fifa-world-cup

February 5, 2024

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly as py
import cufflinks as cf
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
players= pd.read_csv(r'WorldCupPlayers.csv', encoding= 'unicode_escape')
matches= pd.read_csv(r'WorldCupMatches.csv', encoding= 'unicode_escape')
world_cup= pd.read_csv(r'WorldCups.csv', encoding= 'unicode_escape')
```

[3]:
```
players.head()
```

[3]:
```
   RoundID  MatchID Team Initials           Coach Name Line-up  Shirt Number  \
0      201     1096           FRA  CAUDRON Raoul (FRA)       S             0
1      201     1096           MEX     LUQUE Juan (MEX)       S             0
2      201     1096           FRA  CAUDRON Raoul (FRA)       S             0
3      201     1096           MEX     LUQUE Juan (MEX)       S             0
4      201     1096           FRA  CAUDRON Raoul (FRA)       S             0

        Player Name Position Event
0       Alex THEPOT       GK   NaN
1   Oscar BONFIGLIO       GK   NaN
2  Marcel LANGILLER      NaN  G40'
3      Juan CARRENO      NaN  G70'
4   Ernest LIBERATI      NaN   NaN
```

[4]:
```
matches.head()
```

[4]:
```
     Year              Datetime    Stage         Stadium         City  \
0  1930.0  13 Jul 1930 - 15:00  Group 1         Pocitos  Montevideo
1  1930.0  13 Jul 1930 - 15:00  Group 4  Parque Central  Montevideo
2  1930.0  14 Jul 1930 - 12:45  Group 2  Parque Central  Montevideo
3  1930.0  14 Jul 1930 - 14:50  Group 3         Pocitos  Montevideo
```

```
4  1930.0  15 Jul 1930 - 16:00   Group 1  Parque Central  Montevideo

  Home Team Name  Home Team Goals  Away Team Goals Away Team Name  \
0         France              4.0              1.0         Mexico
1            USA              3.0              0.0        Belgium
2     Yugoslavia              2.0              1.0         Brazil
3        Romania              3.0              1.0           Peru
4      Argentina              1.0              0.0         France

  Win conditions  Attendance  Half-time Home Goals  Half-time Away Goals  \
0                     4444.0                   3.0                   0.0
1                    18346.0                   2.0                   0.0
2                    24059.0                   2.0                   0.0
3                     2549.0                   1.0                   0.0
4                    23409.0                   0.0                   0.0

                  Referee               Assistant 1  \
0  LOMBARDI Domingo (URU)     CRISTOPHE Henry (BEL)
1       MACIAS Jose (ARG)  MATEUCCI Francisco (URU)
2     TEJADA Anibal (URU)   VALLARINO Ricardo (URU)
3   WARNKEN Alberto (CHI)       LANGENUS Jean (BEL)
4     REGO Gilberto (BRA)      SAUCEDO Ulises (BOL)

                 Assistant 2  RoundID  MatchID Home Team Initials  \
0         REGO Gilberto (BRA)    201.0   1096.0                FRA
1       WARNKEN Alberto (CHI)    201.0   1090.0                USA
2         BALWAY Thomas (FRA)    201.0   1093.0                YUG
3    MATEUCCI Francisco (URU)    201.0   1098.0                ROU
4  RADULESCU Constantin (ROU)    201.0   1085.0                ARG

  Away Team Initials
0                MEX
1                BEL
2                BRA
3                PER
4                FRA
```

[5]:
```
matches.tail()
```

[5]:
```
      Year Datetime Stage Stadium City Home Team Name  Home Team Goals  \
4567   NaN      NaN   NaN     NaN  NaN            NaN              NaN
4568   NaN      NaN   NaN     NaN  NaN            NaN              NaN
4569   NaN      NaN   NaN     NaN  NaN            NaN              NaN
4570   NaN      NaN   NaN     NaN  NaN            NaN              NaN
4571   NaN      NaN   NaN     NaN  NaN            NaN              NaN

      Away Team Goals Away Team Name Win conditions  Attendance  \
4567              NaN            NaN            NaN         NaN
4568              NaN            NaN            NaN         NaN
4569              NaN            NaN            NaN         NaN
4570              NaN            NaN            NaN         NaN
4571              NaN            NaN            NaN         NaN

      Half-time Home Goals  Half-time Away Goals Referee Assistant 1  \
4567                   NaN                   NaN     NaN         NaN
4568                   NaN                   NaN     NaN         NaN
4569                   NaN                   NaN     NaN         NaN
4570                   NaN                   NaN     NaN         NaN
4571                   NaN                   NaN     NaN         NaN

     Assistant 2  RoundID  MatchID Home Team Initials Away Team Initials
4567         NaN      NaN      NaN                NaN                NaN
4568         NaN      NaN      NaN                NaN                NaN
4569         NaN      NaN      NaN                NaN                NaN
4570         NaN      NaN      NaN                NaN                NaN
4571         NaN      NaN      NaN                NaN                NaN
```

[6]:

```
world_cup.head()
```

[6]:

```
   Year      Country      Winner      Runners-Up    Third      Fourth  \
0  1930      Uruguay     Uruguay       Argentina      USA  Yugoslavia
1  1934        Italy       Italy  Czechoslovakia  Germany     Austria
2  1938       France       Italy         Hungary   Brazil      Sweden
3  1950       Brazil     Uruguay          Brazil   Sweden       Spain
4  1954  Switzerland  Germany FR         Hungary  Austria     Uruguay

   GoalsScored  QualifiedTeams  MatchesPlayed Attendance
0           70              13             18    590.549
1           70              16             17    363.000
2           84              15             18    375.700
3           88              13             22  1.045.246
4          140              16             26    768.607
```

[7]:

```
matches.dropna(subset=['Year'], inplace=True)
```

[8]:

```
matches.tail()
```

[8]:

```
       Year              Datetime                     Stage  \
847  2014.0  05 Jul 2014 - 17:00             Quarter-finals
848  2014.0  08 Jul 2014 - 17:00                Semi-finals
849  2014.0  09 Jul 2014 - 17:00                Semi-finals
850  2014.0  12 Jul 2014 - 17:00   Play-off for third place
851  2014.0  13 Jul 2014 - 16:00                      Final
```

```
                 Stadium            City Home Team Name  Home Team Goals  \
847     Arena Fonte Nova        Salvador    Netherlands              0.0   
848     Estadio Mineirao  Belo Horizonte         Brazil              1.0   
849   Arena de Sao Paulo       Sao Paulo    Netherlands              0.0   
850     Estadio Nacional        Brasilia         Brazil              0.0   
851  Estadio do Maracana  Rio De Janeiro        Germany              1.0   

     Away Team Goals Away Team Name                          Win conditions  \
847              0.0     Costa Rica  Netherlands win on penalties (4 - 3)    
848              7.0        Germany                                          
849              0.0      Argentina    Argentina win on penalties (2 - 4)    
850              3.0    Netherlands                                          
851              0.0      Argentina          Germany win after extra time    

     Attendance  Half-time Home Goals  Half-time Away Goals  \
847     51179.0                   0.0                   0.0   
848     58141.0                   0.0                   5.0   
849     63267.0                   0.0                   0.0   
850     68034.0                   0.0                   2.0   
851     74738.0                   0.0                   0.0   

                    Referee                  Assistant 1  \
847   Ravshan IRMATOV (UZB)  RASULOV Abduxamidullo (UZB)   
848   RODRIGUEZ Marco (MEX)      TORRENTERA Marvin (MEX)   
849  Cï¿½neyt ï¿½AKIR (TUR)         DURAN Bahattin (TUR)   
850   HAIMOUDI Djamel (ALG)         ACHIK Redouane (MAR)   
851    Nicola RIZZOLI (ITA)        Renato FAVERANI (ITA)   

                  Assistant 2   RoundID      MatchID Home Team Initials  \
847  KOCHKAROV Bakhadyr (KGZ)  255953.0  300186488.0                NED   
848     QUINTERO Marcos (MEX)  255955.0  300186474.0                BRA   
849         ONGUN Tarik (TUR)  255955.0  300186490.0                NED   
850   ETCHIALI Abdelhak (ALG)  255957.0  300186502.0                BRA   
851      Andrea STEFANI (ITA)  255959.0  300186501.0                GER   

    Away Team Initials  
847                CRC  
848                GER  
849                ARG  
850                NED  
851                ARG  
```

[9]:
```
matches['Home Team Name'].value_counts()
```

[9]:
```
Home Team Name
Brazil                        82
Italy                         57
```

```
Argentina                    54
Germany FR                   43
England                      35
                             ..
Wales                         1
Norway                        1
rn">United Arab Emirates      1
Haiti                         1
rn">Bosnia and Herzegovina    1
Name: count, Length: 78, dtype: int64
```

[10]:

```
names = matches[matches['Home Team Name'].str.contains('rn">')]['Home Team␣
↪Name'].value_counts()
names
```

[10]:

```
Home Team Name
rn">Republic of Ireland       5
rn">United Arab Emirates      1
rn">Trinidad and Tobago       1
rn">Serbia and Montenegro     1
rn">Bosnia and Herzegovina    1
Name: count, dtype: int64
```

[11]:

```
wrong = list(names.index)
wrong
```

[11]:

```
['rn">Republic of Ireland',
 'rn">United Arab Emirates',
 'rn">Trinidad and Tobago',
 'rn">Serbia and Montenegro',
 'rn">Bosnia and Herzegovina']
```

[12]:

```
correct = [name.split('>')[1] for name in wrong]
correct
```

[12]:

```
['Republic of Ireland',
 'United Arab Emirates',
 'Trinidad and Tobago',
 'Serbia and Montenegro',
 'Bosnia and Herzegovina']
```

[13]:

```
old_name = ['Germany FR', 'Maracan  - Est dio Jornalista M rio Filho', 'Estadio␣
↪do Maracana']
new_name = ['Germany', 'Maracan Stadium', 'Maracan Stadium']
```

[14]:

```
wrong = wrong + old_name
correct = correct + new_name
```

[15]:
```
wrong, correct
```

[15]:
```
(['rn">Republic of Ireland',
  'rn">United Arab Emirates',
  'rn">Trinidad and Tobago',
  'rn">Serbia and Montenegro',
  'rn">Bosnia and Herzegovina',
  'Germany FR',
  'Maracan  - Est dio Jornalista M rio Filho',
  'Estadio do Maracana'],
 ['Republic of Ireland',
  'United Arab Emirates',
  'Trinidad and Tobago',
  'Serbia and Montenegro',
  'Bosnia and Herzegovina',
  'Germany',
  'Maracan Stadium',
  'Maracan Stadium'])
```

[16]:
```
for index, wr in enumerate(wrong):
    world_cup = world_cup.replace(wrong[index], correct[index])

for index, wr in enumerate(wrong):
    matches = matches.replace(wrong[index], correct[index])

for index, wr in enumerate(wrong):
    players = players.replace(wrong[index], correct[index])
```

[17]:
```
names = matches[matches['Home Team Name'].str.contains('rn">')]['Home Team␣
 ↪Name'].value_counts()
names
```

[17]:
```
Series([], Name: count, dtype: int64)
```

[18]:
```
winner = world_cup['Winner'].value_counts()
winner
```

[18]:
```
Winner
Brazil       5
Italy        4
Germany      4
Uruguay      2
Argentina    2
England      1
France       1
Spain        1
Name: count, dtype: int64
```

[19]:
```
runnerup = world_cup['Runners-Up'].value_counts()
runnerup
```

[19]:
```
Runners-Up
Germany           4
Argentina         3
Netherlands       3
Czechoslovakia    2
Hungary           2
Brazil            2
Italy             2
Sweden            1
France            1
Name: count, dtype: int64
```

[20]:
```
third = world_cup['Third'].value_counts()
third
```

[20]:
```
Third
Germany        4
Brazil         2
Sweden         2
France         2
Poland         2
USA            1
Austria        1
Chile          1
Portugal       1
Italy          1
Croatia        1
Turkey         1
Netherlands    1
Name: count, dtype: int64
```

[21]:
```
teams = pd.concat([winner, runnerup, third], axis=1)
teams.fillna(0, inplace=True)
teams = teams.astype(int)
teams
```

[21]:
```
                count  count  count
Brazil              5      2      2
Italy               4      2      1
Germany             4      4      4
Uruguay             2      0      0
Argentina           2      3      0
England             1      0      0
France              1      1      2
```

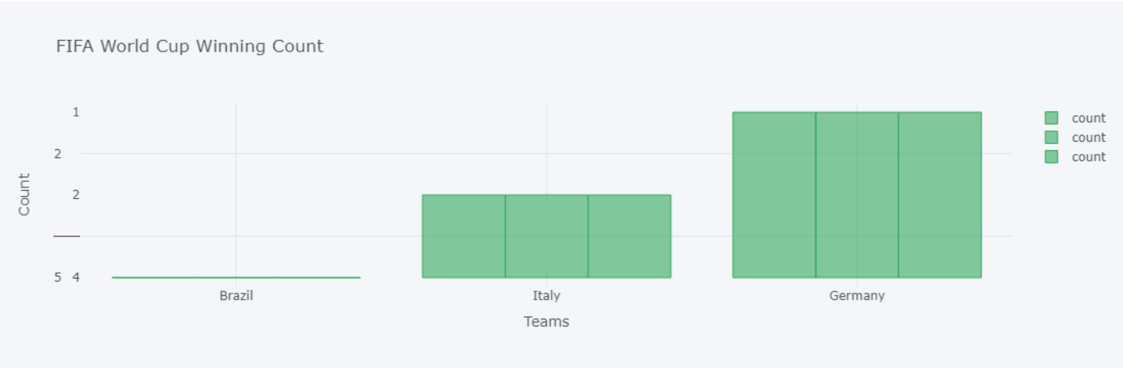
```
Spain                1      0      0
Netherlands          0      3      1
Czechoslovakia       0      2      0
Hungary              0      2      0
Sweden               0      1      2
Poland               0      0      2
USA                  0      0      1
Austria              0      0      1
Chile                0      0      1
Portugal             0      0      1
Croatia              0      0      1
Turkey               0      0      1
```

[22]:

```
from plotly.offline import iplot
py.offline.init_notebook_mode(connected=True)
cf.go_offline()
```

[23]:

```
teams.iplot(kind = 'bar', xTitle='Teams', yTitle='Count', title='FIFA World Cup␣
  ↪Winning Count')
```



[24]:

```
matches.head(2)
```

[24]:

```
     Year               Datetime    Stage         Stadium        City  \
0  1930.0  13 Jul 1930 - 15:00   Group 1         Pocitos  Montevideo
1  1930.0  13 Jul 1930 - 15:00   Group 4  Parque Central  Montevideo

  Home Team Name  Home Team Goals  Away Team Goals Away Team Name  \
0         France              4.0              1.0         Mexico
1            USA              3.0              0.0        Belgium

  Win conditions  Attendance  Half-time Home Goals  Half-time Away Goals  \
0                     4444.0                   3.0                   0.0
1                    18346.0                   2.0                   0.0
```

```
                 Referee              Assistant 1            Assistant 2  \
0  LOMBARDI Domingo (URU)     CRISTOPHE Henry (BEL)    REGO Gilberto (BRA)
1       MACIAS Jose (ARG)  MATEUCCI Francisco (URU)  WARNKEN Alberto (CHI)

   RoundID  MatchID Home Team Initials Away Team Initials
0    201.0   1096.0                FRA                MEX
1    201.0   1090.0                USA                BEL
```

[25]:
```python
home = matches[['Home Team Name', 'Home Team Goals']].dropna()
away = matches[['Away Team Name', 'Away Team Goals']].dropna()
```

[26]:
```python
home.columns = ['Countries', 'Goals']
away.columns = home.columns
```

[27]:
```python
goals = home.append(away, ignore_index = True)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_22620\2748964524.py in ?()
----> 1 goals = home.append(away, ignore_index = True)

~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\generic
 ↪py in ?(self, name)
   6200                 and name not in self._accessors
   6201                 and self._info_axis.
 ↪_can_hold_identifiers_and_holds_name(name)
   6202             ):
   6203                 return self[name]
-> 6204         return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'append'
```

[28]:
```python
goals = goals.groupby('Countries').sum()
goals
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 goals = goals.groupby('Countries').sum()
      2 goals

NameError: name 'goals' is not defined
```

[29]:
```python
goals = goals.sort_values(by = 'Goals', ascending=False)
goals
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 goals = goals.sort_values(by = 'Goals', ascending=False)
      2 goals

NameError: name 'goals' is not defined
```

[30]:
```
goals[:20].iplot(kind='bar', xTitle = 'Country Names', yTitle = 'Goals', title␣
 ↪= 'Countries Hits Number of Goals')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[30], line 1
----> 1 goals[:20].iplot(kind='bar', xTitle = 'Country Names', yTitle = 'Goals' ␣
 ↪title = 'Countries Hits Number of Goals')

NameError: name 'goals' is not defined
```

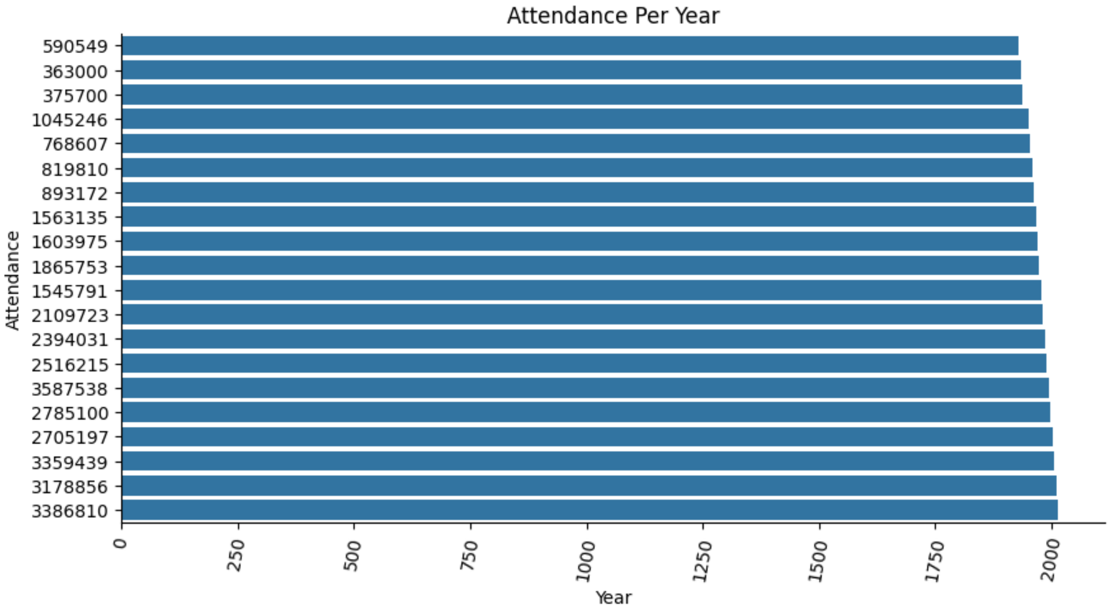
[31]:
```
world_cup['Attendance'] = world_cup['Attendance'].str.replace(".", "")
```

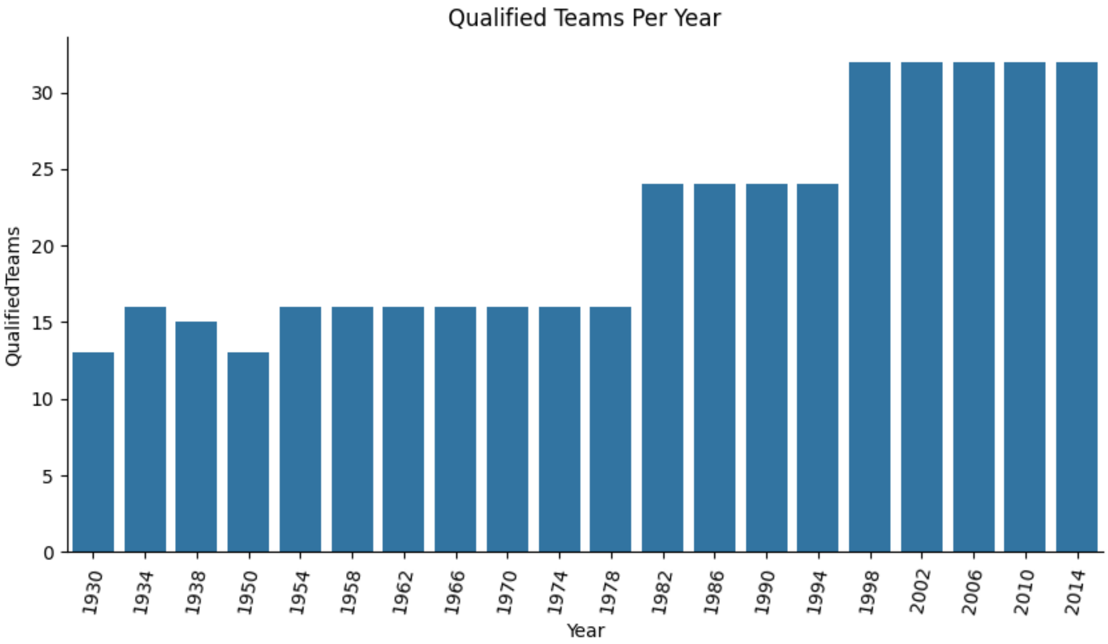
[32]:
```
world_cup.head()
```

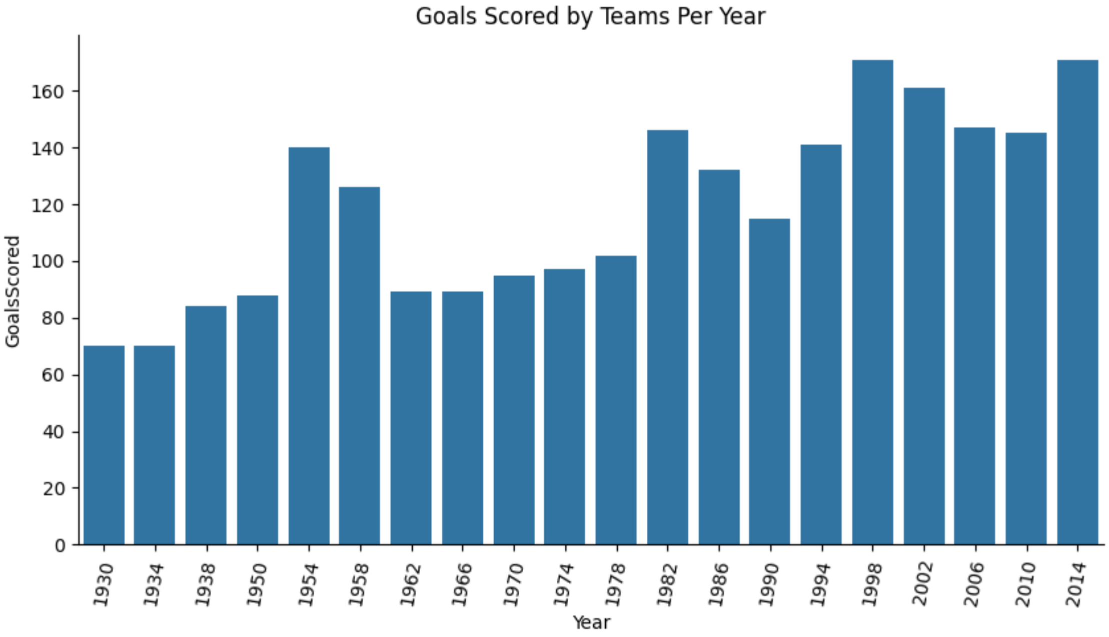
[32]:
```
   Year      Country   Winner      Runners-Up    Third      Fourth  \
0  1930      Uruguay  Uruguay       Argentina      USA  Yugoslavia
1  1934        Italy    Italy  Czechoslovakia  Germany     Austria
2  1938       France    Italy         Hungary   Brazil      Sweden
3  1950       Brazil  Uruguay          Brazil   Sweden       Spain
4  1954  Switzerland  Germany         Hungary  Austria     Uruguay

   GoalsScored  QualifiedTeams  MatchesPlayed Attendance
0           70              13             18     590549
1           70              16             17     363000
2           84              15             18     375700
3           88              13             22    1045246
4          140              16             26     768607
```

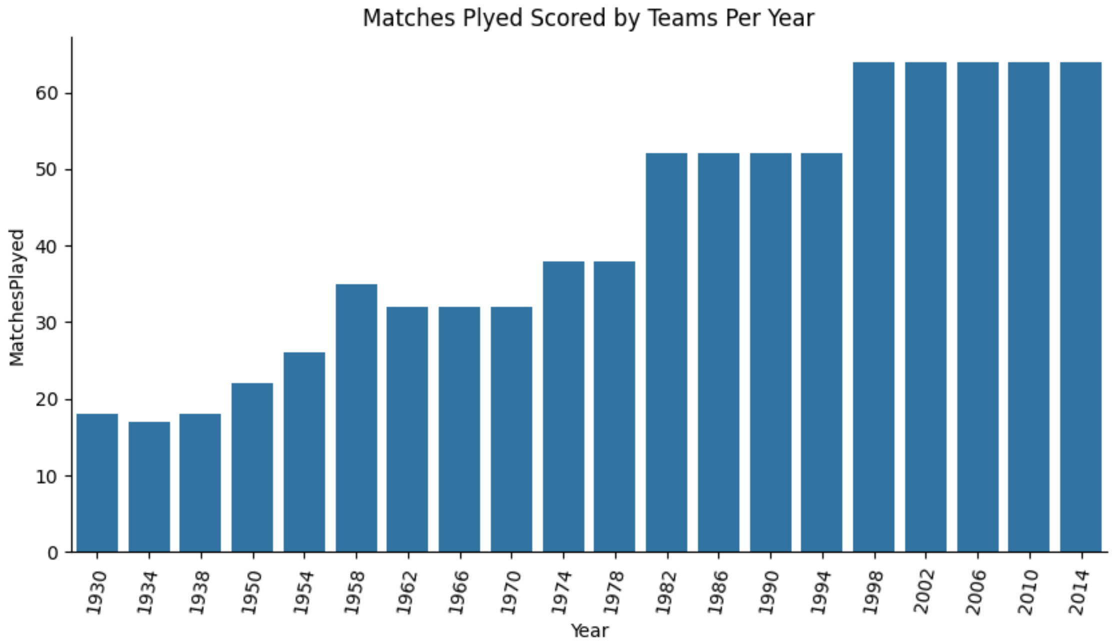
[33]:
```
fig, ax = plt.subplots(figsize = (10,5))
sns.despine(right = True)
g = sns.barplot(x = 'Year', y = 'Attendance', data = world_cup)
g.set_xticklabels(g.get_xticklabels(), rotation = 80)
g.set_title('Attendance Per Year')

#===========================================
fig, ax = plt.subplots(figsize = (10,5))
sns.despine(right = True)
g = sns.barplot(x = 'Year', y = 'QualifiedTeams', data = world_cup)
g.set_xticklabels(g.get_xticklabels(), rotation = 80)
g.set_title('Qualified Teams Per Year')

#==========================================

fig, ax = plt.subplots(figsize = (10,5))
sns.despine(right = True)
g = sns.barplot(x = 'Year', y = 'GoalsScored', data = world_cup)
g.set_xticklabels(g.get_xticklabels(), rotation = 80)
g.set_title('Goals Scored by Teams Per Year')


#==========================================


fig, ax = plt.subplots(figsize = (10,5))
sns.despine(right = True)
g = sns.barplot(x = 'Year', y = 'MatchesPlayed', data = world_cup)
g.set_xticklabels(g.get_xticklabels(), rotation = 80)
g.set_title('Matches Plyed Scored by Teams Per Year')
```

```
[33]: Text(0.5, 1.0, 'Matches Plyed Scored by Teams Per Year')
```




Qualified Teams Per Year
QualifiedTeams
0
5
10
15
20
25
30
Year
1930
1934
1938
1950
1954
1958
1962
1966
1970
1974
1978
1982
1986
1990
1994
1998
2002
2006
2010
2014


Goals Scored by Teams Per Year
GoalsScored
0
20
40
60
80
100
120
140
160
Year
1930
1934
1938
1950
1954
1958
1962
1966
1970
1974
1978
1982
1986
1990
1994
1998
2002
2006
2010
2014



[34]:
```
matches.head(2)
```

[34]:
```
     Year              Datetime    Stage         Stadium         City  \
0  1930.0  13 Jul 1930 - 15:00   Group 1         Pocitos  Montevideo
1  1930.0  13 Jul 1930 - 15:00   Group 4  Parque Central  Montevideo

  Home Team Name  Home Team Goals  Away Team Goals Away Team Name  \
0         France              4.0              1.0         Mexico
1            USA              3.0              0.0        Belgium

  Win conditions  Attendance  Half-time Home Goals  Half-time Away Goals  \
0                     4444.0                   3.0                   0.0
1                    18346.0                   2.0                   0.0

                  Referee               Assistant 1            Assistant 2  \
0  LOMBARDI Domingo (URU)     CRISTOPHE Henry (BEL)    REGO Gilberto (BRA)
1       MACIAS Jose (ARG)  MATEUCCI Francisco (URU)  WARNKEN Alberto (CHI)

   RoundID  MatchID Home Team Initials Away Team Initials
0    201.0   1096.0                FRA                MEX
1    201.0   1090.0                USA                BEL
```

[35]:
```
home = matches.groupby(['Year', 'Home Team Name'])['Home Team Goals'].sum()
home
```

```
[35]: Year    Home Team Name
      1930.0  Argentina         16.0
              Brazil             4.0
              Chile              4.0
              France             4.0
              Paraguay           1.0
                                …
      2014.0  Russia             1.0
              Spain              1.0
              Switzerland        4.0
              USA                2.0
              Uruguay            3.0
      Name: Home Team Goals, Length: 366, dtype: float64
```

```
[36]: away = matches.groupby(['Year', 'Away Team Name'])['Away Team Goals'].sum()
      away
```

```
[36]: Year    Away Team Name
      1930.0  Argentina         2.0
              Belgium           0.0
              Bolivia           0.0
              Brazil            1.0
              Chile             1.0
                               …
      2014.0  Russia            1.0
              Spain             3.0
              Switzerland       3.0
              USA               4.0
              Uruguay           1.0
      Name: Away Team Goals, Length: 411, dtype: float64
```

```
[37]: goals = pd.concat([home, away], axis=1)
      goals.fillna(0, inplace=True)
      goals['Goals'] = goals['Home Team Goals'] + goals['Away Team Goals']
      goals = goals.drop(labels = ['Home Team Goals', 'Away Team Goals'], axis = 1)
      goals
```

```
[37]:                  Goals
      Year
      1930.0 Argentina   18.0
             Brazil       5.0
             Chile        5.0
             France       4.0
             Paraguay     1.0
      …                  …
      1998.0 Iran         2.0
             Mexico       8.0
```

```
       Norway       5.0
       Tunisia      1.0
2006.0 IR Iran      0.0

[427 rows x 1 columns]
```

```
[38]: goals = goals.reset_index()
```

```
[39]: goals.columns = ['Year', 'Country', 'Goals']
      goals = goals.sort_values(by = ['Year', 'Goals'], ascending = [True, False])
      goals
```

```
[39]:        Year     Country  Goals
      0    1930.0   Argentina   18.0
      7    1930.0     Uruguay   15.0
      6    1930.0         USA    7.0
      8    1930.0  Yugoslavia    7.0
      1    1930.0      Brazil    5.0
      ..      …          …     …
      355  2014.0       Japan    2.0
      361  2014.0      Russia    2.0
      340  2014.0    Cameroon    1.0
      352  2014.0    Honduras    1.0
      353  2014.0     IR Iran    1.0

      [427 rows x 3 columns]
```

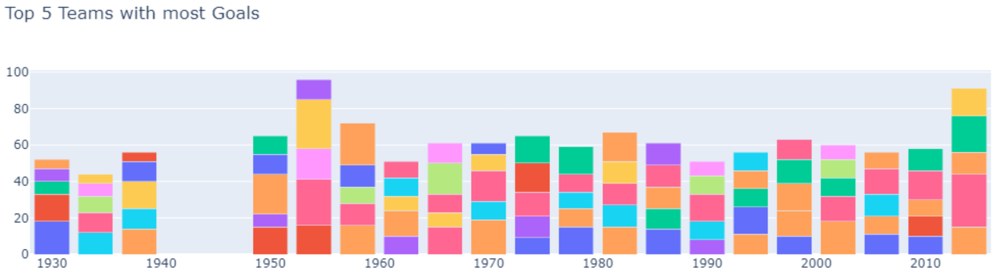
```
[40]: top5 = goals.groupby('Year').head()
      top5.head(10)
```

```
[40]:       Year         Country  Goals
      0   1930.0       Argentina   18.0
      7   1930.0         Uruguay   15.0
      6   1930.0             USA    7.0
      8   1930.0      Yugoslavia    7.0
      1   1930.0          Brazil    5.0
      13  1934.0           Italy   12.0
      11  1934.0         Germany   11.0
      10  1934.0  Czechoslovakia    9.0
      9   1934.0         Austria    7.0
      12  1934.0         Hungary    5.0
```

```
[41]: import plotly.graph_objects as go
```

```
[42]: x, y = goals['Year'].values, goals['Goals'].values
```

```
[43]: data = []
      for team in top5['Country'].drop_duplicates().values:
          year = top5[top5['Country'] == team]['Year']
          goal = top5[top5['Country'] == team]['Goals']

          data.append(go.Bar(x = year, y = goal, name = team))
      layout = go.Layout(barmode = 'stack', title = 'Top 5 Teams with most Goals',␣
       ↪showlegend = False)

      fig = go.Figure(data = data, layout = layout)
      fig.show()
```



```
[44]: matches['Datetime'] = pd.to_datetime(matches['Datetime'])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[44], line 1
----> 1 matches['Datetime'] = pd.to_datetime(matches['Datetime'])

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\tools\datetimes.
 ↪py:1108, in to_datetime(arg, errors, dayfirst, yearfirst, utc, format, exact, ␣
 ↪unit, infer_datetime_format, origin, cache)
   1106             result = arg.tz_localize("utc")
   1107 elif isinstance(arg, ABCSeries):
-> 1108     cache_array = _maybe_cache(arg, format, cache, convert_listlike)
   1109     if not cache_array.empty:
   1110         result = arg.map(cache_array)

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\tools\datetimes.
 ↪py:254, in _maybe_cache(arg, format, cache, convert_listlike)
    252 unique_dates = unique(arg)
    253 if len(unique_dates) < len(arg):
--> 254     cache_dates = convert_listlike(unique_dates, format)
```

```
    255     # GH#45319
    256     try:

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\tools\datetimes.
 ↪py:488, in _convert_listlike_datetimes(arg, format, name, utc, unit, errors,␣
 ↪dayfirst, yearfirst, exact)
    486 # `format` could be inferred, or user didn't ask for mixed-format␣
 ↪parsing.
    487 if format is not None and format != "mixed":
--> 488     return␣
 ↪_array_strptime_with_fallback(arg, name, utc, format, exact, errors)
    490 result, tz_parsed = objects_to_datetime64ns(
    491     arg,
    492     dayfirst=dayfirst,
   (…)
    496     allow_object=True,
    497 )
    499 if tz_parsed is not None:
    500     # We can take a shortcut since the datetime64 numpy array
    501     # is in UTC

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\tools\datetimes.
 ↪py:519, in _array_strptime_with_fallback(arg, name, utc, fmt, exact, errors)
    508 def _array_strptime_with_fallback(
    509     arg,
    510     name,
   (…)
    514     errors: str,
    515 ) -> Index:
    516     """
    517     Call array_strptime, with fallback behavior depending on 'errors'.
    518     """
--> 519     result, timezones =␣
 ↪array_strptime(arg, fmt, exact=exact, errors=errors, utc=utc)
    520     if any(tz is not None for tz in timezones):
    521         return _return_parsed_timezone_results(result, timezones, utc,␣
 ↪name)

File strptime.pyx:534, in pandas._libs.tslibs.strptime.array_strptime()

File strptime.pyx:355, in pandas._libs.tslibs.strptime.array_strptime()

ValueError: time data "17 June 1970 - 16:00 " doesn't match format "%d %b %Y -␣
 ↪%H:%M ", at position 103. You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but not␣
 ↪necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for each element␣
 ↪individually. You might want to use `dayfirst` alongside this.
```

[45]:
```
matches['Datetime'] = matches['Datetime'].apply(lambda x: x.strftime('%d %b,␣
 ↪%y'))
```

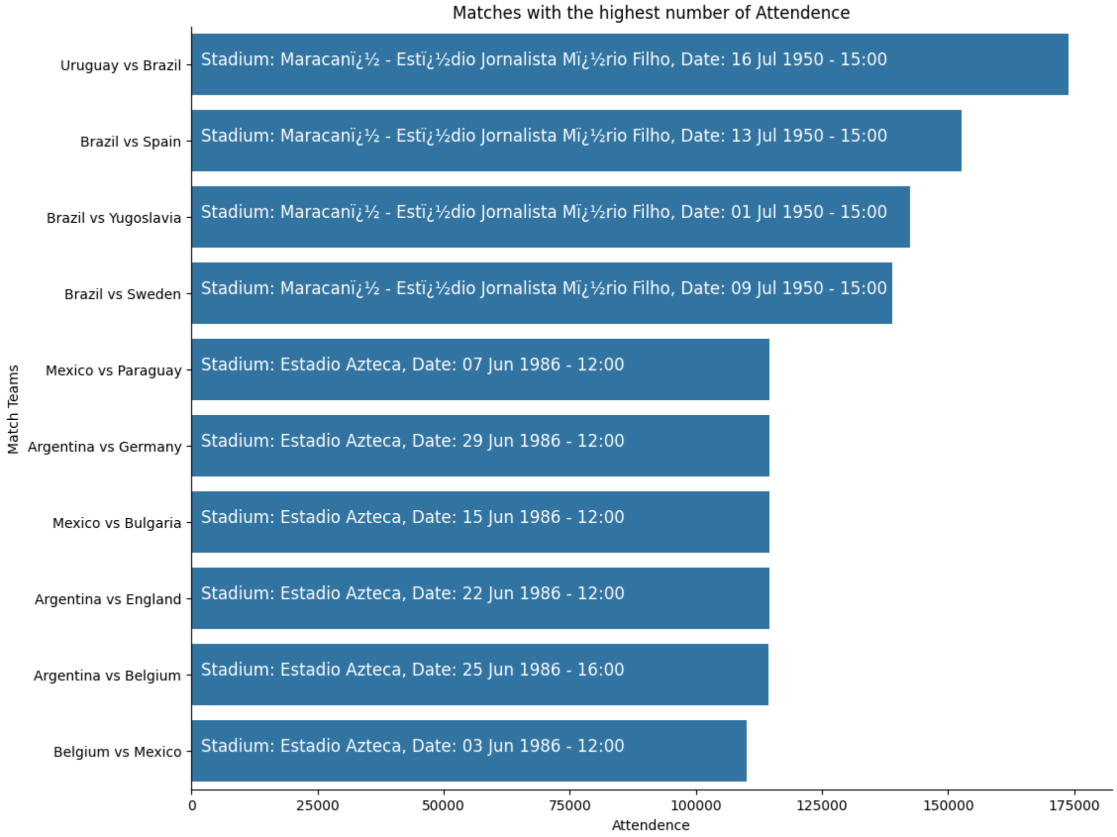
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[45], line 1
----> 1 matches['Datetime'] =␣
 ↪matches['Datetime'].apply(lambda x: x.strftime('%d %b, %y'))

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\serie
 ↪py:4760, in Series.apply(self, func, convert_dtype, args, by_row, **kwargs)
   4625 def apply(
   4626     self,
   4627     func: AggFuncType,
   (…)
   4632     **kwargs,
   4633 ) -> DataFrame | Series:
   4634     """
   4635     Invoke function on values of Series.
   4636 
   (…)
   4751     dtype: float64
   4752     """
   4753     return SeriesApply(
   4754         self,
   4755         func,
   4756         convert_dtype=convert_dtype,
   4757         by_row=by_row,
   4758         args=args,
   4759         kwargs=kwargs,
-> 4760     ).apply()

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\apply
 ↪py:1207, in SeriesApply.apply(self)
   1204     return self.apply_compat()
   1206 # self.func is Callable
-> 1207 return self.apply_standard()

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\apply
 ↪py:1287, in SeriesApply.apply_standard(self)
```

```
   1281 # row-wise access
   1282 # apply doesn't have a `na_action` keyword and for backward compat␣
  ↪reasons
   1283 # we need to give `na_action="ignore"` for categorical data.
   1284 # TODO: remove the `na_action="ignore"` when that default has been␣
  ↪changed in
   1285 #  Categorical (GH51645).
   1286 action = "ignore" if isinstance(obj.dtype, CategoricalDtype) else None
-> 1287 mapped = obj._map_values(
   1288     mapper=curried, na_action=action, convert=self.convert_dtype
   1289 )
   1291 if len(mapped) and isinstance(mapped[0], ABCSeries):
   1292     # GH#43986 Need to do list(mapped) in order to get treated as neste
   1293     #  See also GH#25959 regarding EA support
   1294     return obj._constructor_expanddim(list(mapped), index=obj.index)

File␣
  ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\base.
  ↪py:921, in IndexOpsMixin._map_values(self, mapper, na_action, convert)
    918 if isinstance(arr, ExtensionArray):
    919     return arr.map(mapper, na_action=na_action)
--> 921 return␣
  ↪algorithms.map_array(arr, mapper, na_action=na_action, convert=convert)

File␣
  ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\algorithms.
  ↪py:1814, in map_array(arr, mapper, na_action, convert)
   1812 values = arr.astype(object, copy=False)
   1813 if na_action is None:
-> 1814     return lib.map_infer(values, mapper, convert=convert)
   1815 else:
   1816     return lib.map_infer_mask(
   1817         values, mapper, mask=isna(values).view(np.uint8), convert=convert
   1818     )

File lib.pyx:2917, in pandas._libs.lib.map_infer()

Cell In[45], line 1, in <lambda>(x)
----> 1 matches['Datetime'] = matches['Datetime'].apply(lambda x: x.strftime('%␣
  ↪%b, %y'))

AttributeError: 'str' object has no attribute 'strftime'
```

[46]:

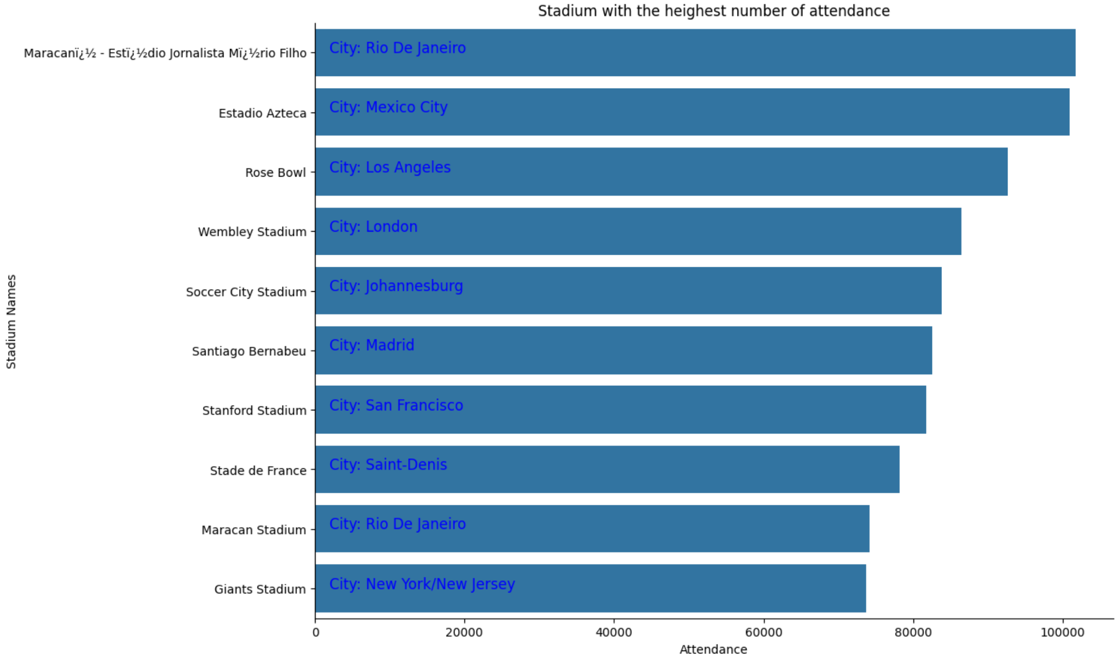
```
top10 = matches.sort_values(by = 'Attendance', ascending = False)[:10]
top10['vs'] = top10['Home Team Name'] + " vs " + top10['Away Team Name']

plt.figure(figsize = (12,10))
```

```
ax = sns.barplot(y = top10['vs'], x = top10['Attendance'])
sns.despine(right = True)

plt.ylabel('Match Teams')
plt.xlabel('Attendence')
plt.title('Matches with the highest number of Attendence')

for i, s in enumerate("Stadium: " + top10['Stadium'] +", Date: " +␣
 ↪top10['Datetime']):
    ax.text(2000, i, s, fontsize = 12, color = 'white')
plt.show()
```

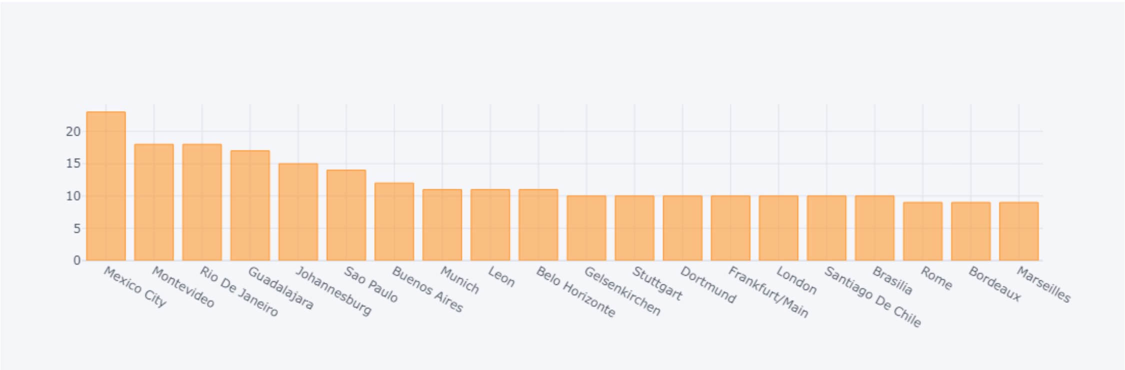


```
[47]: matches['Year'] = matches['Year'].astype(int)

std = matches.groupby(['Stadium', 'City'])['Attendance'].mean().reset_index().
 ↪sort_values(by = 'Attendance', ascending =False)

top10 = std[:10]
```

```
plt.figure(figsize = (12,9))
ax = sns.barplot(y = top10['Stadium'], x = top10['Attendance'])
sns.despine(right = True)

plt.ylabel('Stadium Names')
plt.xlabel('Attendance')
plt.title('Stadium with the heighest number of attendance')
for i, s in enumerate("City: " + top10['City']):
        ax.text(2000, i, s, fontsize = 12, color = 'b')

plt.show()
```



[48]:
```
matches['City'].value_counts()[:20].iplot(kind = 'bar')
```



[49]:

```
gold = world_cup["Winner"]
silver = world_cup["Runners-Up"]
bronze = world_cup["Third"]

gold_count = pd.DataFrame.from_dict(gold.value_counts())
silver_count = pd.DataFrame.from_dict(silver.value_counts())
bronze_count = pd.DataFrame.from_dict(bronze.value_counts())
podium_count = gold_count.join(silver_count, how='outer').join(bronze_count,␣
  ↪how='outer')
podium_count = podium_count.fillna(0)
podium_count.columns = ['WINNER', 'SECOND', 'THIRD']
podium_count = podium_count.astype('int64')
podium_count = podium_count.sort_values(by=['WINNER', 'SECOND', 'THIRD'],␣
  ↪ascending=False)

podium_count.plot(y=['WINNER', 'SECOND', 'THIRD'], kind="bar",
                  color =['gold','silver','brown'], figsize=(15, 6),␣
  ↪fontsize=14,
                 width=0.8, align='center')
plt.xlabel('Countries')
plt.ylabel('Number of podium')
plt.title('Number of podium by country')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[49], line 8
      6 silver_count = pd.DataFrame.from_dict(silver.value_counts())
      7 bronze_count = pd.DataFrame.from_dict(bronze.value_counts())
----> 8 podium_count = gold_count.join(silver_count, how='outer').
 ↪join(bronze_count, how='outer')
      9 podium_count = podium_count.fillna(0)
     10 podium_count.columns = ['WINNER', 'SECOND', 'THIRD']

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame
 ↪py:10415, in DataFrame.join(self, other, on, how, lsuffix, rsuffix, sort,␣
 ↪validate)
  10405     if how == "cross":
  10406         return merge(
  10407             self,
  10408             other,
   (…)
  10413             validate=validate,
  10414         )
> 10415     return merge(
  10416         self,
  10417             other,
  10418             left_on=on,
  10419             how=how,
  10420             left_index=on is None,
  10421             right_index=True,
  10422             suffixes=(lsuffix, rsuffix),
  10423             sort=sort,
  10424             validate=validate,
  10425         )
  10426 else:
  10427     if on is not None:

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\reshape\merge.
 ↪py:183, in merge(left, right, how, on, left_on, right_on, left_index,␣
 ↪right_index, sort, suffixes, copy, indicator, validate)
    168 else:
    169     op = _MergeOperation(
    170         left_df,
    171         right_df,
   (…)
    181         validate=validate,
    182     )
--> 183     return op.get_result(copy=copy)

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\reshape\merge.
 ↪py:885, in _MergeOperation.get_result(self, copy)
    881     self.left, self.right = self._indicator_pre_merge(self.left, self.
 ↪right)
    883 join_index, left_indexer, right_indexer = self._get_join_info()
--> 885 result = self._reindex_and_concat(
    886     join_index, left_indexer, right_indexer, copy=copy
    887 )
    888 result = result.__finalize__(self, method=self._merge_type)
    890 if self.indicator:

File␣
 ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\reshape\merge.
 ↪py:837, in _MergeOperation._reindex_and_concat(self, join_index, left_indexer ␣
 ↪right_indexer, copy)
    834 left = self.left[:]
    835 right = self.right[:]
--> 837 llabels, rlabels = _items_overlap_with_suffix(
    838     self.left._info_axis, self.right._info_axis, self.suffixes
    839 )
    841 if left_indexer is not None and not is_range_indexer(left_indexer,␣
 ↪len(left)):
    842     # Pinning the index here (and in the right code just below) is not
    843     #  necessary, but makes the `.take` more performant if we have e.g.
    844     #  a MultiIndex for left.index.
    845     lmgr = left._mgr.reindex_indexer(
    846         join_index,
    847         left_indexer,
   (…)
    852         use_na_proxy=True,
    853     )

File␣
  ↪~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\reshape\merge.
  ↪py:2655, in _items_overlap_with_suffix(left, right, suffixes)
   2652 lsuffix, rsuffix = suffixes
   2654 if not lsuffix and not rsuffix:
-> 2655     raise ValueError(f"columns overlap but no suffix specified:␣
  ↪{to_rename}")
   2657 def renamer(x, suffix: str | None):
   2658     """
   2659     Rename the left and right indices.
   2660
   (…)
   2671     x : renamed column name
   2672     """

ValueError: columns overlap but no suffix specified: Index(['count'],␣
  ↪dtype='object')
```

[50]:

```
#world_cups_matches['Win conditions'].value_counts()
home = matches[['Home Team Name', 'Home Team Goals']].dropna()
away = matches[['Away Team Name', 'Away Team Goals']].dropna()

goal_per_country = pd.DataFrame(columns=['countries', 'goals'])
goal_per_country = goal_per_country.append(home.rename(index=str,␣
 ↪columns={'Home Team Name': 'countries', 'Home Team Goals': 'goals'}))
goal_per_country = goal_per_country.append(away.rename(index=str,␣
 ↪columns={'Away Team Name': 'countries', 'Away Team Goals': 'goals'}))

goal_per_country['goals'] = goal_per_country['goals'].astype('int64')
goal_per_country = goal_per_country.groupby(['countries'])['goals'].sum().
 ↪sort_values(ascending=False)

goal_per_country[:10].plot(x=goal_per_country.index, y=goal_per_country.values,␣
 ↪kind="bar", figsize=(12, 6), fontsize=14)
plt.xlabel('Countries')
plt.ylabel('Number of goals')
plt.title('Top 10 of Number of goals by country')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_22620\2203313070.py in ?()
      2 home = matches[['Home Team Name', 'Home Team Goals']].dropna()
      3 away = matches[['Away Team Name', 'Away Team Goals']].dropna()
      4 
      5 goal_per_country = pd.DataFrame(columns=['countries', 'goals'])
----> 6 goal_per_country = goal_per_country.append(home.rename(index=str,␣
 ↪columns={'Home Team Name': 'countries', 'Home Team Goals': 'goals'}))
      7 goal_per_country = goal_per_country.append(away.rename(index=str,␣
 ↪columns={'Away Team Name': 'countries', 'Away Team Goals': 'goals'}))
      8 
      9 goal_per_country['goals'] = goal_per_country['goals'].astype('int64')

~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\generic
 ↪py in ?(self, name)
   6200             and name not in self._accessors
   6201             and self._info_axis.
 ↪_can_hold_identifiers_and_holds_name(name)
   6202         ):
   6203             return self[name]
-> 6204         return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'append'
```

[51]:

```
def get_labels(matches):
    if matches['Home Team Goals'] > matches['Away Team Goals']:
        return 'Home Team Win'
    if matches['Home Team Goals'] < matches['Away Team Goals']:
        return 'Away Team Win'
    return 'DRAW'
```

[52]:

```
matches['outcome'] = matches.apply(lambda x: get_labels(x), axis=1)
```

[53]:

```
matches.head()
```

[53]:

```
   Year              Datetime    Stage         Stadium         City  \
0  1930  13 Jul 1930 - 15:00  Group 1         Pocitos  Montevideo
1  1930  13 Jul 1930 - 15:00  Group 4  Parque Central  Montevideo
2  1930  14 Jul 1930 - 12:45  Group 2  Parque Central  Montevideo
3  1930  14 Jul 1930 - 14:50  Group 3         Pocitos  Montevideo
4  1930  15 Jul 1930 - 16:00  Group 1  Parque Central  Montevideo

  Home Team Name  Home Team Goals  Away Team Goals Away Team Name  \
0         France              4.0              1.0         Mexico
1            USA              3.0              0.0        Belgium
```

```
2     Yugoslavia              2.0              1.0           Brazil
3        Romania              3.0              1.0             Peru
4      Argentina              1.0              0.0           France

  Win conditions  …  Half-time Home Goals  Half-time Away Goals  \
0                 …                   3.0                   0.0
1                 …                   2.0                   0.0
2                 …                   2.0                   0.0
3                 …                   1.0                   0.0
4                 …                   0.0                   0.0

                 Referee                Assistant 1  \
0  LOMBARDI Domingo (URU)      CRISTOPHE Henry (BEL)
1       MACIAS Jose (ARG)  MATEUCCI Francisco (URU)
2     TEJADA Anibal (URU)   VALLARINO Ricardo (URU)
3   WARNKEN Alberto (CHI)        LANGENUS Jean (BEL)
4     REGO Gilberto (BRA)      SAUCEDO Ulises (BOL)

                 Assistant 2 RoundID  MatchID  Home Team Initials  \
0        REGO Gilberto (BRA)   201.0   1096.0                 FRA
1      WARNKEN Alberto (CHI)   201.0   1090.0                 USA
2        BALWAY Thomas (FRA)   201.0   1093.0                 YUG
3   MATEUCCI Francisco (URU)   201.0   1098.0                 ROU
4  RADULESCU Constantin (ROU)   201.0   1085.0                 ARG

  Away Team Initials        outcome
0                MEX  Home Team Win
1                BEL  Home Team Win
2                BRA  Home Team Win
3                PER  Home Team Win
4                FRA  Home Team Win

[5 rows x 21 columns]
```

[54]:

```
mt = matches['outcome'].value_counts()
mt
```

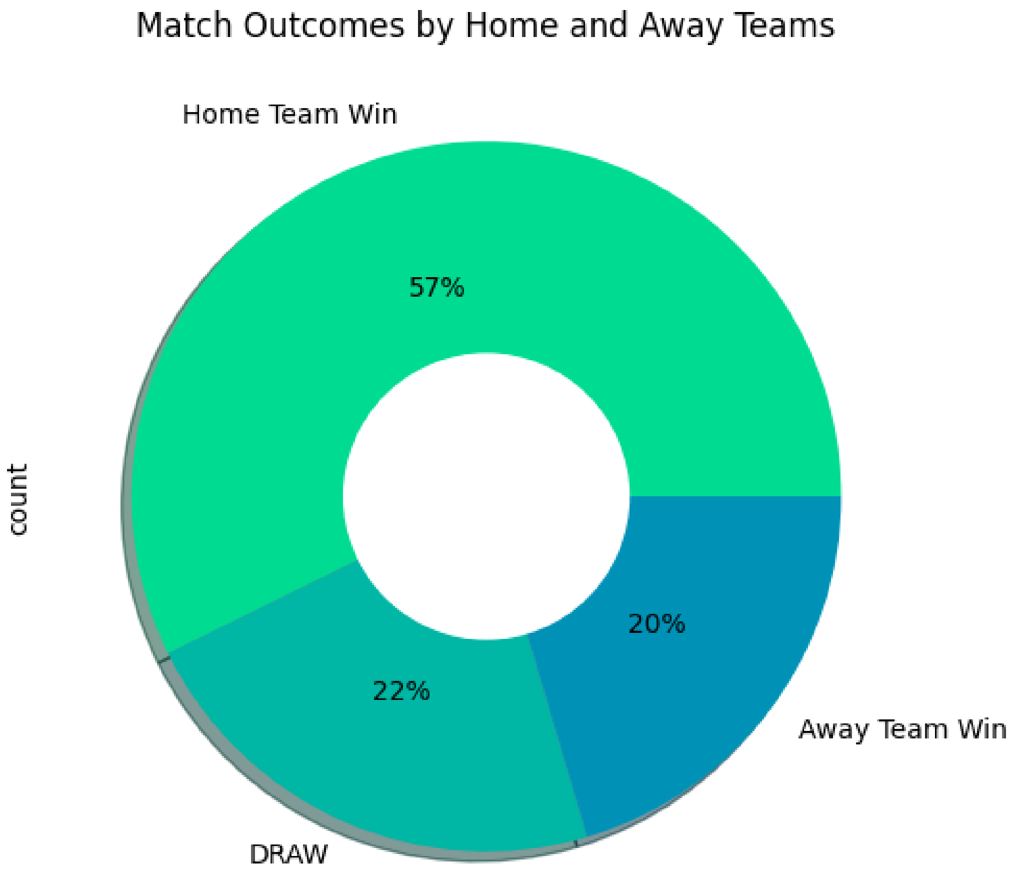
[54]:

```
outcome
Home Team Win    488
DRAW             190
Away Team Win    174
Name: count, dtype: int64
```

[55]:

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

mt.plot.pie(autopct = "%1.0f%%", colors = sns.color_palette('winter_r'), shadow␣
  ↪= True)
```

```
c = plt.Circle((0,0), 0.4, color =  'white')
plt.gca().add_artist(c)
plt.title('Match Outcomes by Home and Away Teams')
plt.show()
```



[ ]:

```
```