FIFA WORLD CUP ANALYSIS

```
pip install cufflinks
```

```
Requirement already satisfied: pygments in /usr/local/lib/python3.11/dist-packages (from ipython>=5.3.0->cufflinks) (2.1
Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (from ipython>=5.3.0->cufflinks) (0.2
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.11/dist-packages (from ipython>=5.3.0->cuffli
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages (from ipython>=5.3.0->cufflinks) (
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.11/dist-packages (from ipywidgets>=7.0.0->cuff
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets>=7.0.
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets>=7.
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets>=7.
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.19.2->c
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.19.2->cufflinks)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.19.2->cufflinks
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly>=4.1.1->cufflinks
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from plotly>=4.1.1->cufflinks) (24.
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidg
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidget
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-packages (from jedi>=0.16->ipython>
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-packages (from pexpect>4.3->ipython>=5.
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.11/dist-packages (from widgetsnbextension~=3.6.
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextensi
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnb
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbex
Requirement already satisfied: jupyter-core>=4.6.1 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->wid
Requirement already satisfied: nbformat in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbexten
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbe
Requirement already satisfied: nest-asyncio>=1.5 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widge
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widge
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widget
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widge
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widget
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.11/dist-packages (from jupyter-core>=4.6.1->n
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.11/dist-packages (from nbclassic>=0.4.7->n
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.11/dist-packages (from bleach[css]!=5.0.0->nbconv
Requirement already satisfied: defusedxml in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebo
Requirement already satisfied: markupsafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=
Requirement already satisfied: mistune<4,>=2.0.3 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->noteb
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.11/dist-packages (from nbformat->notebook>
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.11/dist-packages (from nbformat->notebook>=4.4.
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.11/dist-packages (from argon2-cffi->notebo
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from bleach!=5.0.0->bleach[css]!
Requirement already satisfied: tinycss2<1.5,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from bleach[css]!=5.0.0-
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=2.6->nbformat-
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.11/dist-packages (from jso
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=2.6->nbf
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=2.6->nbformat
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/python3.11/dist-packages (from notebook-shim>=0.
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from argon2-cffi-bindings->argon2
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->nbconvert>
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.0.1->argon2-cffi-bindi
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from jupyter-server<3,>=1.8->
Requirement already satisfied: websocket-client in /usr/local/lib/python3.11/dist-packages (from jupyter-server<3,>=1.8-
Requirement already satisfied: typing-extensions>=4.4.0 in /usr/local/lib/python3.11/dist-packages (from referencing>=0.
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<4,>=3.1.0->jupyter-serve
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<4,>=3.1.0->jupyter-se
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly as py
import cufflinks as cf
```

Data Cleaning:

Preprocessed raw FIFA World Cup data for consistency and accuracy.

Exploratory Data Analysis (EDA):

Identified key trends and patterns in tournament data. Visualized historical performance using Python libraries such as Matplotlib and Seaborn.

Insights Generation:

Highlighted standout teams, players and tournaments.

```
from google.colab import files
uploaded = files.upload()

import pandas as pd

matches = pd.read_csv("WorldCupMatches.csv")
players = pd.read_csv("WorldCupPlayers.csv")
world_cup = pd.read_csv("WorldCups.csv")
```

Choose files 3 files

- **WorldCupMatches.csv**(text/csv) - 239003 bytes, last modified: 14/01/2025 - 100% done
- **WorldCupPlayers.csv**(text/csv) - 2150588 bytes, last modified: 14/01/2025 - 100% done
- **WorldCups.csv**(text/csv) - 1412 bytes, last modified: 14/01/2025 - 100% done

```
Saving WorldCupMatches.csv to WorldCupMatches (2).csv
Saving WorldCupPlayers.csv to WorldCupPlayers (2).csv
Saving WorldCups.csv to WorldCups (2).csv
```

```
matches.head()
```

	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	Refere
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		4444.0	3.0	0.0	LOMBAR Doming (URI
1	1930.0	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium		18346.0	2.0	0.0	MACIA Jose (AR(
2	1930.0	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil		24059.0	2.0	0.0	TEJAD Anib (URI
3	1930.0	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		2549.0	1.0	0.0	WARNKE Alber (CH
4	1930.0	15 Jul 1930 - 16:00	Group 1	Parque Central	Montevideo	Argentina	1.0	0.0	France		23409.0	0.0	0.0	REG Gilber (BR.

Next steps: Generate code with matches | View recommended plots | New interactive sheet

```
matches.tail()
```

	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	Assistar
4567	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4568	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4569	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4570	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4571	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

```
players.head()
```

	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'
3	201	1096	MEX	LUQUE Juan (MEX)	S	0	Juan CARRENO	NaN	G70'
4	201	1096	FRA	CAUDRON Raoul (FRA)	S	0	Ernest LIBERATI	NaN	NaN

Next steps: Generate code with players | View recommended plots | New interactive sheet

```
players.tail()
```

	RoundID	MatchID	Team Initials	Coach Name	Line-up	Shirt Number	Player Name	Position	Event
37779	255959	300186501	ARG	SABELLA Alejandro (ARG)	N	19	ALVAREZ	NaN	NaN
37780	255959	300186501	GER	LOEW Joachim (GER)	N	6	KHEDIRA	NaN	NaN
37781	255959	300186501	ARG	SABELLA Alejandro (ARG)	N	20	AGUERO	NaN	IH46' Y65'
37782	255959	300186501	GER	LOEW Joachim (GER)	N	21	MUSTAFI	NaN	NaN

```
world_cup.head()
```

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams	MatchesPlayed	Attendance
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	13	18	590.549
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	16	17	363.000
2	1938	France	Italy	Hungary	Brazil	Sweden	84	15	18	375.700
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	13	22	1.045.246
4	1954	Switzerland	Germany FR	Hungary	Austria	Uruguay	140	16	26	768.607

Next steps: Generate code with world_cup | View recommended plots | New interactive sheet

```
world_cup.tail()
```

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams	MatchesPlayed	Attendance
15	1998	France	France	Brazil	Croatia	Netherlands	171	32	64	2.785.100
16	2002	Korea/Japan	Brazil	Germany	Turkey	Korea Republic	161	32	64	2.705.197
17	2006	Germany	Italy	France	Germany	Portugal	147	32	64	3.359.439
18	2010	South Africa	Spain	Netherlands	Germany	Uruguay	145	32	64	3.178.856

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

```
matches.tail()
```

	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
847	2014.0	05 Jul 2014 - 17:00	Quarter-finals	Arena Fonte Nova	Salvador	Netherlands	0.0	0.0	Costa Rica	Netherlands win on penalties (4 - 3)	51179.0	0.0	0.0
848	2014.0	08 Jul 2014 - 17:00	Semi-finals	Estadio Mineirao	Belo Horizonte	Brazil	1.0	7.0	Germany		58141.0	0.0	5.0
849	2014.0	09 Jul 2014 - 17:00	Semi-finals	Arena de Sao Paulo	Sao Paulo	Netherlands	0.0	0.0	Argentina	Argentina win on penalties (2 - 4)	63267.0	0.0	0.0
850	2014.0	12 Jul 2014 - 17:00	Play-off for third place	Estadio Nacional	Brasilia	Brazil	0.0	3.0	Netherlands		68034.0	0.0	2.0
851	2014.0	13 Jul 2014 - 16:00	Final	Estadio do Maracana	Rio De Janeiro	Germany	1.0	0.0	Argentina	Germany win after extra time	74738.0	0.0	0.0

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

Home Team Name	count
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

78 rows × 1 columns

dtype: int64

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

Home Team Name	count
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

dtype: int64

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

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

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

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

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

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

wrong, correct
```

```
(['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'])


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])


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

Home Team Name	count

dtype: int64

Attendance, Number of Teams, Goals, and Matches per Cup

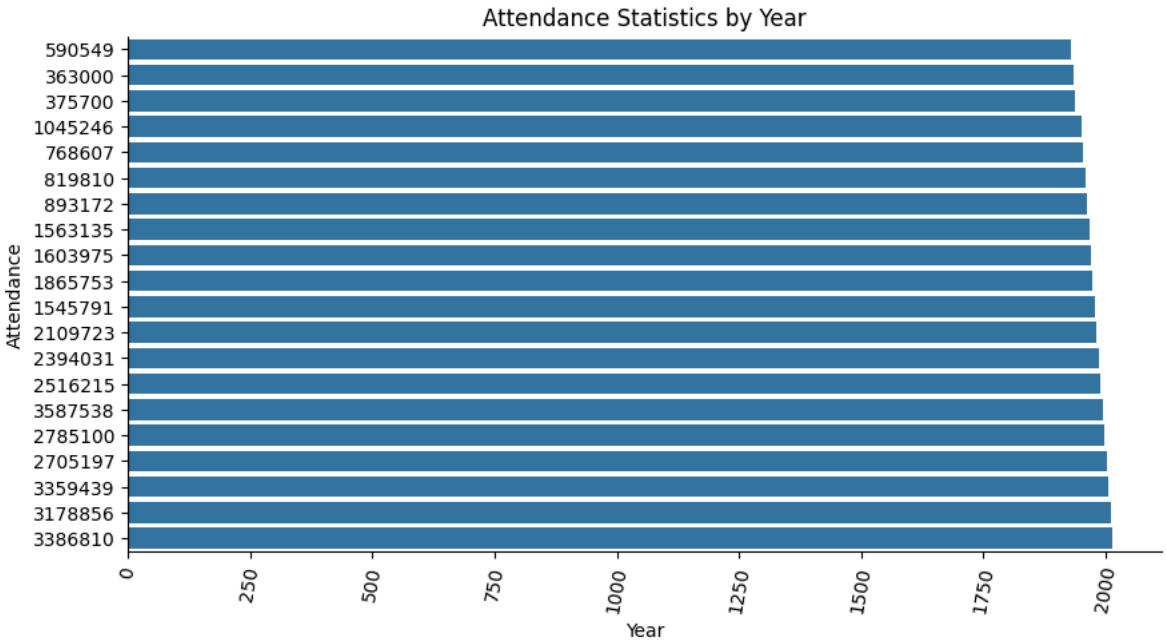
```
world_cup['Attendance'] = world_cup['Attendance'].str.replace(".", "")


world_cup.head()
```

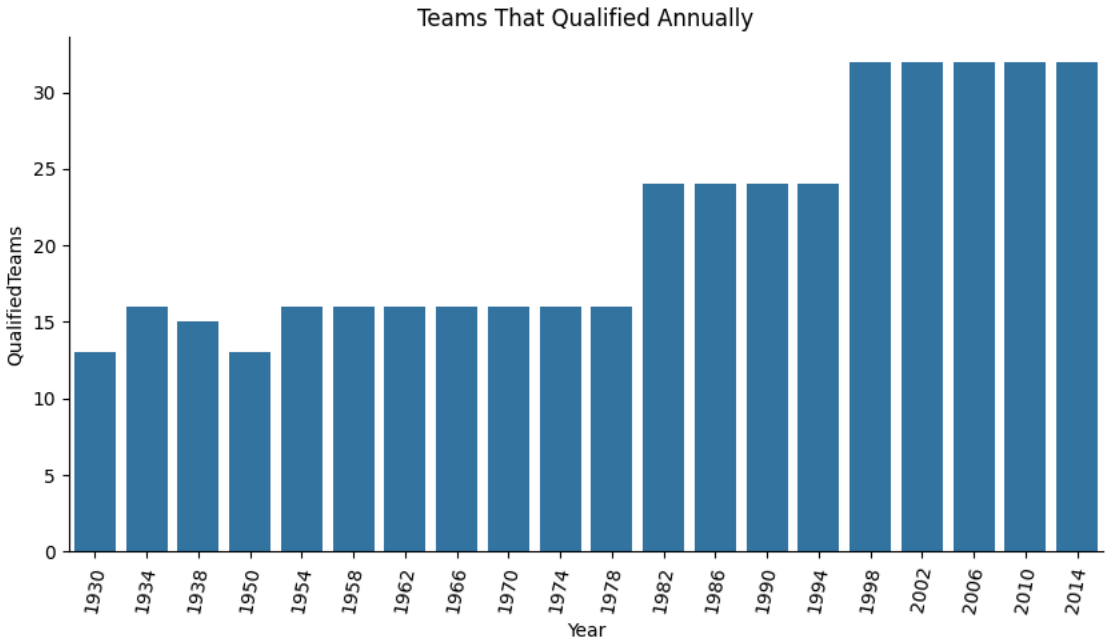
	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	QualifiedTeams	MatchesPlayed	Attendance
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	13	18	590549
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	16	17	363000
2	1938	France	Italy	Hungary	Brazil	Sweden	84	15	18	375700
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	13	22	1045246
4	1954	Switzerland	Germany	Hungary	Austria	Uruguay	140	16	26	768607

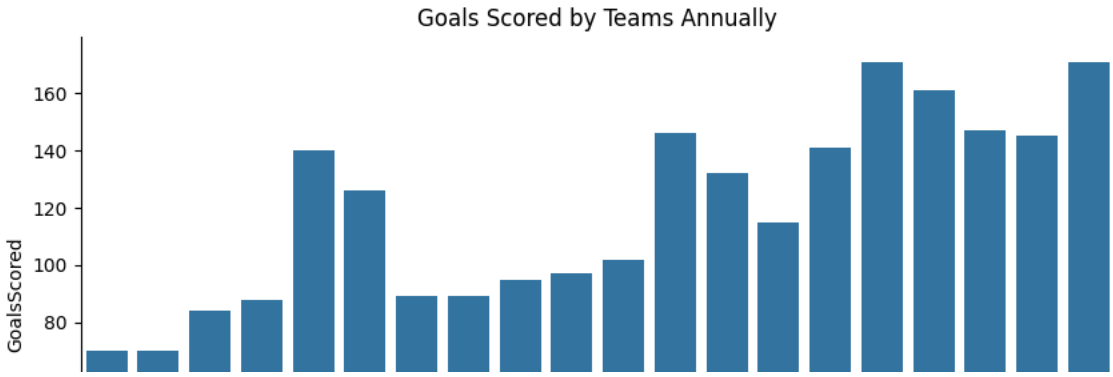
Next steps: Generate code with world_cup | View recommended plots | New interactive sheet

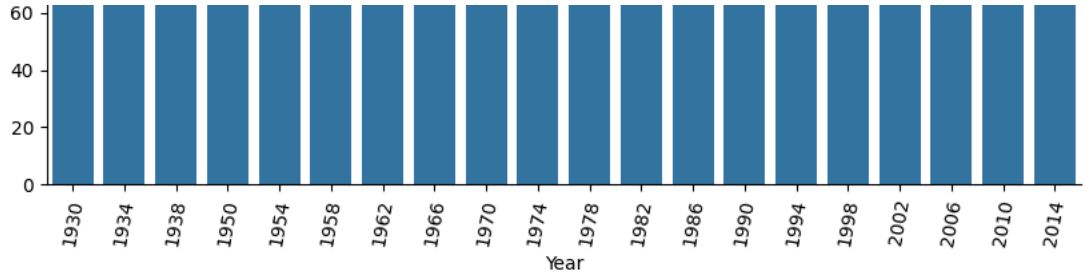
```
fig, ax = plt.subplots(figsize = (10,5))
sns.despine(right = True)
g = sns.barplot(x = 'Year', y = 'Attendance', data = world_cup)
g.set_xticklabels(g.get_xticklabels(), rotation = 80)
g.set_title('Attendance Statistics by Year')


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('Teams That Qualified Annually')


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 Annually')


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('Team-Wise Match Participation and Scores Across Different Years')
```

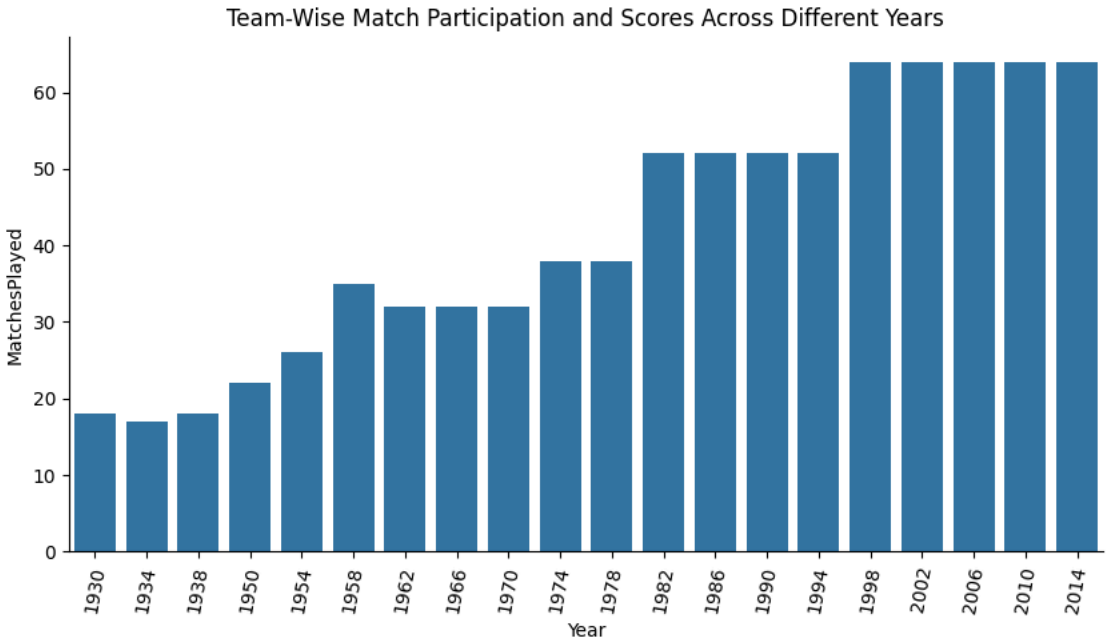
```
<ipython-input-170-f5f0e8585d45>:4: UserWarning:

set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.

<ipython-input-170-f5f0e8585d45>:11: UserWarning:

set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.

<ipython-input-170-f5f0e8585d45>:18: UserWarning:

set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.

<ipython-input-170-f5f0e8585d45>:25: UserWarning:

set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.

Text(0.5, 1.0, 'Team-Wise Match Participation and Scores Across Different Years')
```











Games with the highest number of spectators (1930 - 2014)

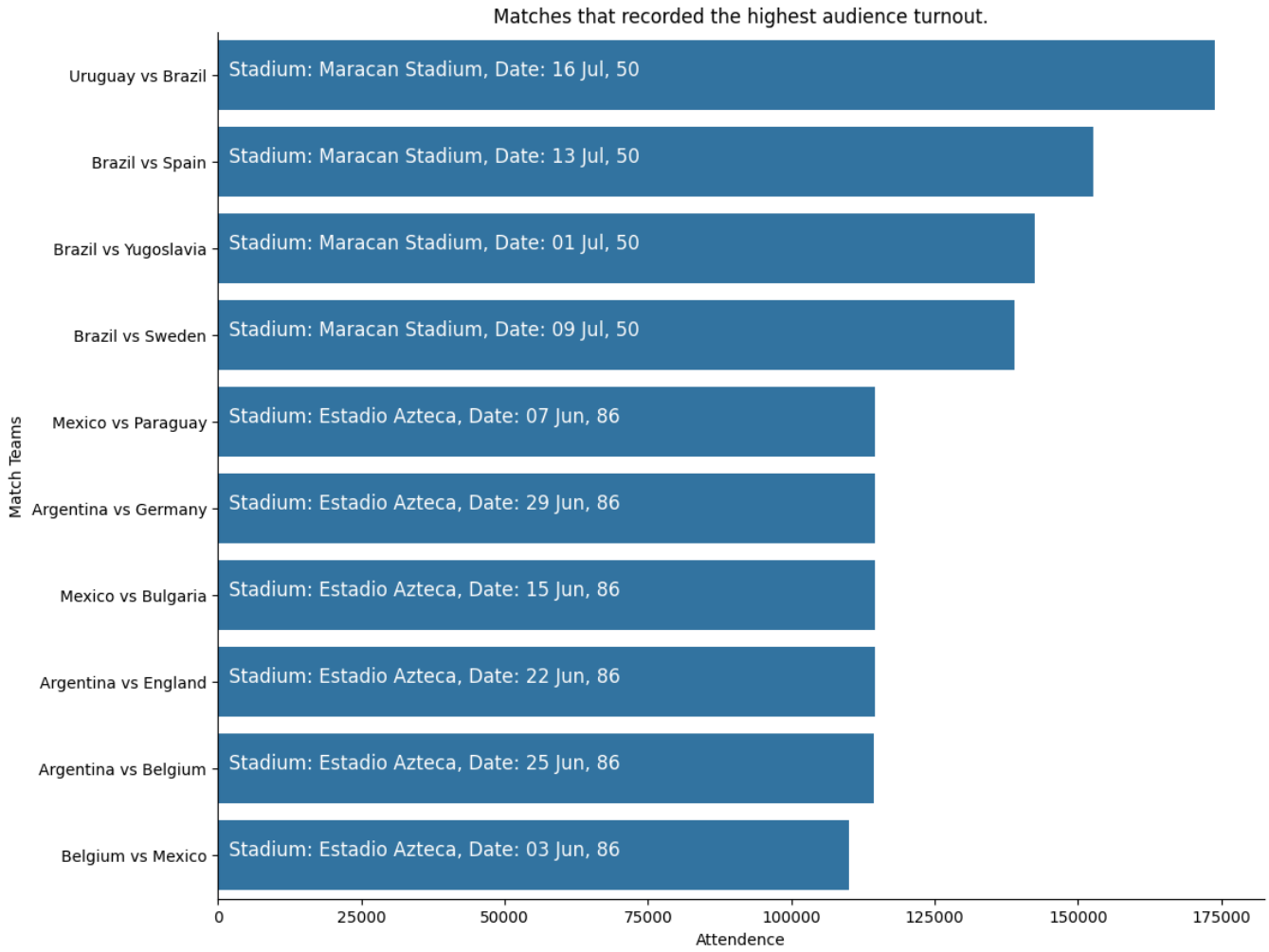
```
matches['Datetime'] = pd.to_datetime(matches['Datetime'], errors='coerce')
invalid_dates = matches[matches['Datetime'].isna()]
print(invalid_dates)
```

```
        Year Datetime        Stage                    Stadium          City  \
229  1970.0      NaT  Semi-finals             Estadio Azteca  Mexico City
251  1974.0      NaT      Group 1           Volksparkstadion      Hamburg
269  1974.0      NaT        Final             Olympiastadion       Munich
588  2002.0      NaT      Group C     Munsu Football Stadium        Ulsan
591  2002.0      NaT      Group C  Gwangju World Cup Stadium      Gwangju
592  2002.0      NaT      Group H       Saitama Stadium 2002      Saitama
602  2002.0      NaT      Group B   Jeonju World Cup Stadium       Jeonju
605  2002.0      NaT      Group G            Kashima Stadium      Ibaraki
613  2002.0      NaT      Group A    Suwon World Cup Stadium        Suwon
633  2002.0      NaT  Round of 16   Jeonju World Cup Stadium       Jeonju

    Home Team Name  Home Team Goals  Away Team Goals Away Team Name  \
229          Italy              4.0              3.0        Germany
251      German DR              1.0              0.0        Germany
269    Netherlands              1.0              2.0        Germany
588         Brazil              2.0              1.0         Turkey
591       China PR              0.0              2.0     Costa Rica
592          Japan              2.0              2.0        Belgium
602          Spain              3.0              1.0       Paraguay
605          Italy              1.0              2.0        Croatia
613        Senegal              3.0              3.0        Uruguay
633         Mexico              0.0              2.0            USA

                  Win conditions  Attendance  Half-time Home Goals  \
229  Italy win after extra time     102444.0                   0.0
251                                  60200.0                   0.0
269                                  78200.0                   1.0
588                                  33842.0                   0.0
591                                  27217.0                   0.0
592                                  55256.0                   0.0
602                                  24000.0                   0.0
605                                  36472.0                   0.0
613                                  33681.0                   3.0
633                                  36380.0                   0.0
```

```
     Half-time Away Goals                        Referee  \
229                   0.0  YAMASAKI MALDONADO Arturo (MEX)
251                   0.0         BARRETO RUIZ Ramon (URU)
269                   2.0                TAYLOR John (ENG)
588                   1.0              KIM Young Joo (KOR)
591                   0.0             VASSARAS Kyros (GRE)
592                   0.0             MATTUS William (CRC)
602                   1.0           EL GHANDOUR Gamal (EGY)
605                   0.0                POLL Graham (ENG)
613                   0.0               WEGEREEF Jan (NED)
633                   1.0         MELO PEREIRA Vitor (POR)

                          Assistant 1                Assistant 2     RoundID  \
229      HORMAZABAL DIAZ Rafael (CHI)  VELASQUEZ Guillermo (COL)       569.0
251             MARQUES Armando (BRA)        PESTARINO Luis (ARG)       262.0
269  GONZALEZ ARCHUNDIA Alfonso (MEX)   BARRETO RUIZ Ramon (URU)       605.0
588              KRISHNAN Visva (SIN)   FERNANDEZ Vladimir (SLV)  43950100.0
591                MATOS Carlos (POR)             POOL Jaap (NED)  43950100.0
592               KOLEIT Haidar (LIB)          DUPANOV Yuri (BLR)  43950100.0
602                 FARAG Wagih (EGY)   MUDZAMIRI Brighton (ZIM)  43950100.0
605                SHARP Philip (ENG)           LARSEN Jens (DEN)  43950100.0
613                   POOL Jaap (NED)       SZEKELY Ferenc (HUN)  43950100.0
```

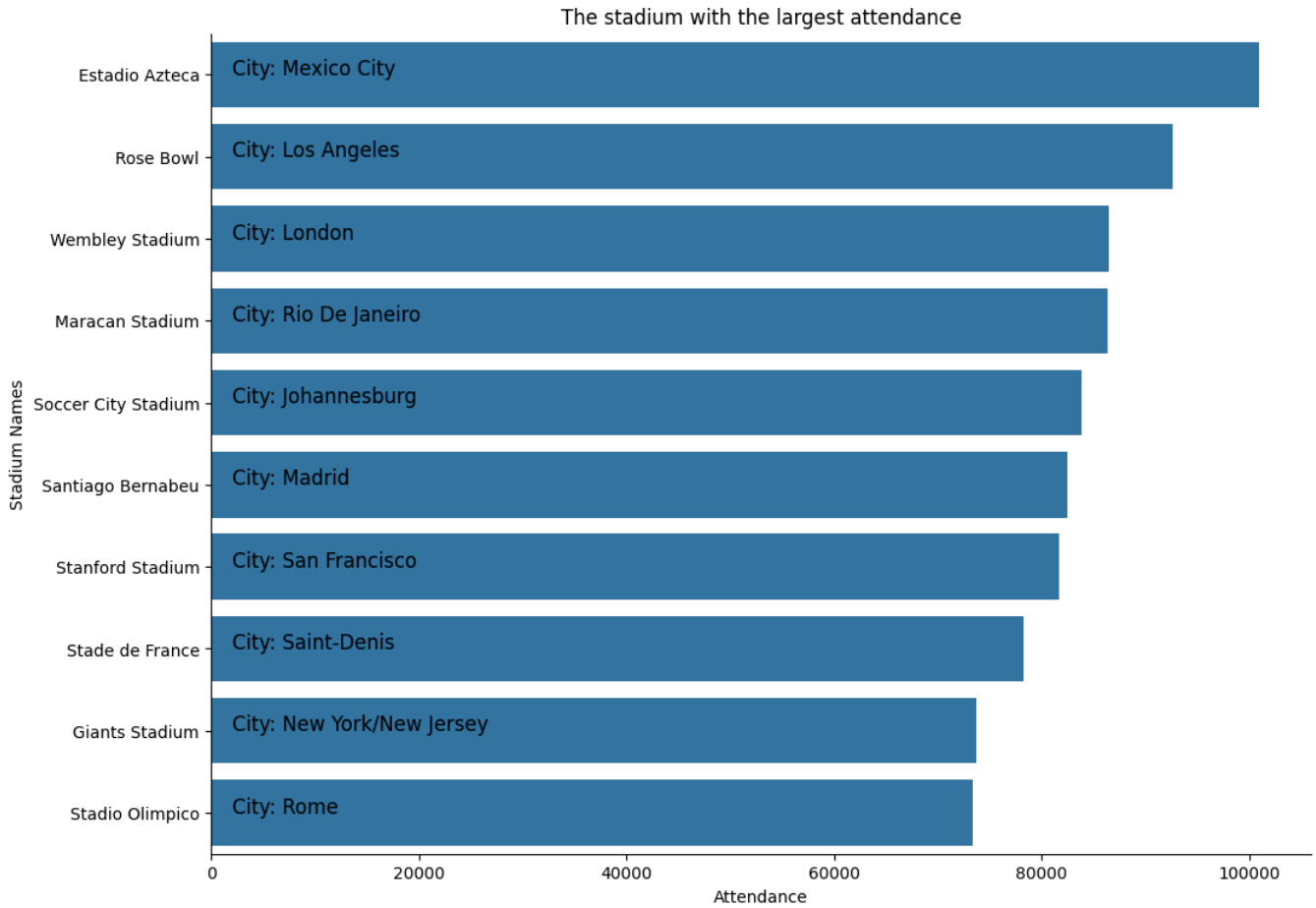
```
matches['Datetime'] = pd.to_datetime(matches['Datetime'], errors='coerce')
matches['Datetime'] = matches['Datetime'].apply(lambda x: x.strftime('%d %b, %y') if pd.notnull(x) else None)
```

```
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 that recorded the highest audience turnout.')

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



The stadium with the highest average crowd attendance.

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

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

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('The stadium with the largest attendance')
for i, s in enumerate("City: " + top10['City']):
        ax.text(2000, i, s, fontsize = 12, color = 'k')

plt.show()
```




```
import cufflinks as cf
import pandas as pd
import plotly.offline as pyo

# Initialize Plotly offline mode                                          showing error
cf.go_offline()
pyo.init_notebook_mode(connected=True)

# Ensure no NaN values and column existence
if 'City' in matches.columns:
    matches['City'].dropna().value_counts()[:20].iplot(kind='bar', title='Top 20 Cities by Match Count')
else:
    print("Column 'City' not found in the dataset.")
```

Goals Scored by Each Team in Every World Cup.

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

	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
0	1930	13 Jul, 30	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		4444.0	3.0	0.0	LOMBARDI Domingo (URU)
1	1930	13 Jul, 30	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium		18346.0	2.0	0.0	MACIAS Jose (ARG)

Next steps: Generate code with matches | View recommended plots | New interactive sheet

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

		Home Team Goals
Year	**Home Team Name**	
1930	**Argentina**	16.0
	Brazil	4.0
	Chile	4.0
	France	4.0
	Paraguay	1.0
...	**...**	...
2014	**Russia**	1.0
	Spain	1.0
	Switzerland	4.0
	USA	2.0
	Uruguay	3.0

366 rows × 1 columns

dtype: float64

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

		Away Team Goals
Year	**Away Team Name**	
1930	**Argentina**	2.0
	Belgium	0.0
	Bolivia	0.0
	Brazil	1.0
	Chile	1.0
...	**...**	...
2014	**Russia**	1.0
	Spain	3.0
	Switzerland	3.0
	USA	4.0
	Uruguay	1.0

411 rows × 1 columns

dtype: float64

```
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
```

		Goals
Year		
1930	**Argentina**	18.0
	Brazil	5.0
	Chile	5.0
	France	4.0
	Paraguay	1.0
...	**...**	...
1998	**Iran**	2.0
	Mexico	8.0
	Norway	5.0
	Tunisia	1.0
2006	**IR Iran**	0.0

427 rows × 1 columns

Next steps: Generate code with goals | View recommended plots | New interactive sheet

```
goals = goals.reset_index()
```

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

	Year	Country	Goals
0	1930	Argentina	18.0
7	1930	Uruguay	15.0
6	1930	USA	7.0
8	1930	Yugoslavia	7.0
1	1930	Brazil	5.0
...	...	...	...
355	2014	Japan	2.0
361	2014	Russia	2.0
340	2014	Cameroon	1.0
352	2014	Honduras	1.0
353	2014	IR Iran	1.0

427 rows × 3 columns

Next steps: Generate code with goals | View recommended plots | New interactive sheet

```
top5 = goals.groupby('Year').head()
top5.head(10)
```

	Year	Country	Goals
0	1930	Argentina	18.0
7	1930	Uruguay	15.0
6	1930	USA	7.0
8	1930	Yugoslavia	7.0
1	1930	Brazil	5.0
13	1934	Italy	12.0
11	1934	Germany	11.0
10	1934	Czechoslovakia	9.0
9	1934	Austria	7.0
12	1934	Hungary	5.0

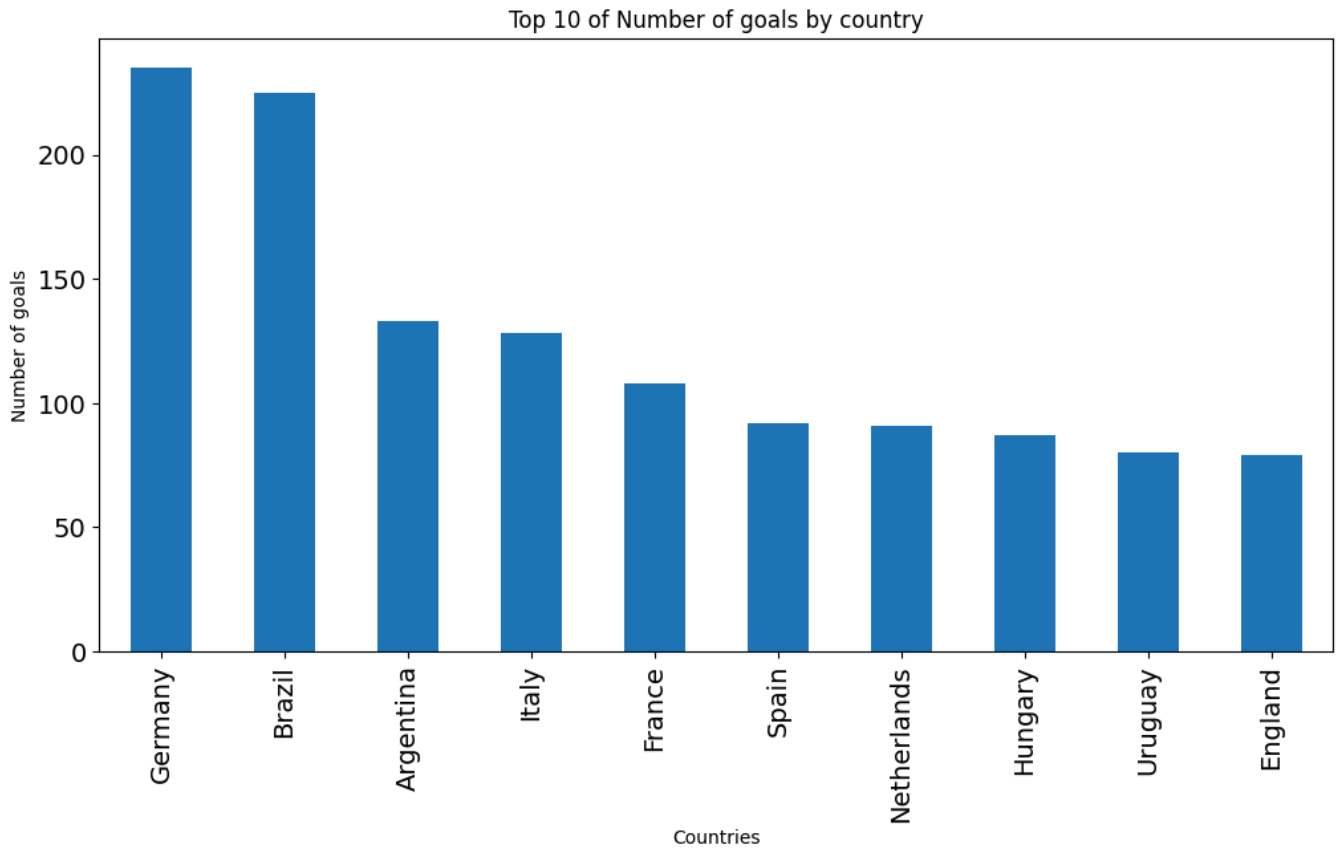
Next steps: Generate code with top5 | View recommended plots | New interactive sheet

```
import plotly.graph_objects as go
import plotly.offline as pyo

# Initialize Plotly offline mode
pyo.init_notebook_mode(connected=True)

# Check the data for potential issues
print(top5.head())  # Display the first few rows
print(top5.isnull().sum())  # Check for missing values

# Filter out rows with missing data
top5 = top5.dropna(subset=['Country', 'Year', 'Goals'])

# Prepare the data for plotting
data = []
for team in top5['Country'].drop_duplicates().values:
    year = top5[top5['Country'] == team]['Year']
    goal = top5[top5['Country'] == team]['Goals']

    # Ensure non-empty data
    if not year.empty and not goal.empty:
        data.append(go.Bar(x=year, y=goal, name=team))

# Create the layout for the plot
layout = go.Layout(barmode='stack', title='Top 5 Teams with Most Goals', showlegend=True)

# Create the figure and display it
fig = go.Figure(data=data, layout=layout)
fig.show()
```

```
   Year     Country  Goals
0  1930   Argentina   18.0
7  1930     Uruguay   15.0
6  1930         USA    7.0
8  1930  Yugoslavia    7.0
1  1930      Brazil    5.0
Year       0
Country    0
Goals      0
dtype: int64
```

Goals scored by each country.

```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming 'matches' is your original DataFrame

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

# Create goal_per_country DataFrame
goal_per_country = pd.DataFrame(columns=['countries', 'goals'])
goal_per_country = pd.concat([
    home.rename(columns={'Home Team Name': 'countries', 'Home Team Goals': 'goals'}),
    away.rename(columns={'Away Team Name': 'countries', 'Away Team Goals': 'goals'})
])

# Convert 'goals' to integer type
goal_per_country['goals'] = goal_per_country['goals'].astype('int64')

# Group by country and sum the goals, then sort
goal_per_country = goal_per_country.groupby('countries')['goals'].sum().sort_values(ascending=False)

# Plot the top 10 countries with the most goals
goal_per_country[:10].plot(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')
plt.show()
```



Match outcomes categorized by home and away teams.

```
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'
```

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

```
matches.head()
```

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	...	Half-time Home Goals	Half-time Away Goals	Referee	Assi
0	1930	13 Jul, 30	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		...	3.0	0.0	LOMBARDI Domingo (URU)	CRIST Henry
1	1930	13 Jul, 30	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium		...	2.0	0.0	MACIAS Jose (ARG)	MATE Fra
2	1930	14 Jul, 30	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil		...	2.0	0.0	TEJADA Anibal (URU)	VALLA R
3	1930	14 Jul, 30	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		...	1.0	0.0	WARNKEN Alberto (CHI)	LANG Jean
4	1930	15 Jul, 30	Group 1	Parque Central	Montevideo	Argentina	1.0	0.0	France		...	0.0	0.0	REGO Gilberto (BRA)	SAU Ulises

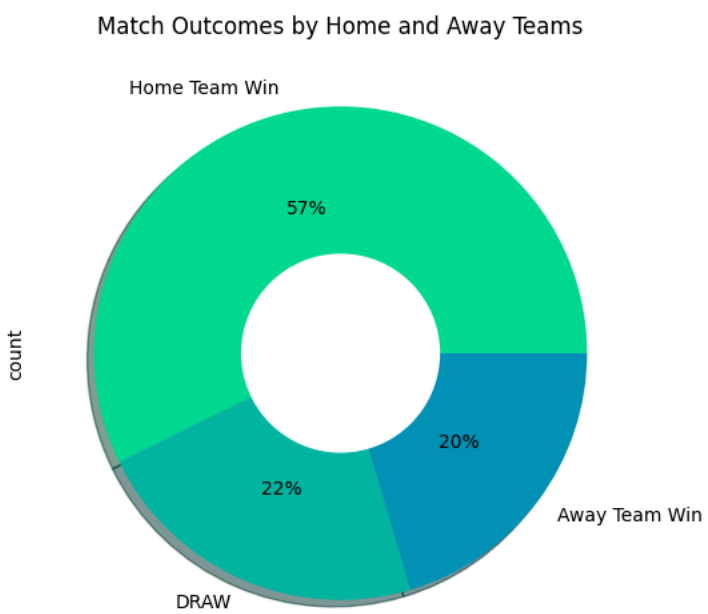
5 rows × 21 columns

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

outcome	count
Home Team Win	488
DRAW	190
Away Team Win	174

dtype: int64

```
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()
```



The team with the most World Cup titles.

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

Winner	count
Brazil	5
Italy	4
Germany	4
Uruguay	2
Argentina	2
England	1
France	1
Spain	1

dtype: int64

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

Runners-Up	count
Germany	4
Argentina	3
Netherlands	3
Czechoslovakia	2
Hungary	2
Brazil	2
Italy	2
Sweden	1
France	1

dtype: int64

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

Third	count
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

dtype: int64

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