Olympic Data Analysis

Importing the required libraries

In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

Loading the dataset

Dataset link : https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results

In [2]:

```
athletes = pd.read_csv("athlete_events.csv")
regions = pd.read_csv("noc_regions.csv")
```

In [3]:

```
athletes.head()
```

Out[3]:

	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 [4]:

```
regions.head()
```

Out[4]:

	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 [5]:

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

Out[5]:

	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 [6]:

```
athletes_df.shape
```

Out[6]:

```
(271116, 17)
```

In [7]:

```
athletes_df.rename(columns = {"region" : "Region",
                              "notes" : "Notes"}, inplace = True)
```

In [8]:

```
athletes_df.head(1)
```

Out[8]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	B B

In [9]:

```
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 [10]:

```
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]:

```
# checking null

nan_values = athletes_df.isna()
nan_columns = nan_values.any()
nan_columns
```

```
Out[11]: 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 [12]: # percentage of the null values present in their respective rows

         (athletes_df.isnull().sum().sort_values(ascending = False).head(7) / athletes_df.size) *
```

```
Out[12]: Notes      5.773023
         Medal      5.019189
         Weight     1.364187
         Height     1.305519
         Age        0.205556
         Region     0.008028
         Season     0.000000
         dtype: float64
```

Viewing India's Data

```
In [13]: athletes_df.query('Team == "India"' ).head()
```

Out[13]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sp
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Athl
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterdam	Athl
895	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Athl
896	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	Los Angeles	Athl
897	512	Shiny Kurisingal Abraham-Wilson	F	23.0	167.0	53.0	India	IND	1988 Summer	1988	Summer	Seoul	Athl

In [14]:

```
athletes_df.query('Team == "India"' ).shape
```

Out[14]:

```
(1400, 17)
```

In [15]:

```
athletes_df.query('Team == "India" & Year == 2008 & Medal == "Gold"')
```

Out[15]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Spo
22004	11601	Abhinav Bindra	M	25.0	173.0	70.0	India	IND	2008 Summer	2008	Summer	Beijing	Shooti

Top countries participating

In [16]:

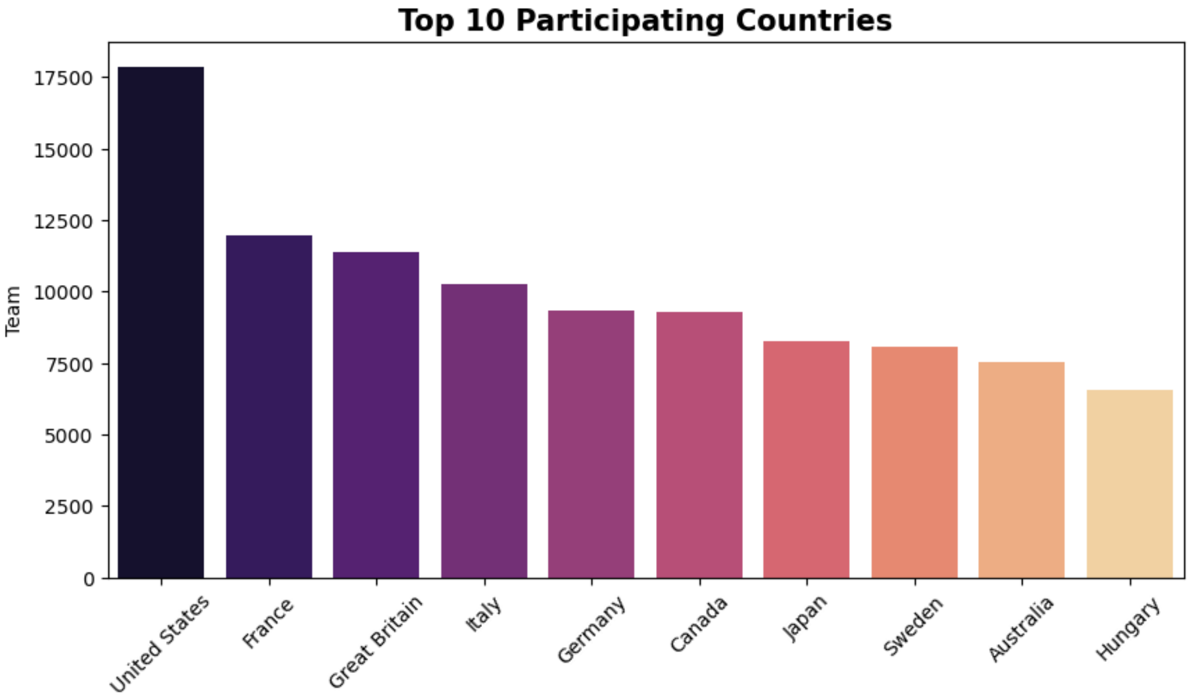
```
top_10_countries = athletes_df.Team.value_counts().sort_values(ascending = False).head(1
```

In [17]:

```
top_10_countries
```

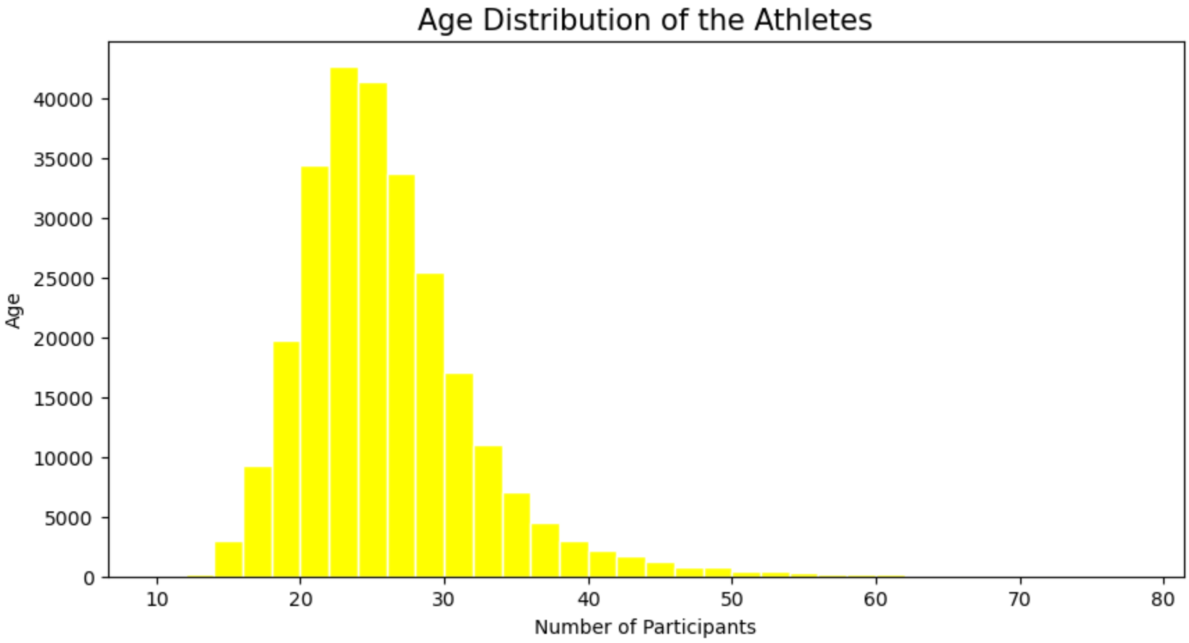
```
Out[17]: United States    17847
         France           11988
         Great Britain    11404
         Italy            10260
         Germany           9326
         Canada            9279
         Japan             8289
         Sweden            8052
         Australia         7513
         Hungary           6547
         Name: Team, dtype: int64
```

```
In [18]: plt.figure(figsize = (10,5))
         plt.title("Top 10 Participating Countries", fontsize = 15, weight = "bold")
         sns.barplot(x = top_10_countries.index, y = top_10_countries, palette = "magma")
         plt.xticks(rotation = 45)
         plt.show()
```



Age distribution of athletes

```
In [19]: plt.figure(figsize = (10,5))
         plt.title("Age Distribution of the Athletes", fontsize = 15)
         plt.xlabel("Number of Participants")
         plt.ylabel("Age")
         plt.hist(athletes_df.Age, bins = np.arange(10,80,2), color = "yellow", edgecolor = "whit
         plt.show()
```



What are the sports played in Winter and Summer Olympics upto now?

```
In [20]: winter_sports = athletes_df[athletes_df.Season == "Winter"].Sport.unique()
         winter_sports
```

```
Out[20]: 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 [21]: summer_sports = athletes_df[athletes_df.Season == "Summer"].Sport.unique()
         summer_sports
```

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

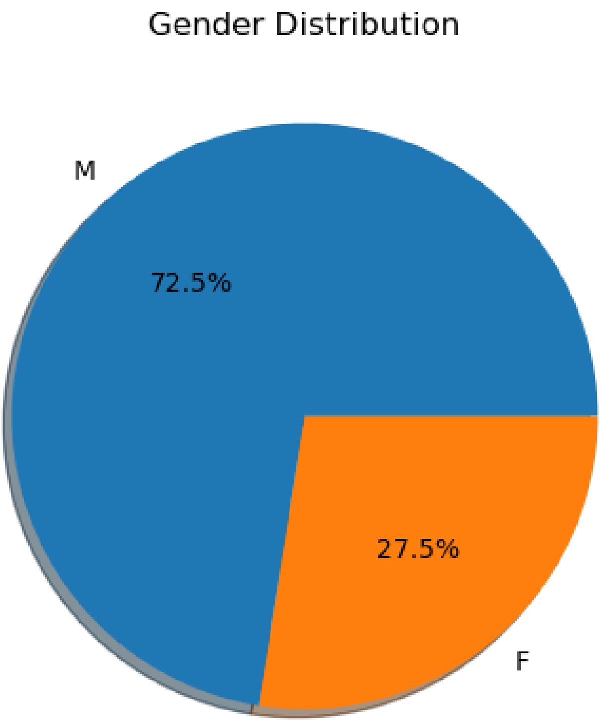
Gender distribution of the athletes

```
In [22]: gender_count = athletes_df.Sex.value_counts()
         gender_count
```

```
Out[22]: M    196594
         F     74522
         Name: Sex, dtype: int64
```

In [23]:

```
plt.figure(figsize = (8,5))
plt.title("Gender Distribution")
plt.pie(gender_count, labels = gender_count.index, autopct = "%1.1f%%", shadow = True)
plt.show()
```



Total Medals Won

In [24]:

```
athletes_df.Medal.value_counts()
```

Out[24]:

```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

Total Female Athletes in each Olympic

In [25]:

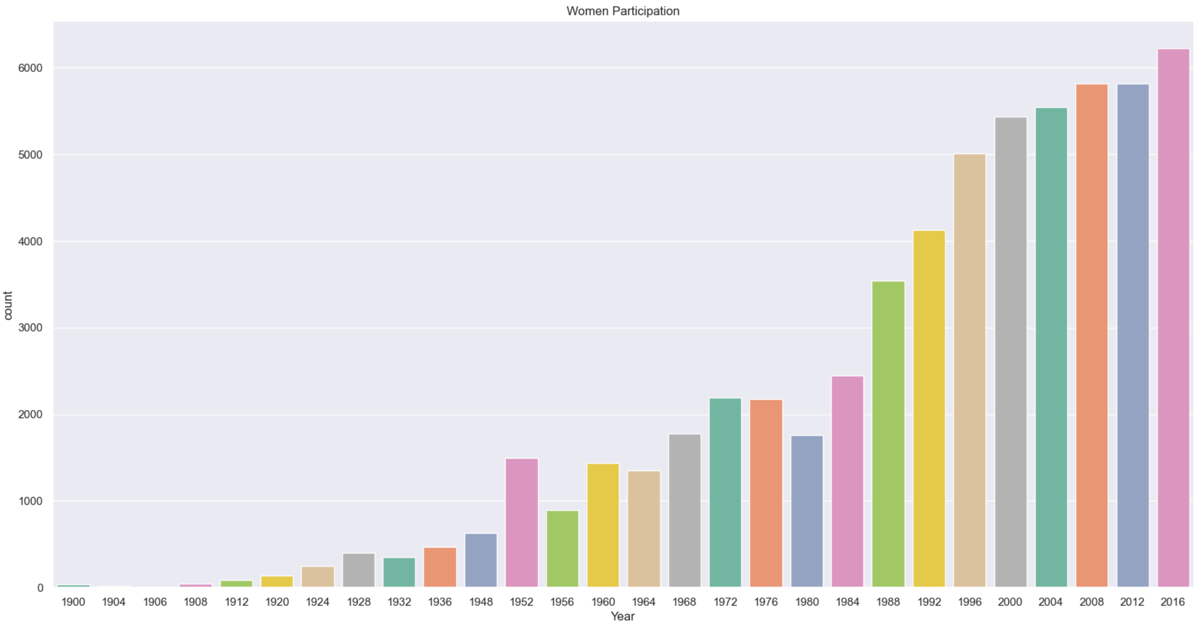
```
female_participants = athletes_df[(athletes_df.Sex == "F") & (athletes_df.Season == "Sum
female_participants = female_participants.groupby("Year").count().reset_index()
female_participants.tail()
```

Out[25]:

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

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

In [27]:
```
sns.set(style = "darkgrid")
plt.figure(figsize = (20,10))
sns.countplot(x = "Year", data = women_olympics, palette = "Set2")
plt.title("Women Participation")
plt.show()
```



Total Male Athletes in each Olympic

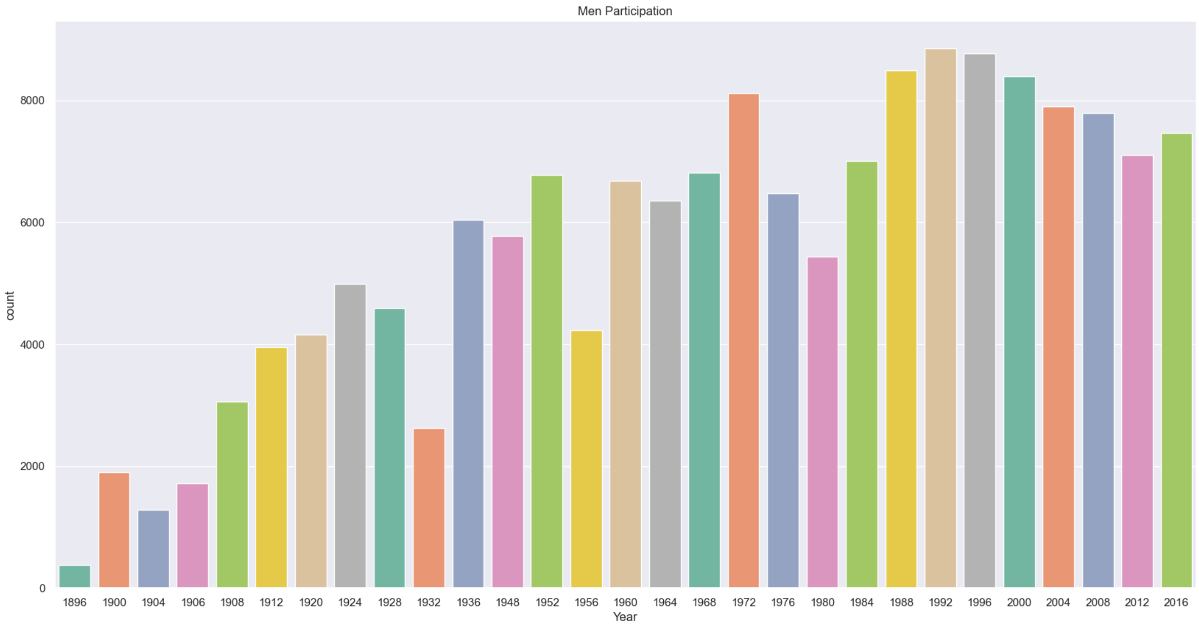
In [28]:
```
male_participants = athletes_df[(athletes_df.Sex == "M") & (athletes_df.Season == "Summe
male_participants = male_participants.groupby("Year").count().reset_index()
male_participants.tail()
```

Out[28]:

	Year	Sex
24	2000	8390
25	2004	7897
26	2008	7786
27	2012	7105
28	2016	7465

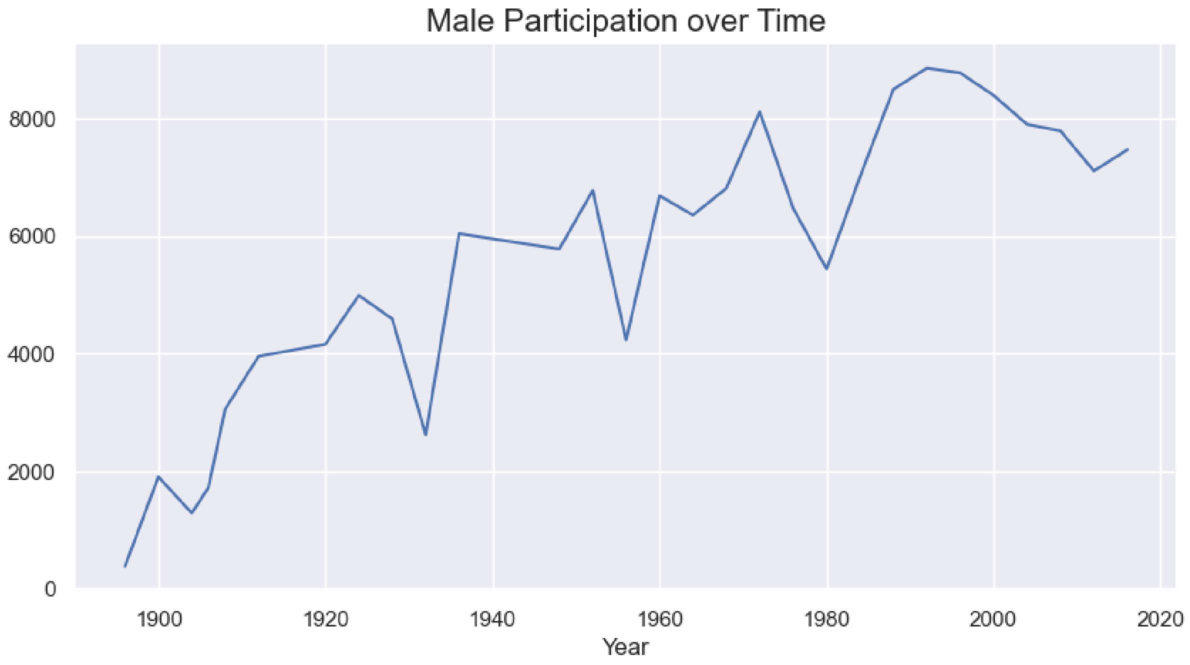
In [29]:
```
men_olympics = athletes_df[(athletes_df.Sex == "M") & (athletes_df.Season == "Summer")]
```

In [30]:
```
sns.set(style = "darkgrid")
plt.figure(figsize = (20,10))
sns.countplot(x = "Year", data = men_olympics, palette = "Set2")
plt.title("Men Participation")
plt.show()
```



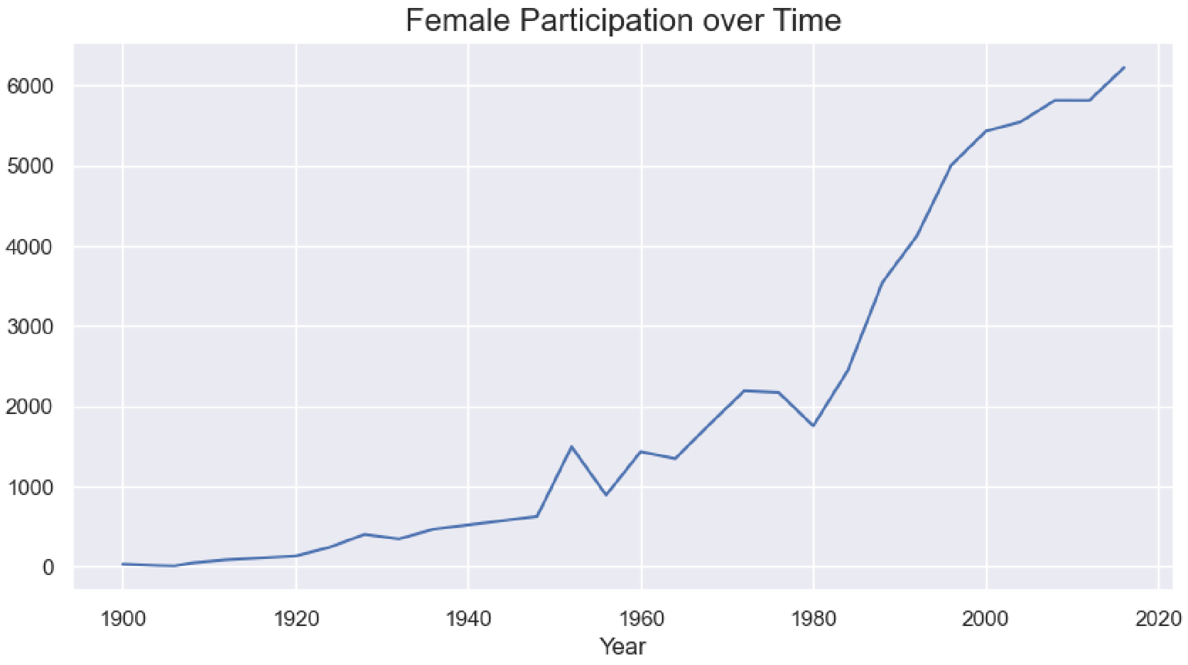
In [31]:

```
part = men_olympics.groupby('Year')['Sex'].value_counts()
plt.figure(figsize=(10,5))
part.loc[:,'M'].plot()
plt.title("Male Participation over Time", size = 16)
plt.show()
```



In [32]:

```
part = women_olympics.groupby('Year')['Sex'].value_counts()
plt.figure(figsize=(10,5))
part.loc[:,'F'].plot()
plt.title("Female Participation over Time", size = 16)
plt.show()
```



Athletes with Gold Medal

In [33]:

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

Out[33]:

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

In [34]:

```
# taking only those who are different from NaN

goldMedal = goldMedal[np.isfinite(goldMedal['Age'])]
```

Gold medal winners who are above 60 years of age

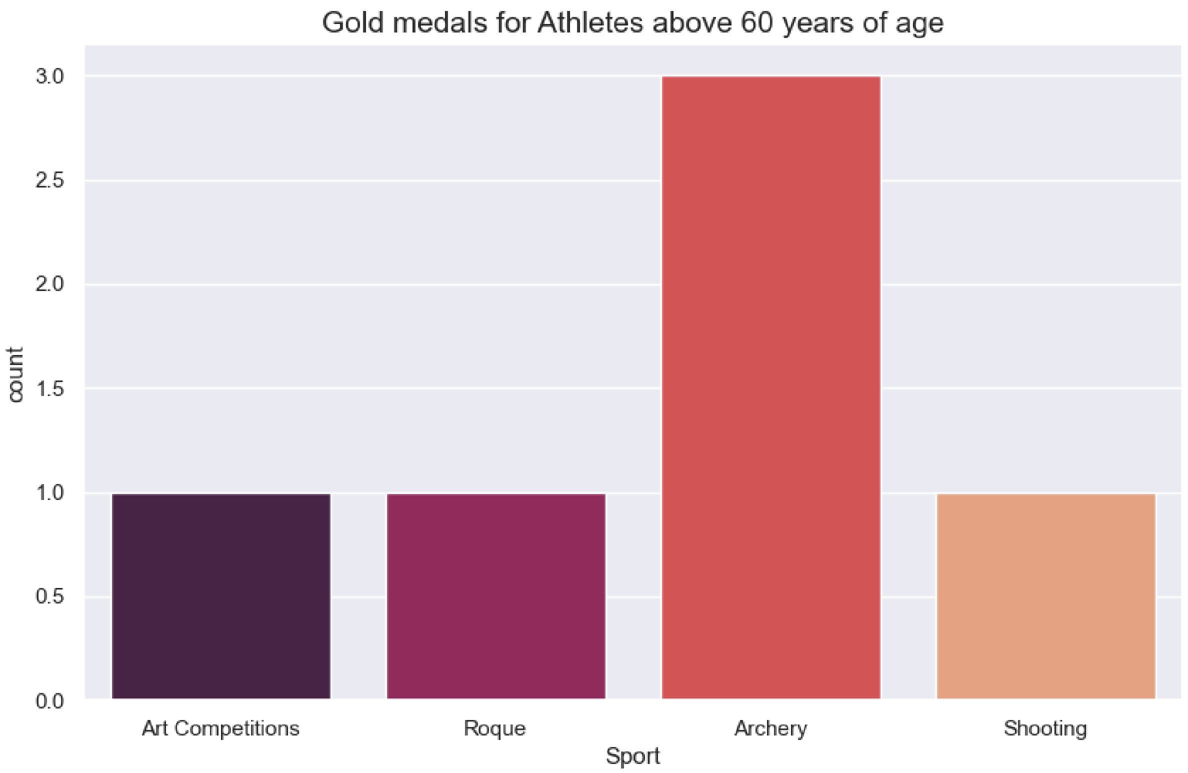
In [35]:
```
goldMedal['Name'][goldMedal['Age'] > 60]
```

Out[35]:
```
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 [36]:
```
sporting_event = goldMedal['Sport'][goldMedal['Age'] > 60]
sporting_event
```

Out[36]:
```
104003    Art Competitions
105199               Roque
190952             Archery
226374             Archery
233390            Shooting
261102             Archery
Name: Sport, dtype: object
```

In [37]:
```
plt.figure(figsize = (10,6))
plt.tight_layout()
ax = sns.countplot(x = sporting_event.index, data = sporting_event, palette = "rocket")
plt.title('Gold medals for Athletes above 60 years of age', size = 15)
plt.show()
```



Gold medal distribution country wise

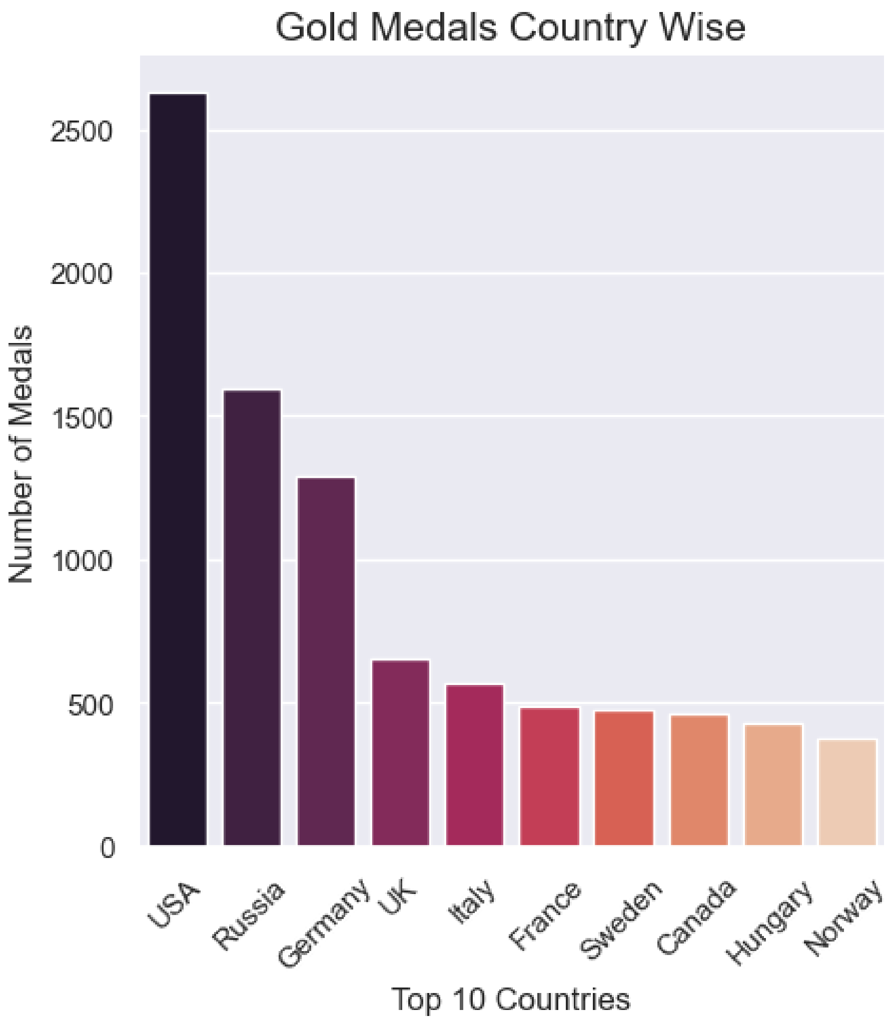
In [38]:
```
goldMedal.Region.value_counts().reset_index(name = 'Medals').head(10)
```

Out[38]:

	index	Medals
0	USA	2627
1	Russia	1599
2	Germany	1293
3	UK	657
4	Italy	567
5	France	491
6	Sweden	479
7	Canada	461
8	Hungary	432
9	Norway	378

In [39]:

```python
totalGoldmedals = goldMedal.Region.value_counts().reset_index(name = 'Medals').head(10)
g = sns.catplot(x = "index", y = "Medals", data = totalGoldmedals, height=5, kind="bar",
g.despine(left = True)
g.set_xlabels("Top 10 Countries")
g.set_ylabels("Number of Medals")
plt.title("Gold Medals Country Wise", size = 15)
plt.xticks(rotation = 45)
plt.show()
```



Rio Olympics 2016

In [40]:
```
most_recent = athletes_df.Year.max()
most_recent
```

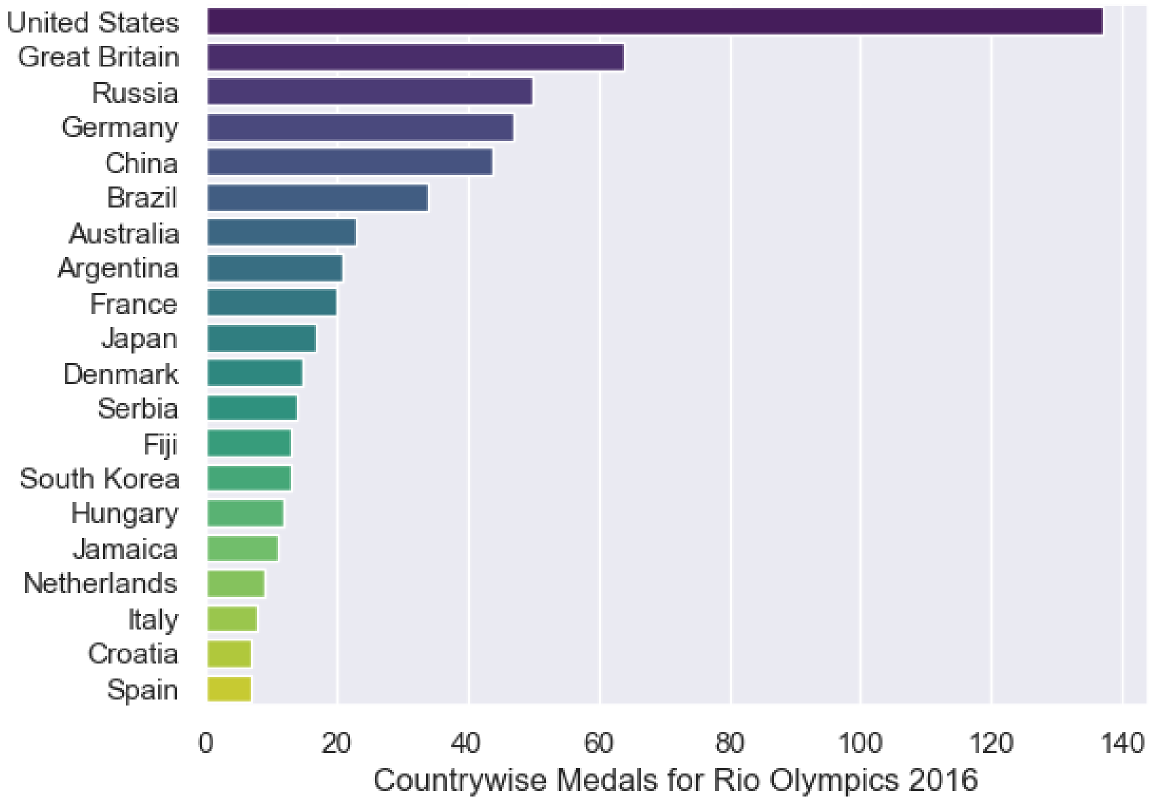
Out[40]:
```
2016
```

In [41]:
```
# Top Gold winning nations in Rio Olympics

team_names = athletes_df[(athletes_df.Year == most_recent) & (athletes_df.Medal == 'Gold
team_names.value_counts().head(10)
```

Out[41]:
```
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 [42]:
```
sns.barplot(x=team_names.value_counts().head(20), y= team_names.value_counts().head(20).
plt.ylabel(None)
plt.xlabel('Countrywise Medals for Rio Olympics 2016')
plt.show()
```

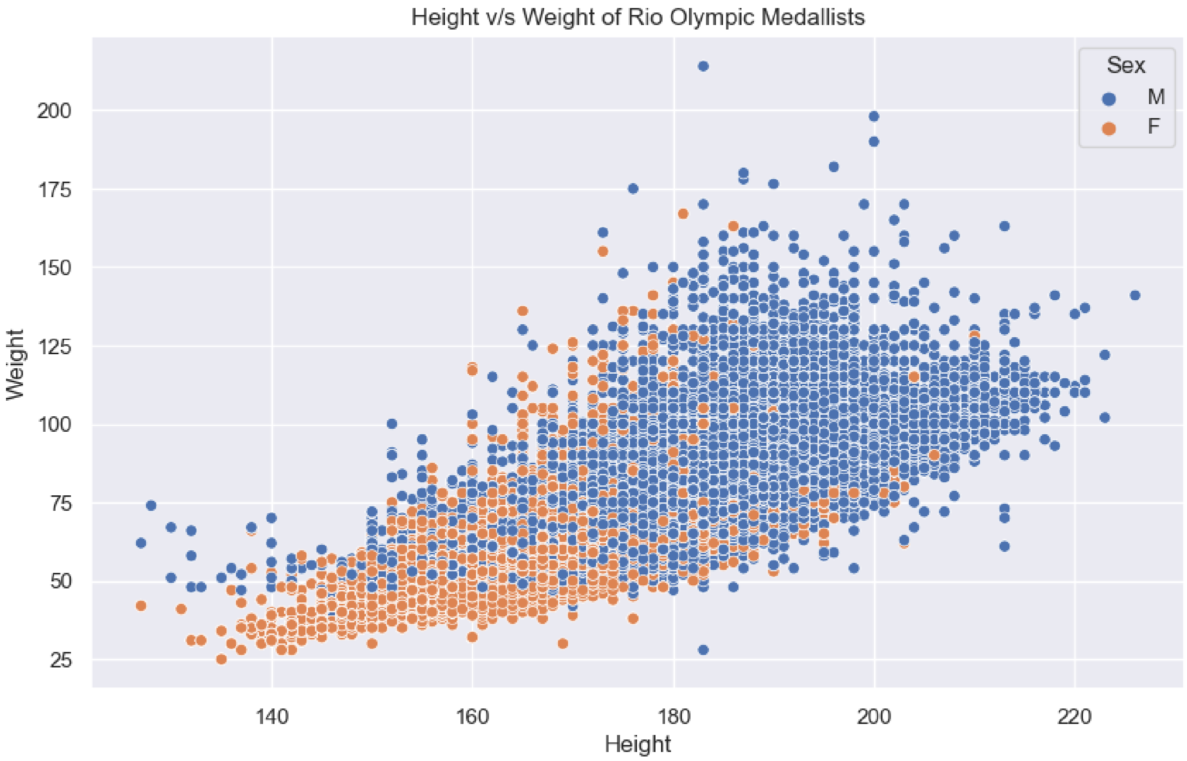


In [43]:
```
# filtering athletes who had won a medal, we are filtering the null elements

not_null_medals = athletes_df[(athletes_df['Height'].notnull()) & (athletes_df['Weight']
```

In [44]:
```
plt.figure(figsize=(10,6))
plt.title("Height v/s Weight of Rio Olympic Medallists")
```

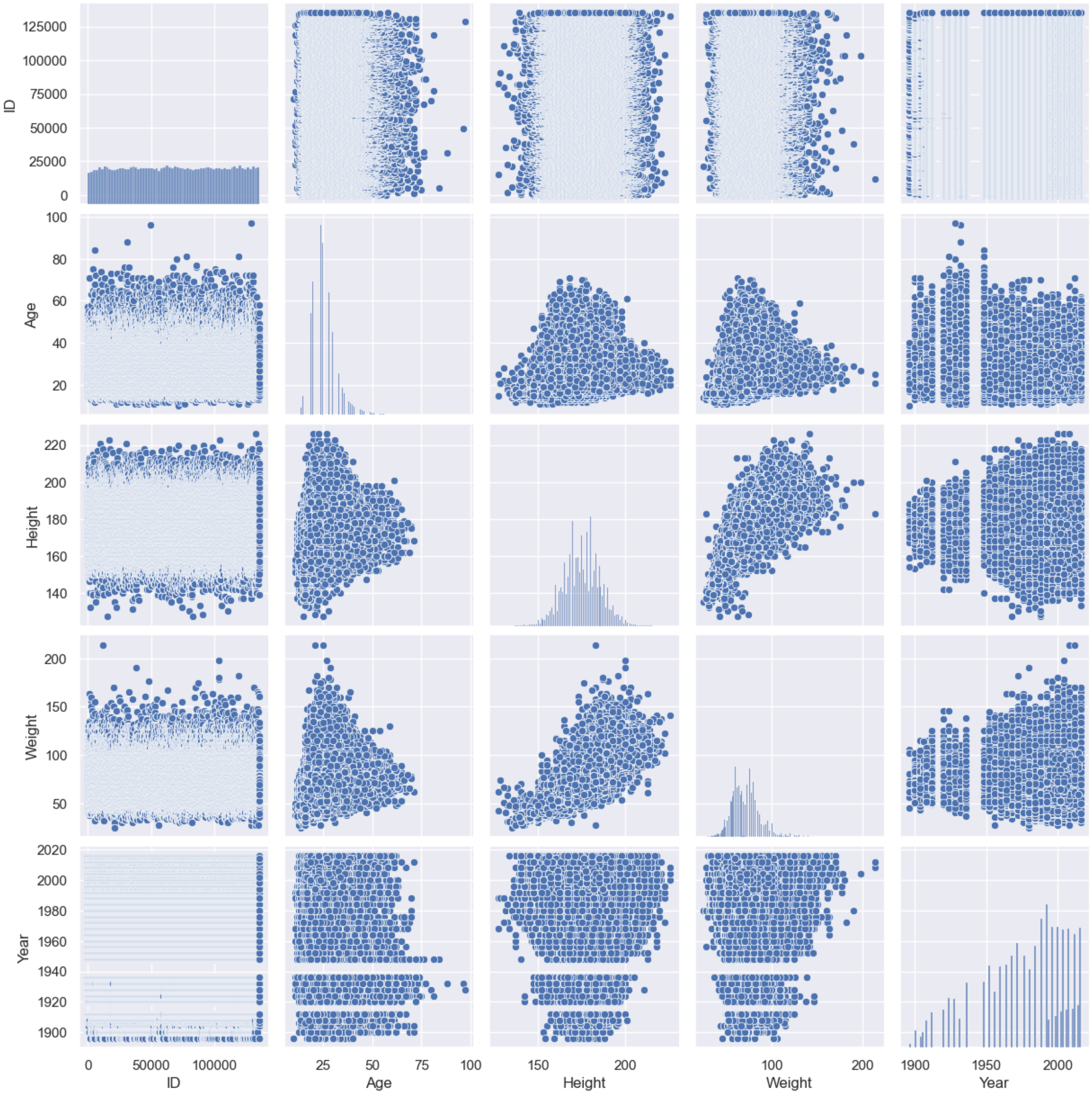
```
axis = sns.scatterplot(x= "Height", y = "Weight", data = not_null_medals, hue = "Sex")
plt.show()
```



In [45]:

```
plt.figure(figsize = (5,10))
sns.pairplot(athletes_df)
plt.show()
```

```
<Figure size 500x1000 with 0 Axes>
```


ID
125000
100000
75000
50000
25000
0
Age
100
80
60
40
20
Height
220
200
180
160
140
Weight
200
150
100
50
Year
2020
2000
1980
1960
1940
1920
1900
0
50000
100000
ID
25
50
75
100
Age
150
200
Height
100
200
Weight
1900
1950
2000
Year