In [1]:
```
# 120 years of Olympics data analysis using python contains
# Country Demographics
# Age Demographics of Athletes
# Gender Demographics of Athletes, specially Women participation over the year
# Medal Demographics


import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [5]:
```
athletes = pd.read_csv('Z:\Datasets\Olympics History/athlete_events.csv')
regions = pd.read_csv('Z:\Datasets\Olympics History/noc_regions.csv')
```

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

Out[6]:

	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 [7]:
```
athletes.head()
```

Out[7]:

	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	London
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen
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 [8]:
```
#Merge 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
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 [9]:
```
athletes_df.shape
```

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

In [10]:
```
athletes_df.rename(columns={'region':'Region', 'notes': 'Notes'}, inplace=True)
athletes_df.head()
```

Out[10]:

	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

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary

In [11]:
```
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 [12]:
```
athletes_df.describe()
```

Out[12]:

	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 [13]:
```
#Null Values
nan_values = athletes_df.isna()
nan_columns = nan_values.any()
nan_columns
```

Out[13]:
```
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 [14]:
```
athletes_df.isnull().sum()
```

Out[14]:
```
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 [15]:
```
nan_values = athletes_df.isna()
nan_columns = nan_values.any()

columns_with_nan = athletes_df.columns[nan_columns].tolist()
print(columns_with_nan)
```

```
['Age', 'Height', 'Weight', 'Medal', 'Region', 'Notes']
```

In [16]:
```
#India
athletes_df.query('Team == "India"').head(5)
```

Out[16]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Ath

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Ath
895	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Ath
896	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Ath
897	512	Shiny Kurisingal Abraham-Wilson	F	23.0	167.0	53.0	India	IND	1988 Summer	1988	Summer	Seoul	Ath

In [17]:
```
#Japan
athletes_df.query('Team == "Japan"').head(5)
```

Out[17]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Spo
625	362	Isao Ko Abe	M	24.0	177.0	75.0	Japan	JPN	1936 Summer	1936	Summer	Berlin	Athleti
629	363	Kazumi Abe	M	28.0	178.0	67.0	Japan	JPN	1976 Winter	1976	Winter	Innsbruck	Bobsleig
630	364	Kazuo Abe	M	25.0	166.0	69.0	Japan	JPN	1960 Summer	1960	Summer	Roma	Wrestlir
631	365	Kinya Abe	M	23.0	168.0	68.0	Japan	JPN	1992 Summer	1992	Summer	Barcelona	Fencir
632	366	Kiyoshi Abe	M	25.0	167.0	62.0	Japan	JPN	1972 Summer	1972	Summer	Munich	Wrestlir

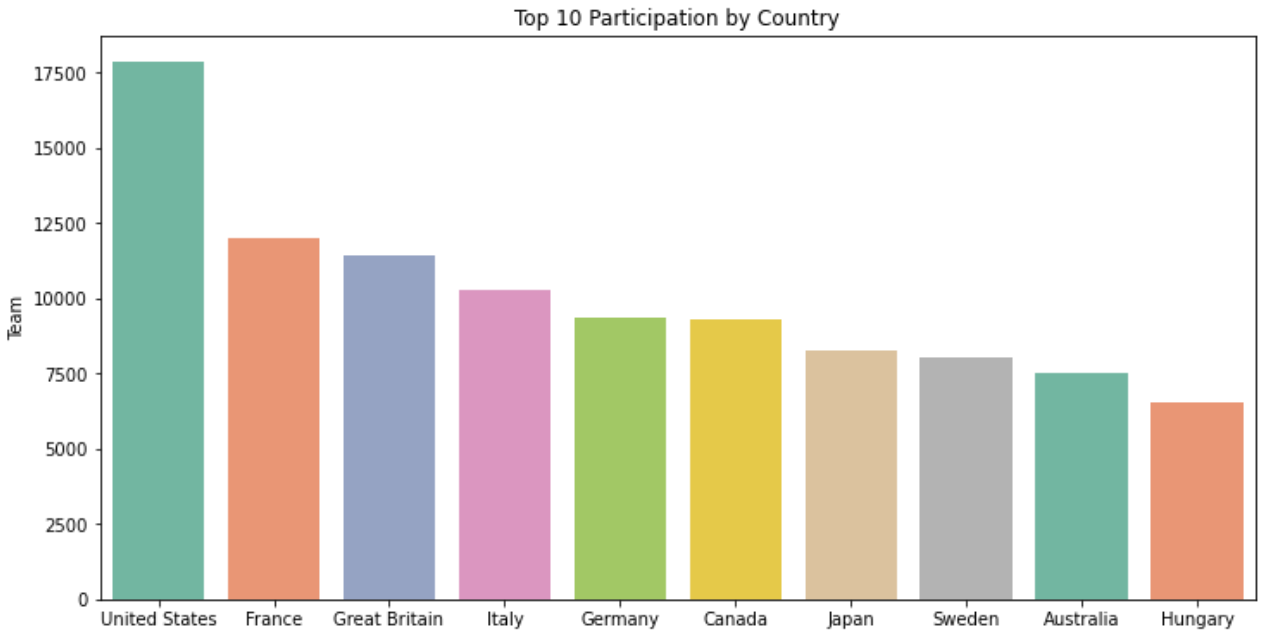
In [18]:
```
#Top Countries participating
top_10_countries = athletes_df.Team.value_counts().sort_values(ascending=False).head(10
top_10_countries
```

Out[18]:
```
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
```

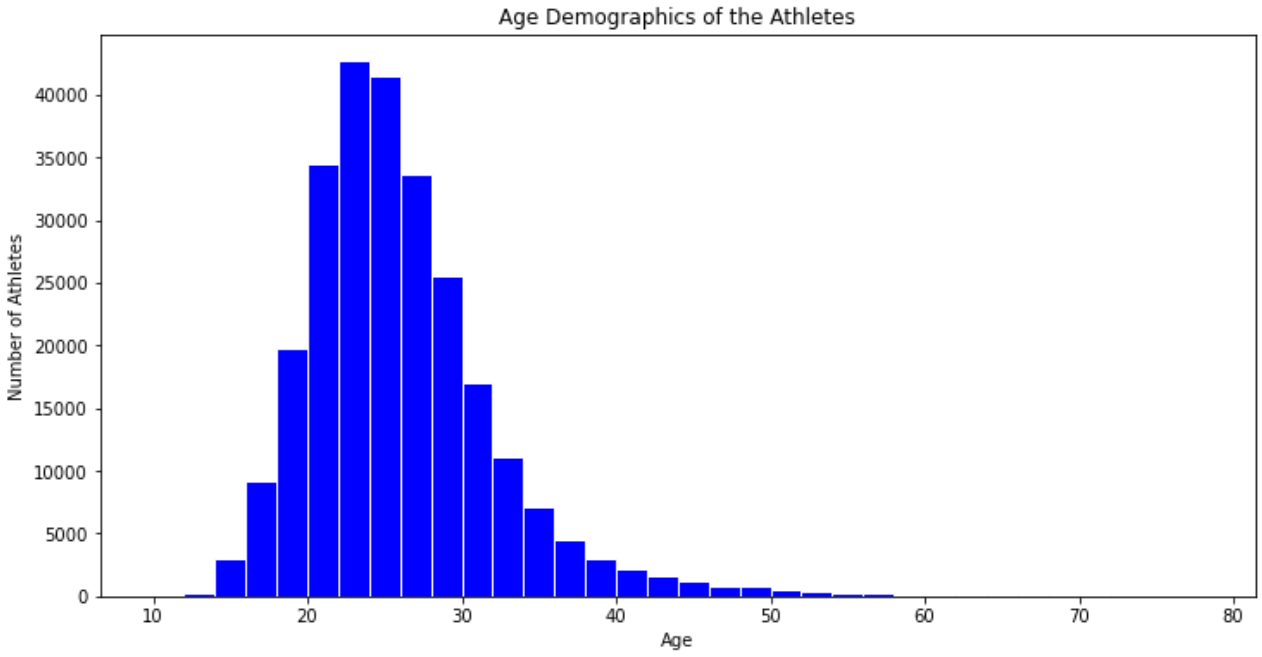
In [19]:

```
#Visualization
plt.figure(figsize=(12,6))
#plt.xticks(rotation=20)
plt.title('Top 10 Participation by Country')
sns.barplot(x=top_10_countries.index, y=top_10_countries, palette = 'Set2');
```



In [20]:

```
#Age Demographics of the Athletes
plt.figure(figsize=(12, 6))
plt.title("Age Demographics of the Athletes")
plt.xlabel('Age')
plt.ylabel('Number of Athletes')
plt.hist(athletes_df.Age, bins = np.arange(10,80,2), color='blue', edgecolor='white');
```



In [21]:
```
winter_sports = athletes_df[athletes_df.Season == 'Winter'].Sport.unique()
winter_sports
```

Out[21]:
```
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)
```

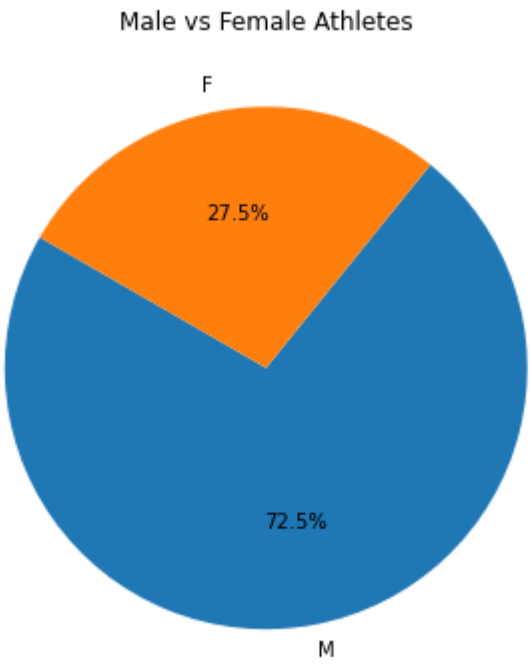
In [22]:
```
summer_sports = athletes_df[athletes_df.Season == 'Summer'].Sport.unique()
summer_sports
```

Out[22]:
```
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 [23]:
```
#Gender Demographics
gender_counts = athletes_df.Sex.value_counts()
gender_counts
```

Out[23]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [24]:
```
plt.figure(figsize=(12,6))
plt.title('Male vs Female Athletes')
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=150);
```



In [25]:
```
#Female Demographics

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

Out[25]:

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

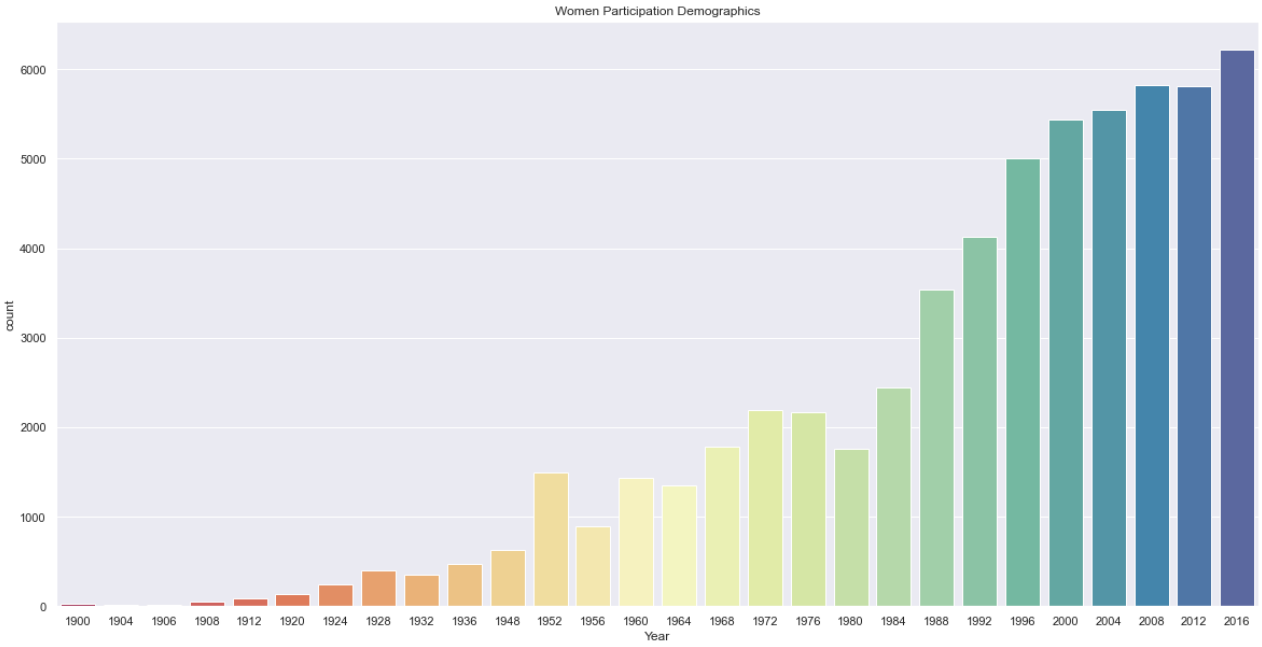
In [26]:
```
female_olympics = athletes_df[(athletes_df.Sex == 'F') & (athletes_df.Season == 'Summer
```

In [27]:
```
#Total Medals
athletes_df.Medal.value_counts()
```

Out[27]:
```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

In [28]:
```
sns.set(style="darkgrid")
plt.figure(figsize=(20,10))
sns.countplot(x='Year', data=female_olympics, palette="Spectral")
plt.title('Women Participation Demographics')
```

Out[28]: Text(0.5, 1.0, 'Women Participation Demographics')



In [29]:
```
part = female_olympics.groupby('Year')[('Sex')].value_counts()
plt.figure(figsize=(20, 10))
part.loc[:,'F'].plot()
plt.title('Women participation Over the time')
```

Out[29]: Text(0.5, 1.0, 'Women participation Over the time')

In [30]:
```
#Medal Demographics


athletes_df.Medal.value_counts()
```

Out[30]:
```
Gold      13372
Bronze    13295
```

```
Silver    13116
Name: Medal, dtype: int64
```

In [31]:
```
goldmedals = athletes_df[(athletes_df.Medal == 'Gold')]
goldmedals = goldmedals[np.isfinite(goldmedals['Age'])] #without NaN
goldmedals.head()
```

Out[31]:

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

In [32]:
```
goldmedals['ID'][goldmedals['Age'] > 60].count()
```

Out[32]: 6

In [33]:
```
sportevents = goldmedals['Sport'][goldmedals['Age'] > 30]
sportevents.count()
```

Out[33]: 2222

In [34]:
```
#Gold Medals Country Demographics

goldmedals.Region.value_counts().reset_index(name='Medal').head(6)
```

Out[34]:

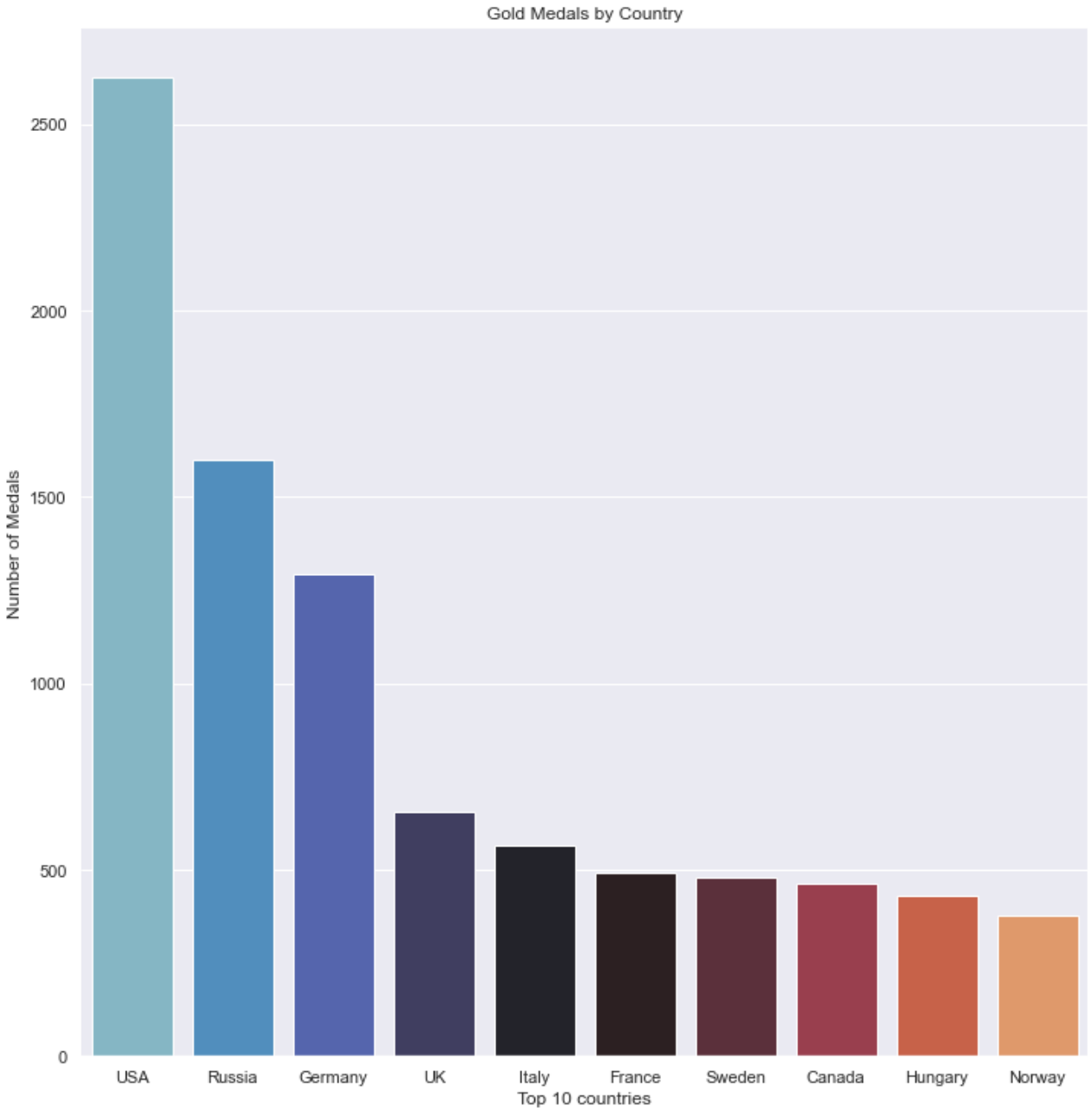
	index	Medal
0	USA	2627
1	Russia	1599
2	Germany	1293

	index	Medal
3	UK	657
4	Italy	567
5	France	491

In [44]:

```
total_goldmedals = goldmedals.Region.value_counts().reset_index(name='Medal').head(10)
g = sns.catplot(x="index", y="Medal", data=total_goldmedals, height=10, kind="bar", pal
g.despine(left=True)
g.set_xlabels("Top 10 countries")
g.set_ylabels("Number of Medals")
plt.title('Gold Medals by Country')
```

Out[44]: Text(0.5, 1.0, 'Gold Medals by Country')



In [36]:
```
goldmedals = athletes_df[(athletes_df.Medal == 'Gold')]
goldmedals = goldmedals[np.isfinite(goldmedals['Age'])] #without NaN
goldmedals.head()
```

Out[36]:

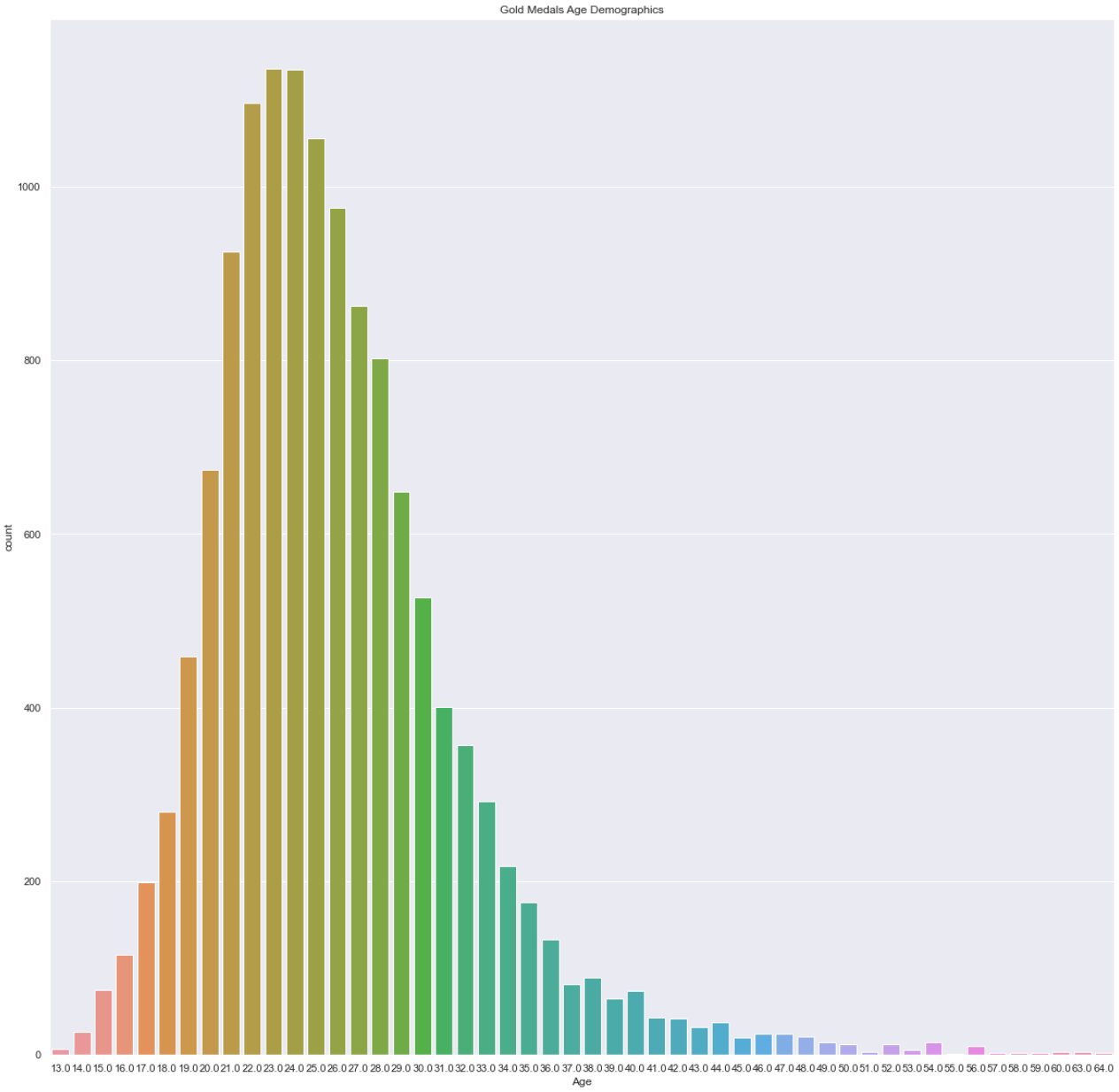
	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	Cit
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Pari
42	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948	Summer	Londo
44	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948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

In [37]:
```
goldmedals = goldmedals[np.isfinite(goldmedals['Age'])]
```

In [47]:
```
plt.figure(figsize=(20, 20))
plt.tight_layout()
sns.countplot(goldmedals['Age'])
plt.title('Gold Medals Age Demographics')
```

```
C:\Users\SRD\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[47]: Text(0.5, 1.0, 'Gold Medals Age Demographics')


Gold Medals Age Demographics
count
0
200
400
600
800
1000
13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0 32.0 33.0 34.0 35.0 36.0 37.0 38.0 39.0 40.0 41.0 42.0 43.0 44.0 45.0 46.0 47.0 48.0 49.0 50.0 51.0 52.0 53.0 54.0 55.0 56.0 57.0 58.0 59.0 60.0 63.0 64.0
Age

In [ ]: