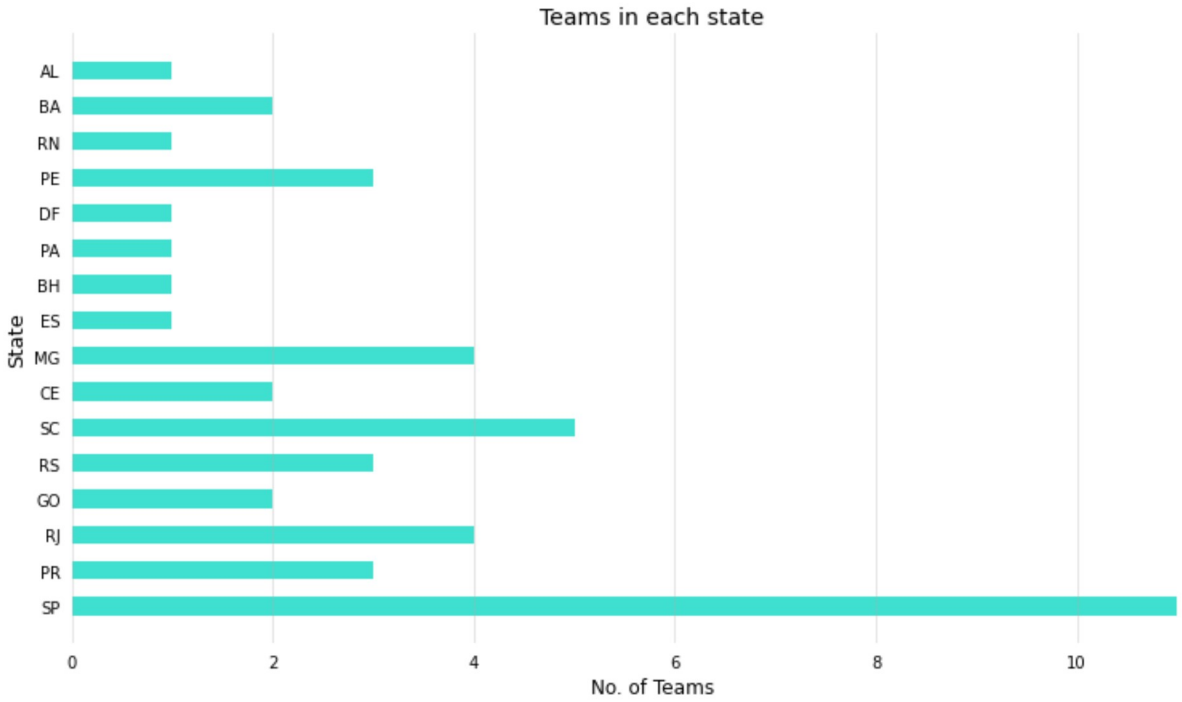
In [137…

```
teams_in_state = {}
states = df['Home team Brazil\'s state'].unique().tolist()
for each in states:
    teams_in_state[each] = teams_in_state.get(each, df[df['Home team Brazil\'s state'] == each]['Home team'].unique().tolist())


teams_in_state
```

Out[137]:
```
{'SP': ['Guarani',
  'Santos',
  'Corinthians',
  'Ponte Preta',
  'São Caetano',
  'São Paulo',
  'Palmeiras',
  'Portuguesa',
  'Santo André',
  'Barueri',
  'Grêmio Prudente'],
 'PR': ['Athletico-PR', 'Coritiba', 'Paraná'],
 'RJ': ['Flamengo', 'Fluminense', 'Vasco', 'Botafogo-RJ'],
 'GO': ['Goiás', 'Atlético-GO'],
 'RS': ['Internacional', 'Juventude', 'Grêmio'],
 'SC': ['Criciúma', 'Figueirense', 'Avaí', 'Chapecoense', 'Joinville'],
 'CE': ['Fortaleza', 'Ceará'],
 'MG': ['Cruzeiro', 'Atlético-MG', 'Ipatinga', 'América-MG'],
 'ES': ['Vitória'],
 'BH': ['Bahia'],
 'PA': ['Paysandu'],
 'DF': ['Brasiliense'],
 'PE': ['Santa Cruz', 'Sport', 'Náutico'],
 'RN': ['América-RN'],
 'BA': ['Vitória', 'Bahia'],
 'AL': ['CSA']}
```

In [139…

```python
def plotBarH(x_axis, y_axis, title):
    plt.barh(x_axis, y_axis, color='turquoise', align='center', height=0.5)
    plt.xlabel("No. of Teams", size=12)
    plt.ylabel("State", size=13)

    plt.grid(axis ='x',alpha=0.4)
    plt.tick_params(left = False, bottom=False)

    for spine in plt.gca().spines.values():
        spine.set_visible(False)

    plt.title(title, size=14)


x = list(teams_in_state.keys())
y = [len(each) for each in teams_in_state.values()]

plt.figure(figsize=(13,7))
plotBarH(x, y, 'Teams in each state')
```



In [ ]:

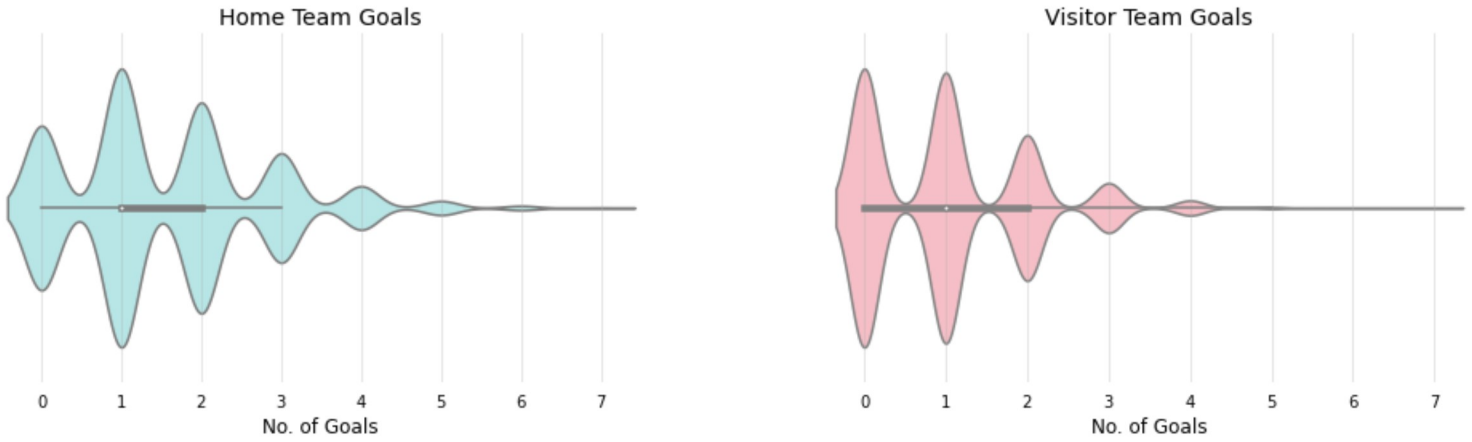
6. Home team's goal and visiting teams Goals Comparision..

In [140…
```
df['Home team Goals'].value_counts()
```

Out[140]:
```
1    2324
2    1759
0    1371
3     912
4     362
5     116
6      36
7       6
Name: Home team Goals, dtype: int64
```

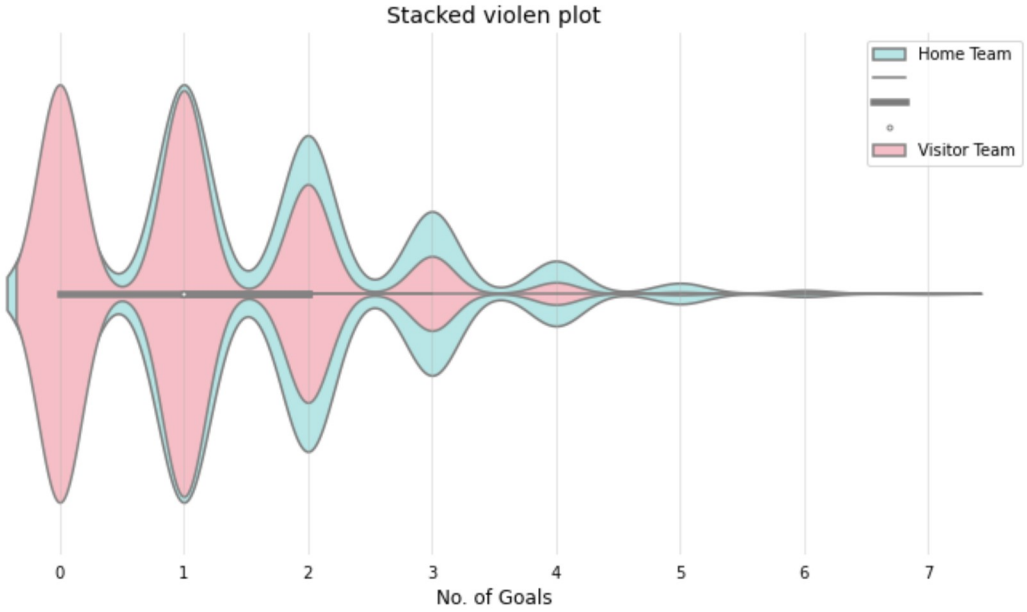
In [141…

```
def plotViolinGraph(data, name, color):
    sns.violinplot(x = data, scale="count", gridsize=1000, color=color)

    plt.grid(axis ='x',alpha=0.4)
    plt.tick_params(left = False, bottom=False)
    plt.xlabel("No. of Goals", size=12)

    for spine in plt.gca().spines.values():
        spine.set_visible(False)

    plt.title(name, size=14)



plt.figure(figsize=(17,4))
plt.subplot(1, 2, 1)
plotViolinGraph(df['Home team Goals'].tolist(), "Home Team Goals", "paleturquoise");

plt.subplot(1, 2, 2)
plotViolinGraph(df['Visitor team Goals'].tolist(), "Visitor Team Goals", "lightpink");
```



In [142…

```
plt.figure(figsize=(12,6))

sns.violinplot(x = df['Home team Goals'].tolist(), scale="count", gridsize=1000, color='paleturquoise')
sns.violinplot(x = df['Visitor team Goals'].tolist(), scale="count", gridsize=1000, color='lightpink')

plt.legend(labels=['Home Team', '', '', '', 'Visitor Team'])

plt.grid(axis ='x',alpha=0.4)
plt.tick_params(left = False, bottom=False)
plt.xlabel("No. of Goals", size=12)

for spine in plt.gca().spines.values():
    spine.set_visible(False)

plt.title("Stacked violen plot", size=14)

plt.show()
```

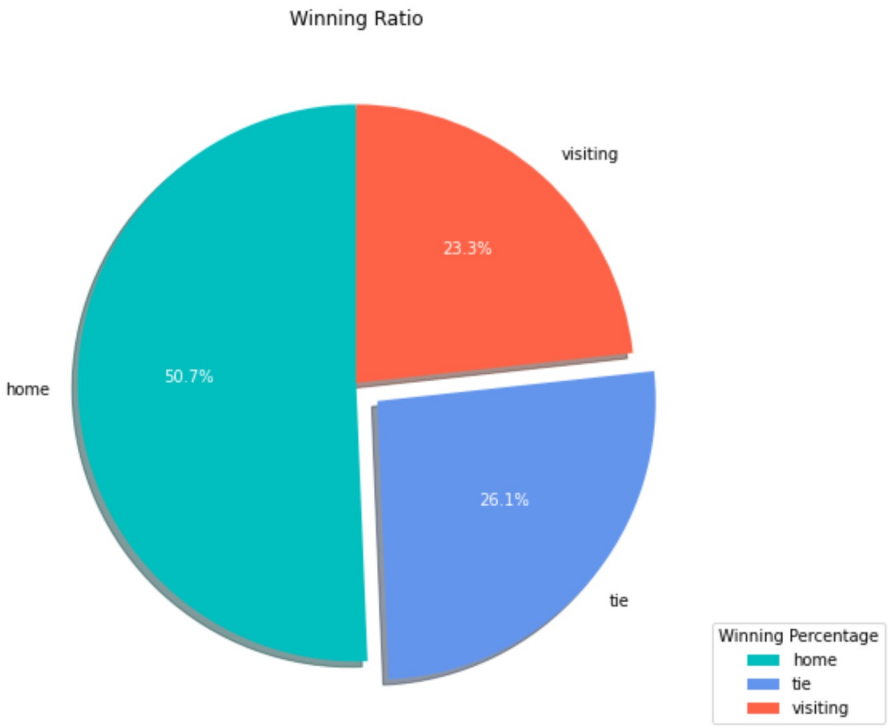


In [ ]:

7. Percentage of Winning Ratio of matches

In [143…

```
def func(pct):
    return "{:.1f}%\n".format(pct)

def plotPeiChart(title, data, names):
    explodes = (0, 0.1, 0)
    colors = ("c", "cornflowerblue", "tomato")

    _, _, autotexts = plt.pie(data,
            labels = names,
            explode = explodes,
            shadow = True,
            colors = colors,
            startangle = 90,
            autopct=lambda pct: func(pct)
           )

    for autotext in autotexts:
        autotext.set_color('white')

    plt.legend(names,
             title="Winning Percentage",
             loc="lower left",
             bbox_to_anchor=(1,0,0.5,1))

    plt.title(title)


plt.figure(figsize=(15,8))

plotPeiChart('Winning Ratio', df['Winner'].value_counts().tolist(), df['Winner'].value_counts().index.tolist())
```



In [ ]:

8. Arena which has most number of home and visitor winners.

In [144…

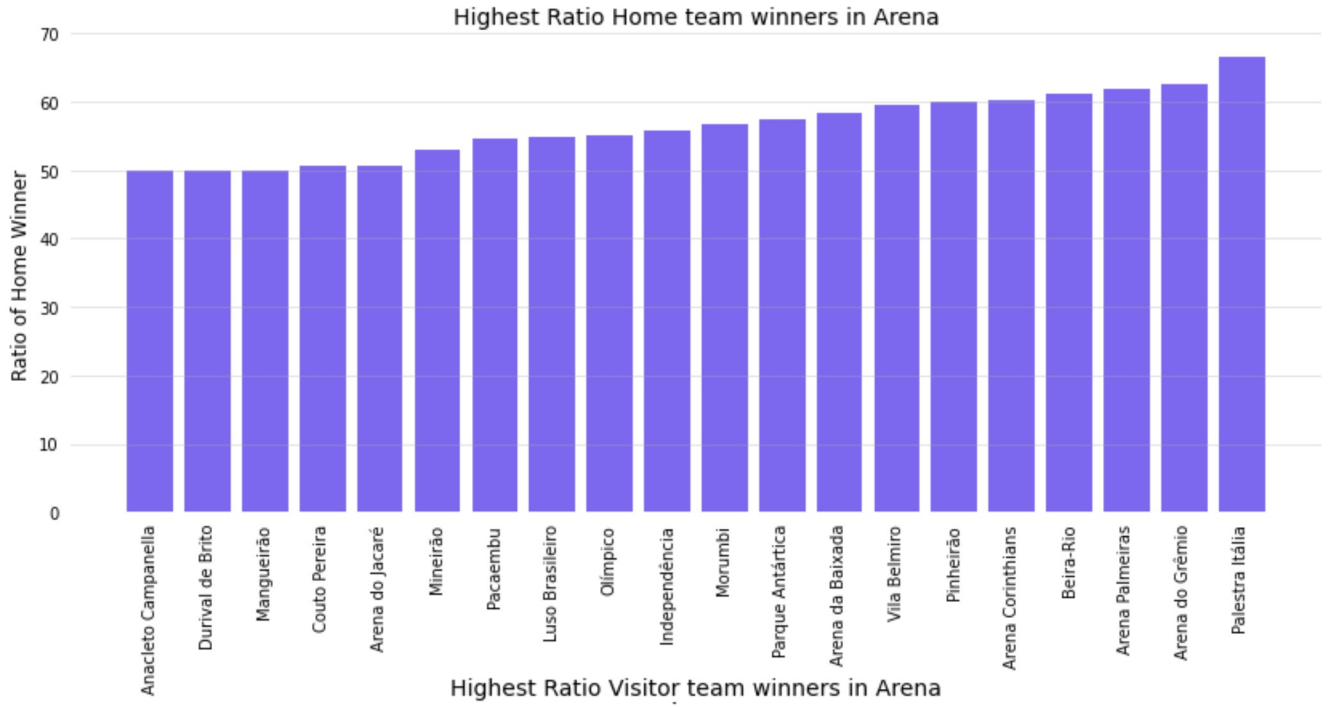
```
arena_count = {}
arena_count_lose = {}
arena = df['Arena'].unique().tolist()
for each in arena:
    try:
        count_wins = df[df['Arena'] == each]['Winner'].value_counts().loc['home']
        count_lose = df[df['Arena'] == each]['Winner'].value_counts().loc['visiting']

        total_rows = df[df['Arena'] == each]['Winner'].shape[0]
        if(count_wins > 10):
            arena_count[each] = arena_count.get(each, (count_wins/total_rows)*100 )

        if(count_lose > 10):
            arena_count_lose[each] = arena_count_lose.get(each, (count_lose/total_rows)*100 )

    except:
        pass


most_home_winners = sorted(arena_count.items(), key=lambda kv:(kv[1], kv[0]))[-20:]
most_visitor_winners = sorted(arena_count_lose.items(), key=lambda kv:(kv[1], kv[0]))[-20:]

most_visitor_winners
```

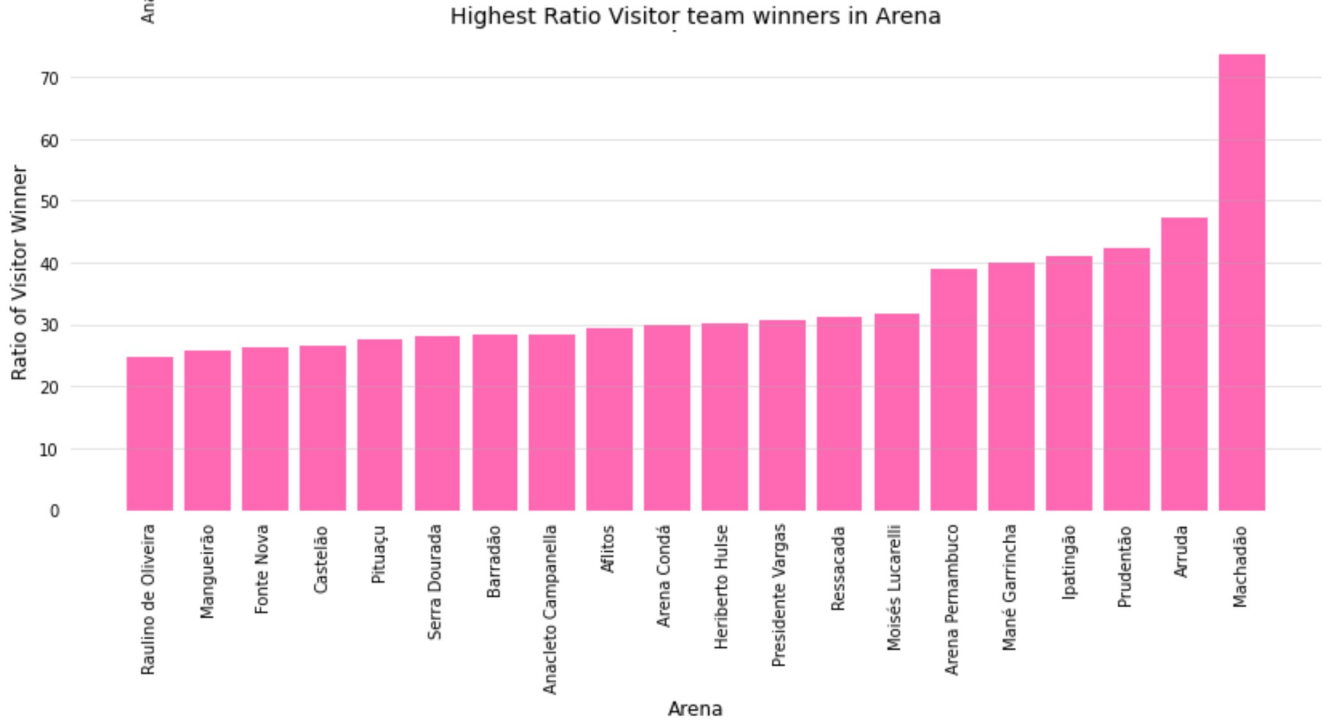
Out[144]:

```
[('Raulino de Oliveira', 24.731182795698924),
 ('Mangueirão', 25.757575757575758),
 ('Fonte Nova', 26.31578947368421),
 ('Castelão', 26.666666666666668),
 ('Pituaçu', 27.659574468085108),
 ('Serra Dourada', 27.986348122866893),
 ('Barradão', 28.26086956521739),
 ('Anacleto Campanella', 28.40909090909091),
 ('Aflitos', 29.333333333333332),
 ('Arena Condá', 29.82456140350877),
 ('Heriberto Hulse', 30.120481927710845),
 ('Presidente Vargas', 30.76923076923077),
 ('Ressacada', 31.30434782608696),
 ('Moisés Lucarelli', 31.843575418994412),
 ('Arena Pernambuco', 38.88888888888889),
 ('Mané Garrincha', 40.0),
 ('Ipatingão', 41.17647058823529),
 ('Prudentão', 42.30769230769231),
 ('Arruda', 47.368421052631575),
 ('Machadão', 73.68421052631578)]
```

In [145…

```
def plotBar(x_axis, y_axis, title, winner, color):
    plt.bar(x_axis, y_axis, color=color)
    plt.xlabel("Arena", size=12)
    plt.ylabel(f"Ratio of {winner} Winner", size=12)
    plt.xticks(rotation=90)


    plt.grid(axis ='y',alpha=0.4)
    plt.tick_params(left = False, bottom=False)

    for spine in plt.gca().spines.values():
        spine.set_visible(False)

    plt.title(title, size=14)


plt.figure(figsize=(14,12))

plt.subplot(2,1,1)
x = [each[0] for each in most_home_winners]
y = [each[1] for each in most_home_winners]
plotBar(x, y, "Highest Ratio Home team winners in Arena", "Home", "mediumslateblue")

plt.subplot(2,1,2)
x = [each[0] for each in most_visitor_winners]
y = [each[1] for each in most_visitor_winners]
plotBar(x, y, "Highest Ratio Visitor team winners in Arena", "Visitor", "hotpink")

plt.subplots_adjust(bottom=0.07,
                    top=0.9,
                    wspace=0.4,
                    hspace=0.4)
```


Highest Ratio Home team winners in Arena
Ratio of Home Winner
70
60
50
40
30
20
10
0
Anacleto Campanella
Durival de Brito
Mangueirão
Couto Pereira
Arena do Jacaré
Mineirão
Pacaembu
Luso Brasileiro
Olímpico
Independência
Morumbi
Parque Antártica
Arena da Baixada
Vila Belmiro
Pinheirão
Arena Corinthians
Beira-Rio
Arena Palmeiras
Arena do Grêmio
Palestra Itália

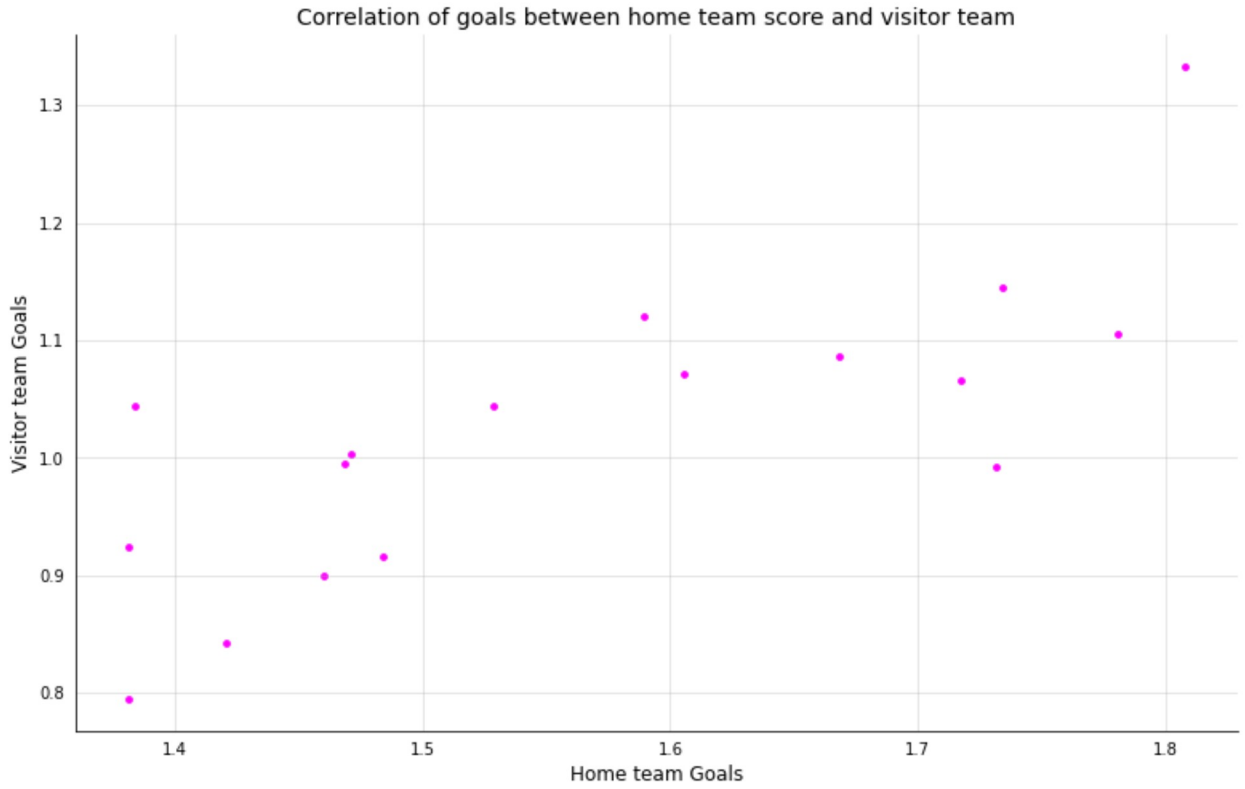

Highest Ratio Visitor team winners in Arena
Ratio of Visitor Winner
70
60
50
40
30
20
10
0
Raulino de Oliveira
Mangueirão
Fonte Nova
Castelão
Pituaçu
Serra Dourada
Barradão
Anacleto Campanella
Áfilitos
Arena Condá
Heriberto Hulse
Presidente Vargas
Ressacada
Moisés Lucarelli
Arena Pernambuco
Mané Garrincha
Ipatingão
Prudentão
Arruda
Machadão
Arena

In [ ]:

9. Correlation between home team score VS visitor team score

In [146…

```
x = df.groupby('Season (Year)').mean()['Home team Goals']
y = df.groupby('Season (Year)').mean()['Visitor team Goals']


plt.figure(figsize=(13,8))
plt.scatter(x, y, color='magenta', s=14)

plt.ylabel("Visitor team Goals", size=12)
plt.xlabel("Home team Goals", size=12)

plt.grid(alpha=0.4)
plt.tick_params(left = False, bottom=False)

count = 1;
for spine in plt.gca().spines.values():
    if(count&1):
        pass;
    else:
        spine.set_visible(False)
    count += 1

plt.title("Correlation of goals between home team score and visitor team", size=14)
plt.show()
```



In [ ]:

10. Comparison between highest goal and lowest goal scored overall.

In [147…

```
df['Total Goals'] = df['Home team Goals'] + df['Visitor team Goals']
df
```

Out[147]:

	ID	Match Date	Season (Year)	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Winner	Arena	Total Goals
0	2003.01.0001	29/03/2003	2003	1	Guarani	Vasco	4	2	SP	RJ	home	Brinco de Ouro	6
1	2003.01.0002	29/03/2003	2003	1	Athletico-PR	Grêmio	2	0	PR	RS	home	Arena da Baixada	2
2	2003.01.0003	30/03/2003	2003	1	Flamengo	Coritiba	1	1	RJ	PR	tie	Maracanã	2
3	2003.01.0004	30/03/2003	2003	1	Goiás	Paysandu	2	2	GO	PA	tie	Serra Dourada	4
4	2003.01.0005	30/03/2003	2003	1	Internacional	Ponte Preta	1	1	RS	SP	tie	Beira-Rio	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6881	2019.38.0376	08/12/2019	2019	38	Vasco	Chapecoense	1	1	RJ	SC	tie	Maracanã	2
6882	2019.38.0377	08/12/2019	2019	38	Botafogo-RJ	Ceará	1	1	RJ	CE	tie	Engenhão	2
6883	2019.38.0378	08/12/2019	2019	38	Avaí	Athletico-PR	0	0	SC	PR	tie	Ressacada	0
6884	2019.38.0379	08/12/2019	2019	38	Goiás	Grêmio	3	2	GO	RS	home	Serra Dourada	5
6885	2019.38.0380	08/12/2019	2019	38	CSA	São Paulo	1	2	AL	SP	visiting	Rei Pelé	3

6886 rows × 13 columns

In [148...

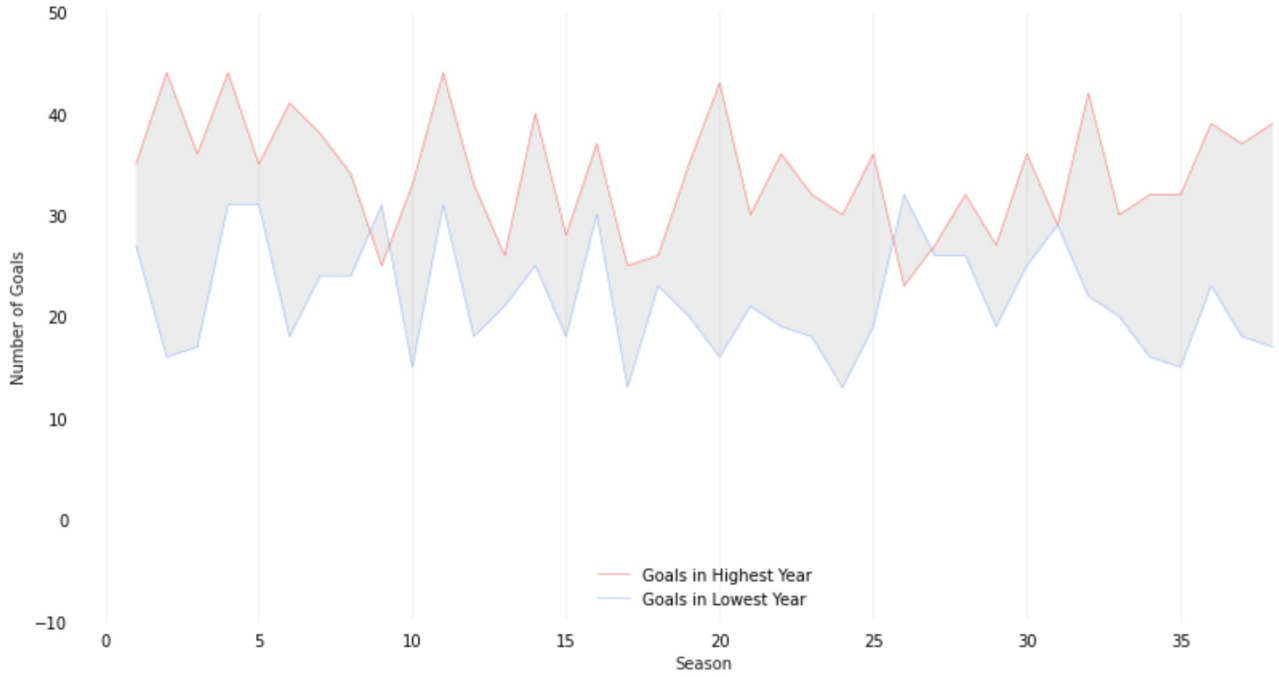
```
df.groupby('Season (Year)').sum()
# 2018 min
# 2003 max
```

Out[148]:

Season (Year)	Season round	Home team Goals	Visitor team Goals	Total Goals
2003	12973	983	610	1593
2004	12972	948	588	1536
2005	9933	835	616	1451
2006	7410	604	426	1030
2007	7410	634	413	1047
2008	7410	658	377	1035
2009	7410	659	435	1094
2010	7410	581	397	978
2011	7410	610	407	1017
2012	7410	559	381	940
2013	7410	558	378	936
2014	7410	540	320	860
2015	7410	555	342	897
2016	7410	564	348	912
2017	7410	526	397	923
2018	7410	525	302	827
2019	7411	525	351	876

In [150…

```
#plotting
plt.figure(figsize=(14,7))
x1 = df[df['Season (Year)'] == 2003].groupby('Season round').sum().index.tolist()[:38]
y1 = df[df['Season (Year)'] == 2003].groupby('Season round').sum()['Total Goals'].tolist()[:38]
plt.plot(x1, y1, c='red', alpha=0.6 , linewidth=0.5, label='Goals in Highest Year')

x2 = df[df['Season (Year)'] == 2018].groupby('Season round').sum().index.tolist()
y2 = df[df['Season (Year)'] == 2018].groupby('Season round').sum()['Total Goals'].tolist()
plt.plot(x2, y2, c='cornflowerblue', alpha=0.7 , linewidth=0.5,label='Goals in Lowest Year')

plt.legend(loc=8, frameon=False)

# changing x-axis' names
plt.ylim([-10,50])
plt.grid(axis='x',alpha=0.15)
plt.xlabel('Season' , alpha=0.9)
plt.ylabel('Number of Goals' , alpha=0.9)

# dejunckifying
plt.tick_params(left = False, bottom=False)
plt.fill_between(x1, y1, y2,
                 facecolor = 'gray', alpha = 0.15)

for spine in plt.gca().spines.values():
    spine.set_visible(False)
```



In [ ]:

In [ ]:

11. Number of home team wins, ties, visitor team wins each year.

In [151…

```
df_wins_year = df.groupby(['Season (Year)', 'Winner']).count()
df_wins_year
```

Out[151]:

Season (Year)	Winner	ID	Match Date	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Arena	Total Goals
2003	**home**	297	297	297	297	297	297	297	297	297	297	297
	tie	142	142	142	142	142	142	142	142	142	142	142
	visiting	113	113	113	113	113	113	113	113	113	113	113
2004	**home**	288	288	288	288	288	288	288	288	288	288	288
	tie	140	140	140	140	140	140	140	140	140	140	140
	visiting	124	124	124	124	124	124	124	124	124	124	124
2005	**home**	235	235	235	235	235	235	235	235	235	235	235
	tie	103	103	103	103	103	103	103	103	103	103	103
	visiting	124	124	124	124	124	124	124	124	124	124	124
2006	**home**	191	191	191	191	191	191	191	191	191	191	191
	tie	97	97	97	97	97	97	97	97	97	97	97
	visiting	92	92	92	92	92	92	92	92	92	92	92
2007	**home**	192	192	192	192	192	192	192	192	192	192	192
	tie	90	90	90	90	90	90	90	90	90	90	90
	visiting	98	98	98	98	98	98	98	98	98	98	98
2008	**home**	208	208	208	208	208	208	208	208	208	208	208
	tie	96	96	96	96	96	96	96	96	96	96	96
	visiting	76	76	76	76	76	76	76	76	76	76	76
2009	**home**	195	195	195	195	195	195	195	195	195	195	195
	tie	102	102	102	102	102	102	102	102	102	102	102
	visiting	83	83	83	83	83	83	83	83	83	83	83
2010	**home**	179	179	179	179	179	179	179	179	179	179	179
	tie	118	118	118	118	118	118	118	118	118	118	118
	visiting	83	83	83	83	83	83	83	83	83	83	83
2011	**home**	184	184	184	184	184	184	184	184	184	184	184
	tie	105	105	105	105	105	105	105	105	105	105	105
	visiting	91	91	91	91	91	91	91	91	91	91	91
2012	**home**	183	183	183	183	183	183	183	183	183	183	183

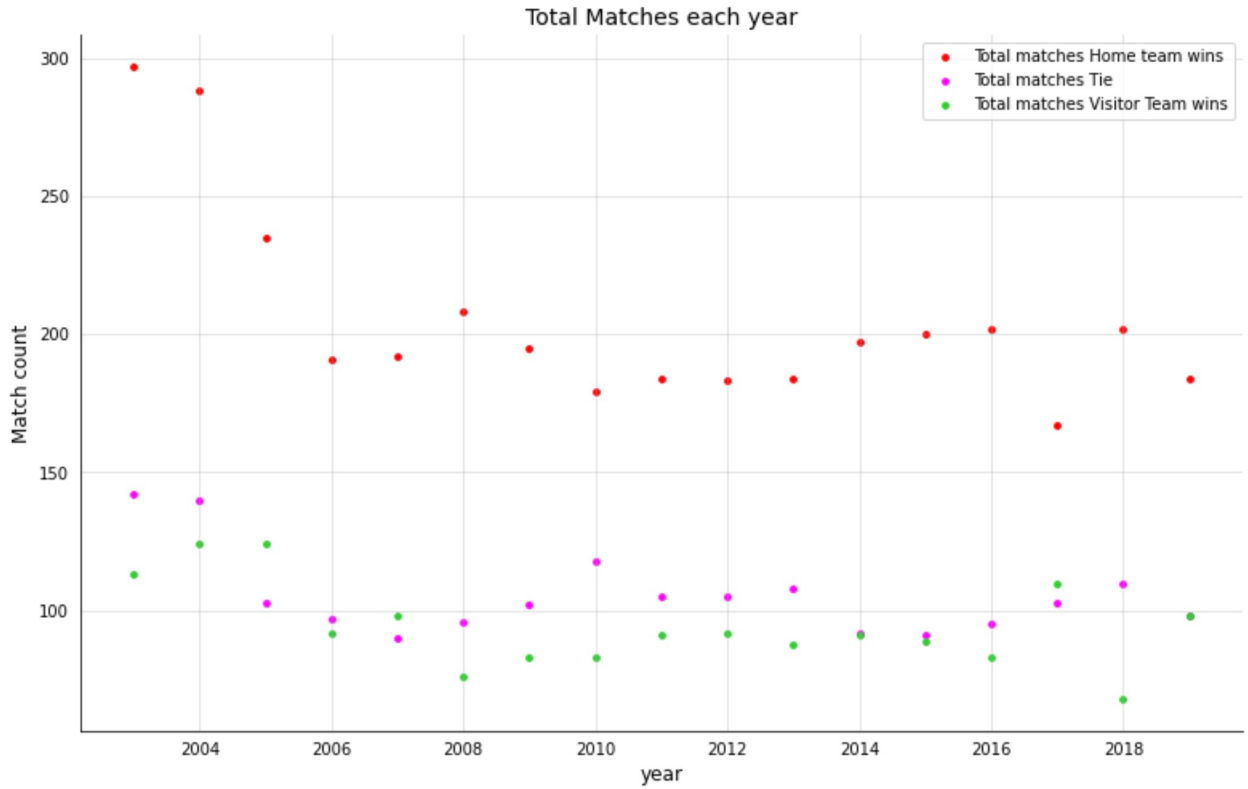
Season (Year)	Winner	ID	Match Date	Season round	Home team	Visitor team	Home team Goals	Visitor team Goals	Home team Brazil's state	Visitor team Brazil's state	Arena	Total Goals
	tie	105	105	105	105	105	105	105	105	105	105	105
	visiting	92	92	92	92	92	92	92	92	92	92	92
2013	home	184	184	184	184	184	184	184	184	184	184	184
	tie	108	108	108	108	108	108	108	108	108	108	108
	visiting	88	88	88	88	88	88	88	88	88	88	88
2014	home	197	197	197	197	197	197	197	197	197	197	197
	tie	92	92	92	92	92	92	92	92	92	92	92
	visiting	91	91	91	91	91	91	91	91	91	91	91
2015	home	200	200	200	200	200	200	200	200	200	200	200
	tie	91	91	91	91	91	91	91	91	91	91	91
	visiting	89	89	89	89	89	89	89	89	89	89	89
2016	home	202	202	202	202	202	202	202	202	202	202	202
	tie	95	95	95	95	95	95	95	95	95	95	95
	visiting	83	83	83	83	83	83	83	83	83	83	83
2017	home	167	167	167	167	167	167	167	167	167	167	167
	tie	103	103	103	103	103	103	103	103	103	103	103
	visiting	110	110	110	110	110	110	110	110	110	110	110
2018	home	202	202	202	202	202	202	202	202	202	202	202
	tie	110	110	110	110	110	110	110	110	110	110	110
	visiting	68	68	68	68	68	68	68	68	68	68	68
2019	home	184	184	184	184	184	184	184	184	184	184	184
	tie	98	98	98	98	98	98	98	98	98	98	98
	visiting	98	98	98	98	98	98	98	98	98	98	98

In [153…

```
count = 0
year = list(range(2003, 2020))
home = []
tie = []
visiting = []

for each in df_wins_year['ID']:
    if(count % 3 == 0):
        home.append(each)

    elif(count % 3 == 1):
        tie.append(each)

    else:
        visiting.append(each)

    count += 1
```

In [154…

```
plt.figure(figsize=(13,8))
plt.scatter(year, home, color='red', label="Total matches Home team wins", s=14)
plt.scatter(year, tie, color='magenta', label="Total matches Tie", s=14)
plt.scatter(year, visiting, color='limegreen', label="Total matches Visitor Team wins", s=14)

plt.ylabel("Match count", size=12)
plt.xlabel("year", size=12)

plt.grid(alpha=0.4)
plt.tick_params(left = False, bottom=False)

count = 1;
for spine in plt.gca().spines.values():
    if(count&1):
        pass;
    else:
        spine.set_visible(False)
    count += 1

plt.title("Total Matches each year", size=14)
plt.legend()
plt.show()
```


Total Matches each year
Total matches Home team wins
Total matches Tie
Total matches Visitor Team wins
Match count
year
300
250
200
150
100
2004
2006
2008
2010
2012
2014
2016
2018

In [ ]:

In [ ]:

In [ ]: