World Cup, formally FIFA World Analysics

In [1]:
```
#import data by kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

In [2]:
```
!kaggle datasets download -d abecklas/fifa-world-cup
```

```
Warning: Your Kaggle API key is readable by other users on this sy
stem! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.jso
n'
Downloading fifa-world-cup.zip to /content
  0% 0.00/349k [00:00<?, ?B/s]
100% 349k/349k [00:00<00:00, 86.9MB/s]
```

In [3]:
```
#file unzip
import zipfile
zip_ref = zipfile.ZipFile('/content/fifa-world-cup.zip')
zip_ref.extractall('/content')
zip_ref.close()
```

In [4]:
```
#importing the Dependinces
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

In [5]:
```
#Read csv file pandas
data = pd.read_csv('/content/WorldCups.csv')
```

In [6]:
```
#check first five rows of the dataset
data.head()
```

Out[6]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	Qualified
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	
2	1938	France	Italy	Hungary	Brazil	Sweden	84	
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	
4	1954	Switzerland	Germany FR	Hungary	Austria	Uruguay	140	

In [7]:

```
#check last five rows of the dataset
data.tail()
```

Out[7]:

	Year	Country	Winner	Runners-Up	Third	Fourth	GoalsScored	Qualifi
15	1998	France	France	Brazil	Croatia	Netherlands	171	
16	2002	Korea/Japan	Brazil	Germany	Turkey	Korea Republic	161	
17	2006	Germany	Italy	France	Germany	Portugal	147	
18	2010	South Africa	Spain	Netherlands	Germany	Uruguay	145	
19	2014	Brazil	Germany	Argentina	Netherlands	Brazil	171	

In [8]:

```
#check shape of the dataset
data.shape
```

Out[8]:

```
(20, 10)
```

In [9]:

```
#check more infomation of the dataset
data.info()
```

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

In [10]:
```
#check mathamtic info
data.describe()
```

Out[10]:

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

In [11]:
```
#check corr relastion of the dataset
data.corr()
```

Out[11]:

	Year	GoalsScored	QualifiedTeams	MatchesPlayed
Year	1.000000	0.829886	0.895565	0.972473
GoalsScored	0.829886	1.000000	0.866201	0.876201
QualifiedTeams	0.895565	0.866201	1.000000	0.949164
MatchesPlayed	0.972473	0.876201	0.949164	1.000000

In [12]:
```
#check missing value of the dataset
data.isnull().sum()
```

Out[12]:
```
Year              0
Country           0
Winner            0
Runners-Up        0
Third             0
Fourth            0
GoalsScored       0
QualifiedTeams    0
MatchesPlayed     0
Attendance        0
dtype: int64
```

In [13]:
```
#check all columns
data.columns
```

Out[13]:
```
Index(['Year', 'Country', 'Winner', 'Runners-Up', 'Third', 'Fourth
',
       'GoalsScored', 'QualifiedTeams', 'MatchesPlayed', 'Attendan
ce'],
      dtype='object')
```

In [14]:
```
data['Attendance'].dtypes #Some problem  with this column. As you c

#There is a problem with this column , that's why the preprocessing
data['Attendance']= data["Attendance"].str.replace('.', '').astype(
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: Fu
tureWarning: The default value of regex will change from True to F
alse in a future version. In addition, single character regular ex
pressions will *not* be treated as literal strings when regex=True
.
  after removing the cwd from sys.path.
```

In [15]:
```
#Add the last worldcup row that wasn't in the csv, the last world c
list2018= [2018,'Russia','France','Croatia','Belgium','England',169
data.loc[len(data)] = list2018
```

In [16]:
```
#Add a column with data about the winner's continent

data['Winner_continent']= ['America','Europe','Europe','America','E
                           'America','Europe','America','Eu
```

In [17]:
```


#'Rename some columns'

data= data.rename(columns={'Country':'Country_host','Runners-Up':'S
```

In [18]:
```
#Add a column to inform if the host is in the best4
data['Host_best4'] = data[['Winner','Second','Third','Fourth']].eq(
```

In [19]:
```
#Turn the Germany FR to Germany
data = data.replace(['Germany FR'],'Germany')
```

In [20]:

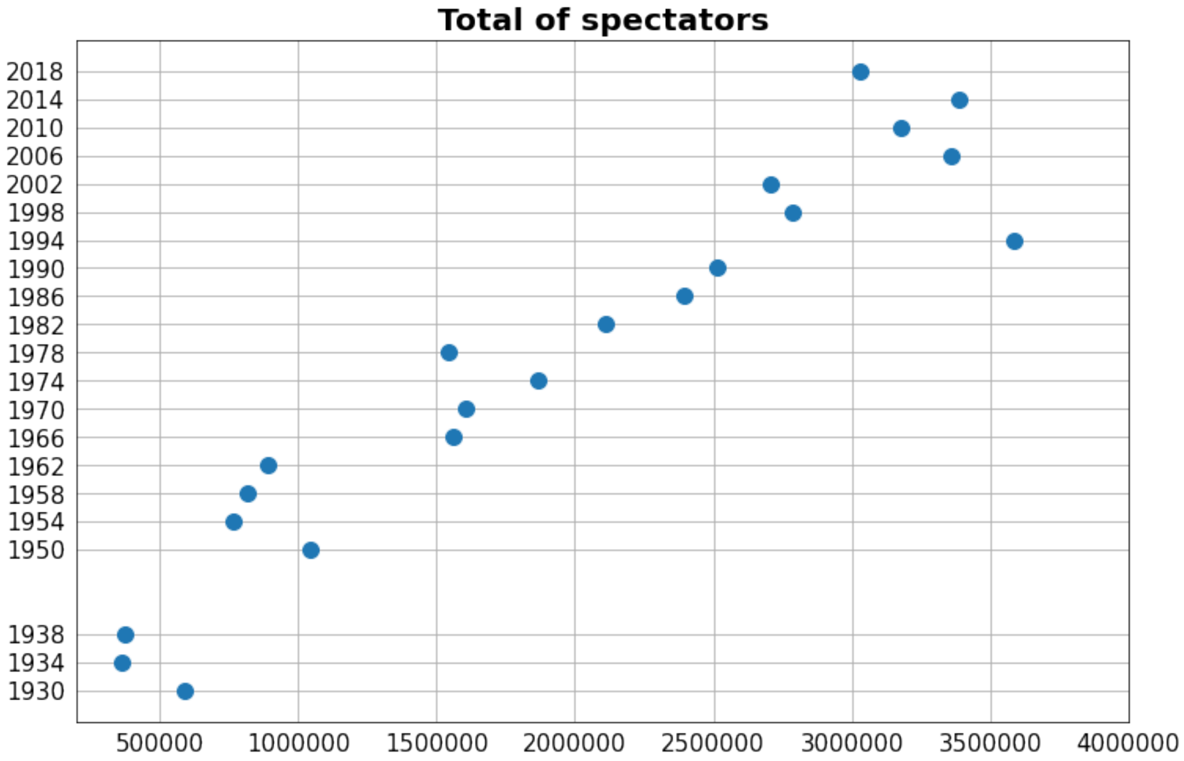
```
#Looking at how it looks like the DF
data.head()
```

Out[20]:

	Year	Country_host	Winner	Second	Third	Fourth	GoalsScored	Qualifie
0	1930	Uruguay	Uruguay	Argentina	USA	Yugoslavia	70	
1	1934	Italy	Italy	Czechoslovakia	Germany	Austria	70	
2	1938	France	Italy	Hungary	Brazil	Sweden	84	
3	1950	Brazil	Uruguay	Brazil	Sweden	Spain	88	
4	1954	Switzerland	Germany	Hungary	Austria	Uruguay	140	

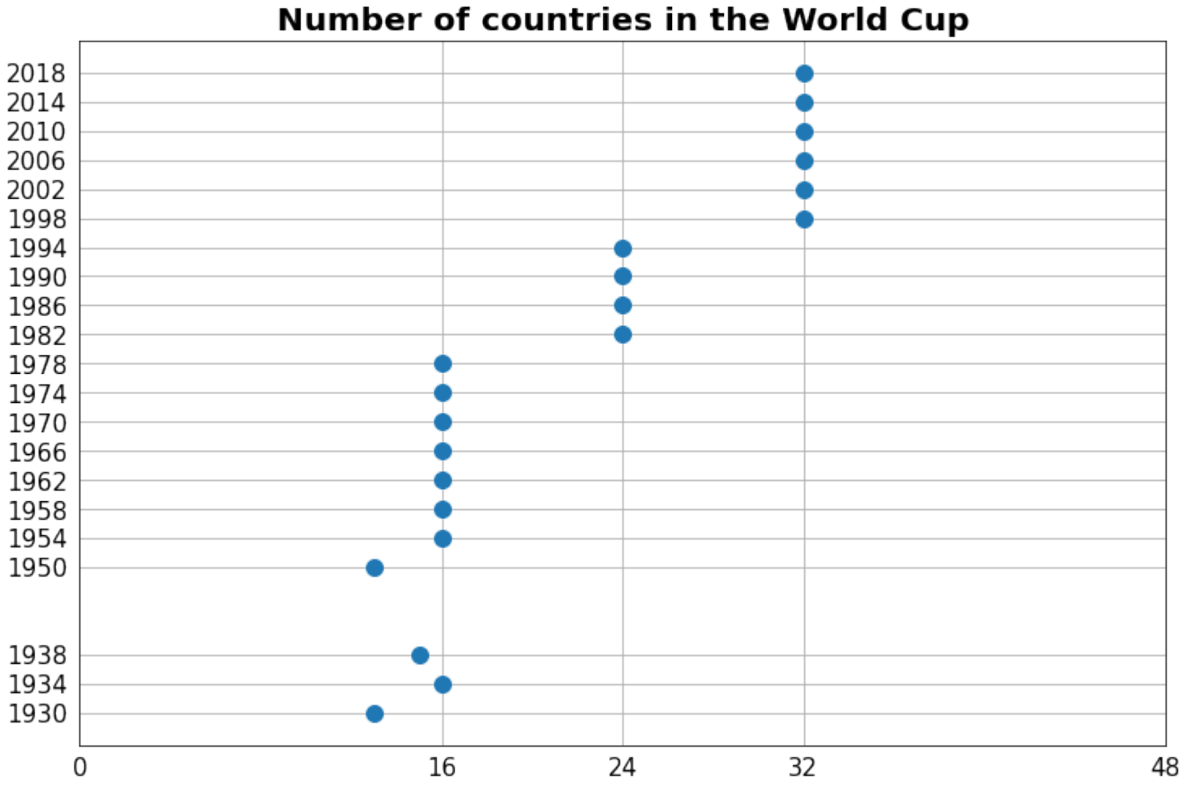
Data Visualization

```
In [21]: #There is a problem with this column , that's why the preprocessing
         #hist_worldcup['Attendance']= hist_worldcup["Attendance"].str.repla
         #hist_worldcup['Attendance']
         fig, ax= plt.subplots(figsize=(12,8))
         plt.title('Total of spectators',size=20,weight='bold')
         data.plot.scatter(x='Attendance',y='Year',ax=ax,zorder=2,s=100)
         #ax.spines[['right', 'top', 'left','bottom']].set_visible(False)
         ax.set_ylabel(None)
         ax.set_xlabel(None)
         ax.grid(visible=True)
         ax.tick_params(axis='both', which='major', labelsize=15)
         ax.set_yticks(data['Year'].tolist())
         ax.set_xticks([500000,1000000,1500000,2000000,2500000,3000000,35000
         ax.ticklabel_format(style='plain')

         plt.tick_params(bottom=False, left=False)
```



Number of countries in the World Cup through years

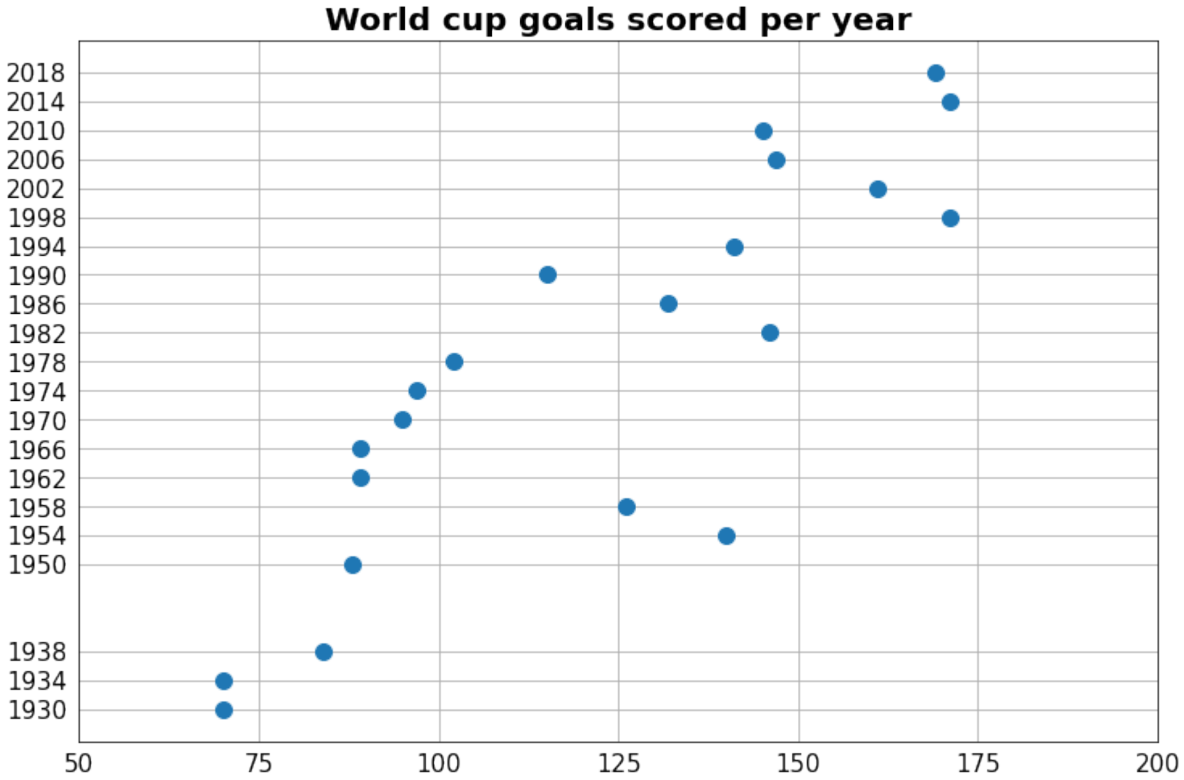
In [22]:
```
fig, ax= plt.subplots(figsize=(12,8))
plt.title('Number of countries in the World Cup',size=20,weight='bo
data.plot.scatter(x='QualifiedTeams',y='Year',ax=ax,zorder=2,s=100)
ax.set_ylabel(None)
ax.set_xlabel(None)
ax.grid(visible=True)
ax.tick_params(axis='both', which='major', labelsize=15)
ax.set_yticks(data['Year'].tolist())
ax.set_xticks([0,16,24 ,32,48])
plt.tick_params(bottom=False, left=False)
```



World Cup goals scored per year

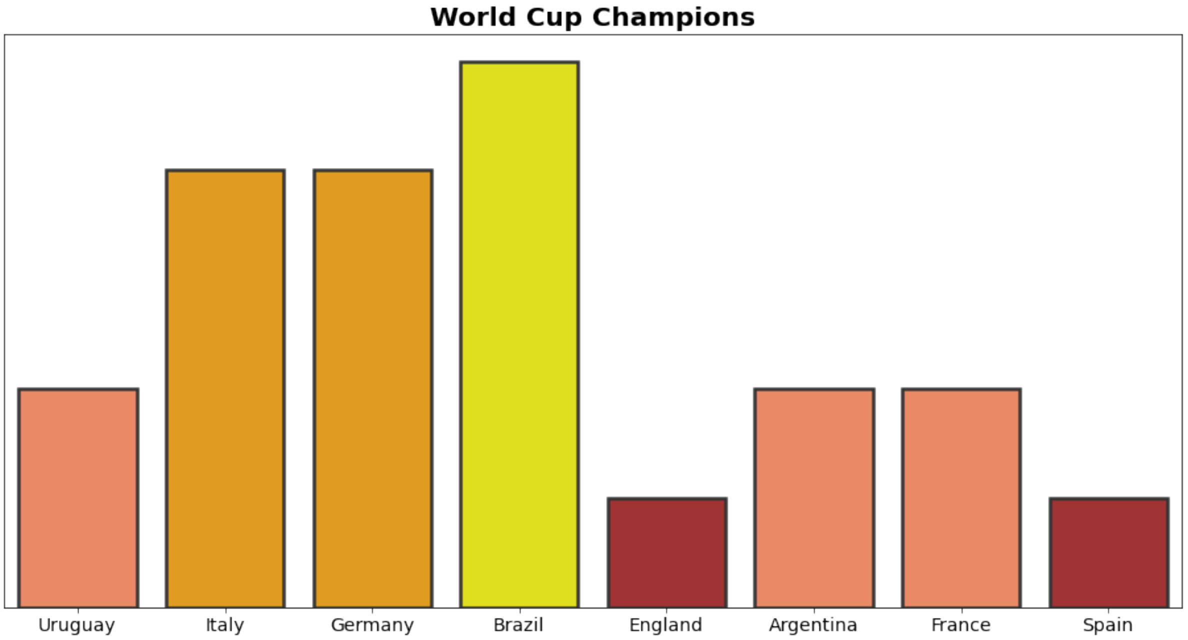
In [23]:

```
fig, ax= plt.subplots(figsize=(12,8))
plt.title('World cup goals scored per year',size=20,weight='bold')
data.plot.scatter(x='GoalsScored',y='Year',ax=ax,zorder=2,s=100)
ax.set_ylabel(None)
ax.set_xlabel(None)
ax.grid(visible=True)
ax.tick_params(axis='both', which='major', labelsize=15)
ax.set_yticks(data['Year'].tolist())
ax.set_xticks([50,75,100,125,150,175,200])
plt.tick_params(bottom=False, left=False)
```



World Cup Champions¶

In [24]:
```
palette=['coral','orange','orange','yellow','firebrick','coral','co
fig, ax= plt.subplots(figsize=(16,8))

plt.title('World Cup Champions',size=20,weight='bold')
sns.countplot(x = data['Winner'], palette=palette,linewidth=2.5, ed
ax.set_ylabel(None)
ax.set_xlabel(None)
plt.tick_params(labelleft=False, left=False,labelsize=14)
```



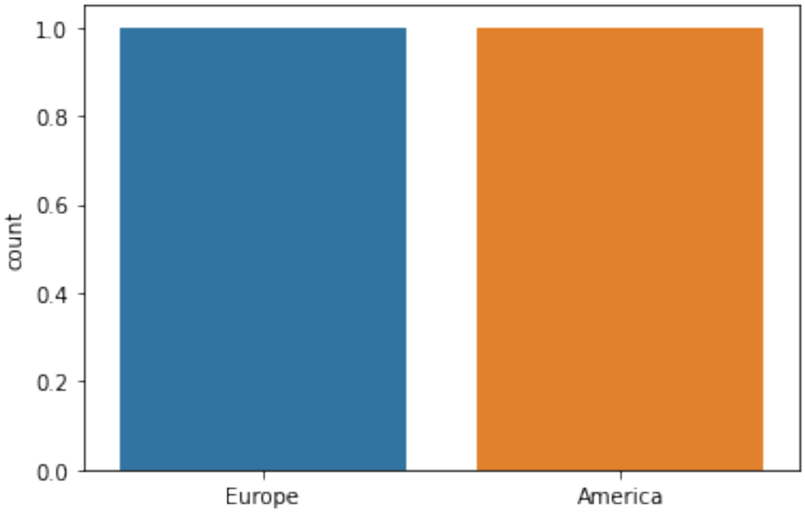
Which continent has got the most amount of World Cups?

In [25]:
```
index1 = data['Winner_continent'].value_counts().index.tolist()
#preprocessing for plotting a pie chart
value1 = data['Winner_continent'].value_counts().values.tolist()
```

```
In [26]: sns.countplot(data['Winner_continent'].value_counts().index.tolist(
```

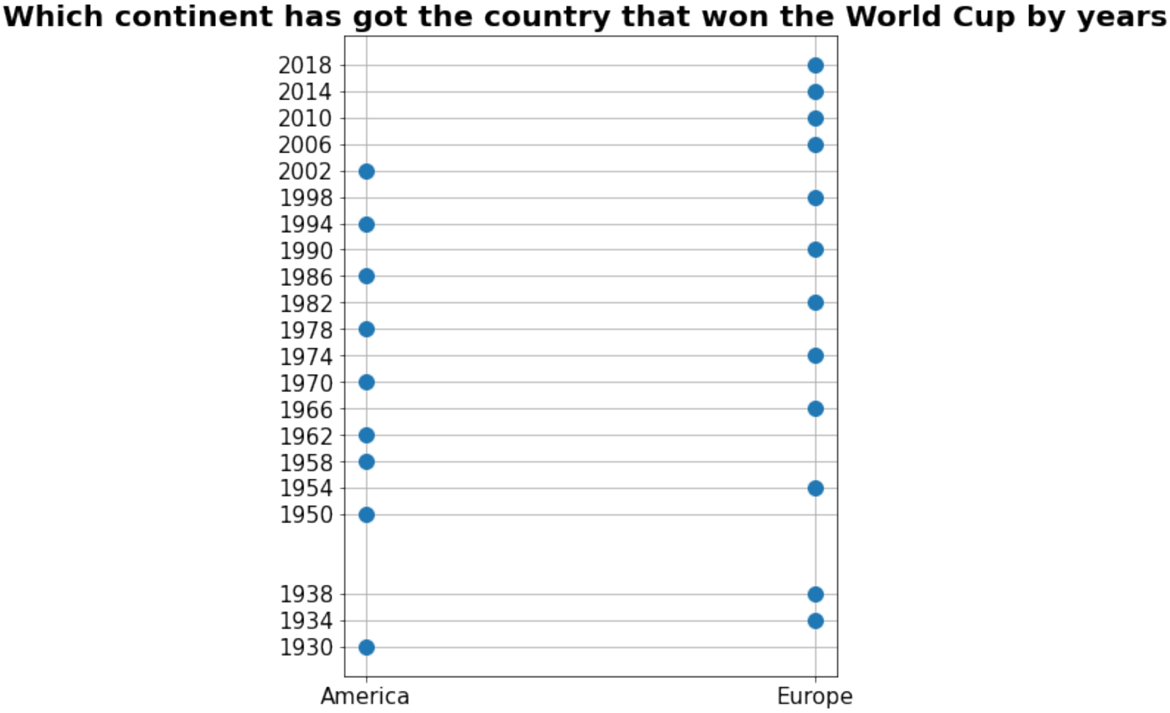
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f58f5d89250>
```



But European Countries has a bit more.

```
In [27]: fig, ax= plt.subplots(figsize=(6,8))
         plt.title('Which continent has got the country that won the World C
         data.plot.scatter(x='Winner_continent',y='Year',ax=ax,zorder=2,s=10
         ax.set_ylabel(None)
         ax.set_xlabel(None)
         ax.grid(visible=True)
         ax.tick_params(axis='both', which='major', labelsize=15)
         ax.set_yticks(data['Year'].tolist());
```



The last Champion was France...

In [28]:

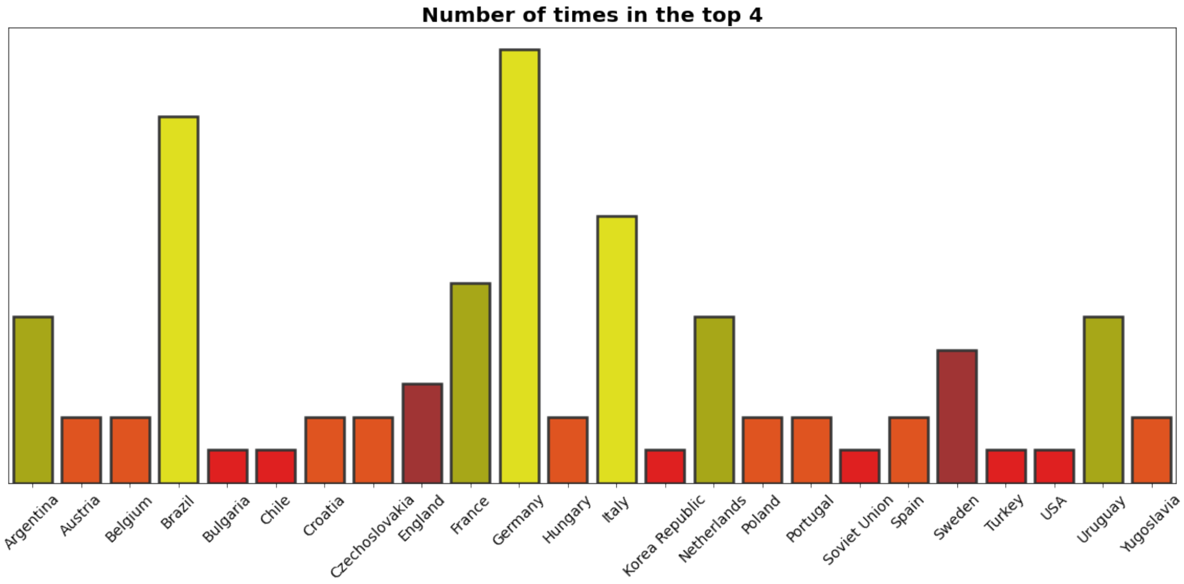
```
col=['Winner','Second','Third','Fourth'] #Preprocessing

countries = data[col].apply(pd.value_counts).reset_index().fillna(0
countries['Total'] = countries['Winner']+countries['Second']+countr
countries['Final'] = countries['Winner']+countries['Second']
countries
```

Out[28]:

	index	Winner	Second	Third	Fourth	Total	Final
0	Argentina	2.0	3.0	0.0	0.0	5.0	5.0
1	Austria	0.0	0.0	1.0	1.0	2.0	0.0
2	Belgium	0.0	0.0	1.0	1.0	2.0	0.0
3	Brazil	5.0	2.0	2.0	2.0	11.0	7.0
4	Bulgaria	0.0	0.0	0.0	1.0	1.0	0.0
5	Chile	0.0	0.0	1.0	0.0	1.0	0.0
6	Croatia	0.0	1.0	1.0	0.0	2.0	1.0
7	Czechoslovakia	0.0	2.0	0.0	0.0	2.0	2.0
8	England	1.0	0.0	0.0	2.0	3.0	1.0
9	France	2.0	1.0	2.0	1.0	6.0	3.0
10	Germany	4.0	4.0	4.0	1.0	13.0	8.0
11	Hungary	0.0	2.0	0.0	0.0	2.0	2.0
12	Italy	4.0	2.0	1.0	1.0	8.0	6.0
13	Korea Republic	0.0	0.0	0.0	1.0	1.0	0.0
14	Netherlands	0.0	3.0	1.0	1.0	5.0	3.0
15	Poland	0.0	0.0	2.0	0.0	2.0	0.0
16	Portugal	0.0	0.0	1.0	1.0	2.0	0.0
17	Soviet Union	0.0	0.0	0.0	1.0	1.0	0.0
18	Spain	1.0	0.0	0.0	1.0	2.0	1.0
19	Sweden	0.0	1.0	2.0	1.0	4.0	1.0
20	Turkey	0.0	0.0	1.0	0.0	1.0	0.0
21	USA	0.0	0.0	1.0	0.0	1.0	0.0
22	Uruguay	2.0	0.0	0.0	3.0	5.0	2.0
23	Yugoslavia	0.0	0.0	0.0	2.0	2.0	0.0

```
In [29]: #Set the Palette
         clrs= ['yellow' if (i>=8) else 'y' if (5<=i<8) else 'firebrick' if

         fig, ax= plt.subplots(figsize=(20,8))
         plt.title('Number of times in the top 4',size=20,weight='bold')
         sns.barplot(data=countries,x='index',y='Total',palette=clrs,linewid
         ax.set_ylabel(None)
         ax.set_xlabel(None)
         plt.tick_params(labelleft=False, left=False,labelsize=14)


         plt.xticks(rotation=45)
```

```
Out[29]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14
         , 15, 16,
                 17, 18, 19, 20, 21, 22, 23]),
          <a list of 24 Text major ticklabel objects>)
```



Number of times in the final

In [30]:

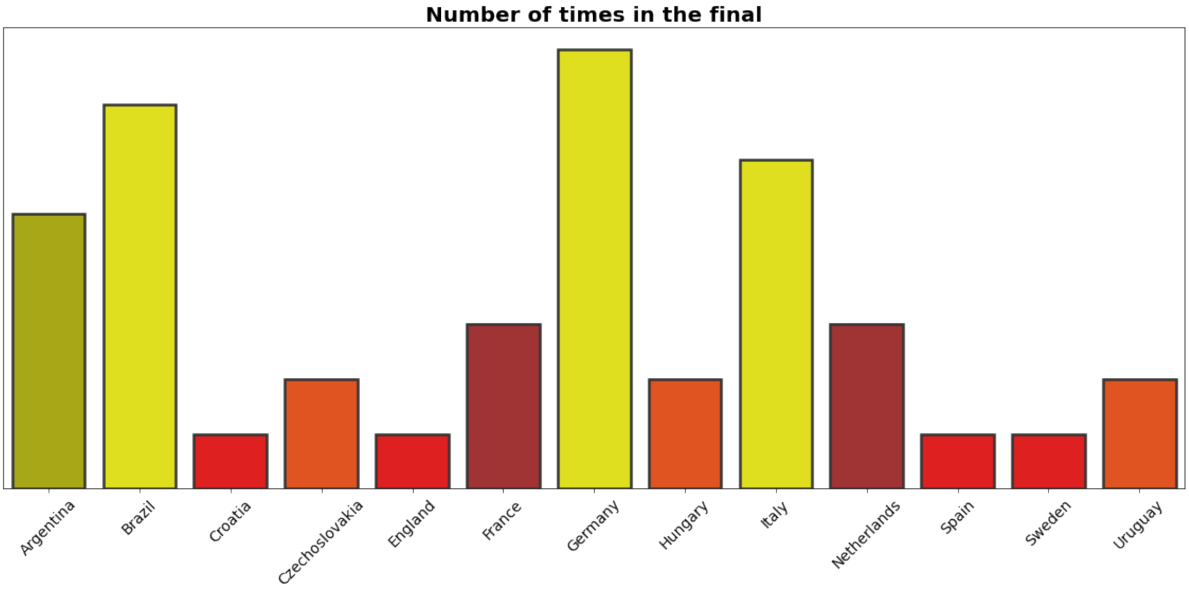
```
finalist = countries.drop(countries[(countries['Winner']==0) & (cou
finalist
```

Out[30]:

	index	Winner	Second	Third	Fourth	Total	Final
0	Argentina	2.0	3.0	0.0	0.0	5.0	5.0
3	Brazil	5.0	2.0	2.0	2.0	11.0	7.0
6	Croatia	0.0	1.0	1.0	0.0	2.0	1.0
7	Czechoslovakia	0.0	2.0	0.0	0.0	2.0	2.0
8	England	1.0	0.0	0.0	2.0	3.0	1.0
9	France	2.0	1.0	2.0	1.0	6.0	3.0
10	Germany	4.0	4.0	4.0	1.0	13.0	8.0
11	Hungary	0.0	2.0	0.0	0.0	2.0	2.0
12	Italy	4.0	2.0	1.0	1.0	8.0	6.0
14	Netherlands	0.0	3.0	1.0	1.0	5.0	3.0
18	Spain	1.0	0.0	0.0	1.0	2.0	1.0
19	Sweden	0.0	1.0	2.0	1.0	4.0	1.0
22	Uruguay	2.0	0.0	0.0	3.0	5.0	2.0

In [31]:
```
#Set the color
clrs= ['yellow' if (i>=6) else 'y' if (i==5) else 'firebrick' if (3


fig, ax= plt.subplots(figsize=(20,8))
plt.title('Number of times in the final',size=20,weight='bold')
sns.barplot(data=finalist,x='index',y='Final',palette=clrs,linewidt
ax.set_ylabel(None)
ax.set_xlabel(None)
plt.tick_params(labelleft=False, left=False,labelsize=14)


plt.xticks(rotation=45)
```

Out[31]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 <a list of 13 Text major ticklabel objects>)
```

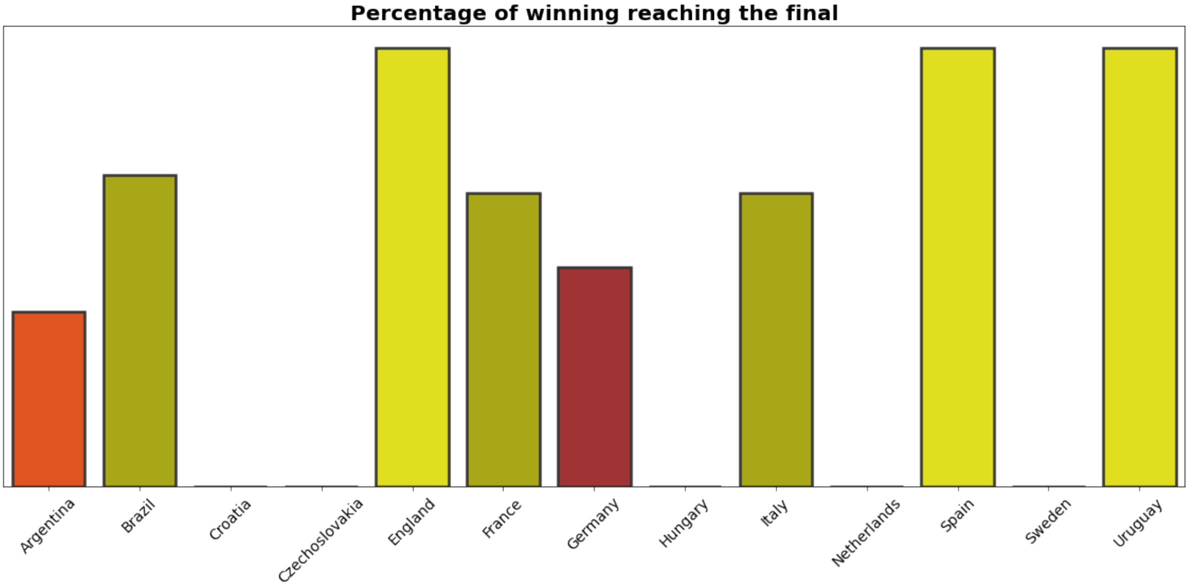


Looking at the relationship between being champion and reaching the final

In [32]:
```
finalist['rel_final'] = finalist['Winner']/finalist['Final'] #prepr
relationship= np.round(finalist[(finalist['Second']>0) | (finalist[
```

In [33]:
```
#Set the color
clrs= ['yellow' if (i==1) else 'y' if (0.5<i<1) else 'firebrick' if


fig, ax= plt.subplots(figsize=(20,8))
plt.title('Percentage of winning reaching the final',size=20,weight
sns.barplot(data=relationship,x='index',y='rel_final',palette=clrs,
ax.set_ylabel(None)
ax.set_xlabel(None)
plt.tick_params(labelleft=False, left=False,labelsize=14)


plt.xticks(rotation=45)
```

Out[33]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 <a list of 13 Text major ticklabel objects>)
```



How many times each country reach at least the first 4 position

In [34]:

```
transpose=countries.T.rename(columns=countries.T.iloc[0]).drop(inde

transpose =transpose.reset_index()[0:4]
transpose
```

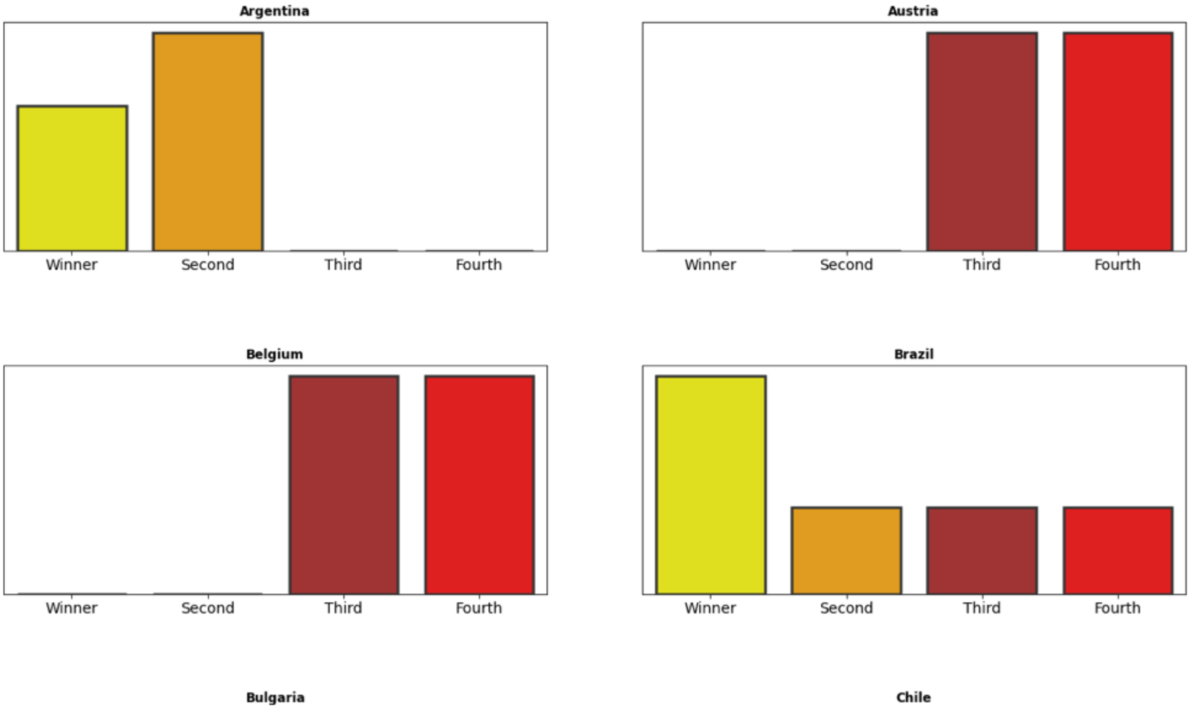
Out[34]:

	index	Argentina	Austria	Belgium	Brazil	Bulgaria	Chile	Croatia	Czechoslovakia	E
0	Winner	2.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	
1	Second	3.0	0.0	0.0	2.0	0.0	0.0	1.0	2.0	
2	Third	0.0	1.0	1.0	2.0	0.0	1.0	1.0	0.0	
3	Fourth	0.0	1.0	1.0	2.0	1.0	0.0	0.0	0.0	

4 rows × 25 columns

In [35]:

```
columns= transpose.columns[1:]
clr= ['yellow','orange','firebrick','red']
fig, axes = plt.subplots(12,2, figsize=(20,60))

fig.subplots_adjust(hspace=.5,top =1, wspace=.175)

for ax, col in zip(axes.flat,columns):
    sns.barplot(data=transpose, x='index',y=col,ax=ax,palette=clr,l
    ax.set_ylabel(None)
    ax.set_xlabel(None)
    ax.tick_params(labelleft=False, left=False,labelsize=14)
    ax.set_title(col,fontweight="bold")
```



Extra Analysis

In [36]:

```
#load matches dataset in pandas dataframe
matches = pd.read_csv('/content/WorldCupMatches.csv')
```

In [37]:

```
#check first five rows of the dataset
matches.head()
```

Out[37]:

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	condi
0	1930.0	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico	
1	1930.0	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium	
2	1930.0	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil	
3	1930.0	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru	
4	1930.0	15 Jul 1930 - 16:00	Group 1	Parque Central	Montevideo	Argentina	1.0	0.0	France	

In [38]:

```
#check last five rows of the dataset
matches.tail()
```

Out[38]:

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	Atte
4567	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4568	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4569	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4570	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4571	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [39]:
```
#check shape of the dataset
matches.shape
```

Out[39]:
```
(4572, 20)
```

In [40]:
```
#checl all columns
matches.columns
```

Out[40]:
```
Index(['Year', 'Datetime', 'Stage', 'Stadium', 'City', 'Home Team
Name',
       'Home Team Goals', 'Away Team Goals', 'Away Team Name',
       'Win conditions', 'Attendance', 'Half-time Home Goals',
       'Half-time Away Goals', 'Referee', 'Assistant 1', 'Assistan
t 2',
       'RoundID', 'MatchID', 'Home Team Initials', 'Away Team Init
ials'],
      dtype='object')
```

In [41]:
```
#check more infomation of the dataset
matches.info()
```

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

In [42]:
```
#chekc mathamtic
matches.describe()
```

Out[42]:

	Year	Home Team Goals	Away Team Goals	Attendance	Half-time Home Goals	Half-time Away Goals	R
count	852.000000	852.000000	852.000000	850.000000	852.000000	852.000000	8.5200
mean	1985.089202	1.811033	1.022300	45164.800000	0.708920	0.428404	1.0661
std	22.448825	1.610255	1.087573	23485.249247	0.937414	0.691252	2.7296
min	1930.000000	0.000000	0.000000	2000.000000	0.000000	0.000000	2.0100
25%	1970.000000	1.000000	0.000000	30000.000000	0.000000	0.000000	2.6200
50%	1990.000000	2.000000	1.000000	41579.500000	0.000000	0.000000	3.3700
75%	2002.000000	3.000000	2.000000	61374.500000	1.000000	1.000000	2.4972
max	2014.000000	10.000000	7.000000	173850.000000	6.000000	5.000000	9.7410

In [43]:
```
#check corr realtion of the dataset
matches.corr()
```

Out[43]:

	Year	Home Team Goals	Away Team Goals	Attendance	Half-time Home Goals	Half-time Away Goals	RoundID	
Year	1.000000	-0.381332	0.075339	0.314698	-0.288909	0.020934	0.343106	
Home Team Goals	-0.381332	1.000000	0.012474	-0.117751	0.729536	-0.009530	-0.110075	-
Away Team Goals	0.075339	0.012474	1.000000	-0.029801	-0.006304	0.693780	-0.005345	
Attendance	0.314698	-0.117751	-0.029801	1.000000	-0.126756	-0.037136	0.069394	
Half-time Home Goals	-0.288909	0.729536	-0.006304	-0.126756	1.000000	0.022204	-0.055303	-
Half-time Away Goals	0.020934	-0.009530	0.693780	-0.037136	0.022204	1.000000	0.011980	
RoundID	0.343106	-0.110075	-0.005345	0.069394	-0.055303	0.011980	1.000000	
MatchID	0.636591	-0.196100	0.082687	0.164686	-0.166201	0.059456	0.071549	

In [44]:
```
#check missing value of the data
matches.isnull().sum()
```

Out[44]:
```
Year                    3720
Datetime                3720
Stage                   3720
Stadium                 3720
City                    3720
Home Team Name          3720
Home Team Goals         3720
Away Team Goals         3720
Away Team Name          3720
Win conditions          3720
Attendance              3722
Half-time Home Goals    3720
Half-time Away Goals    3720
Referee                 3720
Assistant 1             3720
Assistant 2             3720
RoundID                 3720
MatchID                 3720
Home Team Initials      3720
Away Team Initials      3720
dtype: int64
```

In [45]:
```
#Drop Rows with all null values
matches = matches.dropna(how='all')
```

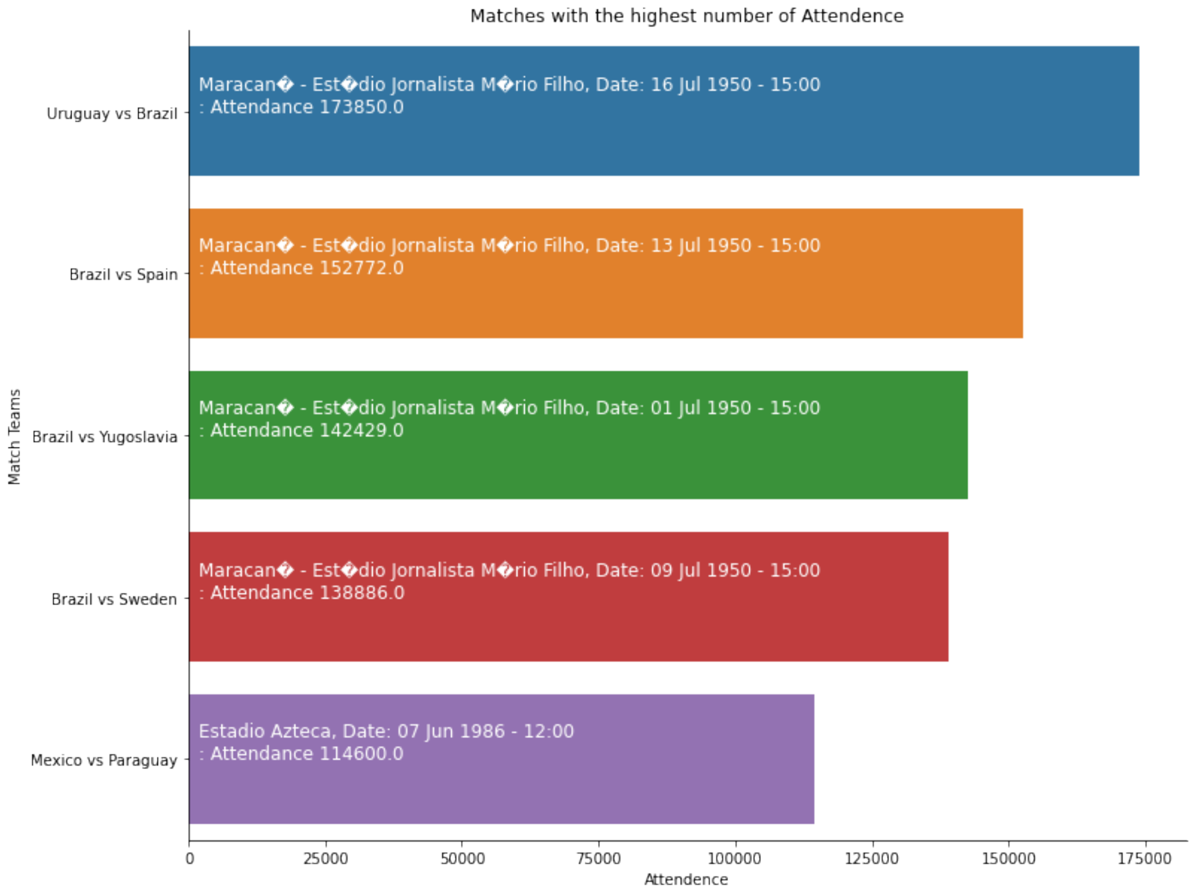
In [46]:
```
matches['Home Team Goals']= matches['Home Team Goals'].astype(int)
matches['Away Team Goals']= matches['Away Team Goals'].astype(int)

matches['result'] = matches['Home Team Goals'].astype(str)+"-"+matc
matches['result']
```

Out[46]:
```
0      4-1
1      3-0
2      2-1
3      3-1
4      1-0
      ...
847    0-0
848    1-7
849    0-0
850    0-3
851    1-0
Name: result, Length: 852, dtype: object
```

Matches with the highest number of Attendance

In [47]:

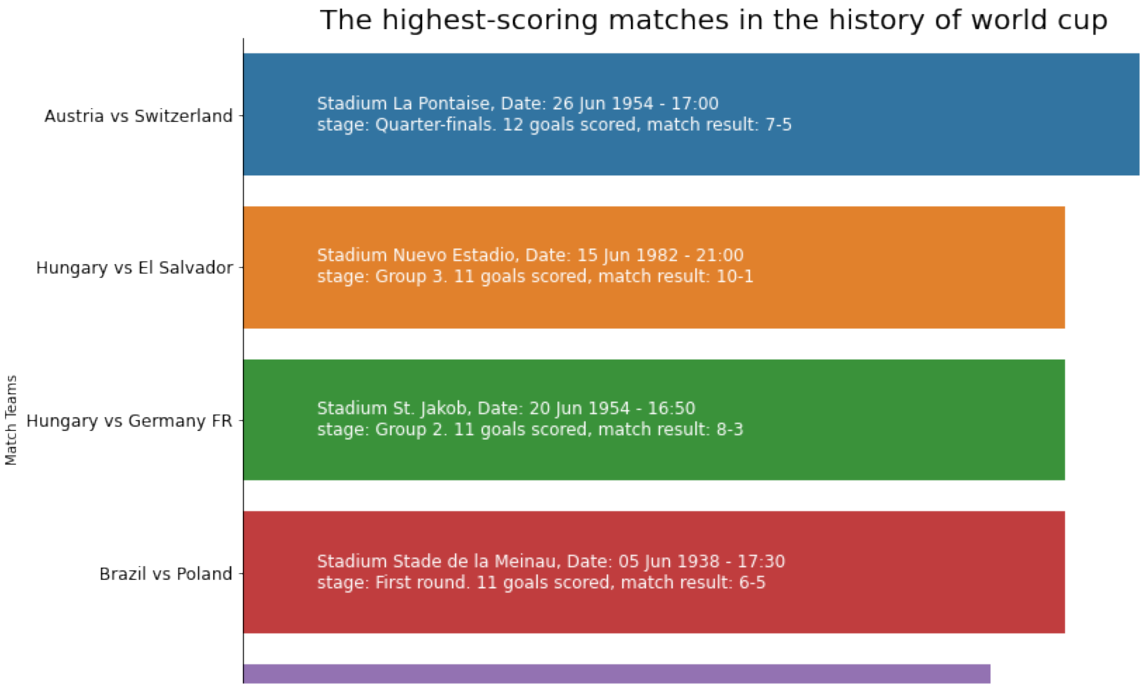
```
top5_attendance = matches.sort_values(by='Attendance',ascending=Fal
top5_attendance

top5_attendance['vs'] = top5_attendance['Home Team Name'] + " vs "

top5_attendance['attend']= top5_attendance['Attendance'].astype(str

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

ax = sns.barplot(y = top5_attendance['vs'], x = top5_attendance['At
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(top5_attendance['Stadium'] +", Date: " + top5
    ax.text(2000, i, s, fontsize = 12, color = 'white')
plt.show()
```

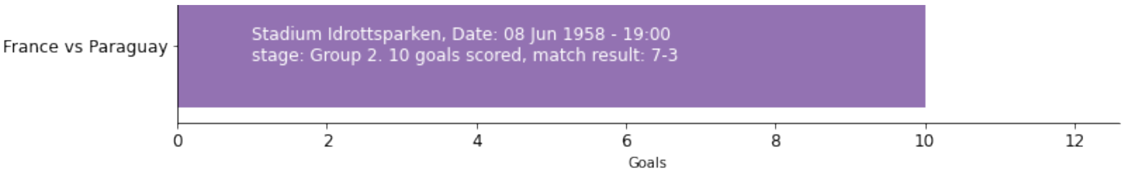


The Highest-Scoring matches in the World Cup

In [48]:

```
matches['total_goals'] = matches['Home Team Goals']+ matches['Away 
matches['vs'] = matches['Home Team Name'] + " vs "+ matches['Away T

top5_goals=matches.sort_values(by='total_goals',ascending=False)[:5

top5_goals['vs'] = top5_goals['Home Team Name'] + " vs " + top5_goa

top5_goals['total_goals_str']= top5_goals['total_goals'].astype(str


top5_goals['Home Team Goals'] = top5_goals['Home Team Goals'].astyp
top5_goals['Away Team Goals'] = top5_goals['Away Team Goals'].astyp


top5_goals['result'] = top5_goals['Home Team Goals'].astype(str)+"-


plt.figure(figsize = (12,10))
ax = sns.barplot(y = top5_goals['vs'], x = top5_goals['total_goals'
sns.despine(right = True)
plt.ylabel('Match Teams')
plt.xlabel('Goals')
plt.yticks(size=12)
plt.xticks(size=12)
plt.title('The highest-scoring matches in the history of world cup'


for i, s in enumerate("Stadium "+top5_goals['Stadium'] +", Date: "
                      top5_goals['total_goals_str']+ ", match resul
    ax.text(1, i ,s,fontsize = 12, color = 'white',va = 'center')



plt.show()
```





Highest difference of goals in a World Cup

In [49]:

```
matches['difference_goals'] = pd.Series.abs(matches['Home Team Goal
top5_difference=matches.sort_values(by='difference_goals',ascending
top5_difference
```

Out[49]:

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	N
80	1954.0	17 Jun 1954 - 18:00	Group 2	Hardturm	Zurich	Hungary	9	0	Rep
243	1974.0	18 Jun 1974 - 19:30	Group 2	Parkstadion	Gelsenkirchen	Yugoslavia	9	0	
312	1982.0	15 Jun 1982 - 21:00	Group 3	Nuevo Estadio	Elche	Hungary	10	1	Sal
66	1950.0	02 Jul 1950 - 15:00	Group 4	Independencia	Belo Horizonte	Uruguay	8	0	B
46	1938.0	12 Jun 1938 - 17:00	Quarter-finals	Fort Carree	Antibes	Sweden	8	0	

5 rows × 24 columns

In [50]:

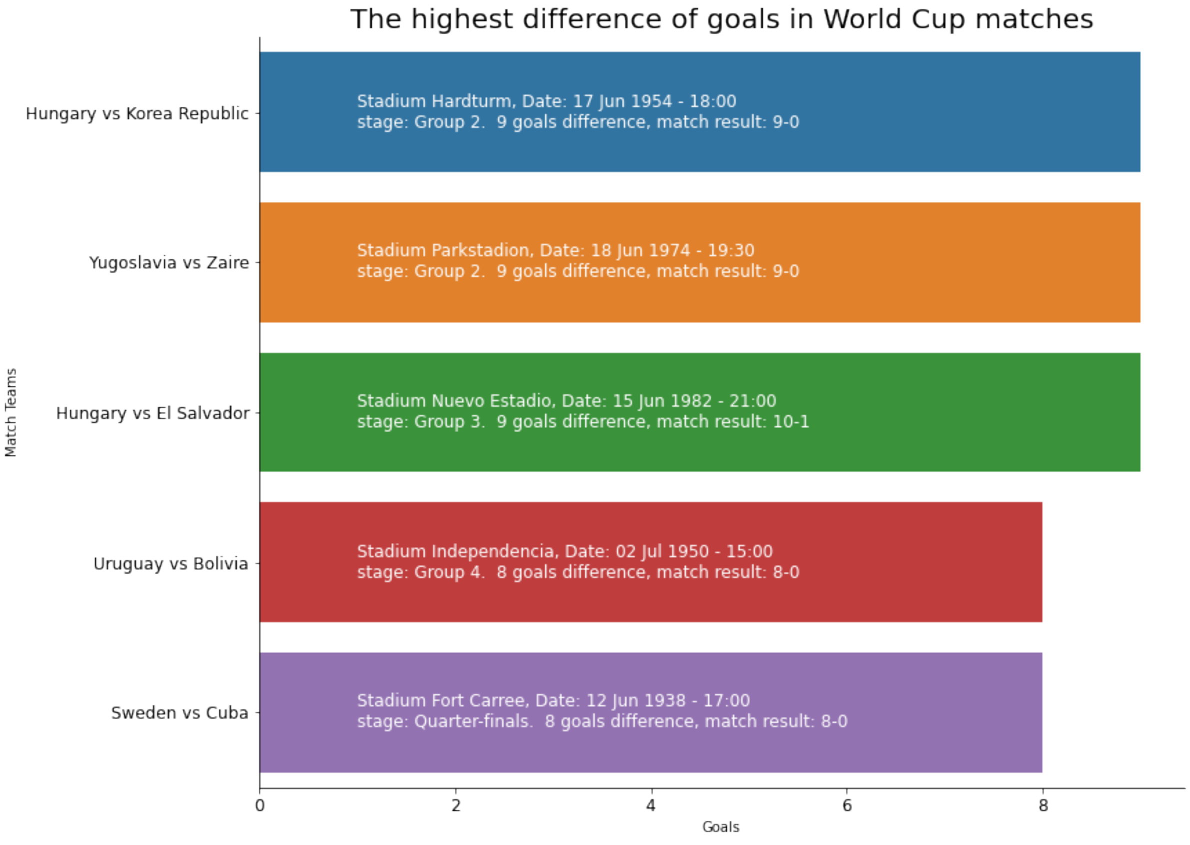
```
top5_difference['result']
```

Out[50]:

```
80      9-0
243     9-0
312    10-1
66      8-0
46      8-0
Name: result, dtype: object
```

In [51]:

```
top5_difference['difference_goals']=top5_difference['difference_goa


top5_difference['difference_goals_str']= top5_difference['differenc




top5_difference['result'] = top5_difference['Home Team Goals'].asty


plt.figure(figsize = (12,10))
ax = sns.barplot(y = top5_difference['vs'], x = top5_difference['di
sns.despine(right = True)
plt.ylabel('Match Teams')
plt.xlabel('Goals')
plt.yticks(size=12)
plt.xticks(size=12)
plt.title('The highest difference of goals in World Cup matches',si


for i, s in enumerate("Stadium "+top5_difference['Stadium'] +", Dat
                      top5_difference['difference_goals_str']+ ", m
    ax.text(1, i ,s,fontsize = 12, color = 'white',va = 'center')




plt.show()
```



Highest Scoring countries

In [52]:
```
matches = matches.replace(['Germany FR'],'Germany') #The same as th
```

In [53]:
```
list_countries =matches['Home Team Name'].unique().tolist()
```

In [54]:
```
lista_home=[]
lista_away=[]
for i in list_countries:

    goals_home = matches.loc[matches['Home Team Name'] == i, 'Home 
    lista_home.append(goals_home)
    goals_away = matches.loc[matches['Away Team Name']== i, 'Away T
    lista_away.append(goals_away)
```

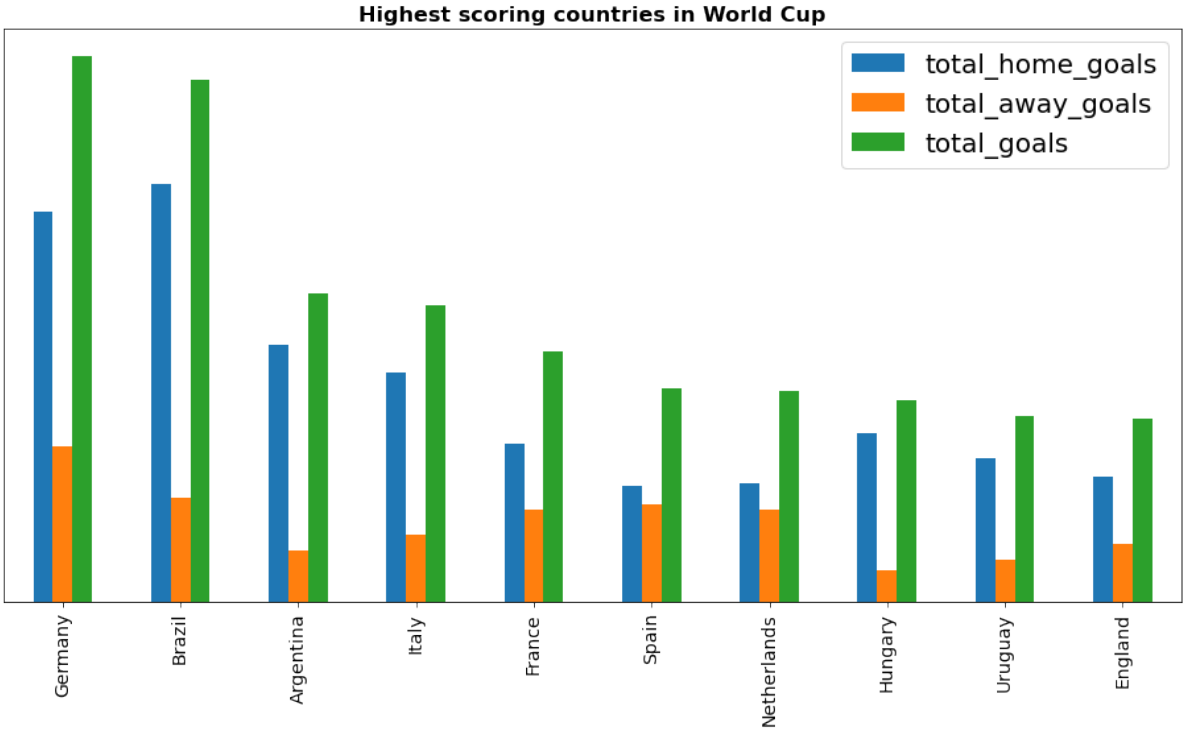
In [55]:
```
df = pd.DataFrame({'country': list_countries,'total_home_goals':lis
df['total_goals'] =df['total_home_goals']+df['total_away_goals']
most_goals=df.sort_values(by='total_goals',ascending=False)[:10]
most_goals
```

Out[55]:

	country	total_home_goals	total_away_goals	total_goals
13	Germany	168	67	235
7	Brazil	180	45	225
4	Argentina	111	22	133
15	Italy	99	29	128
0	France	68	40	108
14	Spain	50	42	92
34	Netherlands	51	40	91
10	Hungary	73	14	87
6	Uruguay	62	18	80
18	England	54	25	79

In [56]:

```
fig, ax= plt.subplots(figsize=(16,8))

plt.title('Highest scoring countries in World Cup',size=16,weight='
most_goals.plot(x="country", y=["total_home_goals", "total_away_goa

#ax.spines[['right', 'top', 'left']].set_visible(False)
ax.set_ylabel(None)
ax.set_xlabel(None)
ax.tick_params(labelleft=False, left=False,labelsize=14)
ax.legend(fontsize=20)


fig.show();
```



Total Goal Conceded of finalist teams

In [57]:
```
matches['Home Team Name'].value_counts()
```

Out[57]:
```
Brazil                        82
Germany                       77
Italy                         57
Argentina                     54
England                       35
                              ..
Wales                          1
Norway                         1
rn">United Arab Emirates       1
Haiti                          1
rn">Bosnia and Herzegovina     1
Name: Home Team Name, Length: 77, dtype: int64
```

In [58]:
```
finalist
```

Out[58]:

	index	Winner	Second	Third	Fourth	Total	Final	rel_final
0	Argentina	2.0	3.0	0.0	0.0	5.0	5.0	0.400000
3	Brazil	5.0	2.0	2.0	2.0	11.0	7.0	0.714286
6	Croatia	0.0	1.0	1.0	0.0	2.0	1.0	0.000000
7	Czechoslovakia	0.0	2.0	0.0	0.0	2.0	2.0	0.000000
8	England	1.0	0.0	0.0	2.0	3.0	1.0	1.000000
9	France	2.0	1.0	2.0	1.0	6.0	3.0	0.666667
10	Germany	4.0	4.0	4.0	1.0	13.0	8.0	0.500000
11	Hungary	0.0	2.0	0.0	0.0	2.0	2.0	0.000000
12	Italy	4.0	2.0	1.0	1.0	8.0	6.0	0.666667
14	Netherlands	0.0	3.0	1.0	1.0	5.0	3.0	0.000000
18	Spain	1.0	0.0	0.0	1.0	2.0	1.0	1.000000
19	Sweden	0.0	1.0	2.0	1.0	4.0	1.0	0.000000
22	Uruguay	2.0	0.0	0.0	3.0	5.0	2.0	1.000000

In [59]:

```
#Looking just the countries that have reached finals, that seem to 
finalista =finalist['index'].tolist()

goalsconceded_home=[]
goalsconceded_away=[]
match1=[]
match2=[]
for i in finalista:

    goalsconc_home = matches.loc[matches['Home Team Name'] == i, 'A
    goalsconceded_home.append(goalsconc_home)
    goalsconc_away = matches.loc[matches['Away Team Name']== i, 'Ho
    goalsconceded_away.append(goalsconc_away)
    counted1 =(matches['Home Team Name']== i).sum()
    counted2 =(matches['Away Team Name']== i).sum()

    match1.append(int(counted1))
    match2.append(int(counted2))
```

In [60]:

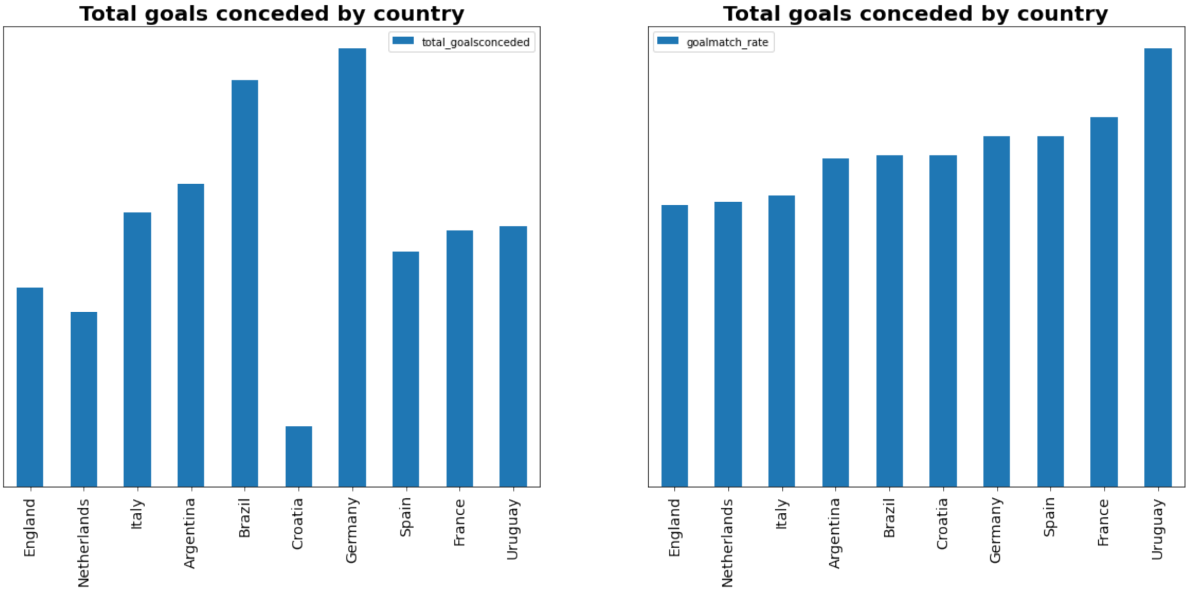
```
#team with fewest goals conceded

df = pd.DataFrame({'country': finalista,'goalsconceded_home':goalsc
                   'matches_home':match1,'matches_away':match2})
df['total_matches'] = df['matches_home']+ df['matches_away']
df['total_goalsconceded'] =df['goalsconceded_home']+df['goalsconced
df['goalmatch_rate'] = (df['total_goalsconceded'] / df['total_match
goals_conceded=df.sort_values(by='goalmatch_rate')[:10]
goals_conceded
```

Out[60]:

	country	goalsconceded_home	goalsconceded_away	matches_home	matches_away
4	England	20	36	35	27
9	Netherlands	21	28	32	22
8	Italy	41	36	57	26
0	Argentina	44	41	54	27
1	Brazil	78	36	82	26
2	Croatia	6	11	3	13
6	Germany	68	55	77	33
10	Spain	30	36	30	29
5	France	31	41	31	30
12	Uruguay	29	44	28	24

In [61]:

```
fig, ax= plt.subplots(nrows=1,ncols=2,figsize=(20,8))

plt.title('Relationship between goals conceded and matches played i
goals_conceded.plot(x="country", y="total_goalsconceded", kind="bar

ax[0].set_title('Total goals conceded by country',size=20,weight='b
ax[0].set_ylabel(None)
ax[0].set_xlabel(None)
ax[0].tick_params(labelleft=False, left=False,labelsize=14)


goals_conceded.plot(x="country", y="goalmatch_rate", kind="bar",ax=

ax[1].set_title('Total goals conceded by country',size=20,weight='b
ax[1].set_ylabel(None)
ax[1].set_xlabel(None)
ax[1].tick_params(labelleft=False, left=False,labelsize=14)
```



In [61]: