In [4]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [5]:
```python
#inserting datasets
athletes = pd.read_csv('C:/Users/achum/Downloads/archive/athlete_events.csv')
regions = pd.read_csv('C:/Users/achum/Downloads/archive/noc_regions.csv')
```

In [6]:
```python
athletes.head()
```

Out[6]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [7]:
```python
regions.head()
```

Out[7]:

	NOC	region	notes
0	AFG	Afghanistan	NaN
1	AHO	Curacao	Netherlands Antilles
2	ALB	Albania	NaN
3	ALG	Algeria	NaN
4	AND	Andorra	NaN

In [8]:
```python
#joining the dataframes

athletes_df = athletes.merge(regions,how = 'left', on = 'NOC')
athletes_df.head()
```

Out[8]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	China	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	China	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	Denmark	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	Denmark	NaN
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN	Netherlands	NaN

In [9]:
```python
#number of rows and coluns
athletes_df.shape
```

Out[9]:
```
(271116, 17)
```

In [10]:
```python
athletes_df.describe()
```

Out[10]:

	ID	Age	Height	Weight	Year
count	271116.000000	261642.000000	210945.000000	208241.000000	271116.000000
mean	68248.954396	25.556898	175.338970	70.702393	1978.378480
std	39022.286345	6.393561	10.518462	14.348020	29.877632
min	1.000000	10.000000	127.000000	25.000000	1896.000000
25%	34643.000000	21.000000	168.000000	60.000000	1960.000000
50%	68205.000000	24.000000	175.000000	70.000000	1988.000000
75%	102097.250000	28.000000	183.000000	79.000000	2002.000000
max	135571.000000	97.000000	226.000000	214.000000	2016.000000

In [11]:
```python
athletes_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 271116 entries, 0 to 271115
Data columns (total 17 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   ID      271116 non-null  int64
 1   Name    271116 non-null  object
 2   Sex     271116 non-null  object
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object
 7   NOC     271116 non-null  object
 8   Games   271116 non-null  object
 9   Year    271116 non-null  int64
 10  Season  271116 non-null  object
 11  City    271116 non-null  object
 12  Sport   271116 non-null  object
 13  Event   271116 non-null  object
 14  Medal   39783 non-null   object
 15  region  270746 non-null  object
 16  notes   5039 non-null    object
dtypes: float64(3), int64(2), object(12)
memory usage: 37.2+ MB
```

In [14]:
```python
#recheck
athletes_df.head()
```

Out[14]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	Region	Notes
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	China	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	China	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	Denmark	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	Denmark	NaN
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN	Netherlands	NaN

In [13]:
```python
#making column name as consistent
athletes_df.rename(columns={'region':'Region', 'notes':'Notes'},inplace=True)
```

In [15]:
```python
#checking null values
null_values = athletes_df.isna()
null_columns = null_values.any()
null_columns
```

Out[15]:
```
ID        False
Name      False
Sex       False
Age        True
Height     True
Weight     True
Team      False
NOC       False
Games     False
Year      False
Season    False
City      False
Sport     False
Event     False
Medal      True
Region     True
Notes      True
dtype: bool
```

In [16]:
```python
#total null values in particluar rows
athletes_df.isnull().sum()
```

Out[16]:
```
ID             0
Name           0
Sex            0
Age         9474
Height     60171
Weight     62875
Team           0
NOC            0
Games          0
Year           0
Season         0
City           0
Sport          0
Event          0
Medal     231333
Region       370
Notes     266077
dtype: int64
```

In [18]:
```python
#getting datas for specific countries using function called "query"
#for INDIA

athletes_df.query('Team == "India"').head(5)
```

Out[18]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	Region	Notes
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Athletics	Athletics Men's 110 metres Hurdles	NaN	India	NaN
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Athletics	Athletics Men's 400 metres Hurdles	NaN	India	NaN
895	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Athletics	Athletics Women's 800 metres	NaN	India	NaN
896	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Athletics	Athletics Women's 4 x 400 metres Relay	NaN	India	NaN
897	512	Shiny Kurisingal Abraham-Wilson	F	23.0	167.0	53.0	India	IND	1988 Summer	1988	Summer	Seoul	Athletics	Athletics Women's 800 metres	NaN	India	NaN

In [19]:
```python
#for JAPAN
athletes_df.query('Team == "Japan"').head(5)
```

Out[19]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	Region	Notes
625	362	Isao Ko Abe	M	24.0	177.0	75.0	Japan	JPN	1936 Summer	1936	Summer	Berlin	Athletics	Athletics Men's Hammer Throw	NaN	Japan	NaN
629	363	Kazumi Abe	M	28.0	178.0	67.0	Japan	JPN	1976 Winter	1976	Winter	Innsbruck	Bobsleigh	Bobsleigh Men's Four	NaN	Japan	NaN
630	364	Kazuo Abe	M	25.0	166.0	69.0	Japan	JPN	1960 Summer	1960	Summer	Roma	Wrestling	Wrestling Men's Lightweight, Freestyle	NaN	Japan	NaN
631	365	Kinya Abe	M	23.0	168.0	68.0	Japan	JPN	1992 Summer	1992	Summer	Barcelona	Fencing	Fencing Men's Foil, Individual	NaN	Japan	NaN
632	366	Kiyoshi Abe	M	25.0	167.0	62.0	Japan	JPN	1972 Summer	1972	Summer	Munich	Wrestling	Wrestling Men's Featherweight, Freestyle	NaN	Japan	NaN

In [20]:
```python
#since 1896 the participation list of top 10 countries

top_countries = athletes_df.Team.value_counts().sort_values(ascending = False).head(10)
top_countries
```

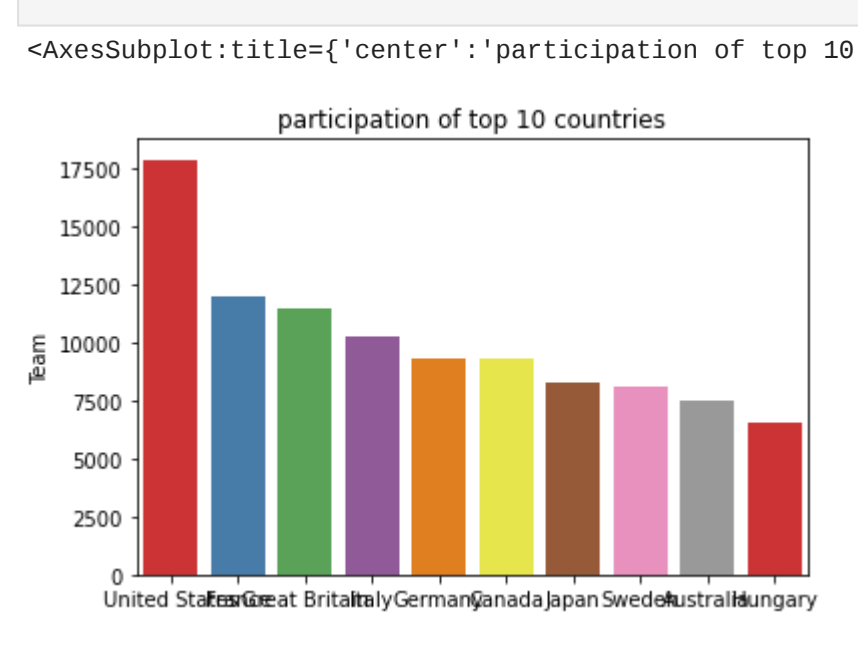
Out[20]:
```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Canada            9279
Japan             8289
Sweden            8052
Australia         7513
Hungary           6547
Name: Team, dtype: int64
```

In [21]:
```python
#plotting for above top 10 countries
plt.figure(figsize=(12,6))
```

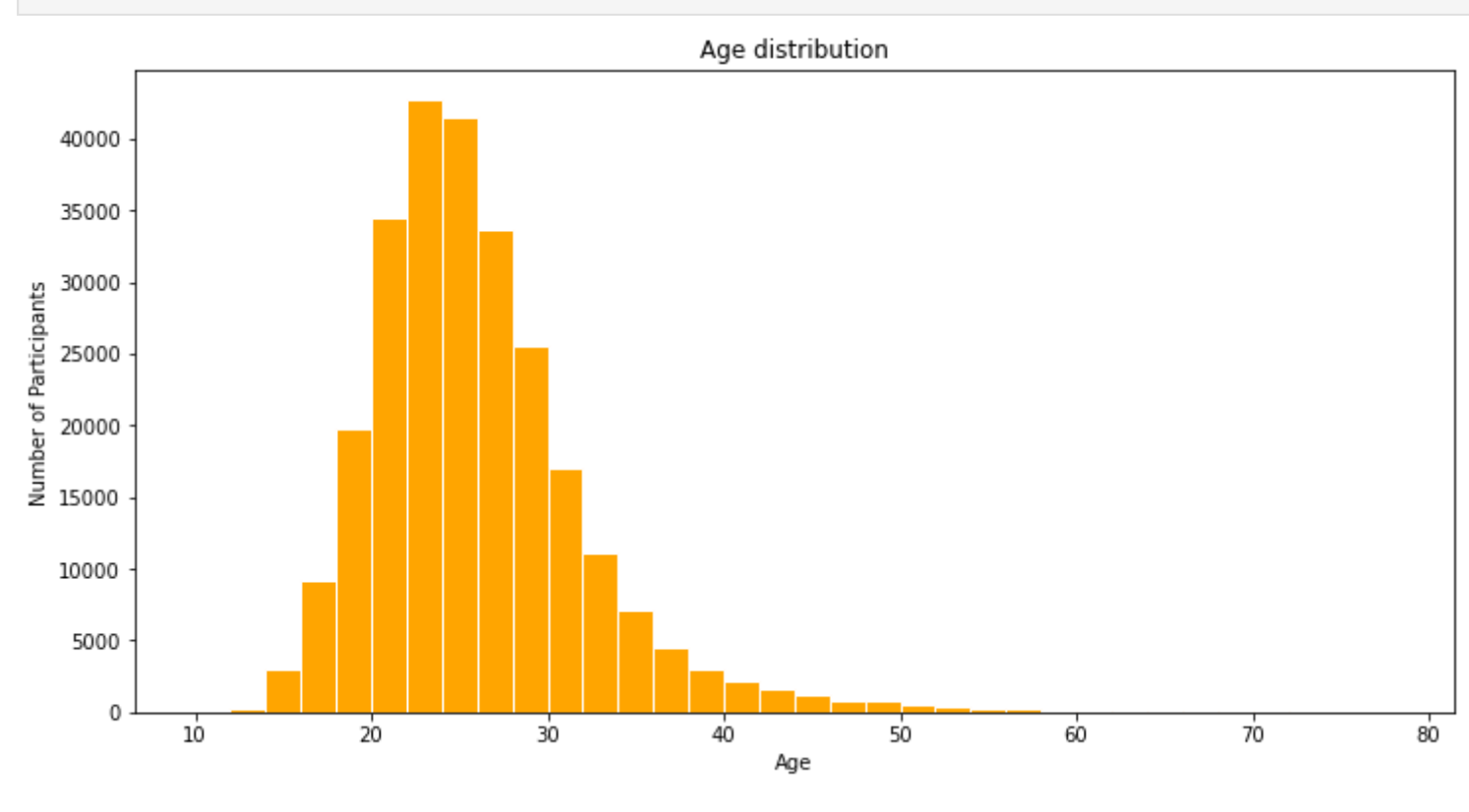
Out[21]:
```
<Figure size 864x432 with 0 Axes>

<Figure size 864x432 with 0 Axes>
```

In [24]:
```python
#plot rotation=20
plt.title('participation of top 10 countries')
sns.barplot(x=top_countries.index, y=top_countries, palette='Set1')
```

Out[24]:
```
<AxesSubplot:title={'center':'participation of top 10 countries'}, ylabel='Team'>
```



In [29]:
```python
#age distribution of the participants
plt.figure(figsize=(12,6))
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('Number of Participants')
plt.hist(athletes_df.Age,bins = np.arange(10,80,2),color='orange',edgecolor='white');
```



In [32]:
```python
#winter sports

winter_sports = athletes_df[athletes_df.Season == 'Winter'].Sport.unique()
winter_sports
```

Out[32]:
```
array(['Speed Skating', 'Cross Country Skiing', 'Ice Hockey', 'Biathlon',
       'Alpine Skiing', 'Luge', 'Bobsleigh', 'Figure Skating',
       'Nordic Combined', 'Freestyle Skiing', 'Ski Jumping', 'Curling',
       'Snowboarding', 'Short Track Speed Skating', 'Skeleton',
       'Military Ski Patrol', 'Alpinism'], dtype=object)
```

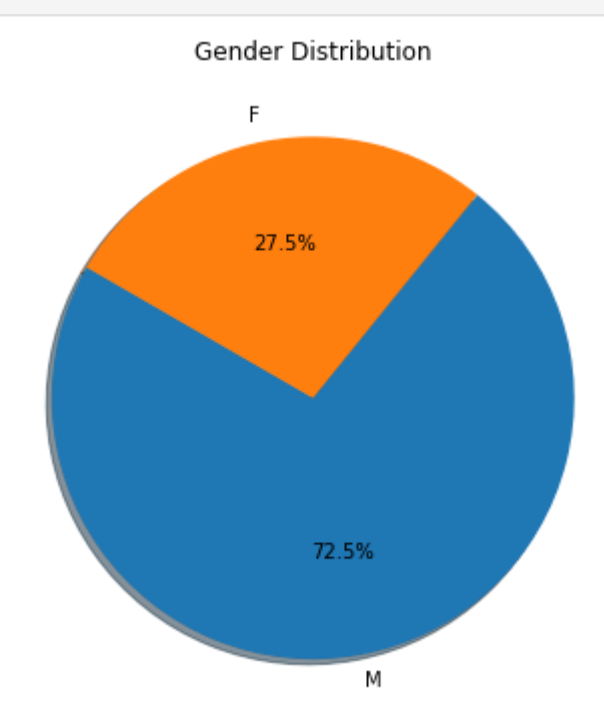
In [33]:
```python
#summer sports

summer_sports = athletes_df[athletes_df.Season == 'Summer'].Sport.unique()
summer_sports
```

Out[33]:
```
array(['Basketball', 'Judo', 'Football', 'Tug-Of-War', 'Athletics',
       'Swimming', 'Badminton', 'Sailing', 'Gymnastics',
       'Art Competitions', 'Handball', 'Weightlifting', 'Wrestling',
       'Water Polo', 'Hockey', 'Rowing', 'Fencing', 'Equestrianism',
       'Shooting', 'Boxing', 'Taekwondo', 'Cycling', 'Diving', 'Canoeing',
       'Tennis', 'Modern Pentathlon', 'Golf', 'Softball', 'Archery',
       'Volleyball', 'Synchronized Swimming', 'Table Tennis', 'Baseball',
       'Rhythmic Gymnastics', 'Rugby Sevens', 'Trampolining',
       'Beach Volleyball', 'Triathlon', 'Rugby', 'Lacrosse', 'Polo',
       'Cricket', 'Ice Hockey', 'Racquets', 'Motorboating', 'Croquet',
       'Figure Skating', 'Jeu De Paume', 'Roque', 'Basque Pelota',
       'Alpinism', 'Aeronautics'], dtype=object)
```

In [36]:
```python
#male and female participants

gender_counts = athletes_df.Sex.value_counts()
gender_counts
```

Out[36]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [39]:
```python
#pie chart for male & female athletes

plt.figure(figsize=(12,6))
plt.title('Gender Distribution')
plt.pie(gender_counts,labels=gender_counts.index,autopct='%1.1f%%',startangle=150,shadow=True);
```



In [38]:
```python
#total medals

athletes_df.Medal.value_counts()
```

Out[38]:
```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

In [43]:
```python
#total number of FEMALE athletes in each olympics

female_participants = athletes_df[(athletes_df.Sex =='F') & (athletes_df.Season == 'Summer')][['Sex','Year']]
female_participants = female_participants.groupby('Year').count().reset_index()
female_participants.head()
```

Out[43]:

	Year	Sex
0	1900	33
1	1904	16
2	1906	11
3	1908	47
4	1912	87

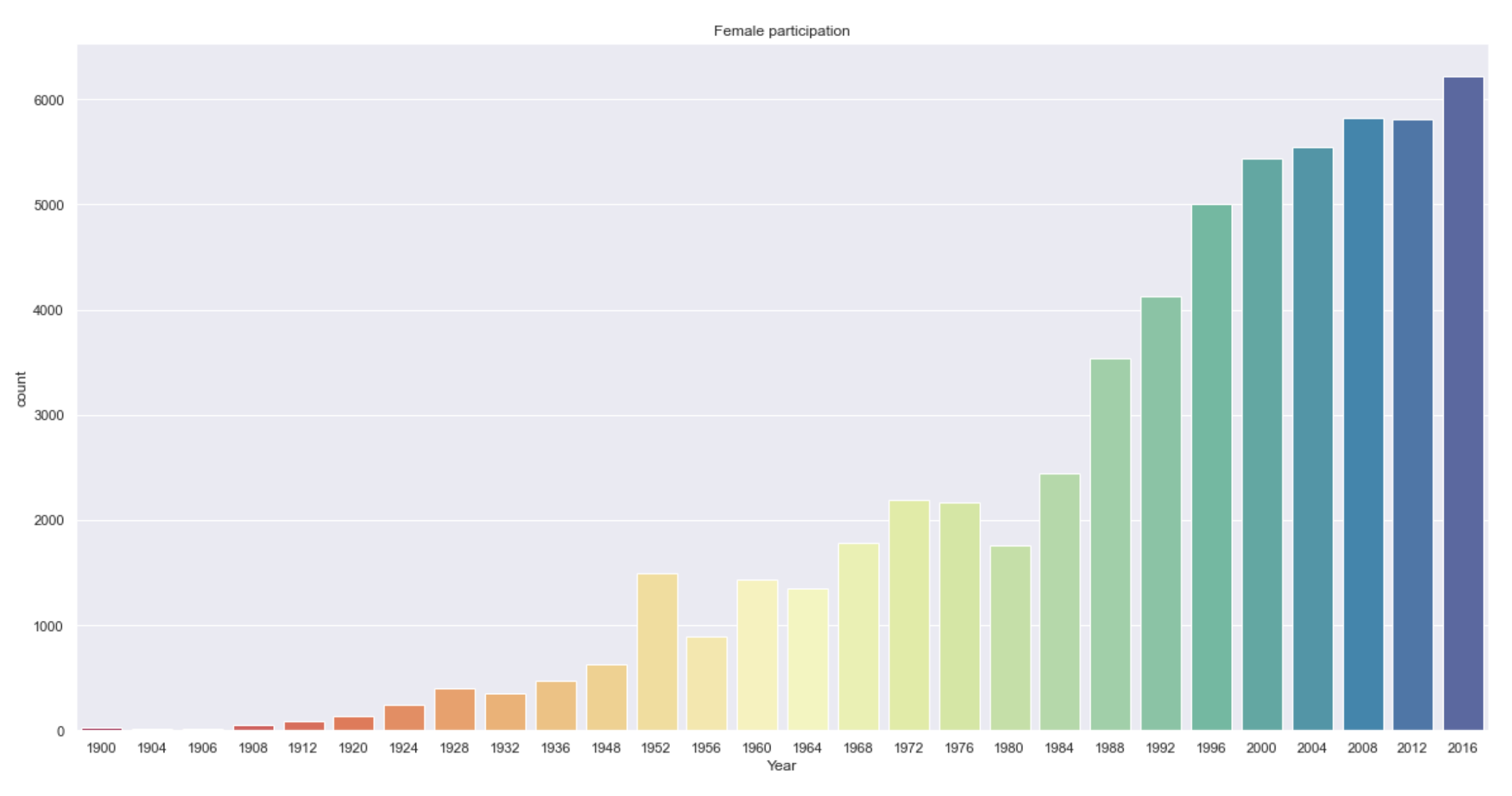
In [44]:
```python
#In recent years counts of female athletes

female_participants = athletes_df[(athletes_df.Sex =='F') & (athletes_df.Season == 'Summer')][['Sex','Year']]
female_participants = female_participants.groupby('Year').count().reset_index()
female_participants.tail()
```

Out[44]:

	Year	Sex
23	2000	5431
24	2004	5546
25	2008	5816
26	2012	5815
27	2016	6223

In [46]:
```python
#bar graph plot for female participation from begining to 2016

women_olympics = athletes_df[(athletes_df.Sex == 'F') & (athletes_df.Season == 'Summer')]
sns.set(style ="darkgrid")
plt.figure(figsize=(20,10))
sns.countplot(x='Year',data = women_olympics,palette = "Spectral")
plt.title('Female participation')
```

Out[46]:
```
Text(0.5, 1.0, 'Female participation')
```



In [16]:
```python
#gold medal athletes

goldmedals = athletes_df[(athletes_df.Medal == 'Gold')]
goldmedals.head()
```

Out[16]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	Denmark	NaN
42	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Team All-Around	Gold	Finland	NaN
44	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Horse Vault	Gold	Finland	NaN
48	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	London	Gymnastics	Gymnastics Men's Pommelled Horse	Gold	Finland	NaN
60	20	Kjetil Andr Aamodt	M	20.0	176.0	85.0	Norway	NOR	1992 Winter	1992	Winter	Albertville	Alpine Skiing	Alpine Skiing Men's Super G	Gold	Norway	NaN

In [17]:
```python
#gold medals beyond age 60

goldmedals['ID'][goldmedals['Age']>60].count()
```

Out[17]:
```
6
```

In [18]:
```python
#for which events they got gold medals

sportevent = goldmedals['Sport'][goldmedals['Age']>60]
sportevent
```

Out[18]:
```
104003    Art Competitions
105199               Roque
190952             Archery
226374             Archery
233390            Shooting
261102             Archery
Name: Sport, dtype: object
```

In [19]:
```python
#name of the athletes who won the gold medals beyond the age of 60

name = goldmedals['Name'][goldmedals['Age']>60]
name
```

Out[19]:
```
104003                         Isaac Lazarus Israls
105199                              Charles Jacobus
190952    Lida Peyton "Eliza" Pollock (McMillen-)
226374                 Galen Carter "G. C." Spencer
233390                            Oscar Gomer Swahn
261102                       Robert W. Williams, Jr.
Name: Name, dtype: object
```

In [27]:
```python
#gold medals for each country

goldmedals.region.value_counts().reset_index(name ='Medal').head(5)
```

Out[27]:

	index	Medal
0	USA	2638
1	Russia	1599
2	Germany	1301
3	UK	678
4	Italy	575

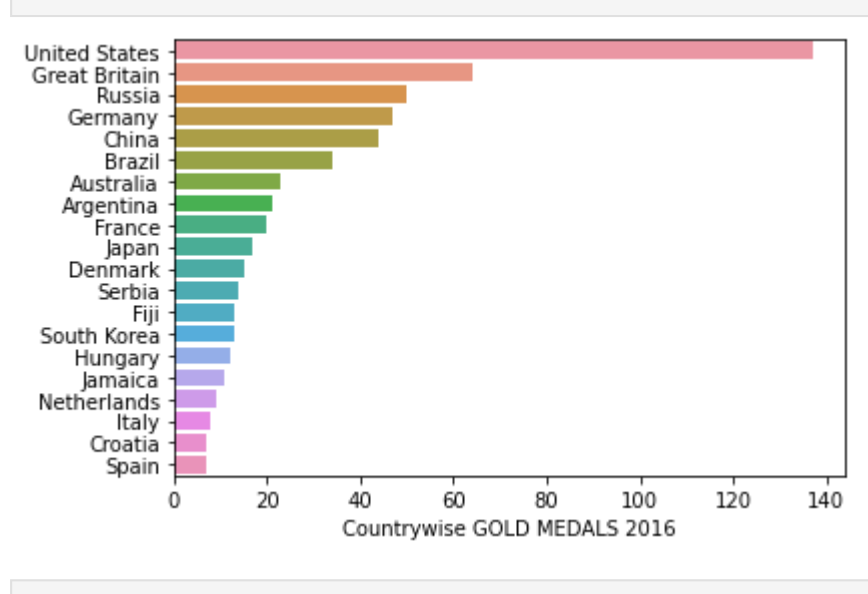
In [29]:
```python
#Rio olympics held on 2016

max_year = athletes.Year.max()
print(max_year)
```

```
2016
```

In [31]:
```python
max_year = athletes.Year.max()
print(max_year)
team_names = athletes_df[(athletes_df.Year == max_year) & (athletes_df.Medal == 'Gold')].Team
team_names.value_counts().head(10)
```

Out[31]:
```
2016
United States    137
Great Britain     64
Russia            50
Germany           47
China             44
Brazil            34
Australia         23
Argentina         21
France            20
Japan             17
Name: Team, dtype: int64
```

In [37]:
```python
#bar graph for above rio olympics

sns.barplot(x=team_names.value_counts().head(20),y=team_names.value_counts().head(20).index)
plt.ylabel(None);
plt.xlabel('Countrywise GOLD MEDALS 2016');
```



In [ ]: