```
from google.colab import files
uploaded = files.upload()
import pandas as pd
df = pd.read_csv('World_Cup.csv')  # use your exact filename here
print(df.head())
print(df.info())
print(df.isnull().sum())
print(df.duplicated().sum())
df = df.dropna()
df = df.drop_duplicates()
print(df.describe())
print(df['Winner'].value_counts())
import matplotlib.pyplot as plt
import seaborn as sns
winners = df['Winner'].value_counts()
plt.figure(figsize=(10,6))
sns.barplot(x=winners.values, y=winners.index, palette='coolwarm')
plt.title('Most Successful FIFA World Cup Winners')
plt.xlabel('Number of Wins')
plt.ylabel('Country')
plt.show()
plt.figure(figsize=(10,6))
sns.lineplot(x='Year', y='GoalsScored', data=df, marker='o')
plt.title('Goals Scored in Each FIFA World Cup')
plt.xlabel('Year')
plt.ylabel('Total Goals Scored')
plt.show()
plt.figure(figsize=(10,6))
sns.lineplot(x='Year', y='QualifiedTeams', data=df, marker='o', color='green')
plt.title('Number of Teams Qualified per FIFA World Cup')
plt.xlabel('Year')
plt.ylabel('Qualified Teams')
plt.show()
plt.figure(figsize=(10,6))
sns.lineplot(x='Year', y='MatchesPlayed', data=df, marker='o', color='orange')
plt.title('Number of Matches Played per FIFA World Cup')
plt.xlabel('Year')
plt.ylabel('Matches Played')
plt.show()
```

Enable browser notifications in Settings to get alerts when executions complete

Choose Files World_Cup.csv

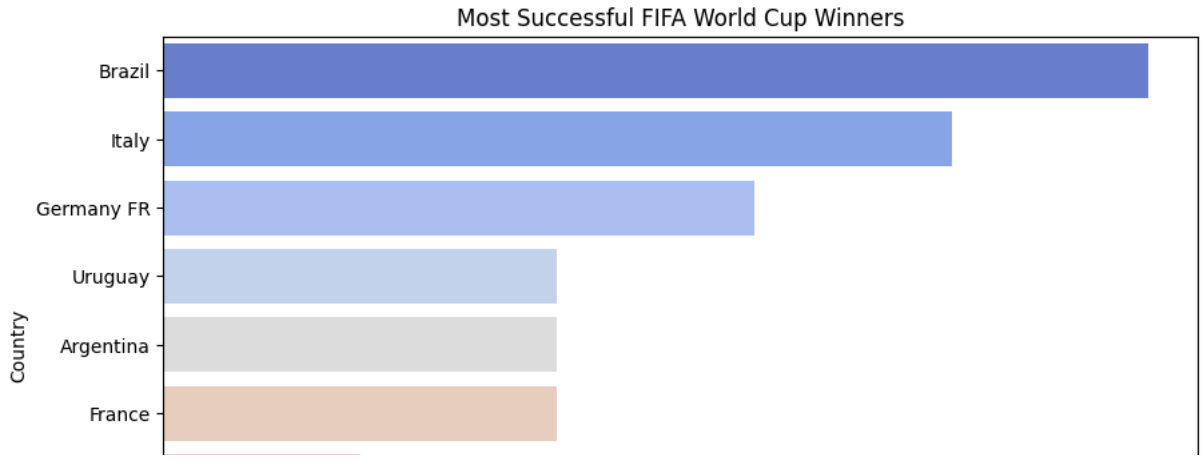
World_Cup.csv(text/csv) - 1288 bytes, last modified: 11/1/2025 - 100% done

```
Saving World_Cup.csv to World_Cup (2).csv
   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
0           70              13             18
1           70              16             17
2           84              15             18
3           88              13             22
4          140              16             26
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Year            21 non-null     int64
 1   Country         21 non-null     object
 2   Winner          21 non-null     object
 3   Runners-Up      21 non-null     object
 4   Third           21 non-null     object
 5   Fourth          21 non-null     object
 6   GoalsScored     21 non-null     int64
 7   QualifiedTeams  21 non-null     int64
 8   MatchesPlayed   21 non-null     int64
dtypes: int64(4), object(5)
memory usage: 1.6+ KB
None
Year              0
Country           0
Winner            0
Runners-Up        0
Third             0
Fourth            0
GoalsScored       0
QualifiedTeams    0
MatchesPlayed     0
dtype: int64
0
              Year  GoalsScored  QualifiedTeams  MatchesPlayed
count    21.000000    21.000000       21.000000      21.000000
mean   1976.857143   121.333333       21.761905      42.857143
std      26.657618    33.943090        7.462605      17.467930
min    1930.000000    70.000000       13.000000      17.000000
25%    1958.000000    89.000000       16.000000      32.000000
50%    1978.000000   126.000000       16.000000      38.000000
75%    1998.000000   146.000000       32.000000      64.000000
max    2018.000000   171.000000       32.000000      64.000000
Winner
Brazil        5
Italy         4
Germany FR    3
Uruguay       2
Argentina     2
France        2
England       1
Spain         1
Germany       1
Name: count, dtype: int64
/tmp/ipython-input-1406377660.py:17: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and

  sns.barplot(x=winners.values, y=winners.index, palette='coolwarm')
```



Enable browser notifications in Settings to get alerts when executions complete