Import libraries

In [66]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from datetime import datetime as dt
import seaborn as sns
```

Datasets

In [5]:

```
cups =pd.read_csv(r"C:\Users\kruna\OneDrive\Desktop\internship\fifa world cup\WorldCups.csv")
cups.head()
```

Out[5]:

	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

In [6]:

```
matches =pd.read_csv(r"C:\Users\kruna\OneDrive\Desktop\internship\fifa world cup\WorldCupMatches.csv")
matches .head()
```

Out[6]:

	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	Ass
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		4444.0	3.0	0.0	LOMBARDI Domingo (URU)	CRIS Henr
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	MACIAS Jose (ARG)	MAT Fr
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	TEJADA Anibal (URU)	VALL
3	1930.0	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		2549.0	1.0	0.0	WARNKEN Alberto (CHI)	LAN Jea
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	REGO Gilberto (BRA)	SAI Ulise

In [7]:

```
players =pd.read_csv(r"C:\Users\kruna\OneDrive\Desktop\internship\fifa world cup\WorldCupPlayers.csv" )
players.head()
```

Out[7]:

	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

User-defined function for data quality che

In [8]:

```
def dataset_integrity_check(df):
    print(df.info())
    print(df.isna().sum())
    print(sum(df.duplicated()))
    print(df.describe())
```

Dataset check - Cups

In [10]:

```
dataset_integrity_check(cups)
cups.head()
```

```
<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
None
Year              0
Country           0
Winner            0
Runners-Up        0
Third             0
Fourth            0
GoalsScored       0
QualifiedTeams    0
MatchesPlayed     0
Attendance        0
dtype: int64
0
              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
```

Out[10]:

	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

Dataset check - Matches

In [11]:

```
dataset_integrity_check(matches)
matches.head()
```

```
<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
None
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
3735
              Year  Home Team Goals  Away Team Goals     Attendance  \
count   852.000000       852.000000       852.000000     850.000000   
mean   1985.089202         1.811033         1.022300   45164.800000   
std      22.448825         1.610255         1.087573   23485.249247   
min    1930.000000         0.000000         0.000000    2000.000000   
25%    1970.000000         1.000000         0.000000   30000.000000   
50%    1990.000000         2.000000         1.000000   41579.500000   
75%    2002.000000         3.000000         2.000000   61374.500000   
max    2014.000000        10.000000         7.000000  173850.000000   

       Half-time Home Goals  Half-time Away Goals       RoundID       MatchID  
count            852.000000            852.000000  8.520000e+02  8.520000e+02  
mean               0.708920              0.428404  1.066177e+07  6.134687e+07  
std                0.937414              0.691252  2.729613e+07  1.110572e+08  
min                0.000000              0.000000  2.010000e+02  2.500000e+01  
25%                0.000000              0.000000  2.620000e+02  1.188750e+03  
50%                0.000000              0.000000  3.370000e+02  2.191000e+03  
75%                1.000000              1.000000  2.497220e+05  4.395006e+07  
max                6.000000              5.000000  9.741060e+07  3.001865e+08  
```

Out[11]:

	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	Ass
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		4444.0	3.0	0.0	LOMBARDI Domingo (URU)	CRIS Henr
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	MACIAS Jose (ARG)	MAT Fr
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	TEJADA Anibal (URU)	VALL
3	1930.0	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		2549.0	1.0	0.0	WARNKEN Alberto (CHI)	LAN Jea
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	REGO Gilberto (BRA)	SAU Ulise

Dataset check - Players

In [12]:

```
dataset_integrity_check(players)
players.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37784 entries, 0 to 37783
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   RoundID        37784 non-null  int64 
 1   MatchID        37784 non-null  int64 
 2   Team Initials  37784 non-null  object
 3   Coach Name     37784 non-null  object
 4   Line-up        37784 non-null  object
 5   Shirt Number   37784 non-null  int64 
 6   Player Name    37784 non-null  object
 7   Position       4143 non-null   object
 8   Event          9069 non-null   object
dtypes: int64(3), object(6)
memory usage: 2.6+ MB
None
RoundID              0
MatchID              0
Team Initials        0
Coach Name           0
Line-up              0
Shirt Number         0
Player Name          0
Position         33641
Event            28715
dtype: int64
736
            RoundID       MatchID  Shirt Number
count  3.778400e+04  3.778400e+04  37784.000000
mean   1.105647e+07  6.362233e+07     10.726022
std    2.770144e+07  1.123916e+08      6.960138
min    2.010000e+02  2.500000e+01      0.000000
25%    2.630000e+02  1.199000e+03      5.000000
50%    3.370000e+02  2.216000e+03     11.000000
75%    2.559310e+05  9.741000e+07     17.000000
max    9.741060e+07  3.001865e+08     23.000000
```

Out[12]:

	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

Null value treatment & de-duplication

In [13]:

```
cups = cups.drop_duplicates().dropna(how = 'all')
players = players.drop_duplicates().dropna(how = 'all')
matches = matches.drop_duplicates().dropna(how = 'all')

print(len(cups))
print(len(players))
print(len(matches))
```

```
20
37048
836
```

In [72]:

```
cups.isnull().sum()
```

Out[72]:

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

In [73]:

```
players.isnull().sum()
```

Out[73]:

```
RoundID                 0
MatchID                 0
Team Initials           0
Coach Name              0
Line-up                 0
Shirt Number            0
Player Name             0
Position            33030
Event               28225
Count               28225
Cards               28225
Penalties           28225
Penalties Scored    28225
Own Goals           28225
dtype: int64
```

In [74]:

```
matches.isnull().sum()
```

Out[74]:

```
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              1
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 [14]:

```
np.unique(cups['Winner'])
```

Out[14]:

```
array(['Argentina', 'Brazil', 'England', 'France', 'Germany',
       'Germany FR', 'Italy', 'Spain', 'Uruguay'], dtype=object)
```

Data treatment - Standardization like Converting 'Germany FR' to 'Germany'

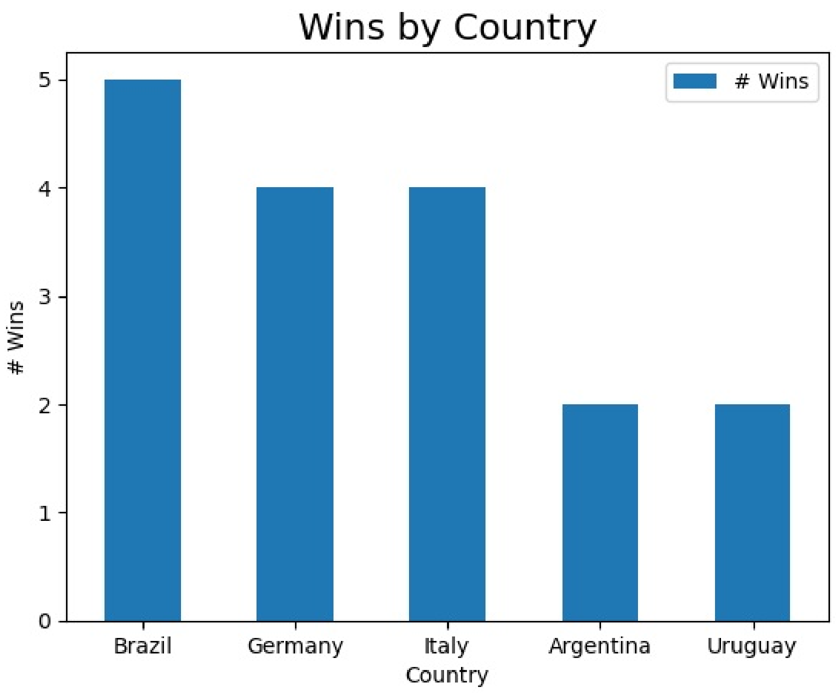
In [15]:

```
cups['Winner'] = np.where(cups['Winner'] == 'Germany FR', 'Germany', cups['Winner'])
```

Top 5 countries who have won the most number of FIFA World Cups

In [17]:

```
top5 = cups.groupby(['Winner'], as_index = False).agg({"Year":"count"}).sort_values(['Year'], ascending = False
top5.columns = ['Team', '# Wins']
plot1 = top5.plot.bar(x='Team', y='# Wins', rot=0)
plot1.set_xlabel('Country')
plot1.set_ylabel('# Wins')
plot1.set_title('Wins by Country', fontsize = 17)
plt.show()
top5
```



Out[17]:

	Team	# Wins
1	Brazil	5
4	Germany	4
5	Italy	4
0	Argentina	2
7	Uruguay	2

many times did the host country win the world cup

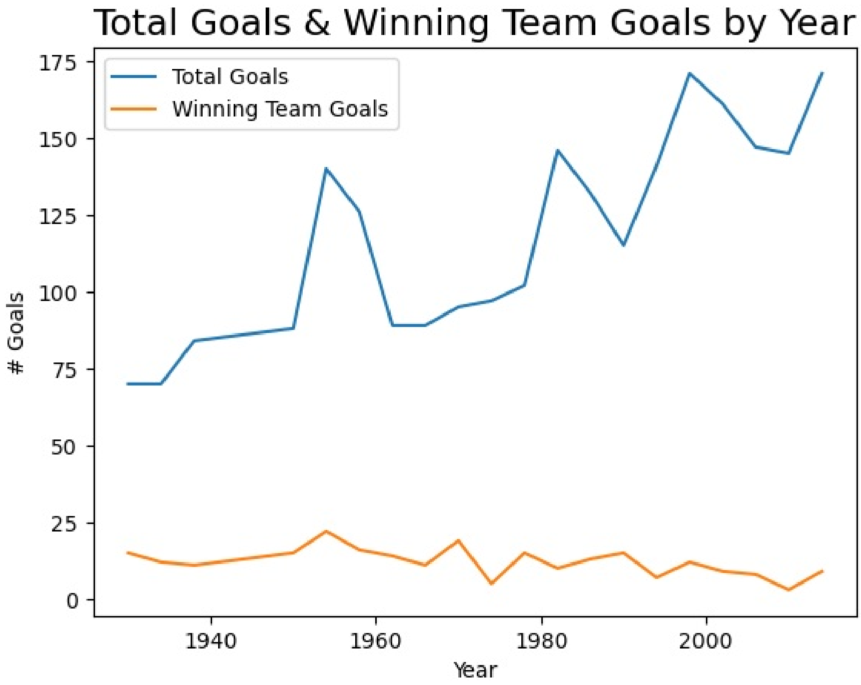
In [32]:

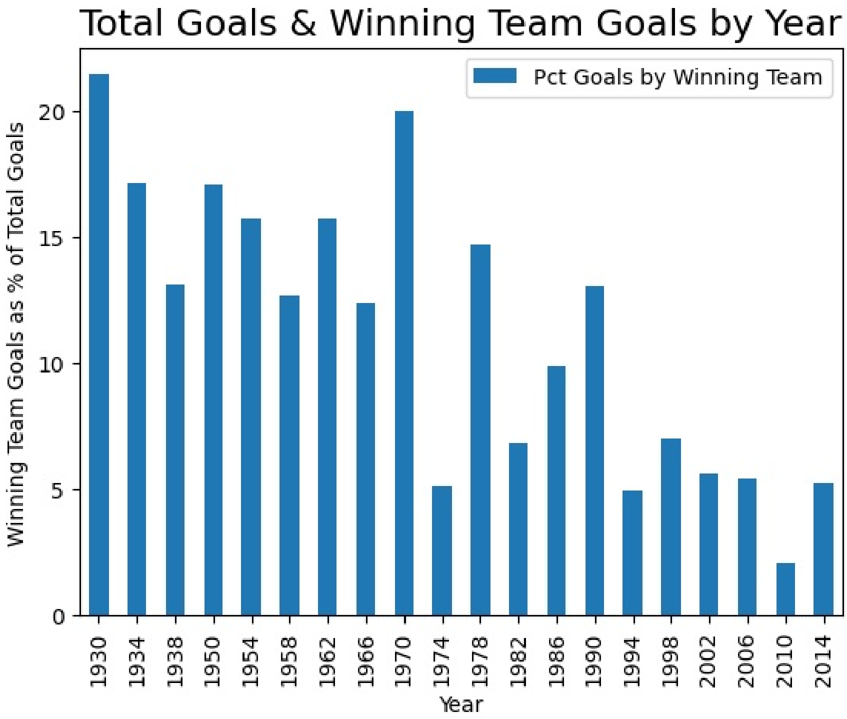
```
print(len(cups[cups['Country']==cups['Winner']]))
```

```
6
```

In [43]:

```
total_goals = cups[['Year', 'Winner', 'GoalsScored']]
total_goals.columns = ['Year', 'Team', 'Total Goals']

hteam_goals = matches[['Year', 'Home Team Name', 'Home Team Goals', 'RoundID', 'MatchID', 'Home Team Initials']]
hteam_goals.columns = ['Year', 'Team', 'Winning Team Goals', 'RoundID', 'MatchID', 'Team Initials']
ateam_goals = matches[['Year', 'Away Team Name', 'Away Team Goals', 'RoundID', 'MatchID', 'Away Team Initials']]
ateam_goals.columns = ['Year', 'Team', 'Winning Team Goals', 'RoundID', 'MatchID', 'Team Initials']
combined_goals = hteam_goals + ateam_goals


goals_agg = goals.groupby(['Year', 'Team'], as_index = False).agg({"Winning Team Goals":"sum"})

final_goal_score = pd.merge(total_goals, goals_agg, how = 'left', on = ['Year', 'Team'])
final_goal_score['Pct Goals by Winning Team'] = np.round(final_goal_score['Winning Team Goals']/final_goal_scor
plot2 = final_goal_score[['Year', 'Total Goals', 'Winning Team Goals']].plot.line(x = 'Year')
plot2.set_xlabel('Year')
plot2.set_ylabel('# Goals')
plot2.set_title('Total Goals & Winning Team Goals by Year', fontsize = 17)
plt.show()

plot3 = final_goal_score[['Year', 'Pct Goals by Winning Team']].plot.bar(x = 'Year')
plot3.set_xlabel('Year')
plot3.set_ylabel('Winning Team Goals as % of Total Goals')
plot3.set_title('Total Goals & Winning Team Goals by Year', fontsize = 17)
plt.show()

final_goal_score[['Year', 'Total Goals', 'Team', 'Winning Team Goals', 'Pct Goals by Winning Team']]
```


Total Goals & Winning Team Goals by Year
Total Goals
Winning Team Goals
# Goals
175
150
125
100
75
50
25
0
1940
1960
1980
2000
Year


Total Goals & Winning Team Goals by Year
Pct Goals by Winning Team
Winning Team Goals as % of Total Goals
20
15
10
5
0
1930
1934
1938
1950
1954
1958
1962
1966
1970
1974
1978
1982
1986
1990
1994
1998
2002
2006
2010
2014
Year

Out[43]:

	Year	Total Goals	Team	Winning Team Goals	Pct Goals by Winning Team
0	1930	70	Uruguay	15.0	21.43
1	1934	70	Italy	12.0	17.14
2	1938	84	Italy	11.0	13.10
3	1950	88	Uruguay	15.0	17.05
4	1954	140	Germany	22.0	15.71
5	1958	126	Brazil	16.0	12.70
6	1962	89	Brazil	14.0	15.73
7	1966	89	England	11.0	12.36
8	1970	95	Brazil	19.0	20.00
9	1974	97	Germany	5.0	5.15
10	1978	102	Argentina	15.0	14.71
11	1982	146	Italy	10.0	6.85
12	1986	132	Argentina	13.0	9.85
13	1990	115	Germany	15.0	13.04
14	1994	141	Brazil	7.0	4.96
15	1998	171	France	12.0	7.02
16	2002	161	Brazil	9.0	5.59
17	2006	147	Italy	8.0	5.44
18	2010	145	Spain	3.0	2.07
19	2014	171	Germany	9.0	5.26

The top scoring players of each season with hey from the winning team

In [47]:

```
players['Count'] = players['Event'].str.count('G|W|P')
players_scored = players[players['Count']>0].groupby(['RoundID', 'MatchID', 'Player Name', 'Team Initials'], as

scored = pd.merge(goals, players_scored, how = 'left', on = ['RoundID', 'MatchID', 'Team Initials'])
scored_agg = scored.groupby(['Year', 'Team', 'Player Name'], as_index = False).agg({"Count":"sum"})
scored_agg.columns = ['Year', 'Player Team', 'Player Name', 'Count']

scored_agg['Rank'] = scored_agg.groupby(['Year'], as_index=False)['Count'].rank("dense", ascending=False)
scored_max = scored_agg[scored_agg['Rank']==1]

top_scorers = pd.merge(final_goal_score, scored_max, how = 'outer', on = ['Year'])
top_scorers['IsWinningTeam'] = np.where(top_scorers['Team']==top_scorers['Player Team'], 'Yes', 'No')
top_scorers[['Year', 'Player Team', 'Player Name', 'Count', 'IsWinningTeam']]
```

Out[47]:

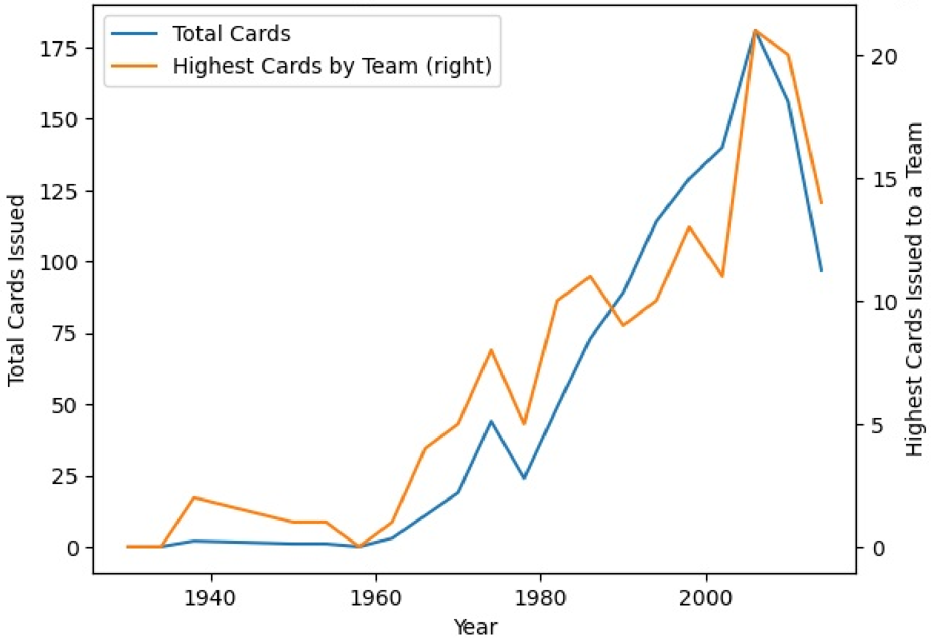
	Year	Player Team	Player Name	Count	IsWinningTeam
0	1930	Argentina	Guillermo STABILE	7.0	No
1	1934	Czechoslovakia	Oldrich NEJEDLY	5.0	No
2	1938	Brazil	LEONIDAS	7.0	No
3	1950	Brazil	ADEMIR	8.0	No
4	1954	Hungary	Sandor KOCSIS	11.0	No
5	1958	France	Just FONTAINE	10.0	No
6	1962	Brazil	GARRINCHA	4.0	Yes
7	1962	Brazil	VAVA	4.0	Yes
8	1962	Hungary	Florian ALBERT	4.0	No
9	1962	Soviet Union	Valentin IVANOV	4.0	No
10	1966	Portugal	EUSEBIO (Eusebio da Silva Ferreira)	8.0	No
11	1970	Germany	Gerd MUELLER	8.0	No
12	1974	Argentina	Rene HOUSEMAN	3.0	No
13	1974	Netherlands	Johan CRUYFF	3.0	No
14	1974	Poland	Grzegorz LATO	3.0	No
15	1974	Sweden	Ralf EDSTROM	3.0	No
16	1974	Yugoslavia	Dusan BAJEVIC	3.0	No
17	1978	Argentina	Mario KEMPES	6.0	Yes
18	1982	Germany	Karl-Heinz RUMMENIGGE	5.0	No
19	1986	England	Gary LINEKER	5.0	No
20	1990	Italy	Salvatore SCHILLACI	6.0	No
21	1994	Germany	KLINSMANN	5.0	No
22	1994	Russia	Oleg SALENKO	5.0	No
23	1998	Argentina	Gabriel BATISTUTA	5.0	No
24	1998	Italy	Christian VIERI	5.0	No
25	2002	Brazil	RONALDO	4.0	Yes
26	2002	Germany	KLOSE	4.0	No
27	2006	Argentina	CRESPO	3.0	No
28	2006	Argentina	RODRIGUEZ	3.0	No
29	2006	Germany	KLOSE	3.0	No
30	2006	Spain	DAVID VILLA	3.0	No
31	2006	Spain	F. TORRES	3.0	No
32	2010	Argentina	HIGUAIN	4.0	No
33	2010	Netherlands	SNEIJDER	4.0	No
34	2010	Spain	DAVID VILLA	4.0	Yes
35	2014	Colombia	JAMES	4.0	No

Wards (Red/Yellow) were issued in each season and team was issued the highest no. of cards in a season

In [49]:

```
players['Cards'] = players['Event'].str.count('R|Y|RSY')
players_fined = players[players['Cards']>0].groupby(['RoundID', 'MatchID', 'Team Initials'], as_index = False).

fined = pd.merge(goals, players_fined, how = 'left', on = ['RoundID', 'MatchID', 'Team Initials'])
fined_agg = fined.groupby(['Year', 'Team'], as_index = False).agg({"Cards":"sum"})
fined_agg.columns = ['Year', 'Player Team', 'Highest Cards by Team']

fined_agg['Rank'] = fined_agg.groupby(['Year'], as_index=False)['Highest Cards by Team'].rank("dense", ascendin
fined_max = fined_agg[fined_agg['Rank']==1]

final_fine_score = fined_agg.groupby(['Year'], as_index = False).agg({"Highest Cards by Team":"sum"})
final_fine_score.columns = ['Year','Total Cards']

top_fined = pd.merge(final_fine_score, fined_max, how = 'outer', on = ['Year'])
# top_scorers['IsWinningTeam'] = np.where(top_scorers['Team']==top_scorers['Player Team'], 'Yes', 'No')

plot4 = top_fined[['Year', 'Total Cards', 'Highest Cards by Team']].plot.line(x = 'Year', secondary_y = 'Highes
plot4.set_xlabel('Year')
plot4.set_ylabel('Total Cards Issued')
plot4.right_ax.set_ylabel('Highest Cards Issued to a Team')
plot4.set_title('Total Cards Issued & Highest Cards Issued to Team  by Year', fontsize = 17)
plt.show()
top_fined[['Year', 'Total Cards', 'Player Team', 'Highest Cards by Team']]
```

Total Cards Issued & Highest Cards Issued to Team by Year



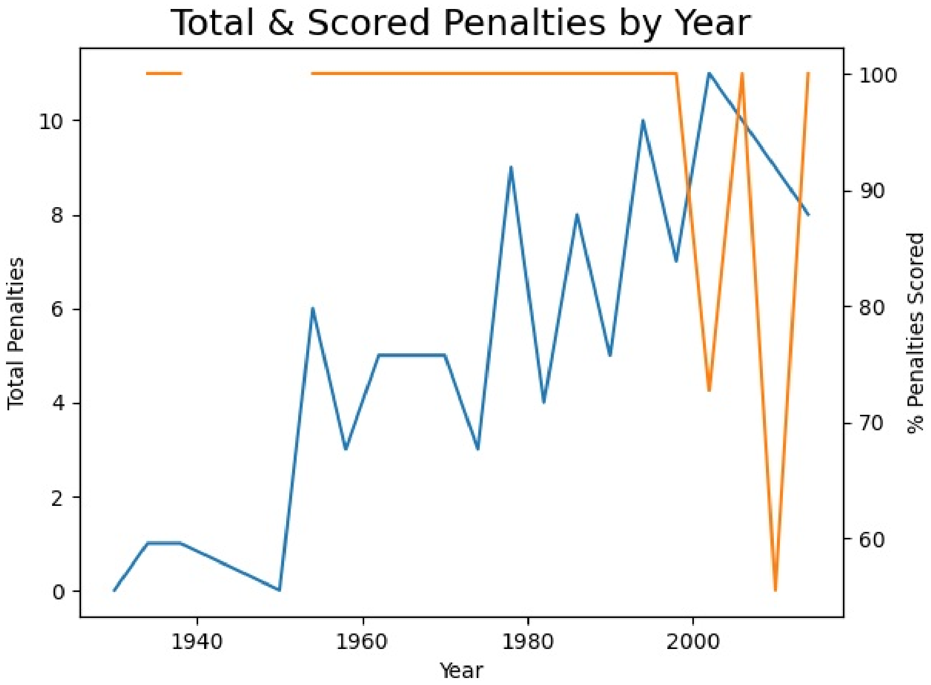
Out[49]:

	Year	Total Cards	Player Team	Highest Cards by Team
0	1930.0	0.0	Argentina	0.0
1	1930.0	0.0	Brazil	0.0
2	1930.0	0.0	Chile	0.0
3	1930.0	0.0	France	0.0
4	1930.0	0.0	Paraguay	0.0
5	1930.0	0.0	Romania	0.0
6	1930.0	0.0	USA	0.0
7	1930.0	0.0	Uruguay	0.0
8	1930.0	0.0	Yugoslavia	0.0
9	1934.0	0.0	Austria	0.0
10	1934.0	0.0	Czechoslovakia	0.0
11	1934.0	0.0	Germany	0.0
12	1934.0	0.0	Hungary	0.0
13	1934.0	0.0	Italy	0.0
14	1934.0	0.0	Spain	0.0
15	1934.0	0.0	Sweden	0.0
16	1934.0	0.0	Switzerland	0.0
17	1938.0	2.0	Brazil	2.0
18	1950.0	1.0	Brazil	1.0
19	1954.0	1.0	Hungary	1.0
20	1958.0	0.0	Argentina	0.0
21	1958.0	0.0	Brazil	0.0
22	1958.0	0.0	Czechoslovakia	0.0
23	1958.0	0.0	England	0.0
24	1958.0	0.0	France	0.0
25	1958.0	0.0	Germany	0.0
26	1958.0	0.0	Hungary	0.0
27	1958.0	0.0	Mexico	0.0
28	1958.0	0.0	Northern Ireland	0.0
29	1958.0	0.0	Paraguay	0.0
30	1958.0	0.0	Soviet Union	0.0
31	1958.0	0.0	Sweden	0.0
32	1958.0	0.0	Wales	0.0
33	1958.0	0.0	Yugoslavia	0.0
34	1962.0	3.0	Argentina	1.0
35	1962.0	3.0	Brazil	1.0
36	1962.0	3.0	Yugoslavia	1.0
37	1966.0	11.0	Germany	4.0
38	1970.0	19.0	Italy	5.0
39	1974.0	44.0	Netherlands	8.0
40	1978.0	24.0	Argentina	5.0
41	1978.0	24.0	Brazil	5.0
42	1982.0	49.0	Italy	10.0
43	1986.0	73.0	Iraq	11.0
44	1990.0	89.0	Austria	9.0
45	1990.0	89.0	Germany	9.0
46	1994.0	114.0	Bulgaria	10.0
47	1994.0	114.0	Romania	10.0
48	1998.0	129.0	Germany	13.0
49	2002.0	140.0	Cameroon	11.0
50	2006.0	181.0	Portugal	21.0
51	2010.0	156.0	Netherlands	20.0
52	2014.0	97.0	Brazil	14.0

penalties were taken in each season with penalties were successful

In [54]:

```
players['Penalties'] = players['Event'].str.count('P')
players['Penalties Scored'] = players['Penalties'] - players['Event'].str.count('MP')

penalties = players[players['Penalties']>0].groupby(['RoundID', 'MatchID', 'Team Initials'], as_index = False).

penalties_scored = pd.merge(goals, penalties, how = 'left', on = ['RoundID', 'MatchID', 'Team Initials'])

final_penalties_scored = penalties_scored.groupby(['Year'], as_index = False).agg({"Penalties":"sum", "Penaltie
final_penalties_scored['Perc Scored'] = np.round(final_penalties_scored['Penalties Scored']/final_penalties_sco
plot5 = final_penalties_scored[['Year', 'Penalties','Perc Scored']].plot.line(x = 'Year', secondary_y = 'Perc S
plot5.set_xlabel('Year')
plot5.set_ylabel('Total Penalties')
plot5.right_ax.set_ylabel('% Penalties Scored')
plot5.set_title('Total & Scored Penalties by Year', fontsize = 17)
plt.show()

final_penalties_scored
```



Out[54]:

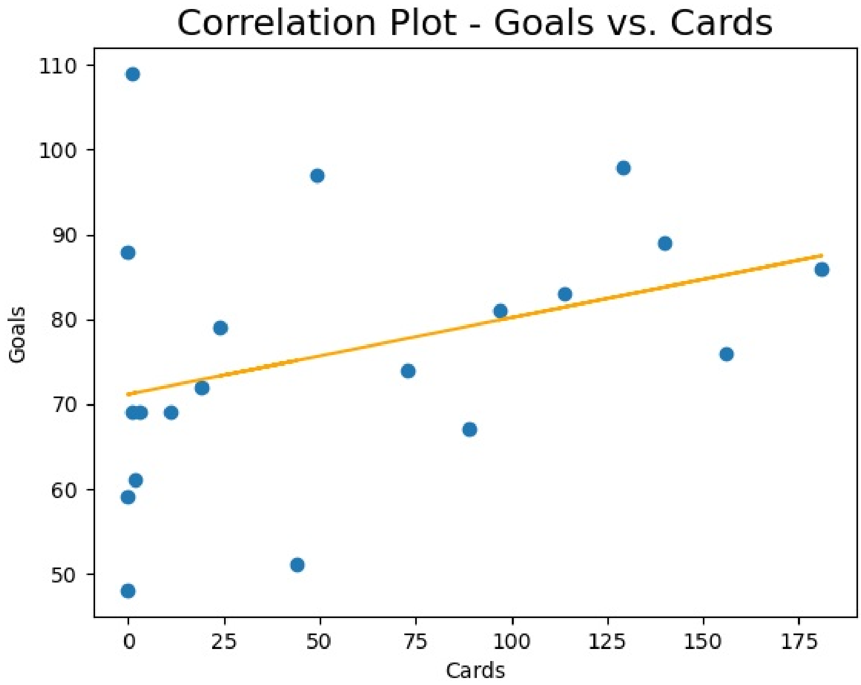
	Year	Penalties	Penalties Scored	Perc Scored
0	1930.0	0.0	0.0	NaN
1	1934.0	1.0	1.0	100.00
2	1938.0	1.0	1.0	100.00
3	1950.0	0.0	0.0	NaN
4	1954.0	6.0	6.0	100.00
5	1958.0	3.0	3.0	100.00
6	1962.0	5.0	5.0	100.00
7	1966.0	5.0	5.0	100.00
8	1970.0	5.0	5.0	100.00
9	1974.0	3.0	3.0	100.00
10	1978.0	9.0	9.0	100.00
11	1982.0	4.0	4.0	100.00
12	1986.0	8.0	8.0	100.00
13	1990.0	5.0	5.0	100.00
14	1994.0	10.0	10.0	100.00
15	1998.0	7.0	7.0	100.00
16	2002.0	11.0	8.0	72.73
17	2006.0	10.0	10.0	100.00
18	2010.0	9.0	5.0	55.56
19	2014.0	8.0	8.0	100.00

Any correlation between number of yellow cards issued to a team in a match with the number of goals scored by the team in that match?

In [56]:

```
card_vs_goals = pd.merge(goals, players_fined, how = 'left', on = ['RoundID', 'MatchID', 'Team Initials'])
card_vs_goals.columns = ['Year', 'Team', 'Goals', 'RoundID', 'MatchID', 'Team Initials', 'Cards']
card_vs_goals_agg = card_vs_goals.groupby(['Year'], as_index = False).agg({"Cards":"sum", "Goals":"sum"})
```

```
card_vs_goals_agg[['Cards', 'Goals']].plot.scatter(x = 'Cards', y = 'Goals')
plt.scatter(card_vs_goals_agg['Cards'], card_vs_goals_agg['Goals'])

z = np.polyfit(card_vs_goals_agg['Cards'], card_vs_goals_agg['Goals'], 1)
p = np.poly1d(z)
plt.plot(card_vs_goals_agg['Cards'],p(card_vs_goals_agg['Cards']),"orange")
plt.title('Correlation Plot - Goals vs. Cards', fontsize = 17)
plt.show()

card_vs_goals_agg[['Cards', 'Goals']].corr()
```



Out[56]:

	Cards	Goals
Cards	1.000000	0.349952
Goals	0.349952	1.000000

own goals were scored in all seasons with the teams that scored own goals, which team scored the highest

In [57]:

```
players['Own Goals'] = players['Event'].str.count('W')

own_goals = players[players['Own Goals']>0].groupby(['RoundID', 'MatchID', 'Team Initials'], as_index = False).
own_goals_scored = pd.merge(goals, own_goals, how = 'left', on = ['RoundID', 'MatchID', 'Team Initials'])
own_goals_scored_agg = own_goals_scored[own_goals_scored['Own Goals']>0].groupby(['Year', 'Team'], as_index = F
print(sum(own_goals_scored_agg['Own Goals']))
own_goals_scored_agg.sort_values(['Own Goals'], ascending = False)[0:1]
```

```
11.0
```

Out[57]:

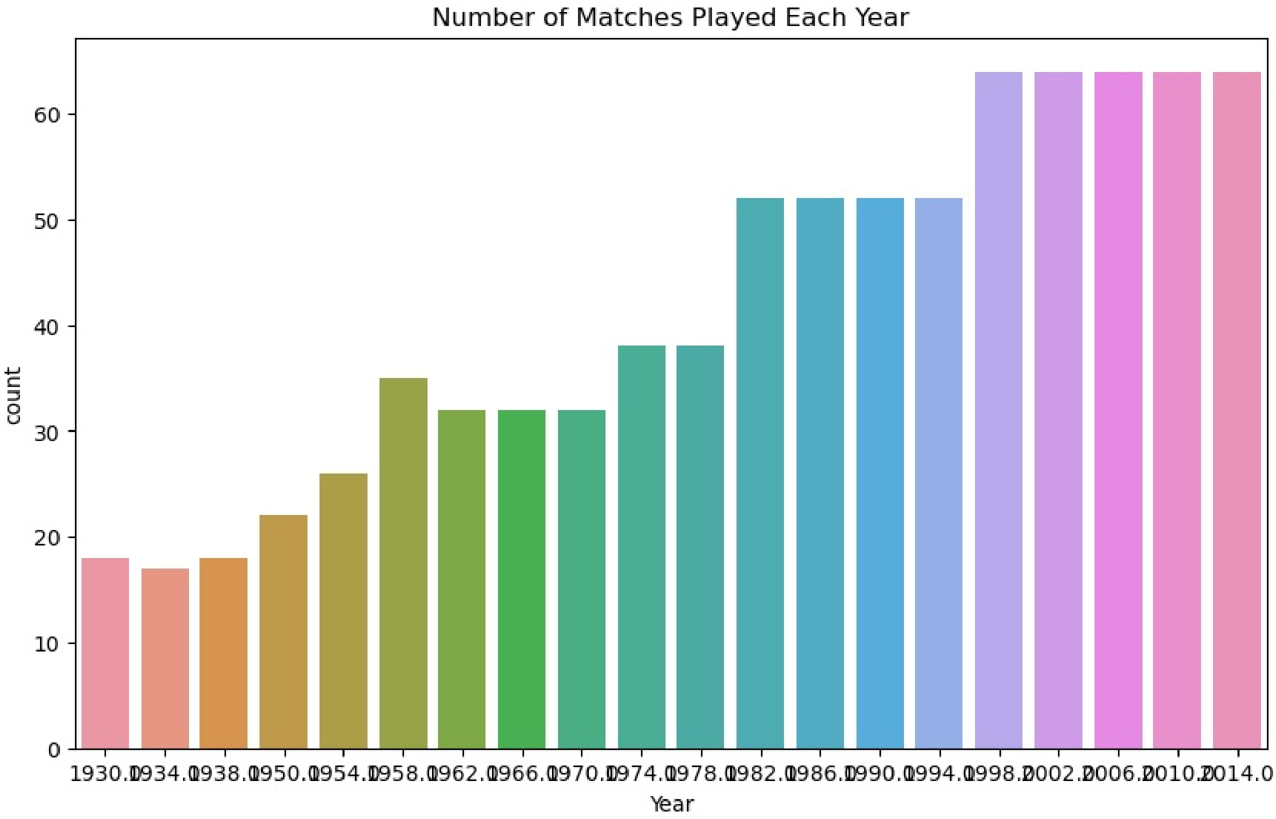
	Year	Team	Own Goals
0	1938.0	Switzerland	1.0

In [ ]:

```
Number of matches played each year
```

In [67]:

```
plt.figure(figsize=(10,6))
sns.countplot(x='Year', data=matches)
plt.title('Number of Matches Played Each Year')
plt.show()
```

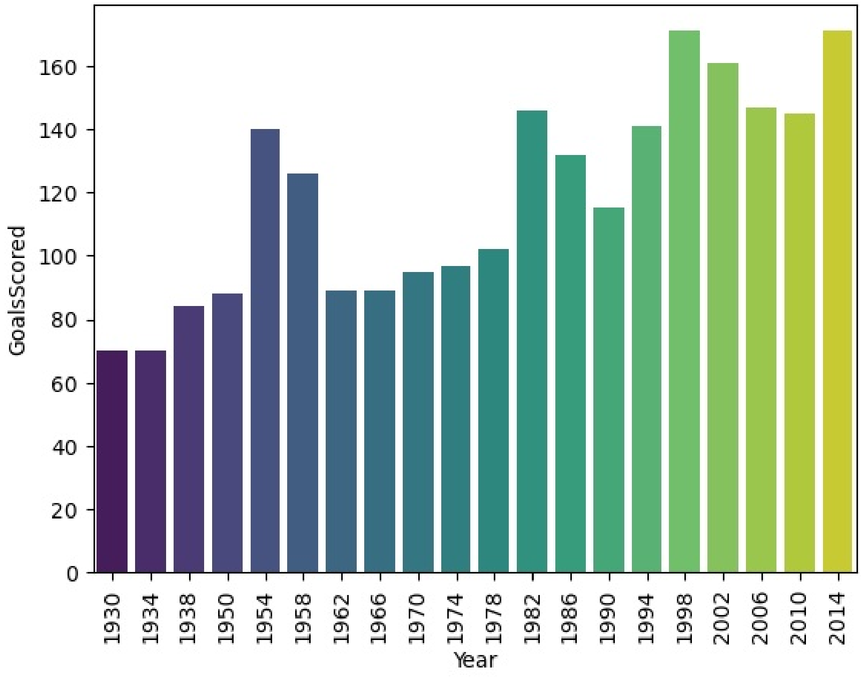


In [71]:

```
sns.barplot(x = cups['Year'], y = cups['GoalsScored'], palette = 'viridis')
plt.xticks(rotation = 90)
```

Out[71]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19]),
 [Text(0, 0, '1930'),
  Text(1, 0, '1934'),
  Text(2, 0, '1938'),
  Text(3, 0, '1950'),
  Text(4, 0, '1954'),
  Text(5, 0, '1958'),
  Text(6, 0, '1962'),
  Text(7, 0, '1966'),
  Text(8, 0, '1970'),
  Text(9, 0, '1974'),
  Text(10, 0, '1978'),
  Text(11, 0, '1982'),
  Text(12, 0, '1986'),
  Text(13, 0, '1990'),
  Text(14, 0, '1994'),
  Text(15, 0, '1998'),
  Text(16, 0, '2002'),
  Text(17, 0, '2006'),
  Text(18, 0, '2010'),
  Text(19, 0, '2014')])
```

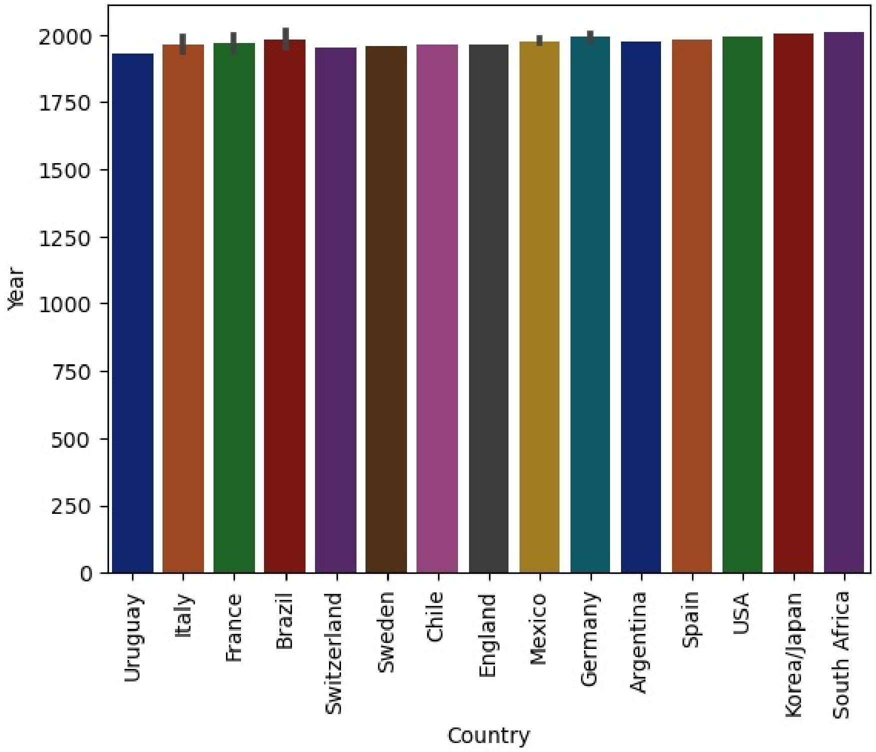


In [77]:

```
sns.barplot(x = cups['Country'], y = cups['Year'], palette = 'dark')
plt.xticks(rotation = 90)
```

Out[77]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 [Text(0, 0, 'Uruguay'),
  Text(1, 0, 'Italy'),
  Text(2, 0, 'France'),
  Text(3, 0, 'Brazil'),
  Text(4, 0, 'Switzerland'),
  Text(5, 0, 'Sweden'),
  Text(6, 0, 'Chile'),
  Text(7, 0, 'England'),
  Text(8, 0, 'Mexico'),
  Text(9, 0, 'Germany'),
  Text(10, 0, 'Argentina'),
  Text(11, 0, 'Spain'),
  Text(12, 0, 'USA'),
  Text(13, 0, 'Korea/Japan'),
  Text(14, 0, 'South Africa')])
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js