In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
matches_df = pd.read_csv('C:/Users/payal/Desktop/Payal/Internkaksha Project/FI
matches_df.head(3)
```

Out[2]:

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico	
1	1930.0	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium	
2	1930.0	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil	

In [3]:
```
players_df = pd.read_csv('C:/Users/payal/Desktop/Payal/Internkaksha Project/FI
players_df.head(3)
```

Out[3]:

	RoundID	MatchID	Team Initials	Coach Name	Line-up	Shirt Number	Player Name	Position	Event
0	201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Alex THEPOT	GK	NaN
1	201	1096	MEX	LUQUE Juan (MEX)	S	0	Oscar BONFIGLIO	GK	NaN
2	201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Marcel LANGILLER	NaN	G40'

In [4]:
```
cups_df = pd.read_csv('C:/Users/payal/Desktop/Payal/Internkaksha Project/FIFA
cups_df .head(3)
```

Out[4]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams	M
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	13	
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	16	
2	1938	France	Italy	Hungary	Brazil	Sweden	84	15	

In [5]:
```
print ("the shape of matches data frame is: ",matches_df.shape)
print ("the shape of players data frame is: ",players_df.shape)
print ("the shape of cups data frame is: ",cups_df.shape)
```

```
the shape of matches data frame is:  (4572, 20)
the shape of players data frame is:  (37784, 9)
the shape of cups data frame is:  (20, 10)
```

In [6]:
```
matches_df.describe()
```

Out[6]:

	Year	Home Team Goals	Away Team Goals	Attendance	Half-time Home Goals	Half-time Away Goals	RoundID
count	852.000000	852.000000	852.000000	850.000000	852.000000	852.000000	8.520000e+0
mean	1985.089202	1.811033	1.022300	45164.800000	0.708920	0.428404	1.066177e+0
std	22.448825	1.610255	1.087573	23485.249247	0.937414	0.691252	2.729613e+0
min	1930.000000	0.000000	0.000000	2000.000000	0.000000	0.000000	2.010000e+0
25%	1970.000000	1.000000	0.000000	30000.000000	0.000000	0.000000	2.620000e+0
50%	1990.000000	2.000000	1.000000	41579.500000	0.000000	0.000000	3.370000e+0
75%	2002.000000	3.000000	2.000000	61374.500000	1.000000	1.000000	2.497220e+0
max	2014.000000	10.000000	7.000000	173850.000000	6.000000	5.000000	9.741060e+0

In [7]:
```
matches_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4572 entries, 0 to 4571
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Year                  852 non-null    float64
 1   Datetime              852 non-null    object 
 2   Stage                 852 non-null    object 
 3   Stadium               852 non-null    object 
 4   City                  852 non-null    object 
 5   Home Team Name        852 non-null    object 
 6   Home Team Goals       852 non-null    float64
 7   Away Team Goals       852 non-null    float64
 8   Away Team Name        852 non-null    object 
 9   Win conditions        852 non-null    object 
 10  Attendance            850 non-null    float64
 11  Half-time Home Goals  852 non-null    float64
 12  Half-time Away Goals  852 non-null    float64
 13  Referee               852 non-null    object 
 14  Assistant 1           852 non-null    object 
 15  Assistant 2           852 non-null    object 
 16  RoundID               852 non-null    float64
 17  MatchID               852 non-null    float64
 18  Home Team Initials    852 non-null    object 
 19  Away Team Initials    852 non-null    object 
dtypes: float64(8), object(12)
memory usage: 714.5+ KB
```

In [8]:
```
# convert the type of Datetime column from object to datetime
matches_df['Datetime'] = pd.to_datetime(matches_df['Datetime'])
# replace all spaces in column names by underscore sign
matches_df.columns = [c.replace(' ', '_') for c in matches_df.columns]
```

In [9]:
```
matches_df.info()
```

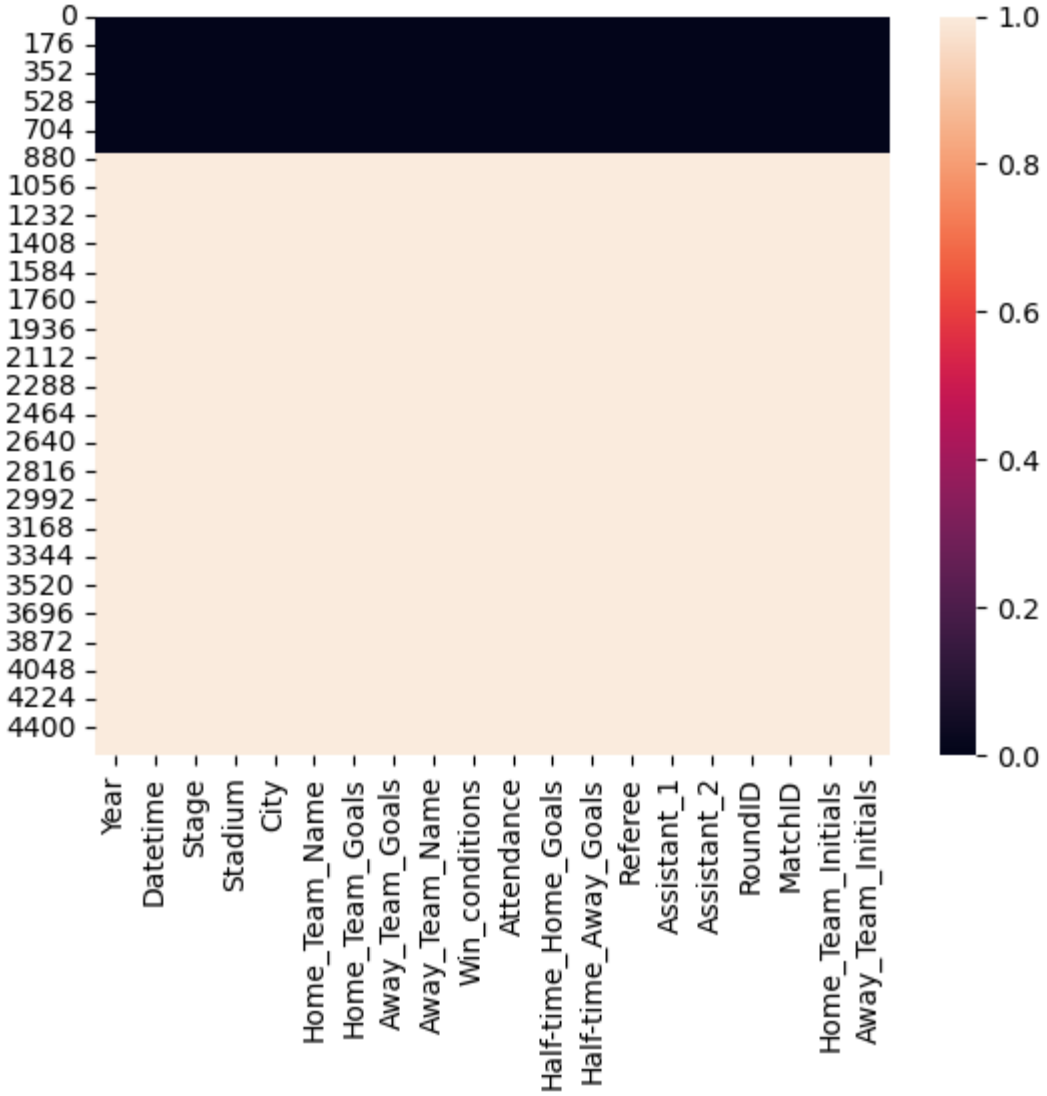
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4572 entries, 0 to 4571
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   Year                  852 non-null    float64       
 1   Datetime              852 non-null    datetime64[ns]
 2   Stage                 852 non-null    object        
 3   Stadium               852 non-null    object        
 4   City                  852 non-null    object        
 5   Home_Team_Name        852 non-null    object        
 6   Home_Team_Goals       852 non-null    float64       
 7   Away_Team_Goals       852 non-null    float64       
 8   Away_Team_Name        852 non-null    object        
 9   Win_conditions        852 non-null    object        
 10  Attendance            850 non-null    float64       
 11  Half-time_Home_Goals  852 non-null    float64       
 12  Half-time_Away_Goals  852 non-null    float64       
 13  Referee               852 non-null    object        
 14  Assistant_1           852 non-null    object        
 15  Assistant_2           852 non-null    object        
 16  RoundID               852 non-null    float64       
 17  MatchID               852 non-null    float64       
 18  Home_Team_Initials    852 non-null    object        
 19  Away_Team_Initials    852 non-null    object        
dtypes: datetime64[ns](1), float64(8), object(11)
memory usage: 714.5+ KB
```

In [10]:
```
# checking if i have some null values
matches_df.isnull().sum()
```

Out[10]:
```
Year                    3720
Datetime                3720
Stage                   3720
Stadium                 3720
City                    3720
Home_Team_Name          3720
Home_Team_Goals         3720
Away_Team_Goals         3720
Away_Team_Name          3720
Win_conditions          3720
Attendance              3722
Half-time_Home_Goals    3720
Half-time_Away_Goals    3720
Referee                 3720
Assistant_1             3720
Assistant_2             3720
RoundID                 3720
MatchID                 3720
Home_Team_Initials      3720
Away_Team_Initials      3720
dtype: int64
```

In [11]:
```
# visualization of null values
import seaborn as sns
sns.heatmap(matches_df.isnull(), cbar=True)
```

Out[11]: <Axes: >



In [12]:
```
matches_df.last_valid_index()
```

Out[12]: 851

In [13]:
```
matches_df[850:860]
```

Out[13]:

	Year	Datetime	Stage	Stadium	City	Home_Team_Name	Home_Team_Goals	Away_
850	2014.0	2014-07-12 17:00:00	Play-off for third place	Estadio Nacional	Brasilia	Brazil	0.0	
851	2014.0	2014-07-13 16:00:00	Final	Estadio do Maracana	Rio De Janeiro	Germany	1.0	
852	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
853	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
854	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
855	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
856	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
857	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
858	NaN	NaT	NaN	NaN	NaN	NaN	NaN	
859	NaN	NaT	NaN	NaN	NaN	NaN	NaN	

In [14]:
```
matches_df = matches_df.dropna()
```

In [15]:
```
matches_df.isnull().sum()
```

Out[15]:
```
Year                    0
Datetime                0
Stage                   0
Stadium                 0
City                    0
Home_Team_Name          0
Home_Team_Goals         0
Away_Team_Goals         0
Away_Team_Name          0
Win_conditions          0
Attendance              0
Half-time_Home_Goals    0
Half-time_Away_Goals    0
Referee                 0
Assistant_1             0
Assistant_2             0
RoundID                 0
MatchID                 0
Home_Team_Initials      0
Away_Team_Initials      0
dtype: int64
```

In [16]:
```
matches_df_dup = matches_df[matches_df.duplicated(keep='last')]
matches_df_dup
```

Out[16]:

	Year	Datetime	Stage	Stadium	City	Home_Team_Name	Home_Team_Goals
820	2014.0	2014-06-28 13:00:00	Round of 16	Estadio Mineirao	Belo Horizonte	Brazil	1.0
821	2014.0	2014-06-28 17:00:00	Round of 16	Estadio do Maracana	Rio De Janeiro	Colombia	2.0
822	2014.0	2014-06-30 13:00:00	Round of 16	Estadio Nacional	Brasilia	France	2.0
824	2014.0	2014-07-04 17:00:00	Quarter-finals	Estadio Castelao	Fortaleza	Brazil	2.0
825	2014.0	2014-07-04 13:00:00	Quarter-finals	Estadio do Maracana	Rio De Janeiro	France	0.0
826	2014.0	2014-07-08 17:00:00	Semi-finals	Estadio Mineirao	Belo Horizonte	Brazil	1.0
827	2014.0	2014-07-12 17:00:00	Play-off for third place	Estadio Nacional	Brasilia	Brazil	0.0
828	2014.0	2014-07-13 16:00:00	Final	Estadio do Maracana	Rio De Janeiro	Germany	1.0
829	2014.0	2014-07-09 17:00:00	Semi-finals	Arena de Sao Paulo	Sao Paulo	Netherlands	0.0
830	2014.0	2014-07-05 17:00:00	Quarter-finals	Arena Fonte Nova	Salvador	Netherlands	0.0
831	2014.0	2014-07-05 13:00:00	Quarter-finals	Estadio Nacional	Brasilia	Argentina	1.0
832	2014.0	2014-06-29 13:00:00	Round of 16	Estadio Castelao	Fortaleza	Netherlands	2.0
833	2014.0	2014-06-29 17:00:00	Round of 16	Arena Pernambuco	Recife	Costa Rica	1.0
834	2014.0	2014-07-01 13:00:00	Round of 16	Arena de Sao Paulo	Sao Paulo	Argentina	1.0
835	2014.0	2014-07-01 17:00:00	Round of 16	Arena Fonte Nova	Salvador	Belgium	2.0

In [17]:
```
matches_df_dup1 = matches_df[matches_df.duplicated(keep='first')]
matches_df_dup1
```

Out[17]:

	Year	Datetime	Stage	Stadium	City	Home_Team_Name	Home_Team_Goals
836	2014.0	2014-06-28 13:00:00	Round of 16	Estadio Mineirao	Belo Horizonte	Brazil	1.0
837	2014.0	2014-06-28 17:00:00	Round of 16	Estadio do Maracana	Rio De Janeiro	Colombia	2.0
838	2014.0	2014-06-29 13:00:00	Round of 16	Estadio Castelao	Fortaleza	Netherlands	2.0
839	2014.0	2014-06-29 17:00:00	Round of 16	Arena Pernambuco	Recife	Costa Rica	1.0
840	2014.0	2014-06-30 13:00:00	Round of 16	Estadio Nacional	Brasilia	France	2.0
842	2014.0	2014-07-01 13:00:00	Round of 16	Arena de Sao Paulo	Sao Paulo	Argentina	1.0
843	2014.0	2014-07-01 17:00:00	Round of 16	Arena Fonte Nova	Salvador	Belgium	2.0
844	2014.0	2014-07-04 13:00:00	Quarter-finals	Estadio do Maracana	Rio De Janeiro	France	0.0
845	2014.0	2014-07-04 17:00:00	Quarter-finals	Estadio Castelao	Fortaleza	Brazil	2.0
846	2014.0	2014-07-05 13:00:00	Quarter-finals	Estadio Nacional	Brasilia	Argentina	1.0
847	2014.0	2014-07-05 17:00:00	Quarter-finals	Arena Fonte Nova	Salvador	Netherlands	0.0
848	2014.0	2014-07-08 17:00:00	Semi-finals	Estadio Mineirao	Belo Horizonte	Brazil	1.0
849	2014.0	2014-07-09 17:00:00	Semi-finals	Arena de Sao Paulo	Sao Paulo	Netherlands	0.0
850	2014.0	2014-07-12 17:00:00	Play-off for third place	Estadio Nacional	Brasilia	Brazil	0.0
851	2014.0	2014-07-13 16:00:00	Final	Estadio do Maracana	Rio De Janeiro	Germany	1.0

In [18]:
```
matches_df_dup.shape
```

Out[18]:
```
(15, 20)
```

In [19]:
```
matches_df = matches_df.drop_duplicates()
```

In [20]:
```
matches_df.shape
```

Out[20]:
```
(835, 20)
```

In [21]:
```
matches_df.columns
```

Out[21]:
```
Index(['Year', 'Datetime', 'Stage', 'Stadium', 'City', 'Home_Team_Name',
       'Home_Team_Goals', 'Away_Team_Goals', 'Away_Team_Name',
       'Win_conditions', 'Attendance', 'Half-time_Home_Goals',
       'Half-time_Away_Goals', 'Referee', 'Assistant_1', 'Assistant_2',
       'RoundID', 'MatchID', 'Home_Team_Initials', 'Away_Team_Initials'],
      dtype='object')
```

In [22]:
```
del matches_df["RoundID"]
```

In [23]:
```
matches_df.columns
```

Out[23]:
```
Index(['Year', 'Datetime', 'Stage', 'Stadium', 'City', 'Home_Team_Name',
       'Home_Team_Goals', 'Away_Team_Goals', 'Away_Team_Name',
       'Win_conditions', 'Attendance', 'Half-time_Home_Goals',
       'Half-time_Away_Goals', 'Referee', 'Assistant_1', 'Assistant_2',
       'MatchID', 'Home_Team_Initials', 'Away_Team_Initials'],
      dtype='object')
```

In [24]:
```
matches_df['Goals'] = matches_df['Home_Team_Goals'] + matches_df['Away_Team_Go
```

In [25]:
```
matches_df.Goals.head()
```

Out[25]:
```
0    5.0
1    3.0
2    3.0
3    4.0
4    1.0
Name: Goals, dtype: float64
```

In [26]:
```
def outcome(matches_df):
    if matches_df['Home_Team_Goals'] > matches_df['Away_Team_Goals']:
        return 'Home_Team_Win'
    if matches_df['Home_Team_Goals'] < matches_df['Away_Team_Goals']:
        return 'Away_Team_Win'
    return 'DRAW'

matches_df['outcome_of_the_match'] = matches_df.apply(lambda x: outcome(x), ax
```

In [27]:
```
matches_df.outcome_of_the_match.head()
```

Out[27]:
```
0    Home_Team_Win
1    Home_Team_Win
2    Home_Team_Win
3    Home_Team_Win
4    Home_Team_Win
Name: outcome_of_the_match, dtype: object
```

In [28]:
```
cups_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Year            20 non-null     int64 
 1   Country         20 non-null     object
 2   Winner          20 non-null     object
 3   Runners-Up      20 non-null     object
 4   Third           20 non-null     object
 5   Fourth          20 non-null     object
 6   GoalsScored     20 non-null     int64 
 7   QualifiedTeams  20 non-null     int64 
 8   MatchesPlayed   20 non-null     int64 
 9   Attendance      20 non-null     object
dtypes: int64(4), object(6)
memory usage: 1.7+ KB
```

In [29]:
```
cups_df.describe()
```

Out[29]:

	Year	GoalsScored	QualifiedTeams	MatchesPlayed
count	20.000000	20.000000	20.000000	20.000000
mean	1974.800000	118.950000	21.250000	41.800000
std	25.582889	32.972836	7.268352	17.218717
min	1930.000000	70.000000	13.000000	17.000000
25%	1957.000000	89.000000	16.000000	30.500000
50%	1976.000000	120.500000	16.000000	38.000000
75%	1995.000000	145.250000	26.000000	55.000000
max	2014.000000	171.000000	32.000000	64.000000

In [30]:
```
cups_df.isnull().sum()
```

Out[30]:
```
Year              0
Country           0
Winner            0
Runners-Up        0
Third             0
Fourth            0
GoalsScored       0
QualifiedTeams    0
MatchesPlayed     0
Attendance        0
dtype: int64
```

In [31]:
```
cups_df
```

Out[31]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	Qualified
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	
2	1938	France	Italy	Hungary	Brazil	Sweden	84	
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	
4	1954	Switzerland	Germany FR	Hungary	Austria	Uruguay	140	
5	1958	Sweden	Brazil	Sweden	France	Germany FR	126	
6	1962	Chile	Brazil	Czechoslovakia	Chile	Yugoslavia	89	
7	1966	England	England	Germany FR	Portugal	Soviet Union	89	
8	1970	Mexico	Brazil	Italy	Germany FR	Uruguay	95	
9	1974	Germany	Germany FR	Netherlands	Poland	Brazil	97	
10	1978	Argentina	Argentina	Netherlands	Brazil	Italy	102	
11	1982	Spain	Italy	Germany FR	Poland	France	146	
12	1986	Mexico	Argentina	Germany FR	France	Belgium	132	
13	1990	Italy	Germany FR	Argentina	Italy	England	115	
14	1994	USA	Brazil	Italy	Sweden	Bulgaria	141	
15	1998	France	France	Brazil	Croatia	Netherlands	171	
16	2002	Korea/Japan	Brazil	Germany	Turkey	Korea Republic	161	
17	2006	Germany	Italy	France	Germany	Portugal	147	
18	2010	South Africa	Spain	Netherlands	Germany	Uruguay	145	
19	2014	Brazil	Germany	Argentina	Netherlands	Brazil	171	

In [32]:
```
cups_df = cups_df.replace(['Germany FR'],'Germany')
cups_df
```

Out[32]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	Qualified
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	
2	1938	France	Italy	Hungary	Brazil	Sweden	84	
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	
4	1954	Switzerland	Germany	Hungary	Austria	Uruguay	140	
5	1958	Sweden	Brazil	Sweden	France	Germany	126	
6	1962	Chile	Brazil	Czechoslovakia	Chile	Yugoslavia	89	
7	1966	England	England	Germany	Portugal	Soviet Union	89	
8	1970	Mexico	Brazil	Italy	Germany	Uruguay	95	
9	1974	Germany	Germany	Netherlands	Poland	Brazil	97	
10	1978	Argentina	Argentina	Netherlands	Brazil	Italy	102	
11	1982	Spain	Italy	Germany	Poland	France	146	
12	1986	Mexico	Argentina	Germany	France	Belgium	132	
13	1990	Italy	Germany	Argentina	Italy	England	115	
14	1994	USA	Brazil	Italy	Sweden	Bulgaria	141	
15	1998	France	France	Brazil	Croatia	Netherlands	171	
16	2002	Korea/Japan	Brazil	Germany	Turkey	Korea Republic	161	
17	2006	Germany	Italy	France	Germany	Portugal	147	
18	2010	South Africa	Spain	Netherlands	Germany	Uruguay	145	
19	2014	Brazil	Germany	Argentina	Netherlands	Brazil	171	

In [33]:
```
!pip install jovian
```

```
Requirement already satisfied: jovian in c:\users\payal\anaconda3\lib\site-pa
ckages (0.2.47)
Requirement already satisfied: requests in c:\users\payal\anaconda3\lib\site-
packages (from jovian) (2.29.0)
Requirement already satisfied: uuid in c:\users\payal\anaconda3\lib\site-pack
ages (from jovian) (1.30)
Requirement already satisfied: pyyaml in c:\users\payal\anaconda3\lib\site-pa
ckages (from jovian) (6.0)
Requirement already satisfied: click in c:\users\payal\anaconda3\lib\site-pac
kages (from jovian) (8.0.4)
Requirement already satisfied: colorama in c:\users\payal\anaconda3\lib\site-
packages (from click->jovian) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\payal\ana
conda3\lib\site-packages (from requests->jovian) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\payal\anaconda3\lib\s
ite-packages (from requests->jovian) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\payal\anacon
da3\lib\site-packages (from requests->jovian) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\payal\anaconda
3\lib\site-packages (from requests->jovian) (2023.5.7)
```

In [34]:
```
import jovian
```

In [ ]:
```

```

Exploratory Analysis and Visualization In this third section i will be perfoming several exploratory analysis and visualisation techniques in order to retrieve usefull information from my datasets, and visualize them in different form of visualization figures. i will try to make at least five visualization figures. At the end i will make a small note of interesting insights based on the information retreived from figures and results of the exploratory analysis.

agenda of this section:

i will be performing some operations like(sum, mean) and some statistics of numeric column distribution of numeric cilumns using histograms relationship between columns using different figures then i will end up making a small note

In [35]:
```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (20, 9)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

TODO - TOTAL NUMBER OF GAMES PLAYED EACH YEAR

In [36]:
```
matches_df.columns
```

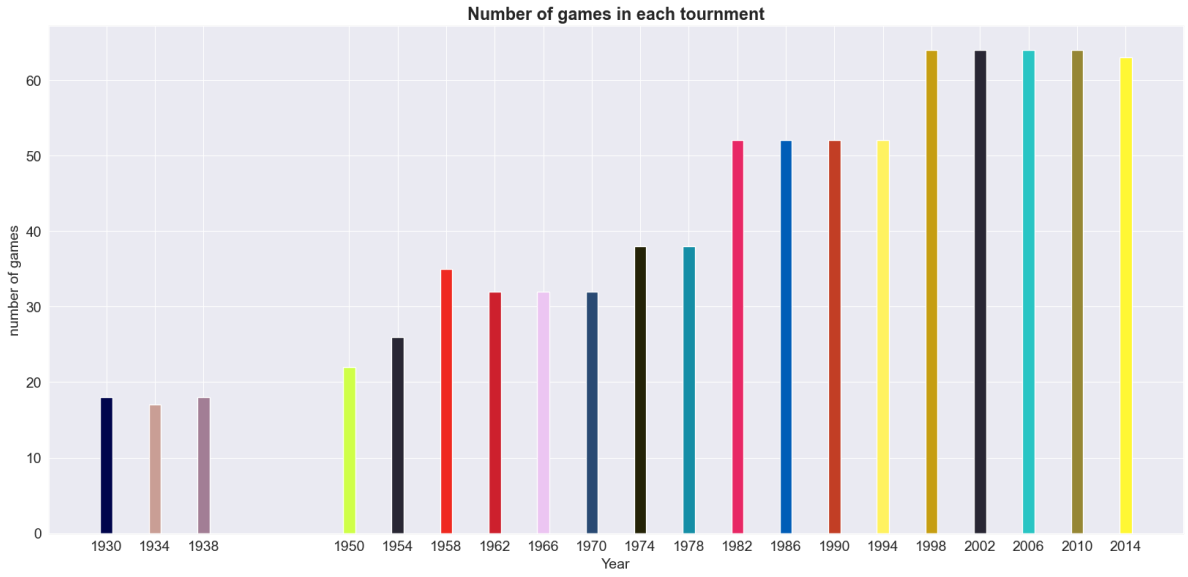
Out[36]:
```
Index(['Year', 'Datetime', 'Stage', 'Stadium', 'City', 'Home_Team_Name',
       'Home_Team_Goals', 'Away_Team_Goals', 'Away_Team_Name',
       'Win_conditions', 'Attendance', 'Half-time_Home_Goals',
       'Half-time_Away_Goals', 'Referee', 'Assistant_1', 'Assistant_2',
       'MatchID', 'Home_Team_Initials', 'Away_Team_Initials', 'Goals',
       'outcome_of_the_match'],
      dtype='object')
```

In [37]:
```
num_of_matches_df = matches_df.groupby('Year')[['MatchID']].count()
num_of_matches_df
```

Out[37]:

Year	MatchID
1930.0	18
1934.0	17
1938.0	18
1950.0	22
1954.0	26
1958.0	35
1962.0	32
1966.0	32
1970.0	32
1974.0	38
1978.0	38
1982.0	52
1986.0	52
1990.0	52
1994.0	52
1998.0	64
2002.0	64
2006.0	64
2010.0	64
2014.0	63

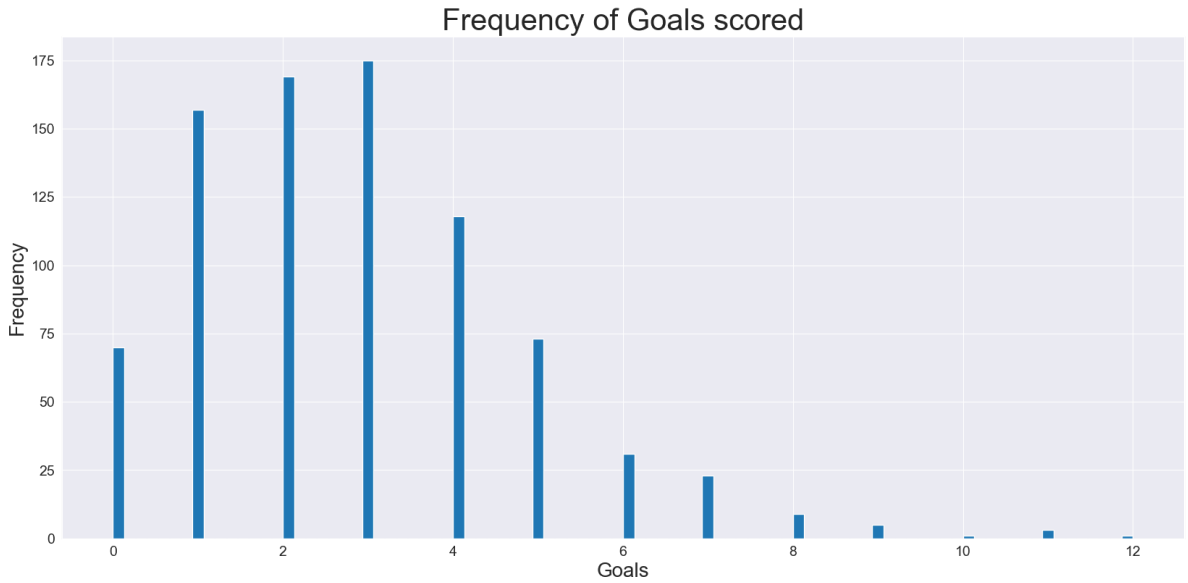
In [38]:
```
plt.title('Number of games in each tournment',fontweight=800)
plt.xlabel('Year')
plt.ylabel('number of games')
plt.xticks(num_of_matches_df.index)
plt.bar(num_of_matches_df.index,num_of_matches_df.MatchID,width=1, color=['#00
                '#294A73','#242307','#158EA6','#E82865','#005DB7','#C23E25','#F
```



This analysis is informing us several things:

timeframe: this datasets have been collected from first world cup of 1930 to the world cup of 2014 in some years, number of matches are equal and in others are different. maybe it's because of different number of team paticipants which can make an increase or decrease of stage. there is a gap between 1938 and 1950, here i think that maybe the cause of this gap is the second world war which that lasted from 1939 to 1945.

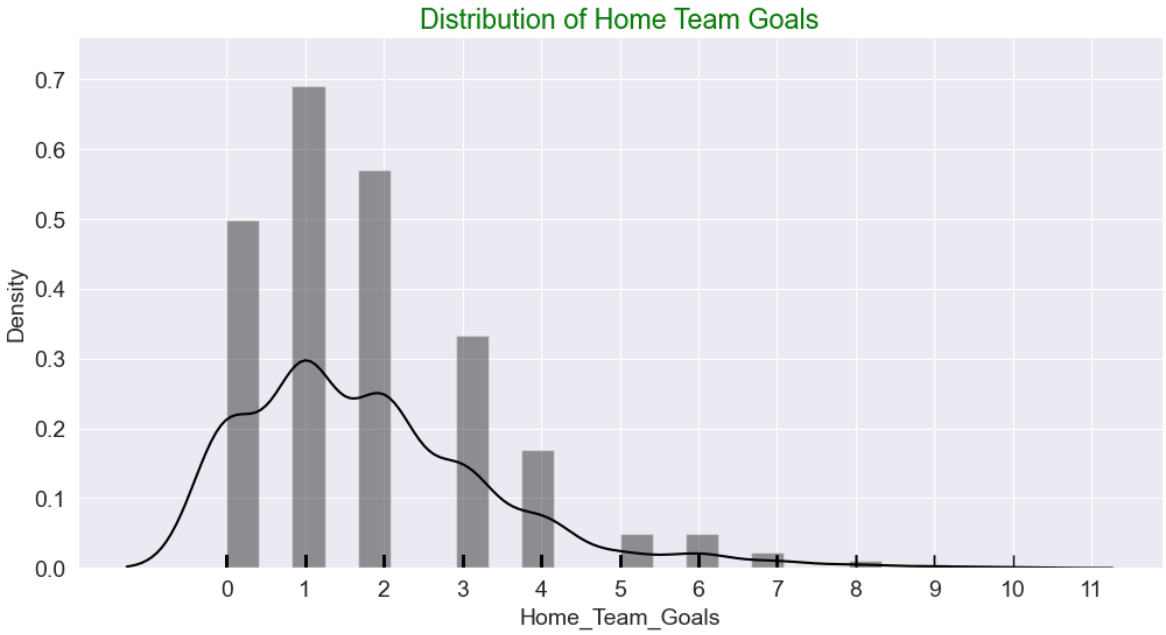
In [39]:
```
matches_df.Goals.plot(kind = 'hist', bins = 90,)
plt.xlabel('Goals', fontsize = 20)
plt.ylabel('Frequency', fontsize = 20)
plt.title('Frequency of Goals scored', fontsize = 30)
plt.show()
```

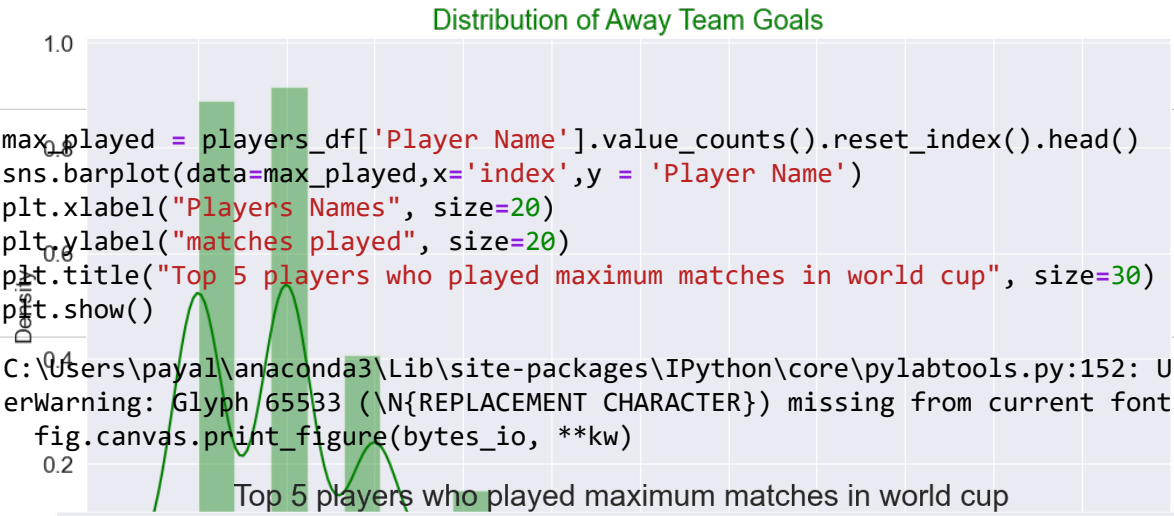


TODO - DISTRIBUTION OF HOME TEAM GEALS AND AWAY TEAM GOALS

In [40]:
```
plt.figure(figsize=(12,13))
plt.subplot(211)
sns.distplot(matches_df["Home_Team_Goals"],color="black",rug=True)
plt.xticks(np.arange(0,12,1))
plt.title("Distribution of Home Team Goals",color='g')


plt.subplot(212)
sns.distplot(matches_df["Away_Team_Goals"],color="g",rug=True)
plt.xticks(np.arange(0,12,1))
plt.title("Distribution of Away Team Goals",color='g')
plt.show()
```

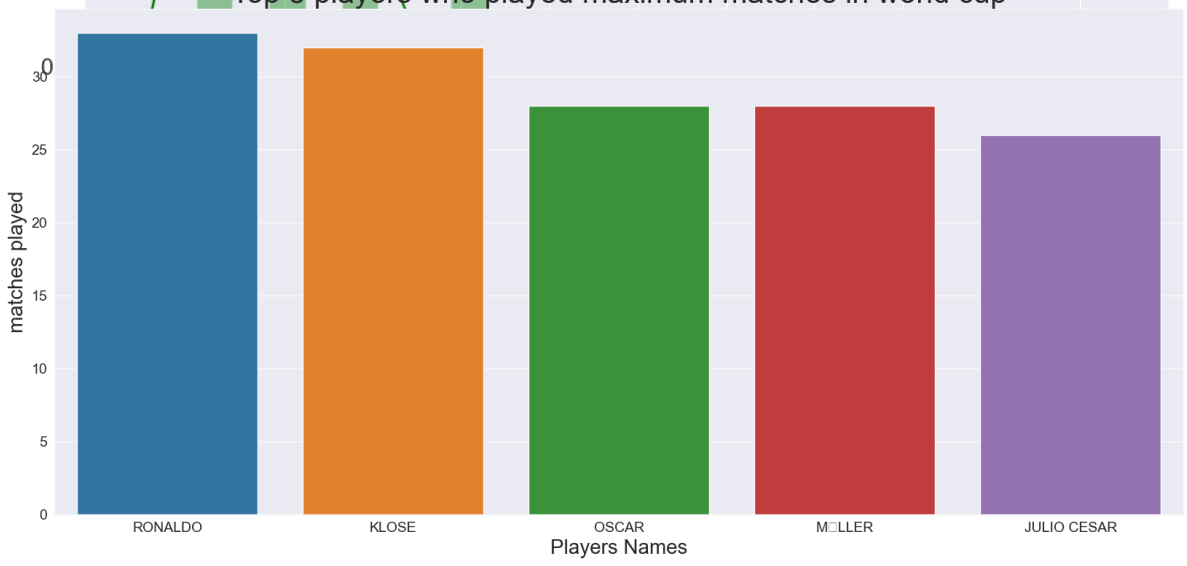
```
C:\Users\payal\AppData\Local\Temp\ipykernel_19640\4176475616.py:3: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(matches_df["Home_Team_Goals"],color="black",rug=True)
C:\Users\payal\AppData\Local\Temp\ipykernel_19640\4176475616.py:9: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(matches_df["Away_Team_Goals"],color="g",rug=True)
```





In [41]:
```
max_played = players_df['Player Name'].value_counts().reset_index().head()
sns.barplot(data=max_played,x='index',y = 'Player Name')
plt.xlabel("Players Names", size=20)
plt.ylabel("matches played", size=20)
plt.title("Top 5 players who played maximum matches in world cup", size=30)
plt.show()
```

```
C:\Users\payal\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: Us
erWarning: Glyph 65533 (\N{REPLACEMENT CHARACTER}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



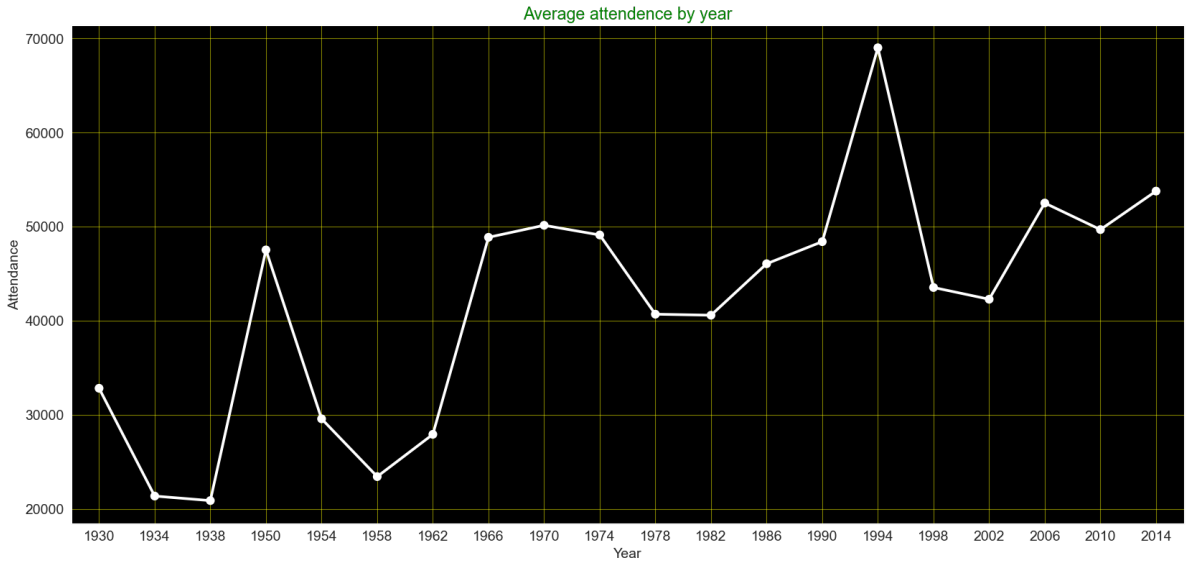
TODO - AVERAGE NUMBER OF ATTENDANTS PER YEAR

In [42]:
```
attendance_df = matches_df.groupby("Year")["Attendance"].mean().reset_index()
attendance_df
```

Out[42]:

	Year	Attendance
0	1930.0	32808.277778
1	1934.0	21352.941176
2	1938.0	20872.222222
3	1950.0	47511.181818
4	1954.0	29561.807692
5	1958.0	23423.142857
6	1962.0	27911.625000
7	1966.0	48847.968750
8	1970.0	50124.218750
9	1974.0	49098.763158
10	1978.0	40678.710526
11	1982.0	40571.596154
12	1986.0	46039.057692
13	1990.0	48388.750000
14	1994.0	68991.115385
15	1998.0	43517.187500
16	2002.0	42268.703125
17	2006.0	52491.234375
18	2010.0	49669.625000
19	2014.0	53758.888889

In [43]:
```
attendance_df["Year"] = attendance_df["Year"].astype(int)
attend = sns.pointplot(x=attendance_df["Year"],y=attendance_df["Attendance"],c
attend.set_facecolor("k")
plt.grid(True,color="yellow",alpha=.5)
plt.title("Average attendence by year",color='g')
plt.show()
```



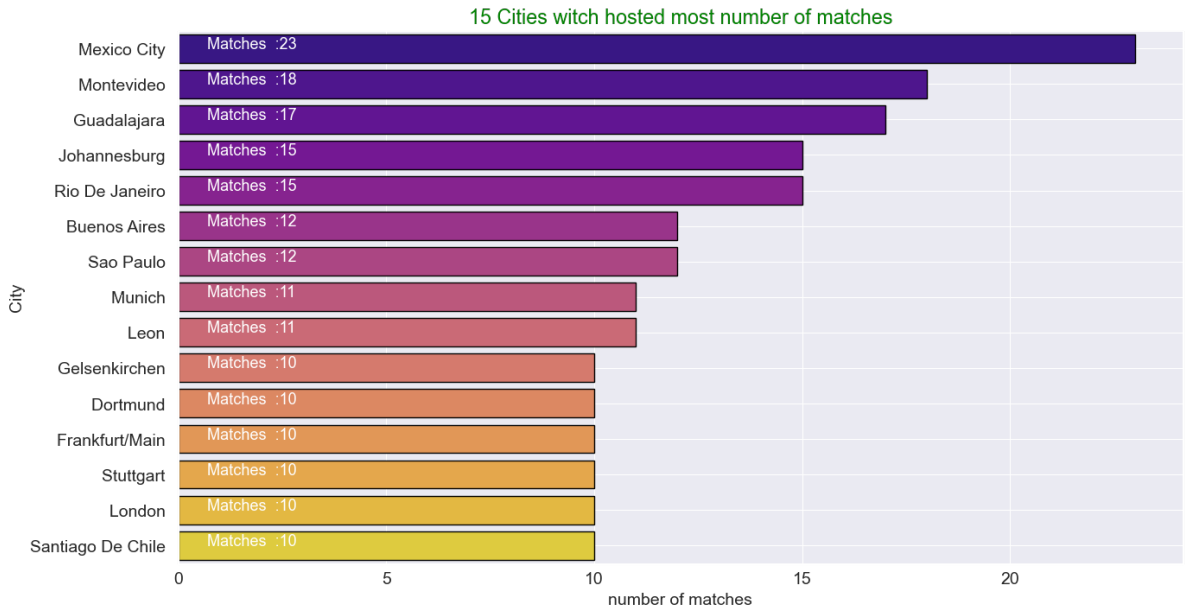
TODO - NUMBER OF GAMES HOSTED BY CITY

In [44]:
```
match_by_city = matches_df["City"].value_counts().reset_index()
match_by_city.head()
```

Out[44]:

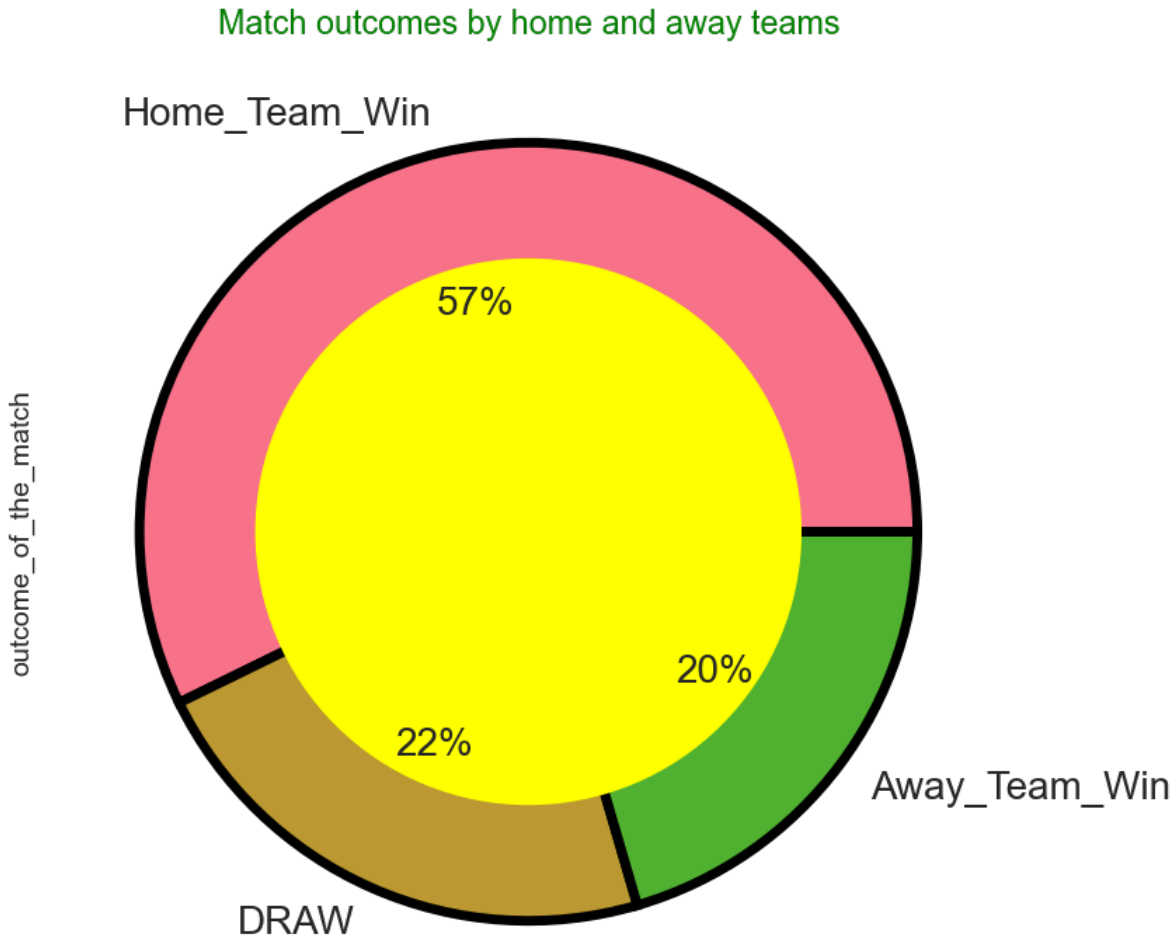
	index	City
0	Mexico City	23
1	Montevideo	18
2	Guadalajara	17
3	Johannesburg	15
4	Rio De Janeiro	15

In [45]:
```
plt.figure(figsize=(15,8))
ax = sns.barplot(y=match_by_city["index"][:15],x = match_by_city["City"][:15],
                 linewidth=1,edgecolor="k"*1)
plt.xlabel("number of matches")
plt.ylabel("City")
plt.grid(True)
plt.title("15 Cities witch hosted most number of matches",color='g')

for i,j in enumerate("Matches  :" + match_by_city["City"][:15].astype(str)):
    ax.text(.7,i,j,fontsize = 13,color="w")
plt.show()
```



TODO - MATCH OUTCOME

In [46]:
```
matches_df["outcome_of_the_match"].value_counts().plot.pie(autopct="%1.0f%%",f
                                                colors = sns.color_palette("husl"),
                                                wedgeprops={"linewidth":5,"edgecolo
                                                shadow=False)
circle = plt.Circle((0,0),.7,color="yellow")
plt.gca().add_artist(circle)
plt.title("Match outcomes by home and away teams",color='g')
plt.show()
```



AFTER DOING SEVERAL EXPLORATORY ANALYSIS AND VISUALIZATIONS, HERE IS SOME INTERESTING INSIGHTS:

The world cup tournment inaugurated in 1930. Brazil is the country having many cups. World cup 1994 is the one with the most number of attendents Mexico city is the city which hosted the most number of matches Home team has a lot of chance to win the match with 57% accuracy

Q1: TODO - Who is the referee who was officiating the most number of FIFA World Cup matches?

In [47]:
```
ref = matches_df[['Referee', 'MatchID']]
ref.head()
```

Out[47]:

	Referee	MatchID
0	LOMBARDI Domingo (URU)	1096.0
1	MACIAS Jose (ARG)	1090.0
2	TEJADA Anibal (URU)	1093.0
3	WARNKEN Alberto (CHI)	1098.0
4	REGO Gilberto (BRA)	1085.0

In [48]:
```
ref = ref.groupby('Referee').count()
ref.head()
```

Out[48]:

Referee	MatchID
ABD EL FATAH Essam (EGY)	1
ADAIR John (NIR)	1
AGNOLIN Luigi (ITA)	4
AGUILAR ELIZALDE Abel (MEX)	1
AGUILAR Joel (SLV)	2

In [49]:
```
ref.sort_values("MatchID", ascending = False).head()
```

Out[49]:

Referee	MatchID
Ravshan IRMATOV (UZB)	9
QUINIOU Joel (FRA)	8
LARRIONDA Jorge (URU)	8
ARCHUNDIA Benito (MEX)	8
LANGENUS Jean (BEL)	7

In [50]:
```
ref.sort_values("MatchID", ascending = False).head(1)
```

Out[50]:

Referee	MatchID
Ravshan IRMATOV (UZB)	9

Q2: TODO - Which country that won the highest number of cups?

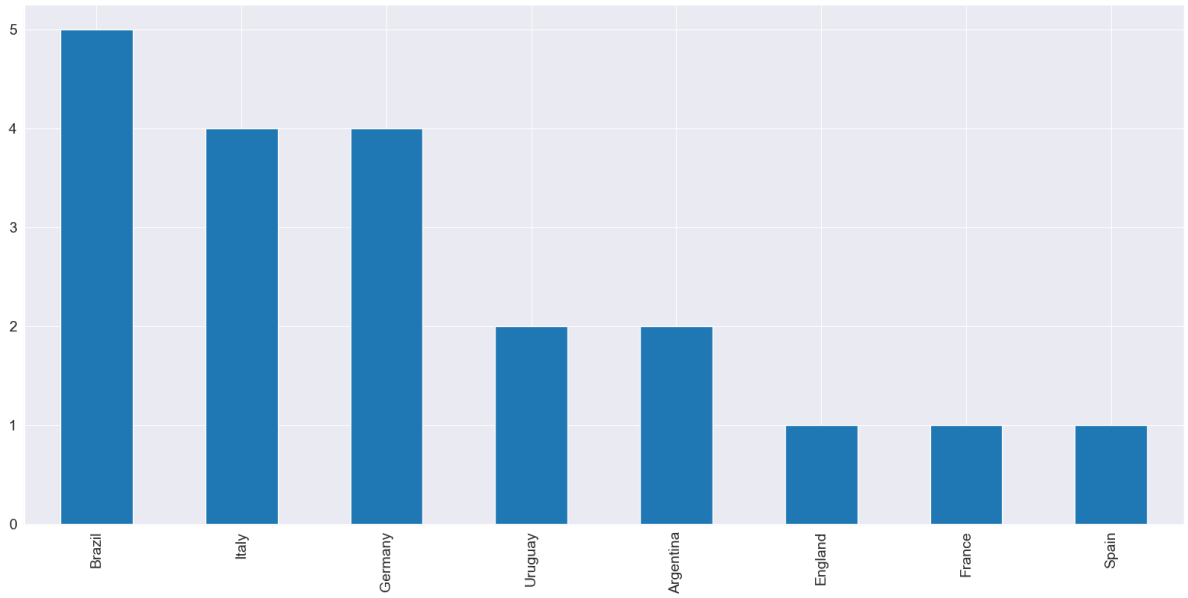
In [51]:
```
cups_winner = cups_df["Winner"].value_counts().reset_index()
cups_winner
```

Out[51]:

	index	Winner
0	Brazil	5
1	Italy	4
2	Germany	4
3	Uruguay	2
4	Argentina	2
5	England	1
6	France	1
7	Spain	1

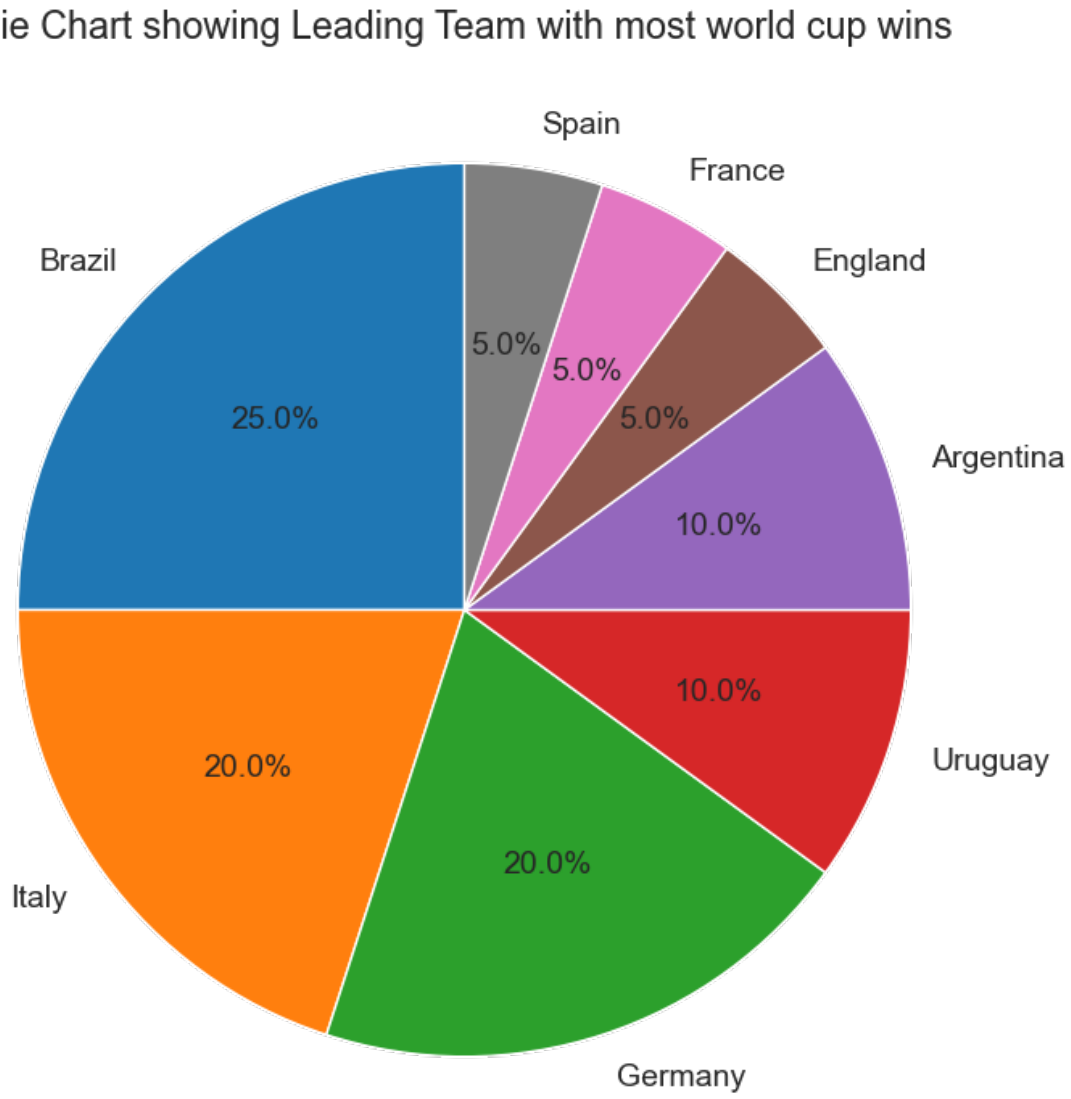
In [52]:
```
cups_df.Winner.value_counts().plot(kind='bar')
```

Out[52]: <Axes: >



Q3: TODO - which country has the highest percentage of FIFA world cup?

In [53]:
```
plt.pie(x=cups_winner['Winner'],labels=cups_winner['index'],autopct='%1.1f%%',
plt.title("Pie Chart showing Leading Team with most world cup wins")
plt.show()
```



Q4: TODO - What are the countries which managed to win a cup as a hoster of the tournament?

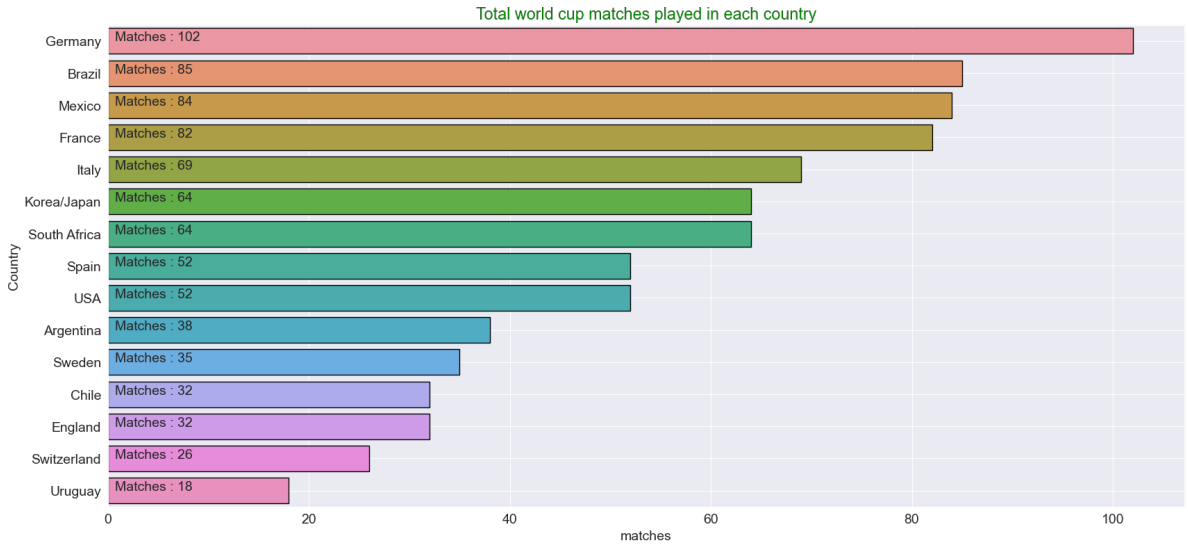
In [54]:
```
hoster_is_winner = cups_df[(cups_df['Winner']==cups_df['Country'])]
hoster_is_winner
```

Out[54]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	1
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	1
7	1966	England	England	Germany	Portugal	Soviet Union	89	1
9	1974	Germany	Germany	Netherlands	Poland	Brazil	97	1
10	1978	Argentina	Argentina	Netherlands	Brazil	Italy	102	1
15	1998	France	France	Brazil	Croatia	Netherlands	171	3

Q5: TODO - which country that hosted alot of number of matches

In [55]:
```
count_host = matches_df[["Year","Stadium","City","MatchID"]]
cup_count  = cups_df[["Year","Country"]]
count_host = count_host.merge(cup_count,left_on="Year",right_on="Year",how="le
count_host["std_cty"] = count_host["Stadium"] +" , "+ count_host["City"]

grouped  =  count_host.groupby("Country")["MatchID"].nunique().reset_index()
grouped  =  grouped.sort_values(by= "MatchID",ascending=False)
grouped

ax = sns.barplot(x="MatchID",y="Country",
                 data=grouped,
                 linewidth=1,
                 edgecolor="k")

for i,j in enumerate("Matches : " + grouped["MatchID"].astype(str)):
    ax.text(.7 ,i,j)

plt.title("Total world cup matches played in each country",color='g')
plt.grid(True)
plt.xlabel("matches")
plt.show()
```



Inferences and Conclusion

From the above analysis i can mention many inferences, here is some of them:

The first think we can draw from the above analysis and visualization is that the world cup started in 1930 and the first country that hosted the Fifa world cup is Uruguay which won Argentina on final. So Uruguay became the first country to win the Fifa word cup as hoster of the tournament Brazil is the first country holding the maximum number of Fifa world cup with 5 cups, followed by Germany and Italy both having 4 cups. Urguay and Argentina have 2 cups each, then England, France and Spain have 1 cups each. An Uzbek Ravshan IRMATOV holds the record for officiating the most FIFA World Cup matches with 9 matches, followed by QUINIOU Joel, LARRIONDA Jorge, ARCHUNDIA Benito with 8 matches each. Germany is leading the list of countries where played maximum number of matches, with 102 matches, followed by Brazil with 85 countries and Mexico with 84 matches A German player Klose is the

one who breaked the record of playing maximum number of matches

In [ ]: