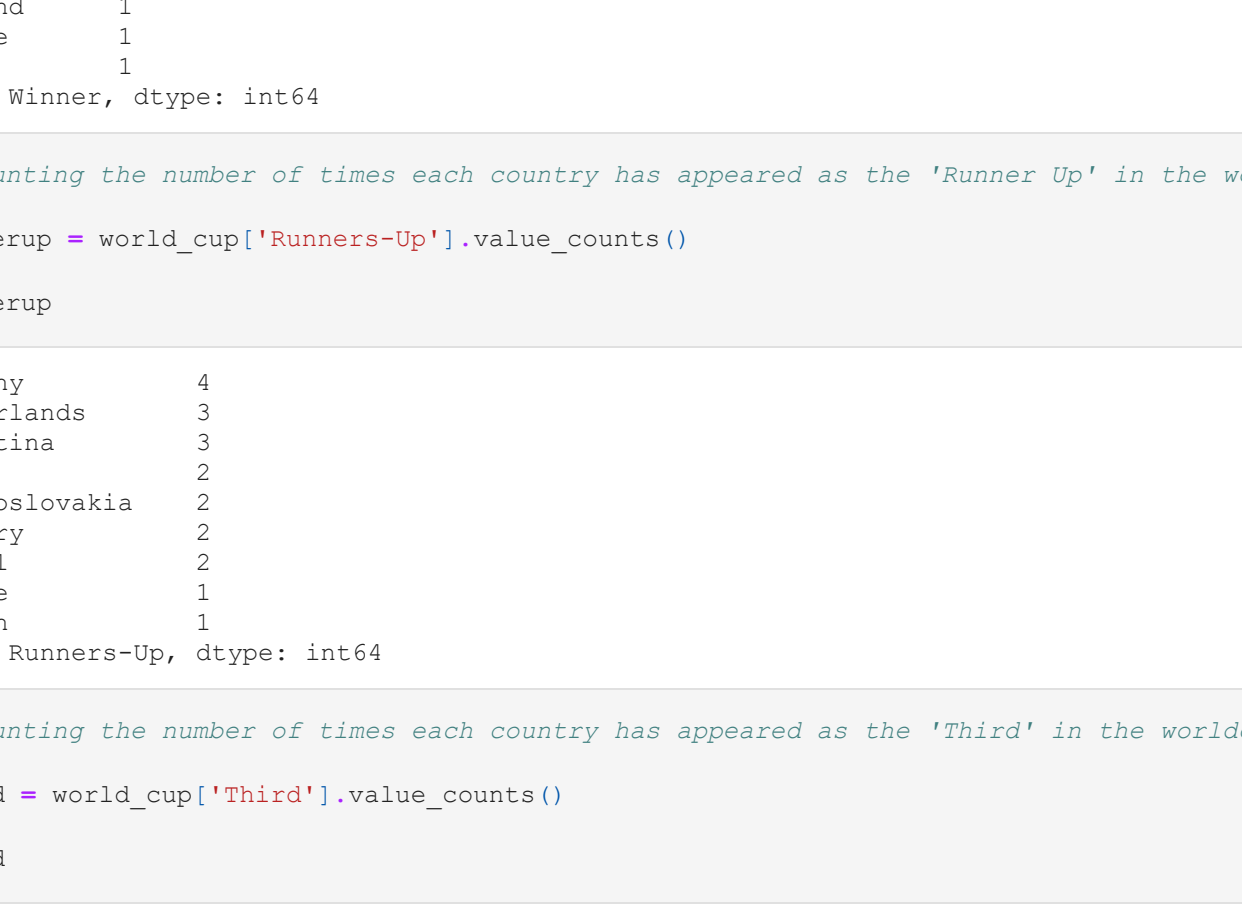
Introduction to Data Science: Football Worldcup Data Analysis

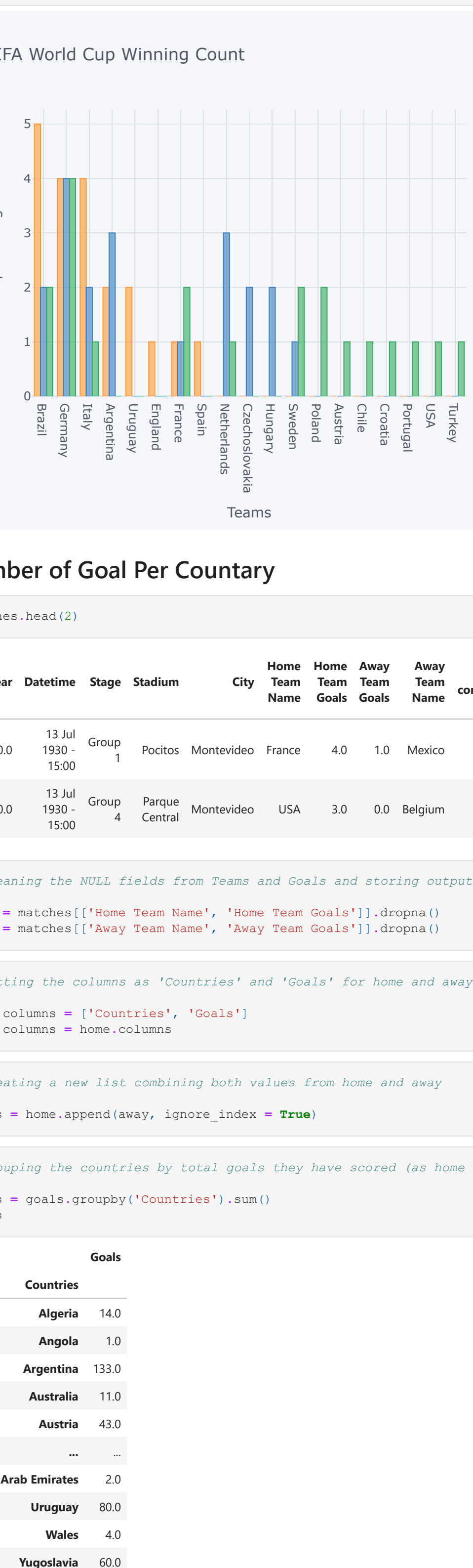
Data Cleaning

Most Number of World Cup Winning Title

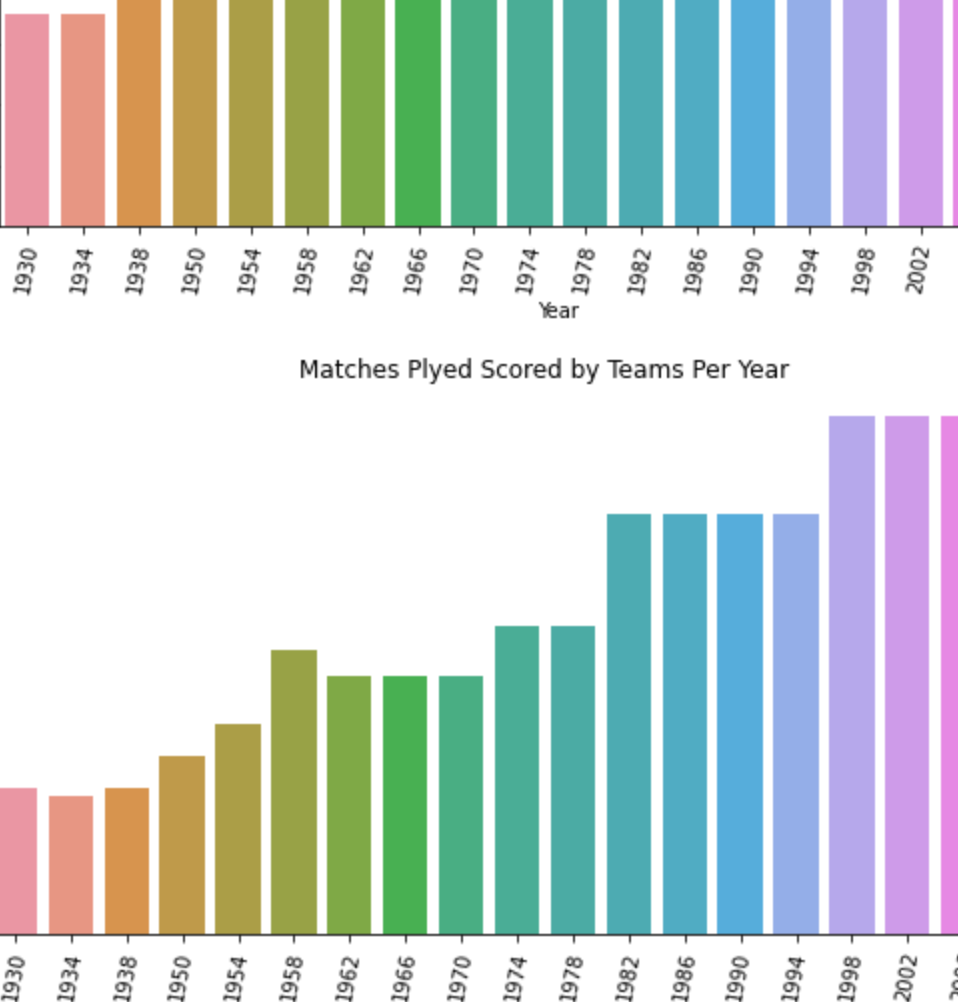
Number of Goal Per Country

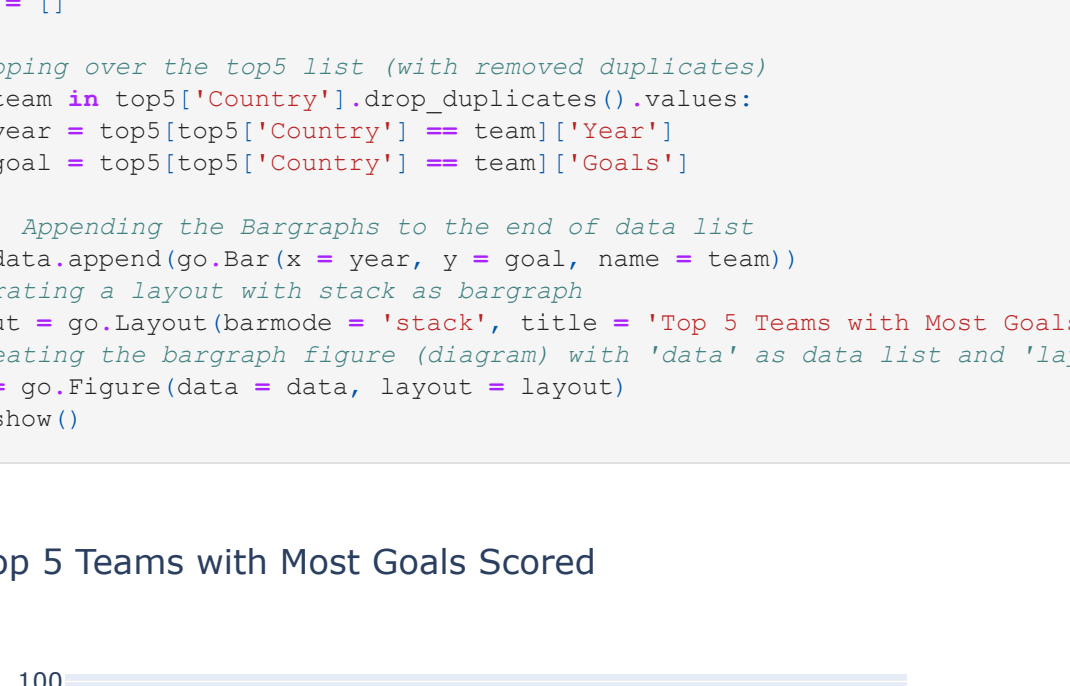


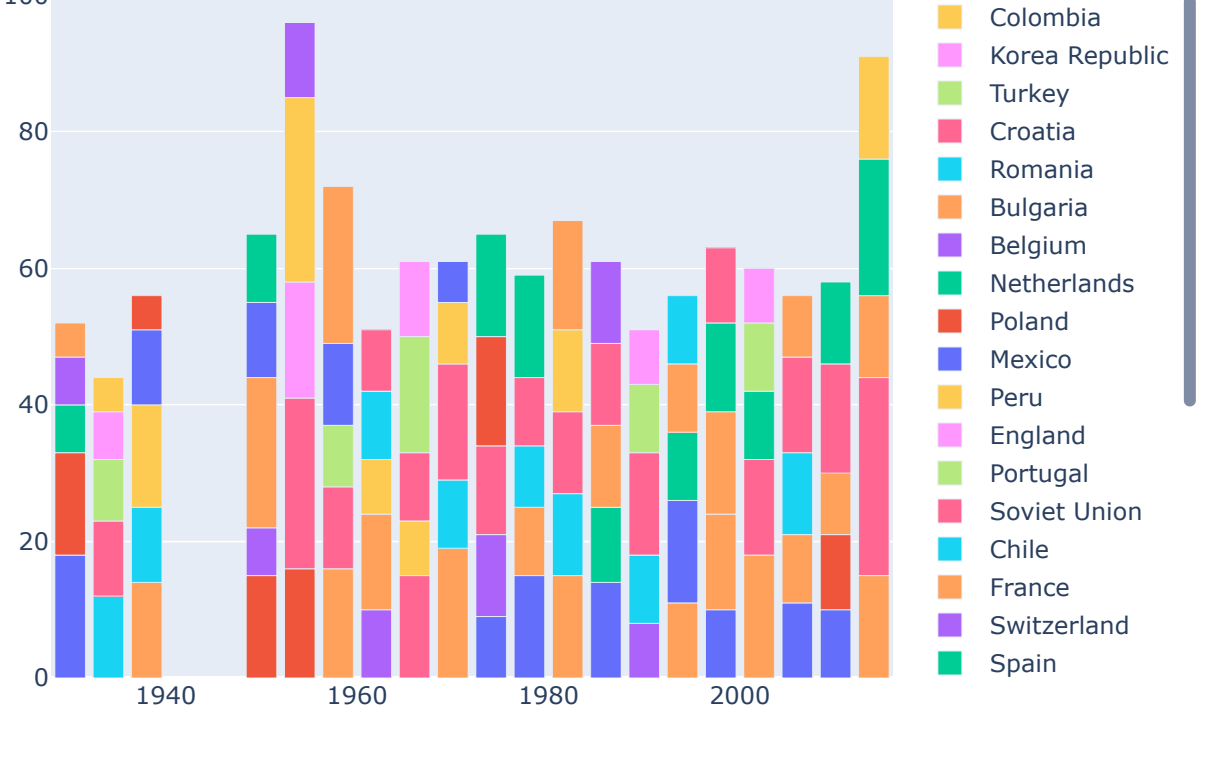
Attendance, Number of Teams, Goals, and Matches per Cup



Goals Per Team Per World Cup







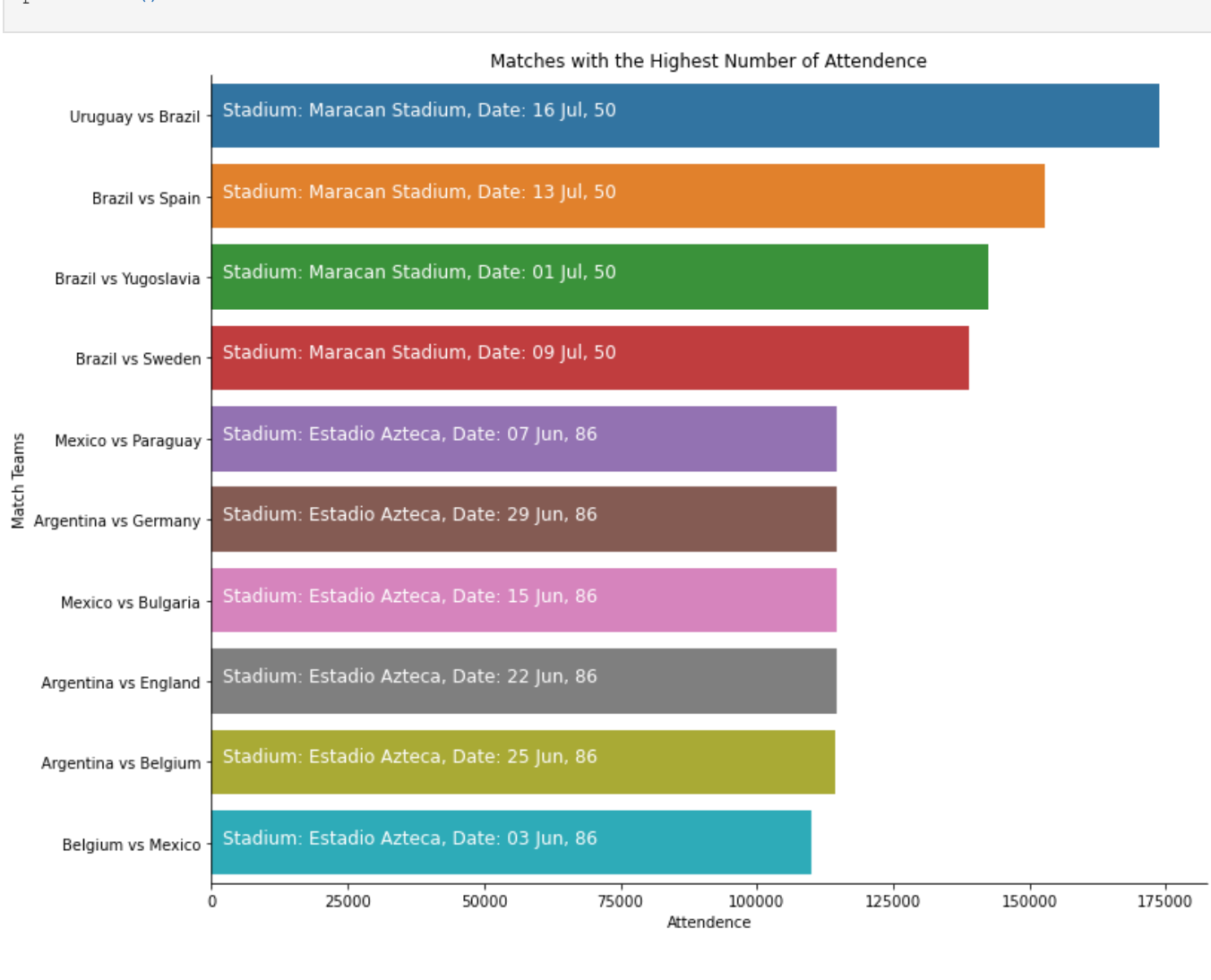
Matches With Highest Number Of Attendance

In [47]:
```python
# Converting the date to datetime dtype
matches['Datetime'] = pd.to_datetime(matches['Datetime'])
```

In [48]:
```python
# Stringifying the date format
# %d - date
# %b - abbrevated month name
# %y - year (YY format)

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

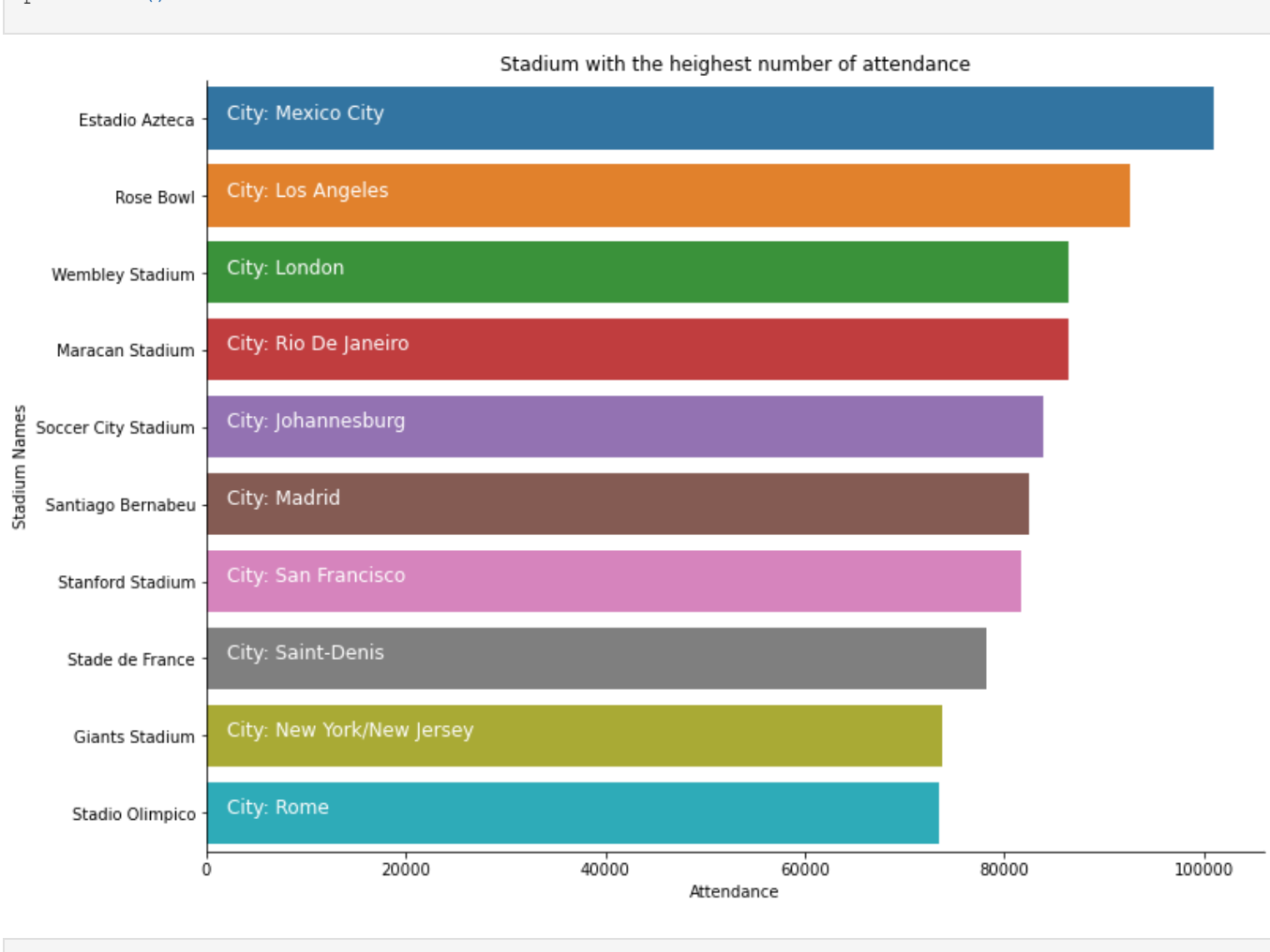
Out[48]:
```
0      13 Jul, 30
1      13 Jul, 30
2      14 Jul, 30
3      14 Jul, 30
4      15 Jul, 30
          ...
847    05 Jul, 14
848    08 Jul, 14
849    09 Jul, 14
850    12 Jul, 14
851    13 Jul, 14
Name: Datetime, Length: 852, dtype: object
```

In [49]:
```python
# Sorting the top 10 matches
top10 = matches.sort_values(by = 'Attendance', ascending = False)[:10]
top10['vs'] = top10['Home Team Name'] + " vs " + top10['Away Team Name']

# Creating the plot figure with size (12, 10)
plt.figure(figsize = (12,10))

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

plt.ylabel('Match Teams')
plt.xlabel('Attendence')
plt.title('Matches with the Highest Number of Attendence')

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

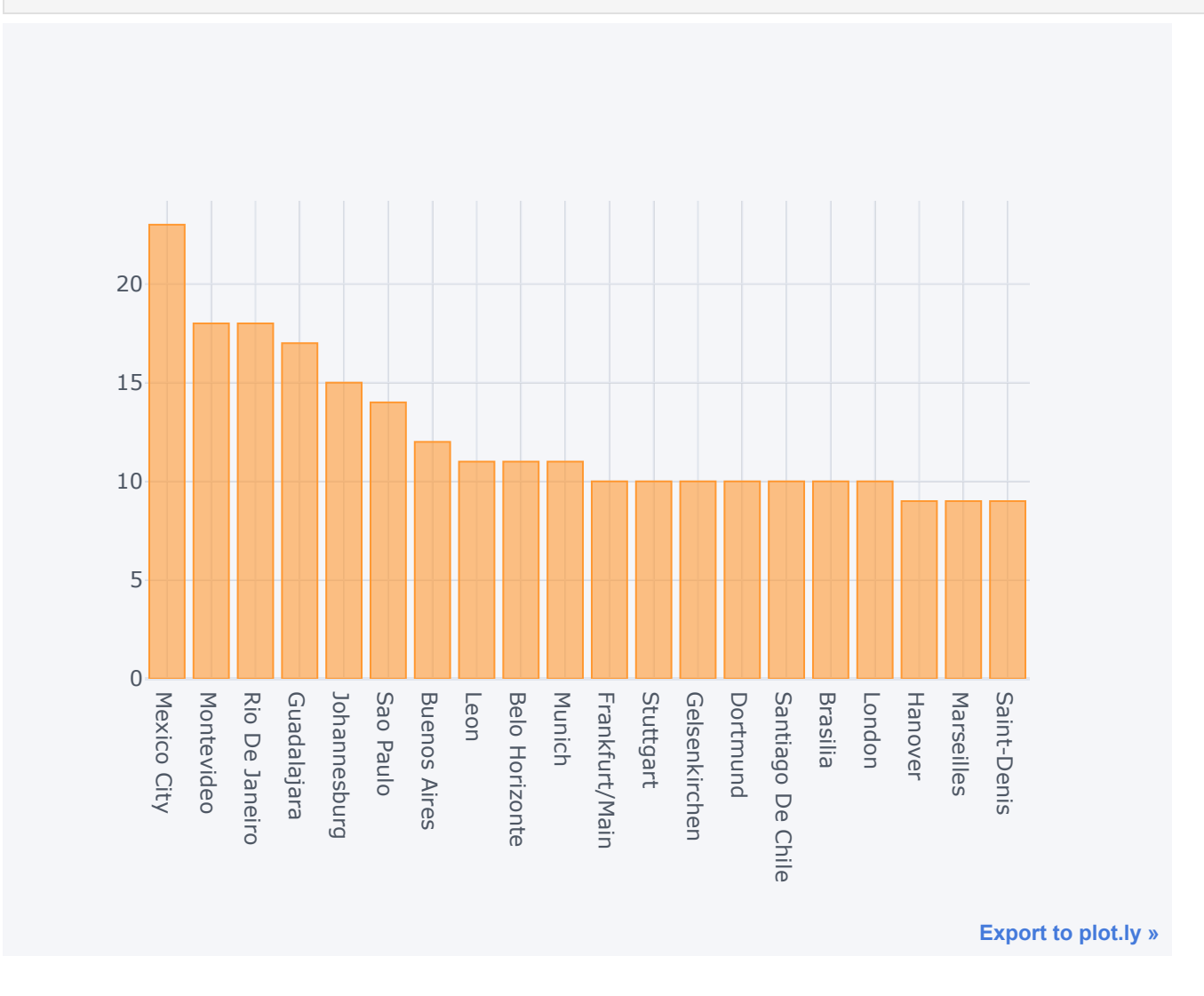


Stadium with Highest Average Attendance

In [50]:
```python
matches['Year'] = matches['Year'].astype(int)

std = matches.groupby(['Stadium', 'City'])['Attendance'].mean().reset_index().sort_va

top10 = std[:10]

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

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

plt.show()
```



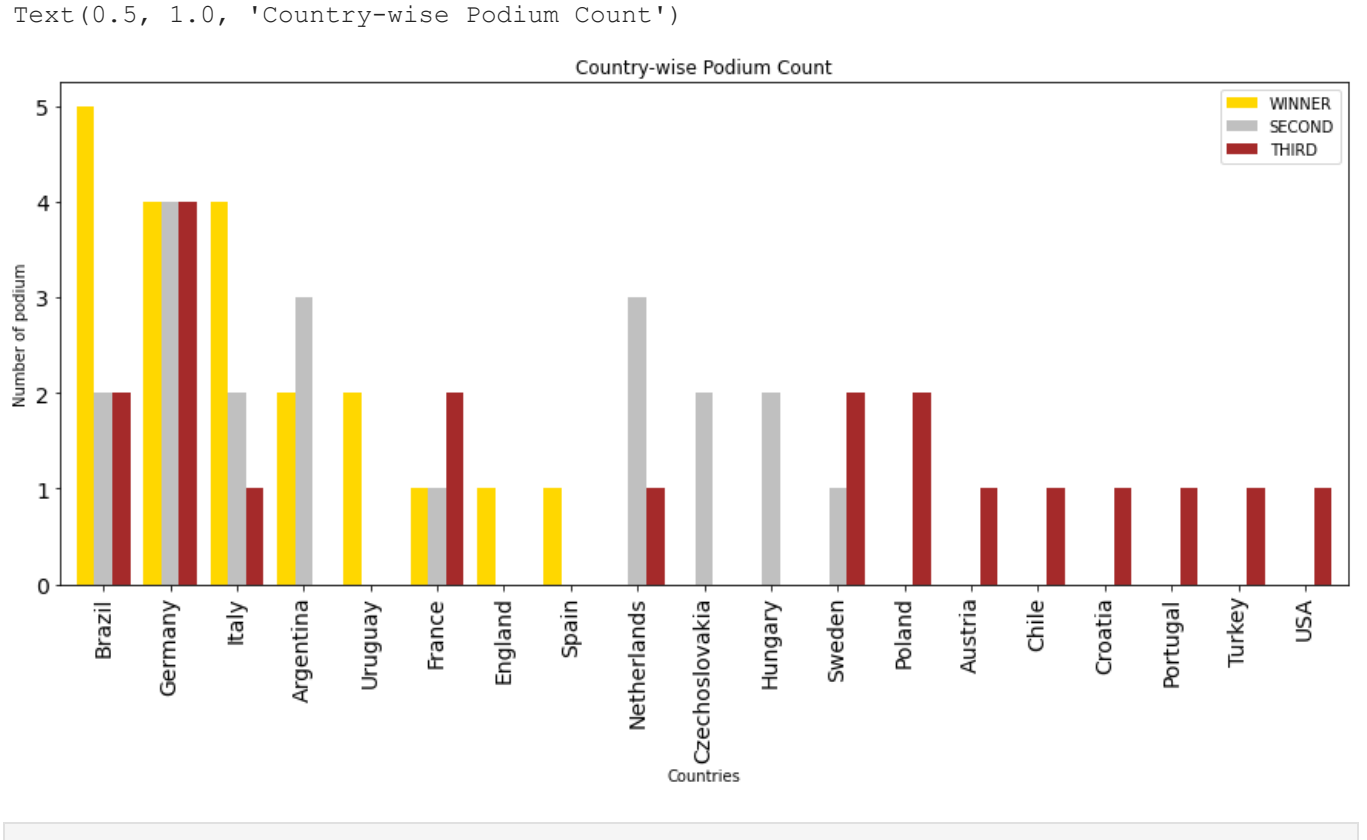
In [51]:
```python
# Most matches were played in which cities
matches['City'].value_counts()[:20].iplot(kind = 'bar')
```



Which countries had won the cup ?

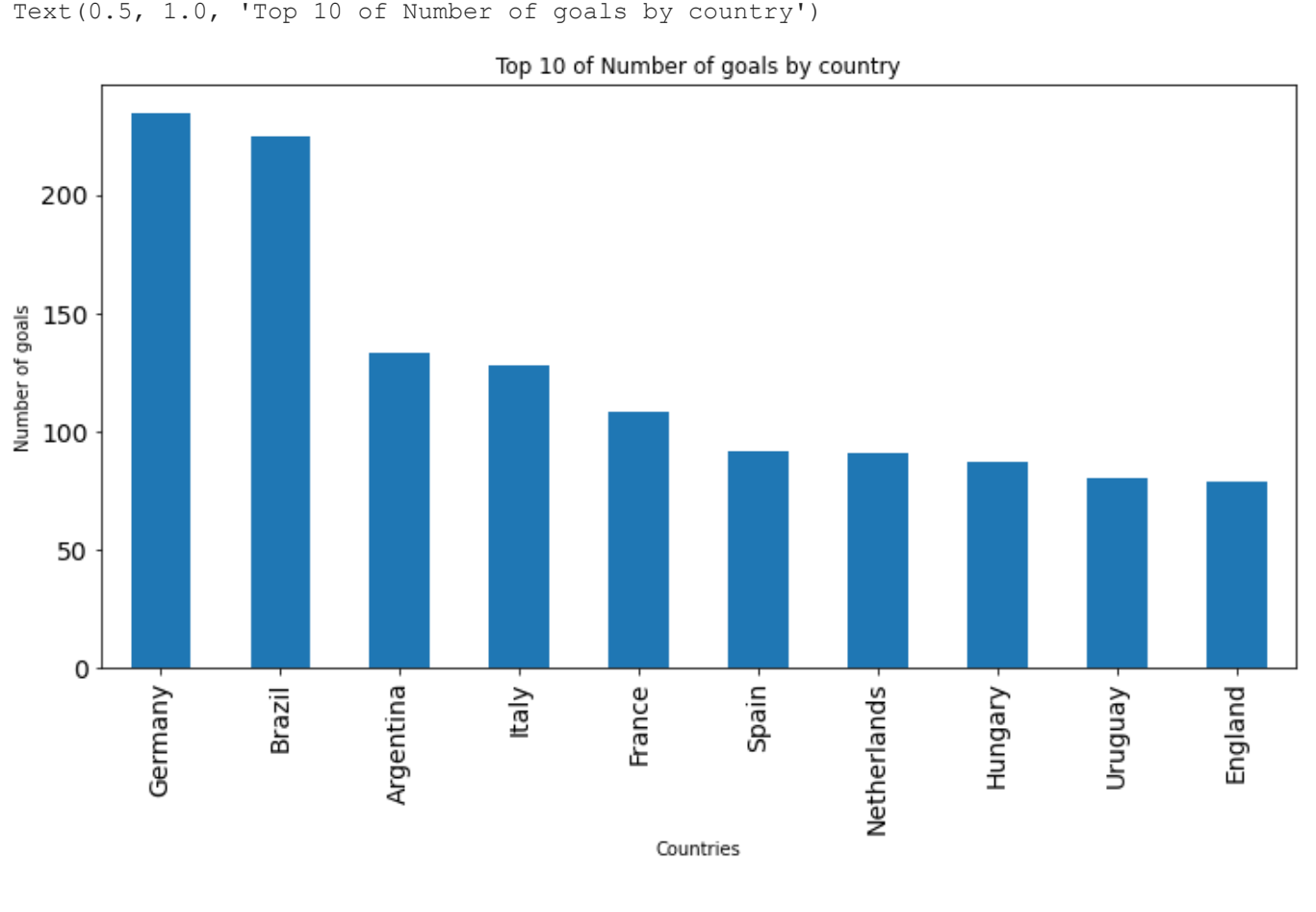
In [52]:
```python
# Creating 3 lists categorising Winner, Runners-Up and Third positions
gold = world_cup["Winner"]
silver = world_cup["Runners-Up"]
bronze = world_cup["Third"]

# Counting values from the lists (we have used from_dict() method as we are creating
gold_count = pd.DataFrame.from_dict(gold.value_counts())
silver_count = pd.DataFrame.from_dict(silver.value_counts())
bronze_count = pd.DataFrame.from_dict(bronze.value_counts())

# Performing Outer join
podium_count = gold_count.join(silver_count, how='outer').join(bronze_count, how='oute
# Filling NULL values with 0
podium_count = podium_count.fillna(0)
# Adding column titles
podium_count.columns = ['WINNER', 'SECOND', 'THIRD']
# Converting the dtype of list to int64
podium_count = podium_count.astype('int64')
# Sorting values in descending order
podium_count = podium_count.sort_values(by=['WINNER', 'SECOND', 'THIRD'], ascending=Fa

# Plotting the bargraph
podium_count.plot(y=['WINNER', 'SECOND', 'THIRD'], kind="bar",
                  color =['gold','silver','brown'], figsize=(15, 6), fontsize=14,
                 width=0.8, align='center')
plt.xlabel('Countries')
plt.ylabel('Number of podium')
plt.title('Country-wise Podium Count')
```

Out[52]: `Text(0.5, 1.0, 'Country-wise Podium Count')`



In [53]:
```python
# Removing NULL values
home = matches[['Home Team Name', 'Home Team Goals']].dropna()
# Removing NULL values
away = matches[['Away Team Name', 'Away Team Goals']].dropna()

# Creating a new DataFrame with countires and goals
goal_per_country = pd.DataFrame(columns=['countries', 'goals'])
# Appending away to home and then assigning the resultant to goal_per_country datafra
goal_per_country = goal_per_country.append(home.rename(index=str, columns={'Home Team
goal_per_country = goal_per_country.append(away.rename(index=str, columns={'Away Team

# Converting the dtype of goals
goal_per_country['goals'] = goal_per_country['goals'].astype('int64')

# Grouping the countries by country name, adding the goal values and then sorting the
goal_per_country = goal_per_country.groupby(['countries'])['goals'].sum().sort_values

# Plotting the graph
goal_per_country[:10].plot(x=goal_per_country.index, y=goal_per_country.values, kind=
plt.xlabel('Countries')
plt.ylabel('Number of goals')
plt.title('Top 10 of Number of goals by country')
```

Out[53]: `Text(0.5, 1.0, 'Top 10 of Number of goals by country')`



Match outcome by home and away teams

In [54]:
```python
# Creating a function get_labels() which takes in a match as argument and returns if
# If both teams scored equally, it returns 'DRAW'
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'
```

In [61]:
```python
# Creating a new column 'outcome' in matches dataframe and performing the get_labels(
matches['outcome'] = matches.apply(lambda x: get_labels(x), axis=1)
```

In [62]:
```python
# Matches dataframe after executing get_label() function
matches.head()
```

Out[62]:

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	...	Half-time Home Goals	Ha ti Av Go
0	1930	13 Jul, 30	Group 1	Pocitos	Montevideo	France	4.0	1.0	Mexico		...	3.0	
1	1930	13 Jul, 30	Group 4	Parque Central	Montevideo	USA	3.0	0.0	Belgium		...	2.0	
2	1930	14 Jul, 30	Group 2	Parque Central	Montevideo	Yugoslavia	2.0	1.0	Brazil		...	2.0	
3	1930	14 Jul, 30	Group 3	Pocitos	Montevideo	Romania	3.0	1.0	Peru		...	1.0	
4	1930	15 Jul, 30	Group 1	Parque Central	Montevideo	Argentina	1.0	0.0	France		...	0.0	

5 rows × 21 columns

In [63]:
```python
# Counting the total result of matchs played
match_outcomes = matches['outcome'].value_counts()
match_outcomes
```

Out[63]:
```
Home Team Win    488
DRAW             190
Away Team Win    174
Name: outcome, dtype: int64
```

In [64]:
```python
# Plotting graph on match_outcomes
plt.figure(figsize = (6,6))

match_outcomes.plot.pie(autopct = "%1.0f%%", colors = sns.color_palette('winter_r'), 

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()
```