120 years of Olympic Data Analysis

In [116]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [117]:
```
# Load Dataset
athlete  = pd.read_csv("D:/data analysis projects/olymipic/athlete_events.csv")
regines = pd.read_csv("D:/data analysis projects/olymipic/noc_regions.csv")
```

In [118]:
```
athlete.head()
```

Out[118]:

	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 [119]:
```
regines.head()
```

Out[119]:

	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 [120]:
```
athlete_df = athlete.merge(regines, how = "left", on = "NOC")
athlete_df.head()
```

Out[120]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region
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
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
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	Denmark
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
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

In [121]:
```
athlete_df.shape
```

Out[121]:
```
(271116, 17)
```

In [122]:
```
athlete_df.rename(columns = {"region": "Region", "notes": "Notes"})
```

Out[122]:

	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	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Innsbruck	Luge	Luge Mixed (Men)'s Doubles	NaN	
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Individual	NaN	
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	Ski Jumping Men's Large Hill, Team	NaN	
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winter	Nagano	Bobsleigh	Bobsleigh Men's Four	NaN	
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winter	Salt Lake City	Bobsleigh	Bobsleigh Men's Four	NaN	

271116 rows × 17 columns

In [123]:
```
athlete_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 [124]:
```
athlete_df.describe()
```

Out[124]:

	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 [125]:
```
# Check for null values
nan_val = athlete_df.isna()
nan_col = nan_val.any()
nan_col
```

Out[125]:
```
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 [126]:
```
athlete_df.isnull().sum()
```

Out[126]:
```
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 [127]:
```
athlete_df.query('Team =="India" ').head()
```

Out[127]:

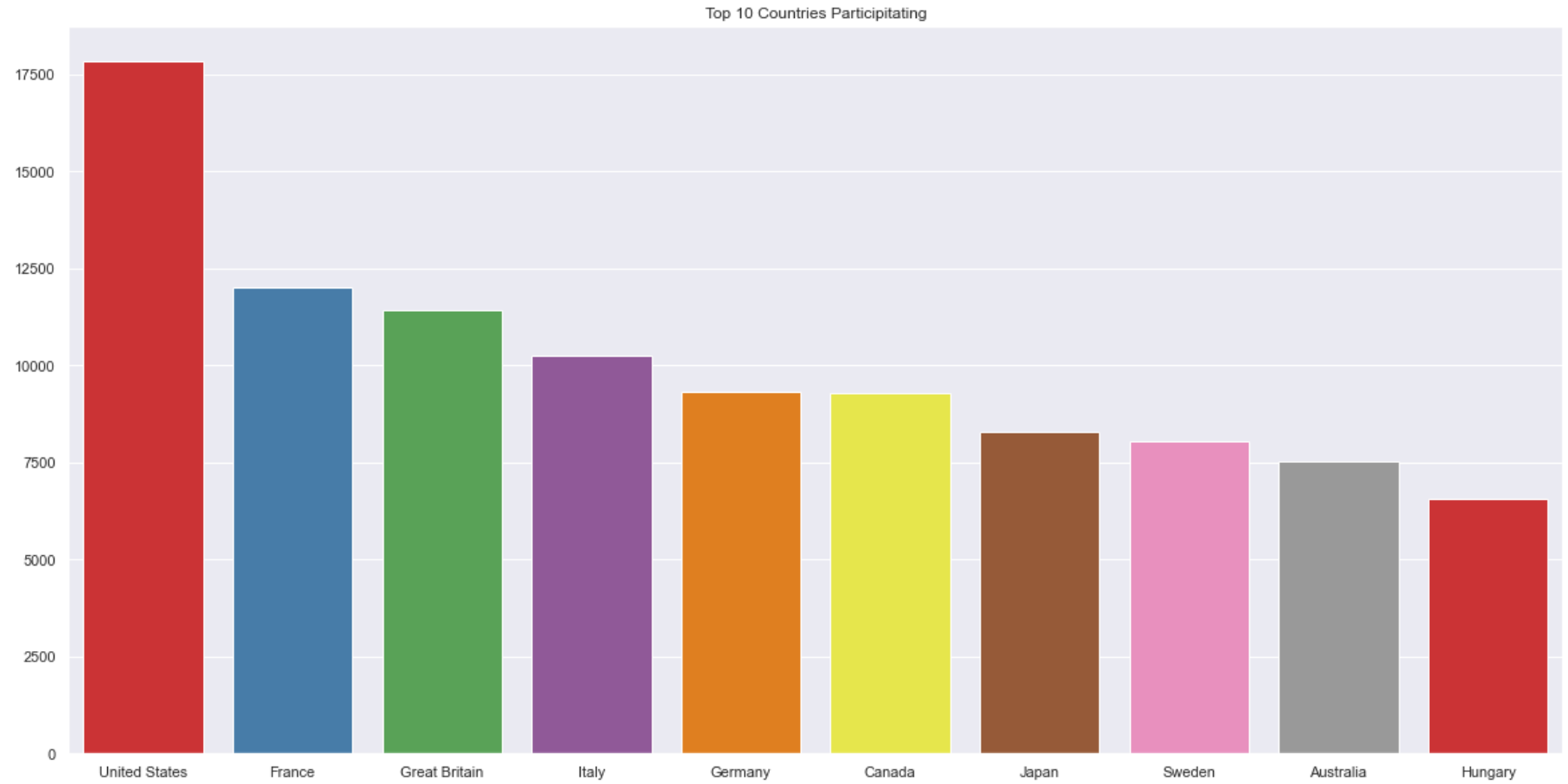
	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Athletics	Athletics Men's 110 metres Hurdles	NaN	India	NaN
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	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 [128]:
```
top_10_countries  = athlete_df.Team.value_counts().sort_values(ascending = False).head(10)
top_10_countries
```

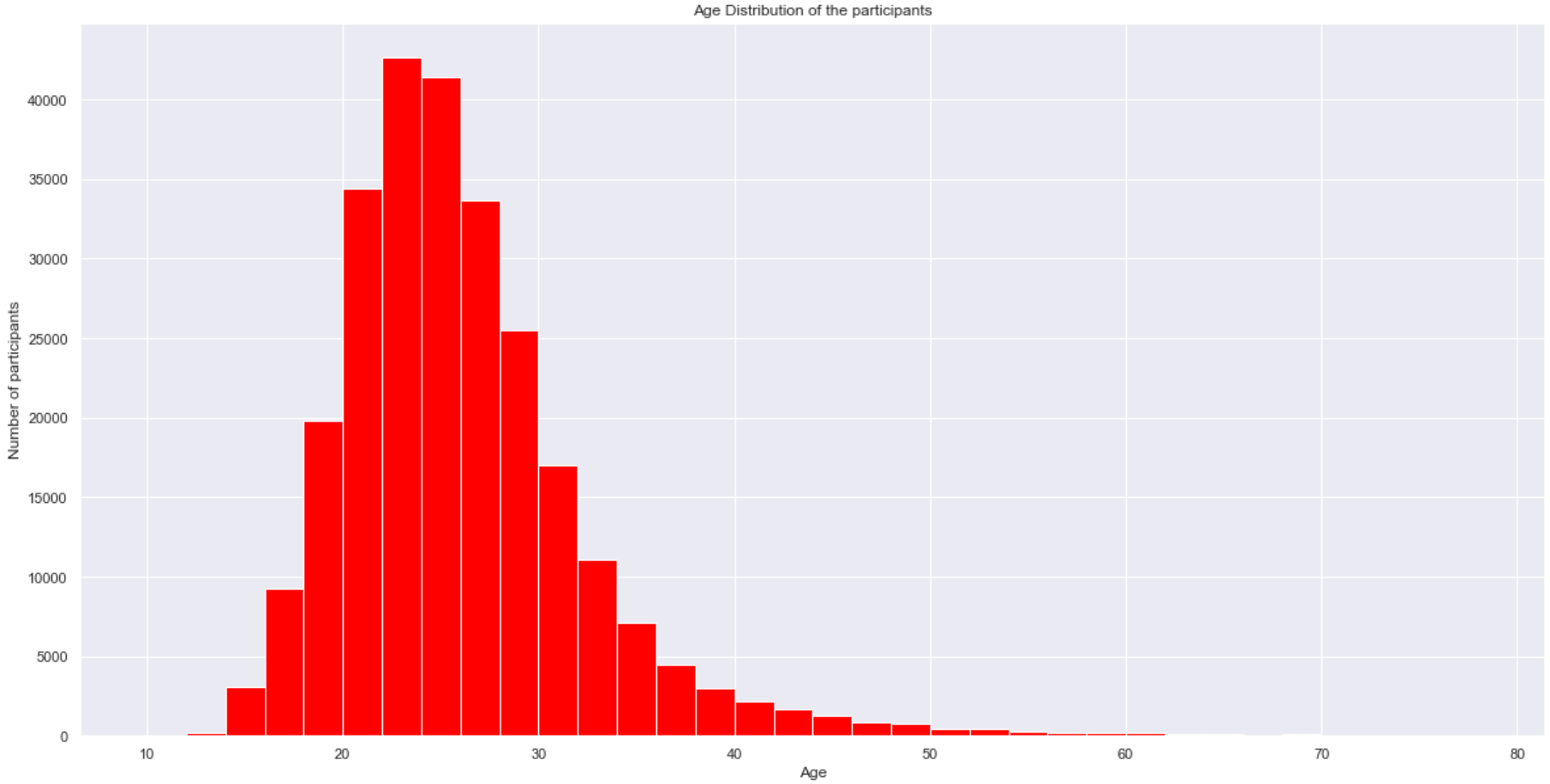
Out[128]:
```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Canada            9279
Japan             8289
Sweden            8052
Australia         7513
Hungary           6547
Name: Team, dtype: int64
```

In [129]:
```
plt.figure(figsize=(20,10))
plt.title("Top 10 Countries Participitating")
sns.barplot(x= top_10_countries.index, y=top_10_countries.values, palette = "Set1")
```

Out[129]: <matplotlib.axes._subplots.AxesSubplot at 0x1e640834820>



In [130]:
```
# Age Distribution of the participants
plt.figure(figsize = (20,10))
plt.title("Age Distribution of the participants")
plt.xlabel("Age")
plt.ylabel("Number of participants")
plt.hist(athlete_df.Age, bins = np.arange(10,80,2), color = "red", edgecolor = "white");
```



In [131]:
```
# Winter Sports
winter_sports = athlete_df[athlete_df.Season == "Winter"].Sport.unique()
winter_sports
```

Out[131]:
```
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)
```

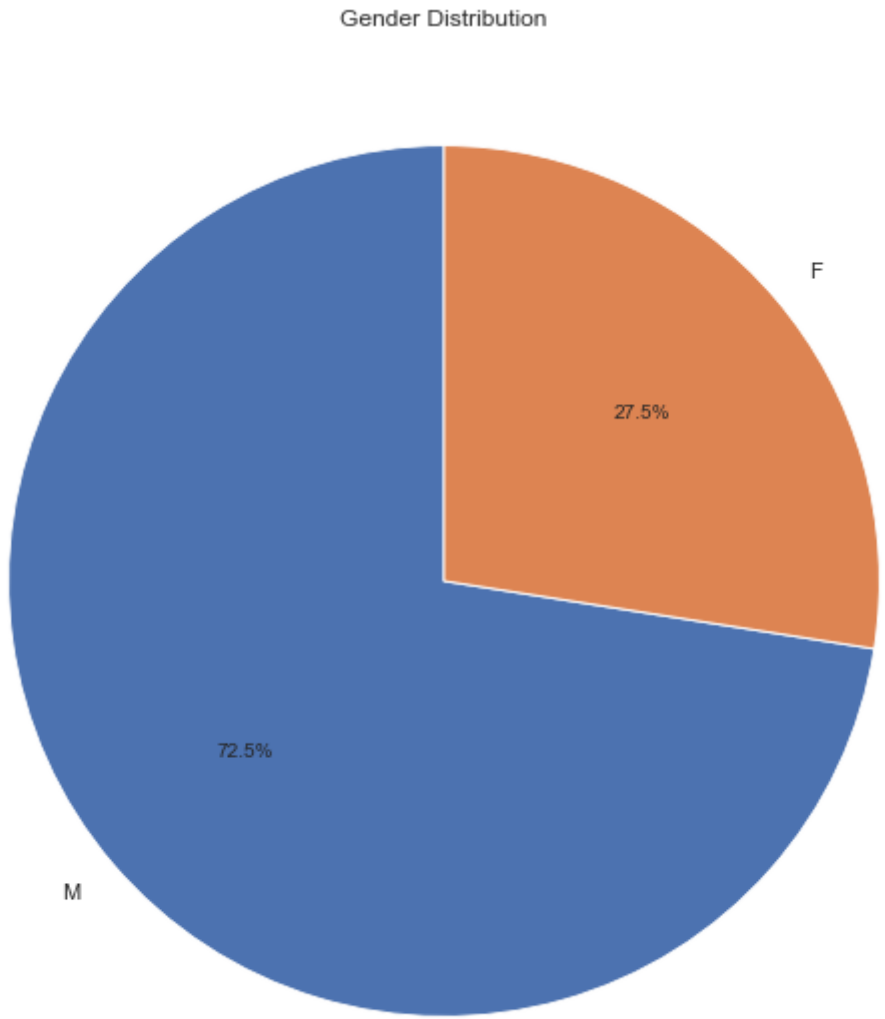
In [132]:
```
# Summer Sports
summer_sports = athlete_df[athlete_df.Season == "Summer"].Sport.unique()
summer_sports
```

Out[132]:
```
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 [133]:
```
# Male and Female participants
gender_count = athlete_df.Sex.value_counts()
gender_count
```

Out[133]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [134]:
```
plt.figure(figsize = (20,10))
plt.title("Gender Distribution")
plt.pie(gender_count, labels = gender_count.index, autopct = "%1.1f%%", startangle =90);
```



In [135]:
```
# Total medals
athlete_df.Medal.value_counts()
```

Out[135]:
```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

In [136]:
```
# Total number of female athletes in each olympics
female_parti = athlete_df[(athlete_df.Sex == "F") & (athlete_df.Season == "Summer")][["Sex", "Year"]]
female_parti = female_parti.groupby("Year").count().reset_index()
female_parti.tail()
```

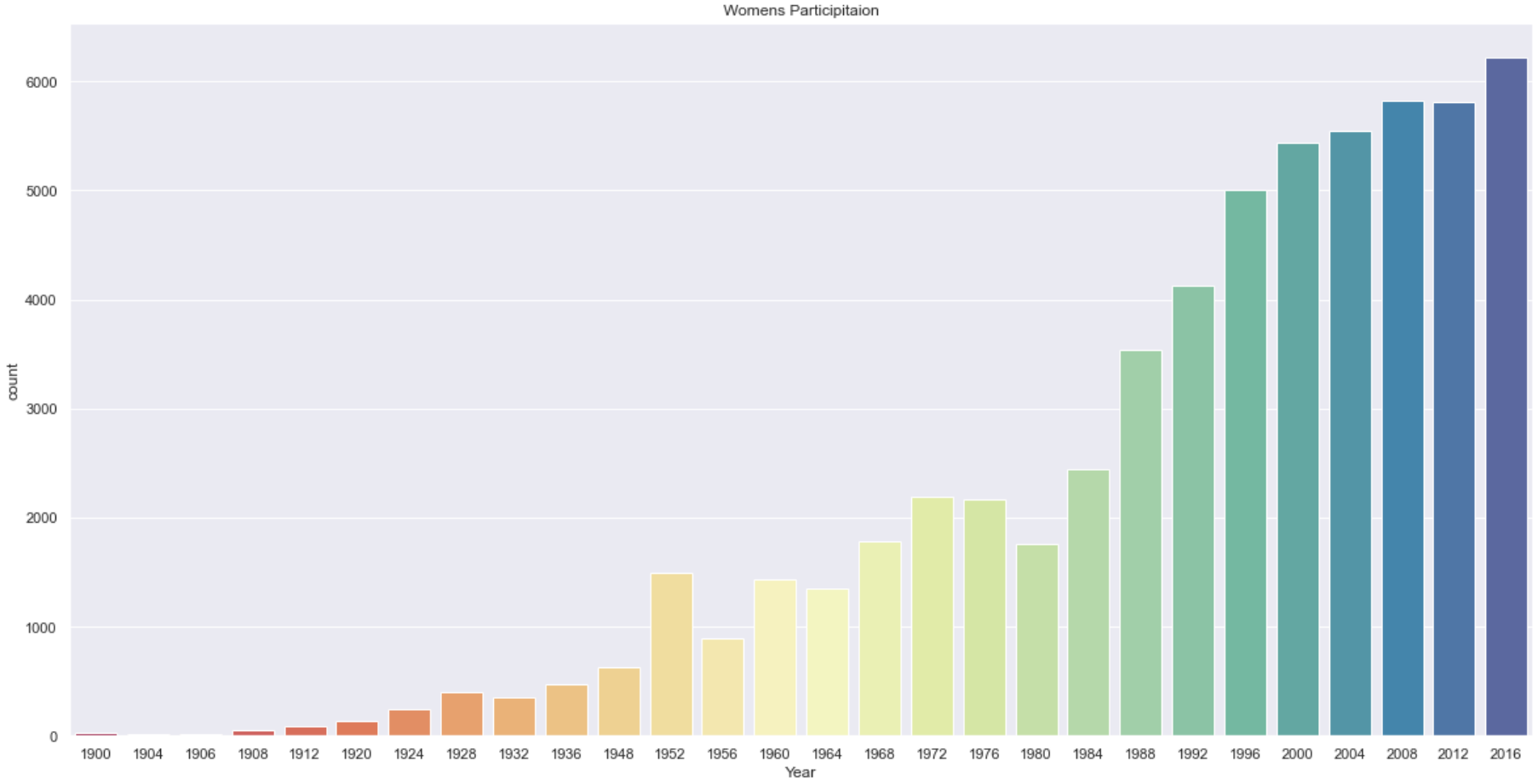
Out[136]:

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

In [137]:
```
women_oly = athlete_df[(athlete_df.Sex=="F") & (athlete_df.Season =="Summer")]
```

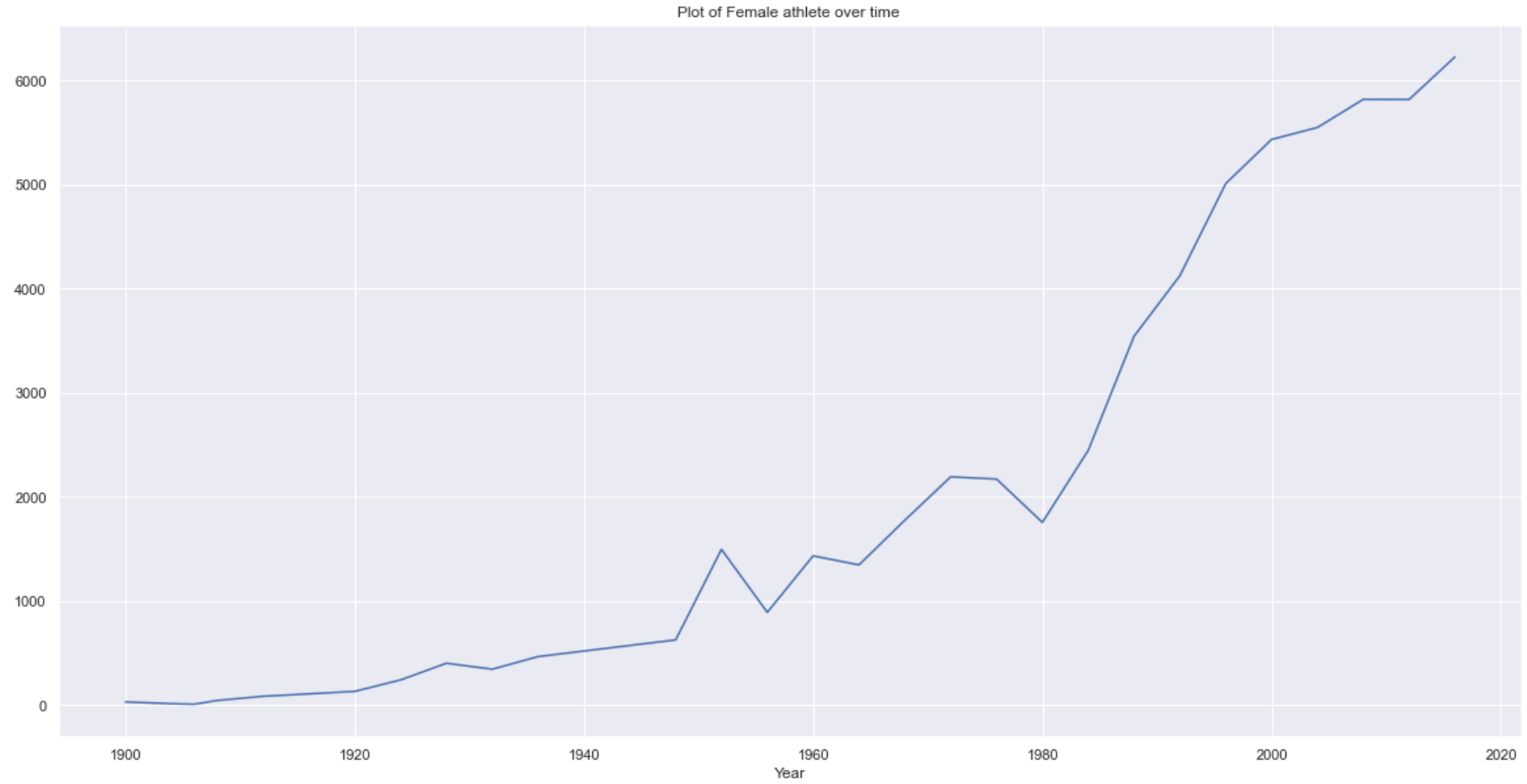
In [138]:
```
sns.set(style = "darkgrid")
plt.figure(figsize=(20,10))
sns.countplot(x="Year", data=women_oly, palette = "Spectral")
plt.title("Womens Participitaion")
```

Out[138]: Text(0.5, 1.0, 'Womens Participitaion')



In [139]:
```
part = women_oly.groupby("Year")["Sex"].value_counts()
plt.figure(figsize=(20,10))
part.loc[:,"F"].plot()
plt.title("Plot of Female athlete over time ")
```

Out[139]: Text(0.5, 1.0, 'Plot of Female athlete over time ')



In [140]:
```
# Gold medal athletes
goldmedals = athlete_df[(athlete_df.Medal=="Gold")]
goldmedals
```

Out[140]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Meda
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	Gol
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	Gol
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	Gol
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	Gol
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	Gol
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
270981	135503	Zurab Zviadauri	M	23.0	182.0	90.0	Georgia	GEO	2004 Summer	2004	Summer	Athina	Judo	Judo Men's Middleweight	Gol
271009	135520	Julia Zwehl	F	28.0	167.0	60.0	Germany	GER	2004 Summer	2004	Summer	Athina	Hockey	Hockey Women's Hockey	Gol
271016	135523	Ronald Ferdinand "Ron" Zwerver	M	29.0	200.0	93.0	Netherlands	NED	1996 Summer	1996	Summer	Atlanta	Volleyball	Volleyball Men's Volleyball	Gol
271049	135545	Henk Jan Zwolle	M	31.0	197.0	93.0	Netherlands	NED	1996 Summer	1996	Summer	Atlanta	Rowing	Rowing Men's Coxed Eights	Gol
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21.0	168.0	80.0	Soviet Union	URS	1952 Summer	1952	Summer	Helsinki	Athletics	Athletics Women's Shot Put	Gol

13372 rows × 17 columns

In [141]:
```
# Take only the values that are different from NaN
goldmedals = goldmedals[np.isfinite(goldmedals["Age"])]
```

In [142]:
```
# Gold medals beyond the age of 60 years
goldmedals["ID"][goldmedals["Age"]>60].count()
```

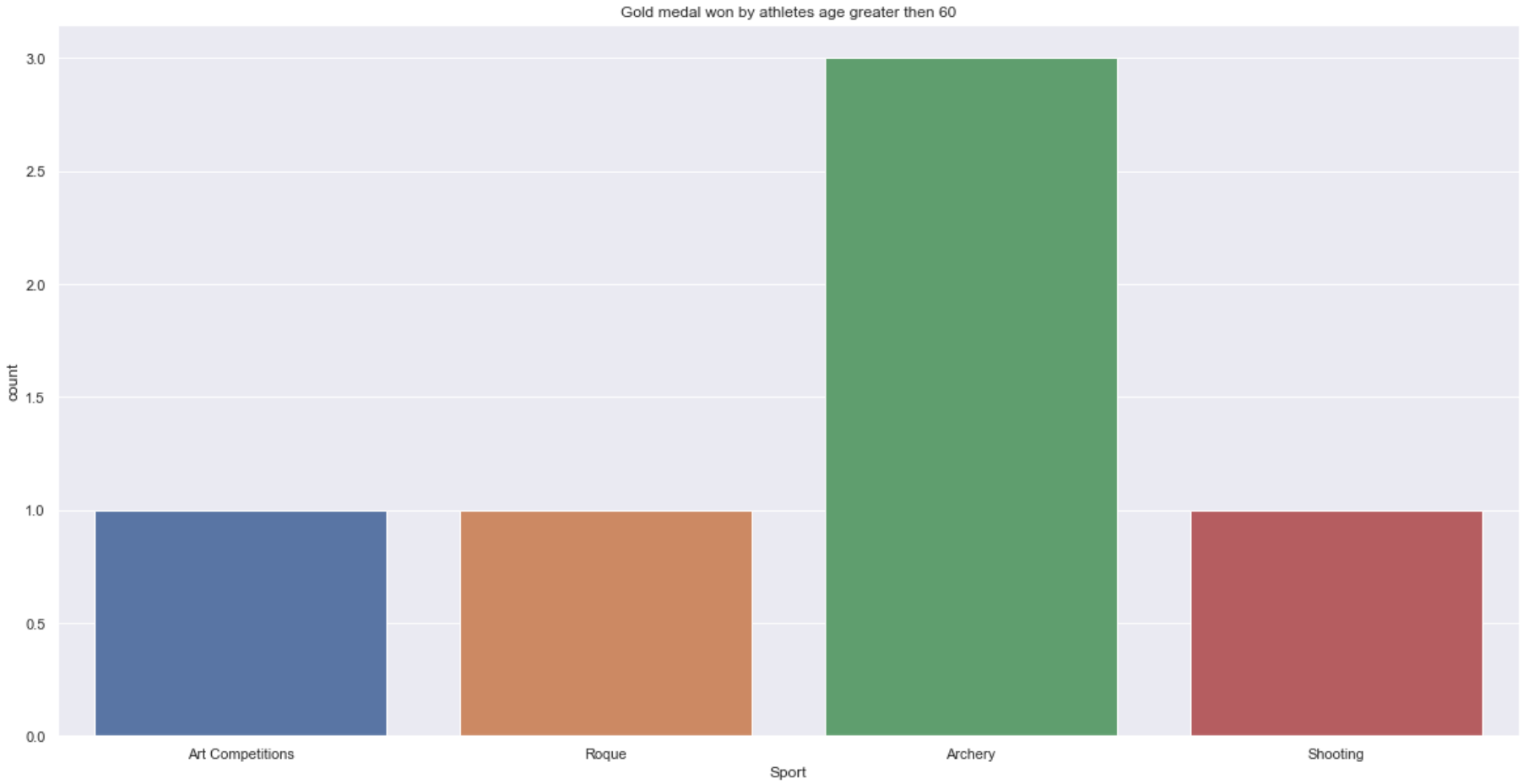
Out[142]:
```
6
```

In [143]:
```
sport_event = goldmedals["Sport"][goldmedals["Age"]>60]
sport_event
```

Out[143]:
```
104003    Art Competitions
105199               Roque
190952             Archery
226374             Archery
233390            Shooting
261102             Archery
Name: Sport, dtype: object
```

In [144]:
```
# Plot fro sport event
plt.figure(figsize = (20,10))
plt.tight_layout()
sns.countplot(sport_event)
plt.title("Gold medal won by athletes age greater then 60")
```

Out[144]:
```
Text(0.5, 1.0, 'Gold medal won by athletes age greater then 60')
```



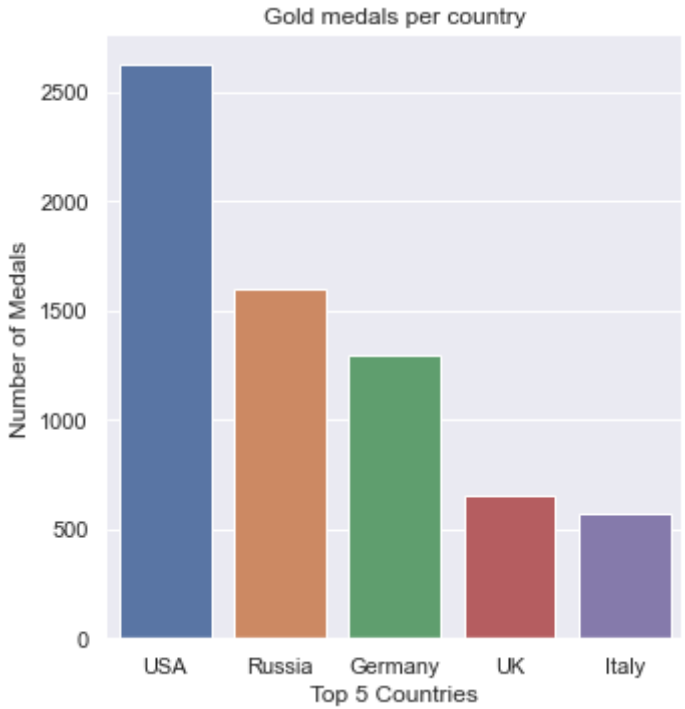
In [145]:
```
# Gold medals from each country
goldmedals.region.value_counts().reset_index(name="Medal").head()
```

Out[145]:

	index	Medal
0	USA	2627
1	Russia	1599
2	Germany	1293
3	UK	657
4	Italy	567

In [146]:
```
total_gold_medals = goldmedals.region.value_counts().reset_index(name="Medal").head()
g = sns.catplot(x="index", y="Medal", data = total_gold_medals,
                height = 5 , kind = "bar")
g.despine(left = True)
g.set_xlabels("Top 5 Countries")
g.set_ylabels("Number of Medals")
plt.title("Gold medals per country")
```

Out[146]: Text(0.5, 1.0, 'Gold medals per country')



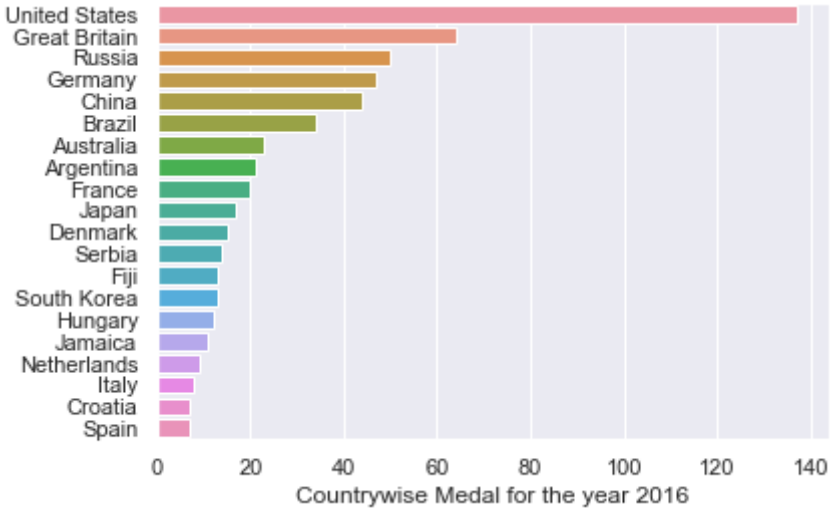
In [147]:
```
# Rio Olympics
max_year = athlete_df.Year.max()
print(max_year)

team_names = athlete_df[(athlete_df.Year == max_year) & (athlete_df.Medal=="Gold")].Team
team_names.value_counts().head(10)
```

```
2016
```

Out[147]:
```
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 [148]:
```
sns.barplot(x= team_names.value_counts().head(20),
            y= team_names.value_counts().head(20).index)
plt.ylabel(None);
plt.xlabel("Countrywise Medal for the year 2016");
```



In [149]:
```
not_null_medal = athlete_df[(athlete_df["Height"].notnull()) & (athlete_df["Weight"].notnull())]
```

In [150]:
```
plt.figure(figsize = (20,10))
axis = sns.scatterplot(x= "Height",
                       y= "Weight",
                       data = not_null_medal,
                       hue = "Sex")
plt.title("Height vs weight of olympics medalist")
```

Out[150]: Text(0.5, 1.0, 'Height vs weight of olympics medalist')