In [1]:
```
# import library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
# read dataset
athlete = pd.read_csv("athlete_events.csv")
region = pd.read_csv("noc_regions.csv")
athlete.head()
```

Out[2]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Londor
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerper
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary

In [3]:
```
region.head()
```

Out[3]:

	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 [4]:
```
# Join the dataframes

athlete_df = athlete.merge(region, how="left", on="NOC")
athlete_df.head()
```

Out[4]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Londor
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerper
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary

In [5]:
```
athlete_df.rename(columns={"region":"Region", "notes":"Notes"}, inplace=True)
```

In [6]:
```
athlete_df.head()
```

Out[6]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Londor
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerper
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary

In [7]:
```
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 [8]:
```
athlete_df.describe()
```

Out[8]:

	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 [9]:
```
# Check null values
athlete_df.isna().sum()
```

Out[9]:
```
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 [10]:
```
# Vietnam details
```

```
athlete_df.query('Team == "Vietnam"').head()
```

Out[10]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
34919	17997	Cao Ngc Phng Trnh	F	23.0	NaN	NaN	Vietnam	VIE	1996 Summer	1996	Summer	Atlanta
41420	21364	Chu Hong Diu Linh	F	18.0	174.0	66.0	Vietnam	VIE	2012 Summer	2012	Summer	London
41550	21433	Chung Th Thanh Lan	F	18.0	156.0	43.0	Vietnam	VIE	1980 Summer	1980	Summer	Moskva
50196	25831	ng Hiu Hin	M	21.0	162.0	48.0	Vietnam	VIE	1988 Summer	1988	Summer	Seoul
50197	25832	ng Th To	F	24.0	167.0	51.0	Vietnam	VIE	1992 Summer	1992	Summer	Barcelona

In [11]:

```
# Laos details

athlete_df.query('Team == "Laos"').head()
```

Out[11]:

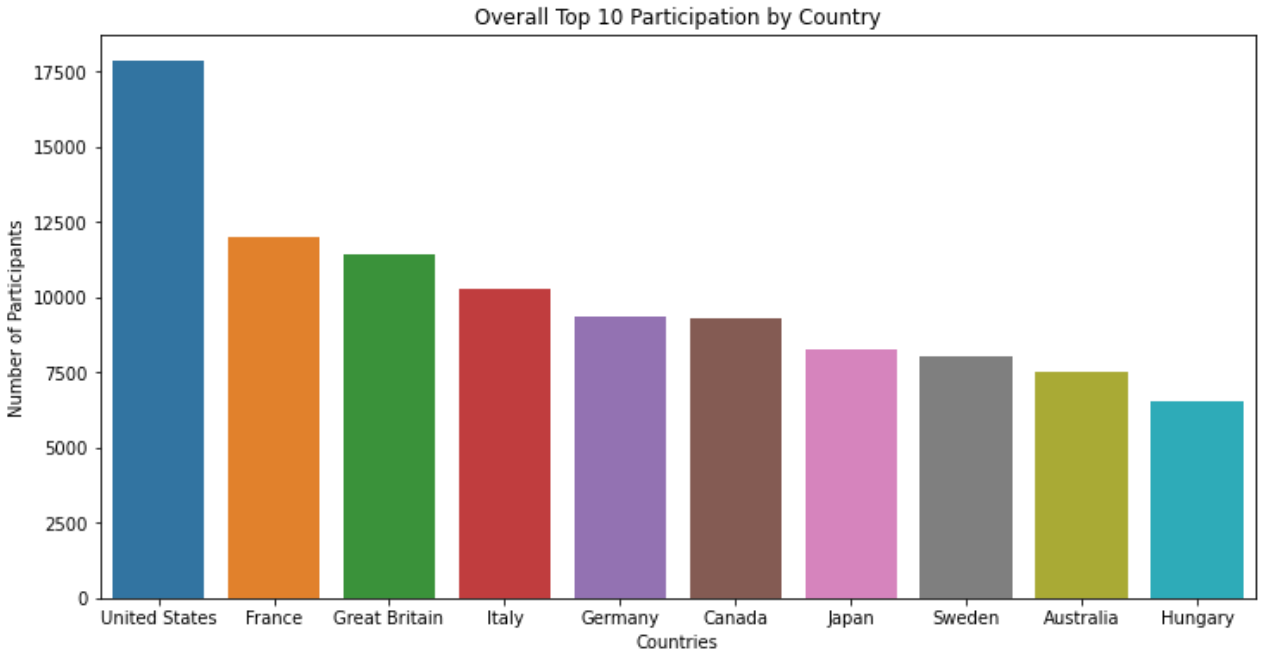
	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
6198	3482	Thongdy Amnouayphone	M	19.0	165.0	60.0	Laos	LAO	1996 Summer	1996	Summer	Atlant
8426	4642	Chamleunesouk Ao-Oudomphonh	M	25.0	168.0	51.0	Laos	LAO	2004 Summer	2004	Summer	Athin
27036	14070	Boualong Boungnavong	F	21.0	155.0	50.0	Laos	LAO	1980 Summer	1980	Summer	Moskv
27037	14071	Khamseua Bounheuang	M	26.0	170.0	63.0	Laos	LAO	1980 Summer	1980	Summer	Moskv
31386	16137	Siri Arun Budcharern	F	14.0	166.0	63.0	Laos	LAO	2016 Summer	2016	Summer	Rio d Janeir

In [12]:
```
# Top countries participating
top_10_countries = athlete_df.Team.value_counts().sort_values(ascending=False).head(10)
top_10_countries
```

Out[12]:
```
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
```

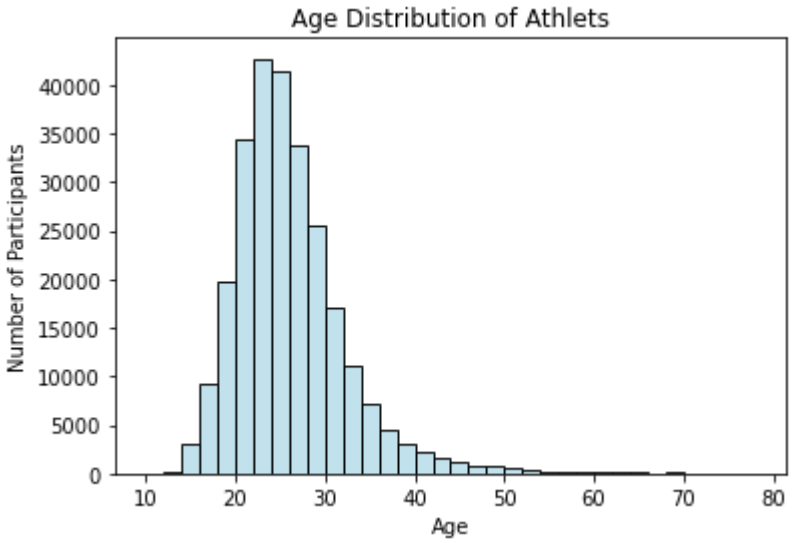
In [13]:
```
# Plot for top 10 countries participants
plt.figure(figsize=(12,6))
sns.barplot(x=top_10_countries.index, y=top_10_countries)
plt.title("Overall Top 10 Participation by Country")
plt.ylabel("Number of Participants")
plt.xlabel("Countries")
```

Out[13]:
```
Text(0.5, 0, 'Countries')
```



In [14]:
```
# Age distribution of participants

sns.histplot(data=athlete_df.Age, bins=np.arange(10, 80, 2), color="lightblue")
plt.title("Age Distribution of Athlets")
plt.ylabel("Number of Participants")
```

Out[14]:
```
Text(0, 0.5, 'Number of Participants')
```



In [15]:
```
# List winter sports

winter_sports = athlete_df[athlete.Season == "Winter"].Sport.unique()
winter_sports
```

Out[15]:
```
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 [16]:
```
# List summer sports

summer_sports = athlete_df[athlete.Season == "Summer"].Sport.unique()
summer_sports
```

Out[16]:
```
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)
```

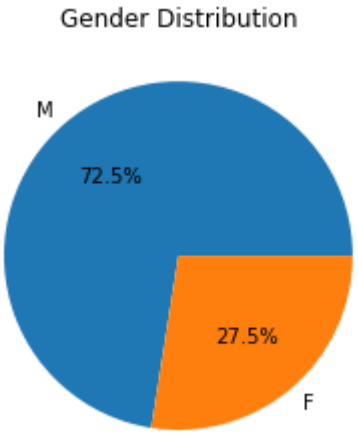
In [17]:
```
# Male and Female Participants

gender_count = athlete_df.Sex.value_counts()
gender_count
```

Out[17]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [18]:
```
# Pie plot for Male and Female Participants
plt.title("Gender Distribution")
plt.pie(gender_count, labels=gender_count.index, autopct="%.1f%%")
```

Out[18]:
```
([<matplotlib.patches.Wedge at 0x16a6fb5bdf0>,
```

```
 <matplotlib.patches.Wedge at 0x16a6fb6a4f0>],
[Text(-0.7147310163003325, 0.8361576252945936, 'M'),
 Text(0.7147309380136029, -0.8361576922125369, 'F')],
[Text(-0.38985328161836313, 0.4560859774334146, '72.5%'),
 Text(0.38985323891651064, -0.45608601393411097, '27.5%')])
```



In [19]:
```
# Total medals

athlete_medals = athlete_df.Medal.value_counts()
athlete_medals
```

Out[19]:
```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

In [20]:
```
# Total of Female Athletes in each Olympic

female_participants = athlete_df.query("Sex == 'F'")[["Sex","Year", "Season"]]
female_participants = female_participants.groupby(["Year", "Season"]).count().reset_ind
female_participants
```

Out[20]:

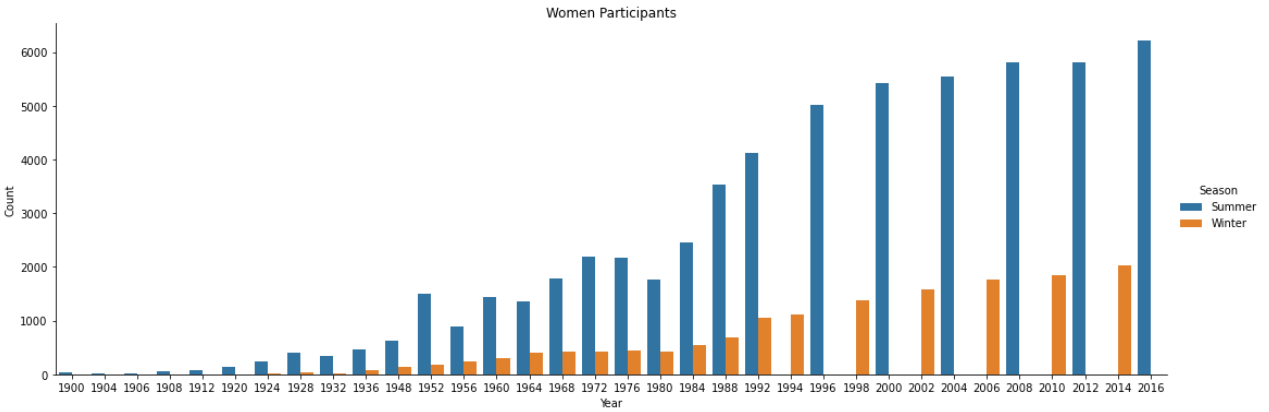
	Year	Season	Sex
0	1900	Summer	33
1	1904	Summer	16
2	1906	Summer	11
3	1908	Summer	47
4	1912	Summer	87
5	1920	Summer	134
6	1924	Summer	244
7	1924	Winter	17
8	1928	Summer	404
9	1928	Winter	33
10	1932	Summer	347
11	1932	Winter	22

	Year	Season	Sex
12	1936	Summer	468
13	1936	Winter	81
14	1948	Summer	628
15	1948	Winter	133
16	1952	Summer	1497
17	1952	Winter	185
18	1956	Summer	893
19	1956	Winter	246
20	1960	Summer	1435
21	1960	Winter	295
22	1964	Summer	1348
23	1964	Winter	404
24	1968	Summer	1777
25	1968	Winter	416
26	1972	Summer	2193
27	1972	Winter	415
28	1976	Summer	2172
29	1976	Winter	434
30	1980	Summer	1756
31	1980	Winter	430
32	1984	Summer	2447
33	1984	Winter	536
34	1988	Summer	3543
35	1988	Winter	680
36	1992	Summer	4124
37	1992	Winter	1054
38	1994	Winter	1105
39	1996	Summer	5008
40	1998	Winter	1384
41	2000	Summer	5431
42	2002	Winter	1582
43	2004	Summer	5546
44	2006	Winter	1757

	Year	Season	Sex
45	2008	Summer	5816
46	2010	Winter	1847
47	2012	Summer	5815
48	2014	Winter	2023
49	2016	Summer	6223

In [21]:
```
sns.catplot(x="Year", y="Sex", data=female_participants, kind="bar", hue="Season", heig
plt.title("Women Participants")
plt.ylabel("Count")
```

Out[21]:
```
Text(9.932233796296302, 0.5, 'Count')
```



In [22]:
```
# Gold medal athletes

goldMedals = athlete_df[(athlete_df.Medal == "Gold")]
goldMedals
```

Out[22]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seasor
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summe
42	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summe
44	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summe
48	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summe
60	20	Kjetil Andr Aamodt	M	20.0	176.0	85.0	Norway	NOR	1992 Winter	1992	Winte
...	...	...	...	...	...	...	...	...	...	...	.

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seasor
270981	135503	Zurab Zviadauri	M	23.0	182.0	90.0	Georgia	GEO	2004 Summer	2004	Summe
271009	135520	Julia Zwehl	F	28.0	167.0	60.0	Germany	GER	2004 Summer	2004	Summe
271016	135523	Ronald Ferdinand "Ron" Zwerver	M	29.0	200.0	93.0	Netherlands	NED	1996 Summer	1996	Summe
271049	135545	Henk Jan Zwolle	M	31.0	197.0	93.0	Netherlands	NED	1996 Summer	1996	Summe
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21.0	168.0	80.0	Soviet Union	URS	1952 Summer	1952	Summe

13372 rows × 17 columns

In [23]:
```
# Gold medal beyond 60
goldMedals['ID'][goldMedals['Age'] > 60].count()
```

Out[23]: 6

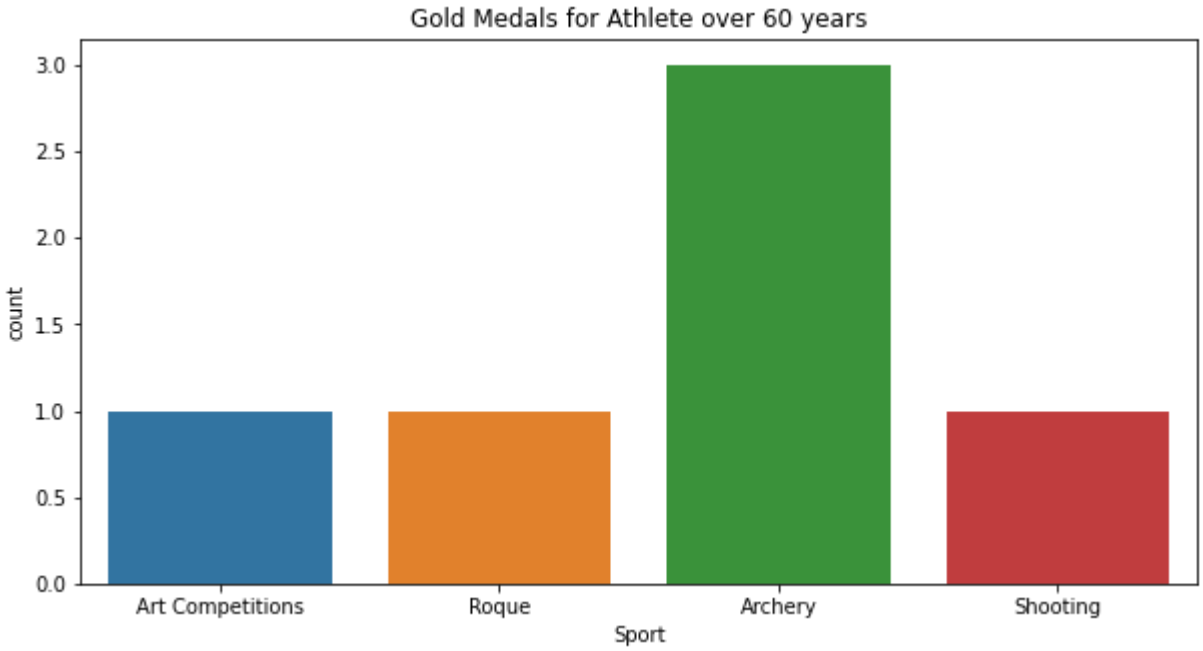
In [24]:
```
sporting_event = goldMedals["Sport"][goldMedals["Age"] > 60]
sporting_event
```

Out[24]:
```
104003    Art Competitions
105199               Roque
190952             Archery
226374             Archery
233390            Shooting
261102             Archery
Name: Sport, dtype: object
```

In [25]:
```
# Plot for sporting_event

plt.figure(figsize=(10,5))
sns.countplot(sporting_event)
plt.title("Gold Medals for Athlete over 60 years")
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an explicit keyword will re
sult in an error or misinterpretation.
  warnings.warn(
```

Out[25]: Text(0.5, 1.0, 'Gold Medals for Athlete over 60 years')



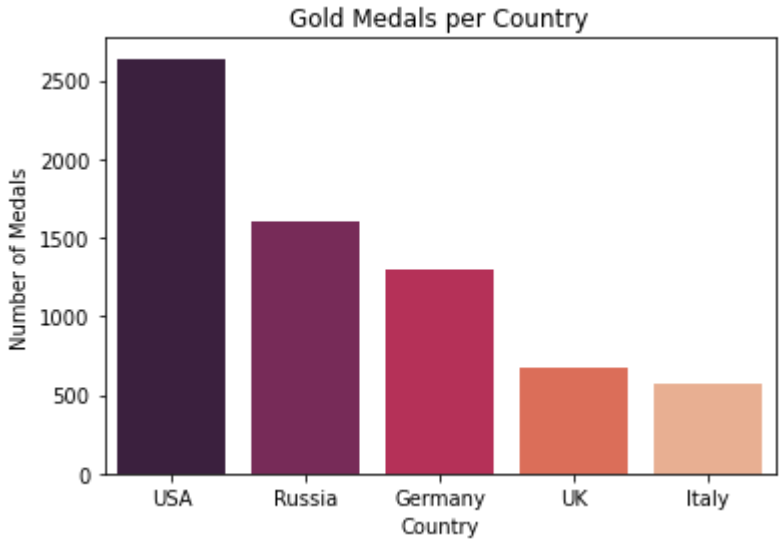
In [26]:
```
# Gold medals from each country

goldMedalsCountry = goldMedals.Region.value_counts().reset_index(name="Medal_num").head
goldMedalsCountry
```

Out[26]:

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

In [27]:
```
# plot top 5 gold medal countries

sns.barplot(x="index", y = "Medal_num", data = goldMedalsCountry, palette="rocket")
plt.title("Gold Medals per Country")
plt.xlabel("Country")
plt.ylabel("Number of Medals")
```

Out[27]:
```
Text(0, 0.5, 'Number of Medals')
```

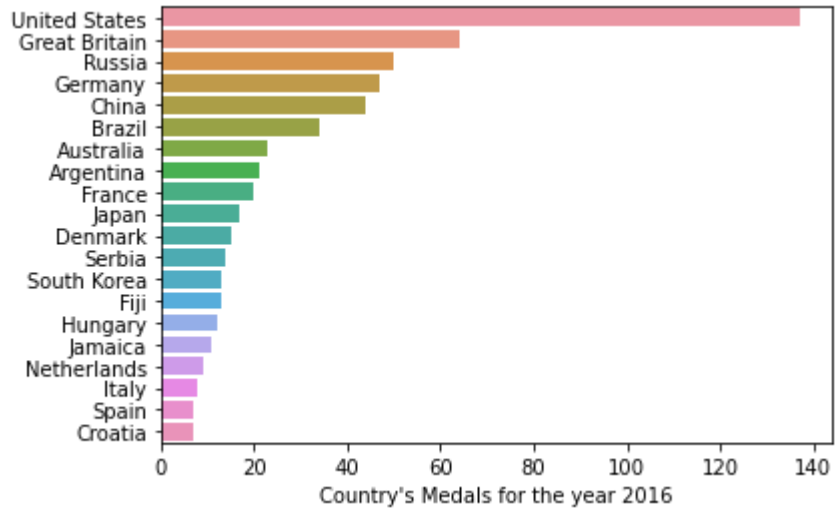


In [28]:
```
max_year = athlete_df.Year.max()

team_names = goldMedals[goldMedals.Year == max_year].Team
team_names.value_counts().head(10)
```

Out[28]:
```
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 [29]:
```
sns.barplot(x=team_names.value_counts().head(20), y=team_names.value_counts().head(20).
plt.xlabel("Country's Medals for the year 2016")
```

Out[29]:
```
Text(0.5, 0, "Country's Medals for the year 2016")
```



In [30]:
```
not_null_medals = athlete_df[(athlete_df.Height.notnull()) & (athlete_df.Weight.notnull
```

In [33]:
```
sns.scatterplot(x="Height", y="Weight", data=not_null_medals, hue="Sex")
```

```
plt.title("Height vs Weight of Olympic Medalists")
```

Out[33]: Text(0.5, 1.0, 'Height vs Weight of Olympic Medalists')