Olympics Data Analysis Using python

In [1]:

```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

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

In [76]:

```
athletes.head()
```

Out[76]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Inn
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winter	N
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winter	Sa

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]:

```
#join the two dataframes
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	
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	B
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Ar
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	

In [6]:

```
#shape of the data
athletes_df.shape
```

Out[6]:

```
(271116, 17)
```

In [7]:

```
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 [8]:

```
athletes_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]:

```
#checking null values -
nan_values = athletes_df.isna()
nan_columns = nan_values.any()
nan_columns
```

Out[9]:

```
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 [10]:

```
athletes_df.isnull().sum()
```

Out[10]:

```
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
```

Details of some countries participating

```
Eg- India and china
```

In [11]:

```
#details of some countries particpating in the olympics
#1) India details
athletes_df.query('Team == "India"').head(3)
```

Out[11]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	Ci
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterda
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterda
895	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	L Angel

In [12]:

```
#2) China
athletes_df.query('Team == "China"').head(3)
```

Out[12]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	B
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	
1072	602	Abudoureheman	M	22.0	182.0	75.0	China	CHN	2000 Summer	2000	Summer	

Participation of countries

In [13]:

```
#displaying top 5 countries participating in the olympics
top_5_countries = athletes_df.Team.value_counts().sort_values(ascending=False).head(5)
top_5_countries
```

Out[13]:

```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Name: Team, dtype: int64
```

In [14]:

```
#displaying least 5 countries participating in the olympics
least_5_countries = athletes_df.Team.value_counts().sort_values(ascending=True).head(5)
least_5_countries
```

Out[14]:

```
Digby               1
Hb-20               1
Fantlet-2           1
Greenoaks Dundee    1
Newfoundland        1
Name: Team, dtype: int64
```

In [15]:

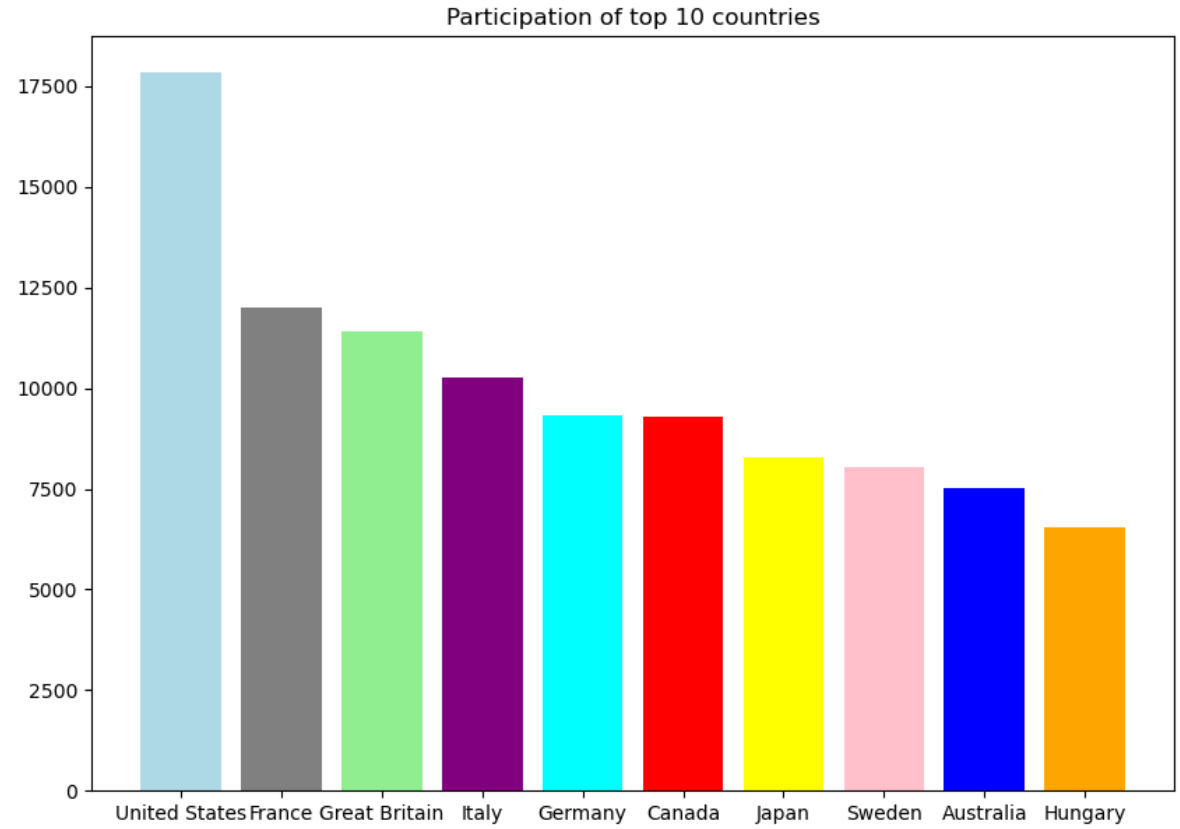
```
#graph for top 10 countries
top_10_countries = athletes_df.Team.value_counts().sort_values(ascending=False).head(10)
top_10_countries
```

Out[15]:

```
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
```

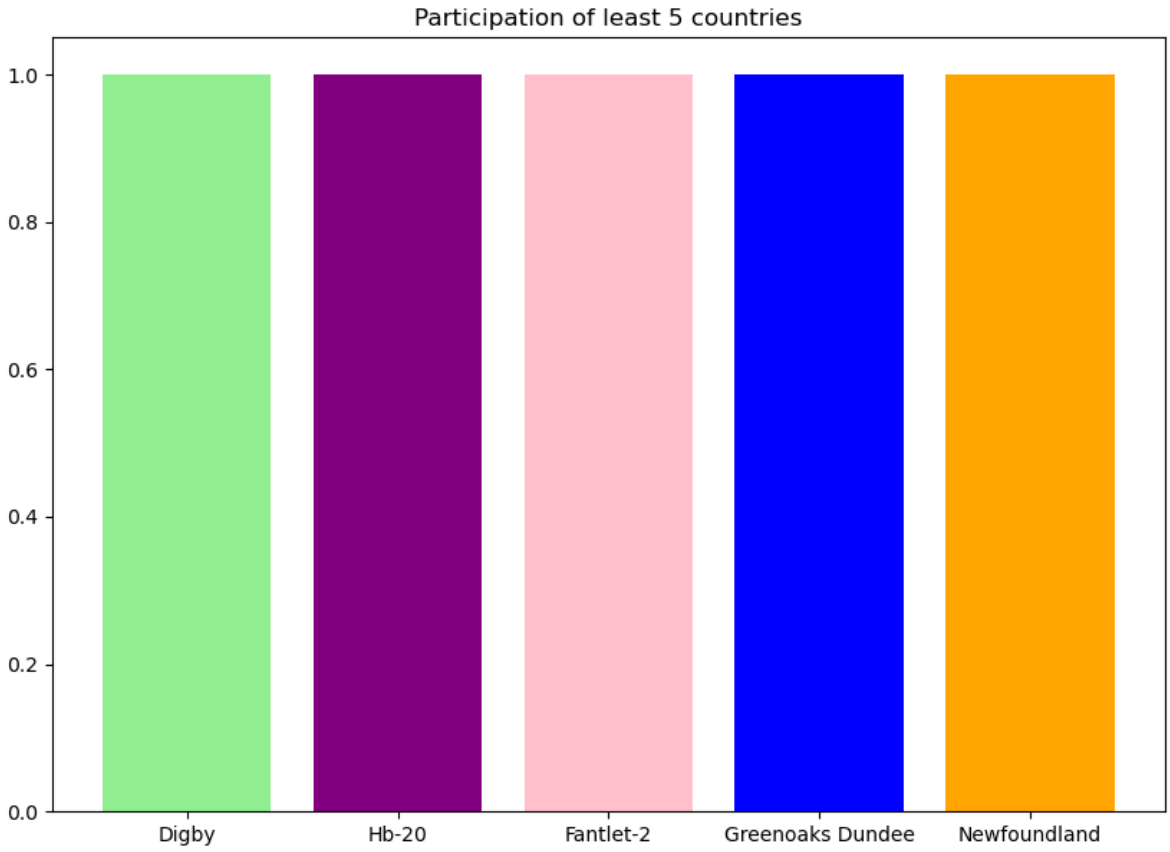
In [16]:

```
plt.figure(figsize=(10,7))
bars = plt.bar(top_10_countries.index,top_10_countries,color=['lightblue', 'grey', 'lightgr
plt.title("Participation of top 10 countries")
plt.show()
```



In [17]:

```
plt.figure(figsize=(10,7))
bars = plt.bar(least_5_countries.index,least_5_countries,color=['lightgreen', 'purple', 'pi
plt.title("Participation of least 5 countries")
plt.show()
```



Sport Season Analysis

In [18]:

```
#summer olympic sports
summer_sports = athletes_df[athletes_df.Season == "Summer"].Sport.unique()
print("Summer sports : ",summer_sports,sep = "\n")
```

```
Summer sports :
['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']
```

In [19]:

```
# Winter Sports
winter_sports = athletes_df[athletes_df.Season == "Winter"].Sport.unique()
print("Winter sports : ",winter_sports,sep = "\n")
```

```
Winter sports :
['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']
```

In [20]:

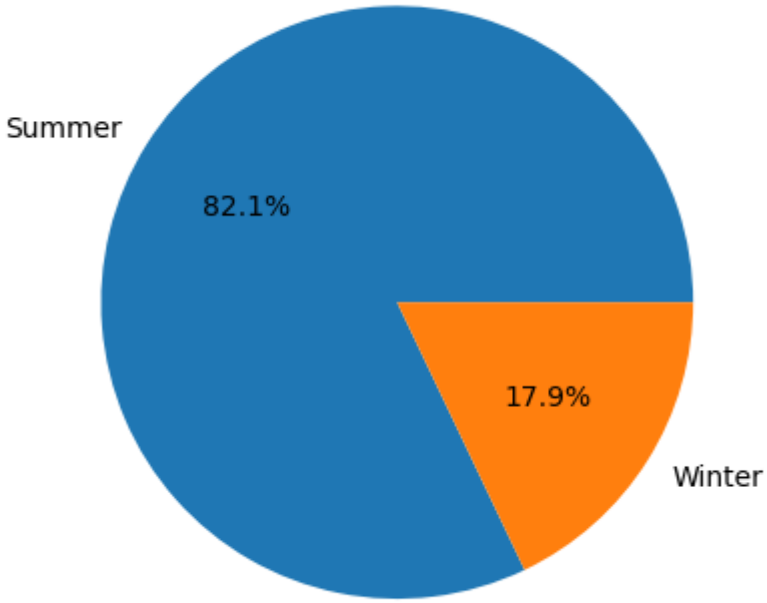
```
label=athletes_df.Season.value_counts().index
count=athletes_df.Season.value_counts().values
```

In [21]:

```
#Players participating in the sports season
plt.pie(count,labels=label,autopct='%1.1f%%')
```

Out[21]:

```
([<matplotlib.patches.Wedge at 0x28548c00580>,
  <matplotlib.patches.Wedge at 0x28548c00d00>],
 [Text(-0.9303751029505434, 0.5868578770109217, 'Summer'),
  Text(0.9303751304233422, -0.5868578334569193, 'Winter')],
 [Text(-0.5074773288821145, 0.3201042965514118, '82.1%'),
  Text(0.5074773438672775, -0.32010427279468323, '17.9%')])
```



In [22]:

```
Diff_seasons = athletes_df.Season.value_counts()
Diff_seasons
```

Out[22]:

```
Summer    222552
Winter     48564
Name: Season, dtype: int64
```

In [23]:

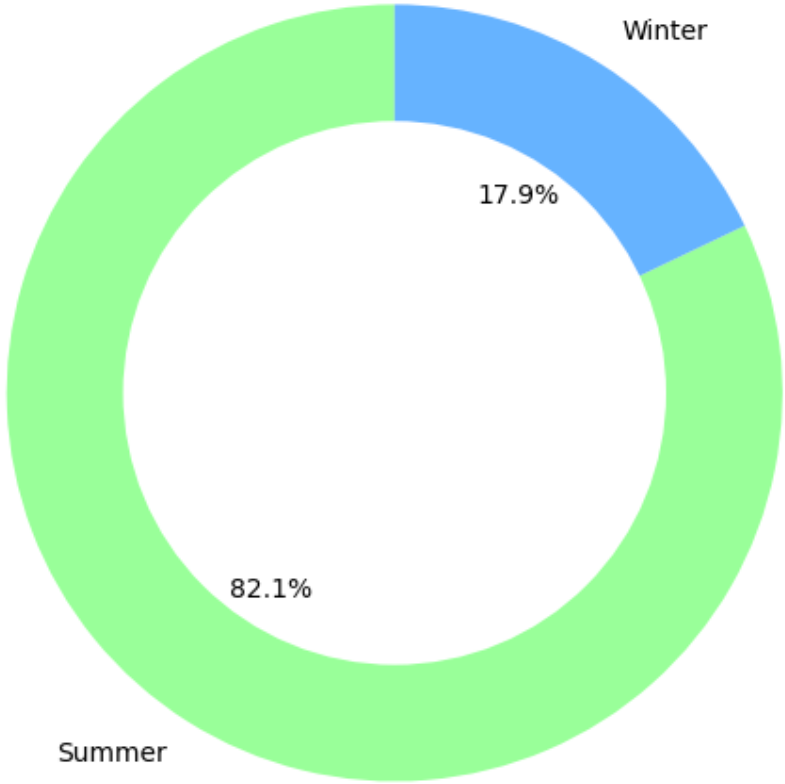
```
fig1, ax1 = plt.subplots()

colors = ['#99ff99','#66b3ff']

ax1.pie(Diff_seasons, colors = colors, labels=Diff_seasons.index, autopct='%1.1f%%', starta

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

ax1.axis('equal')
plt.tight_layout()
plt.show()
```



Gender Analysis

In [24]:

```
#male and female participants in the olympics
gender_counts = athletes_df.Sex.value_counts()
gender_counts
```

Out[24]:

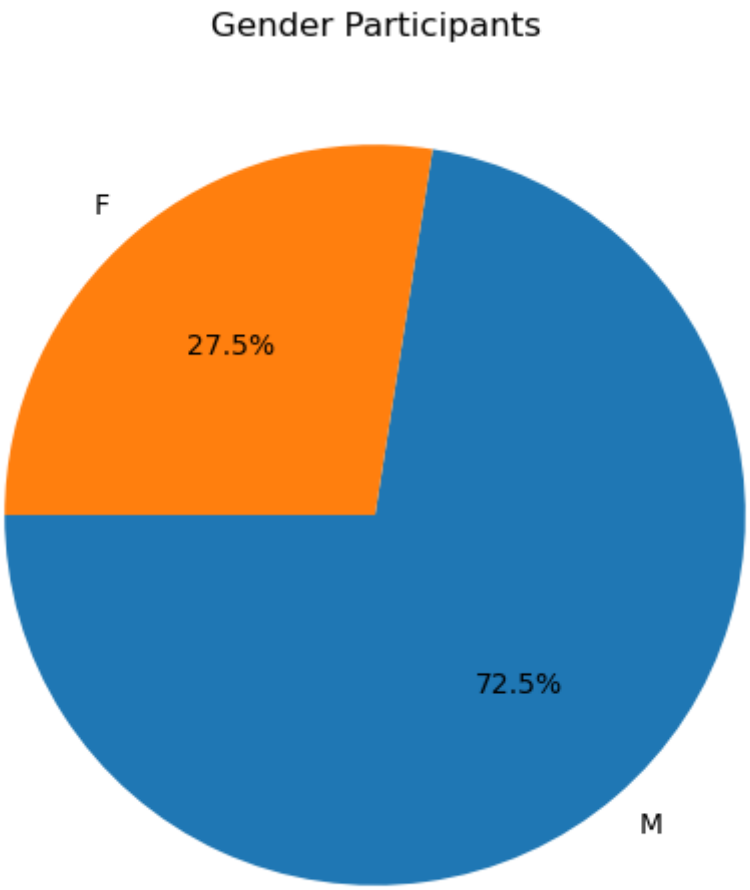
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [25]:

```
plt.figure(figsize=(12,6))
plt.title("Gender Participants")
plt.pie(gender_counts,labels=gender_counts.index, autopct='%1.1f%%',startangle=180)
```

Out[25]:

```
([<matplotlib.patches.Wedge at 0x28549f85df0>,
  <matplotlib.patches.Wedge at 0x28549f93580>],
 [Text(0.7147310163003329, -0.8361576252945934, 'M'),
  Text(-0.7147309380136028, 0.836157692212537, 'F')],
 [Text(0.38985328161836336, -0.4560859774334145, '72.5%'),
  Text(-0.3898532389165105, 0.456086013934111, '27.5%')])
```



In [26]:

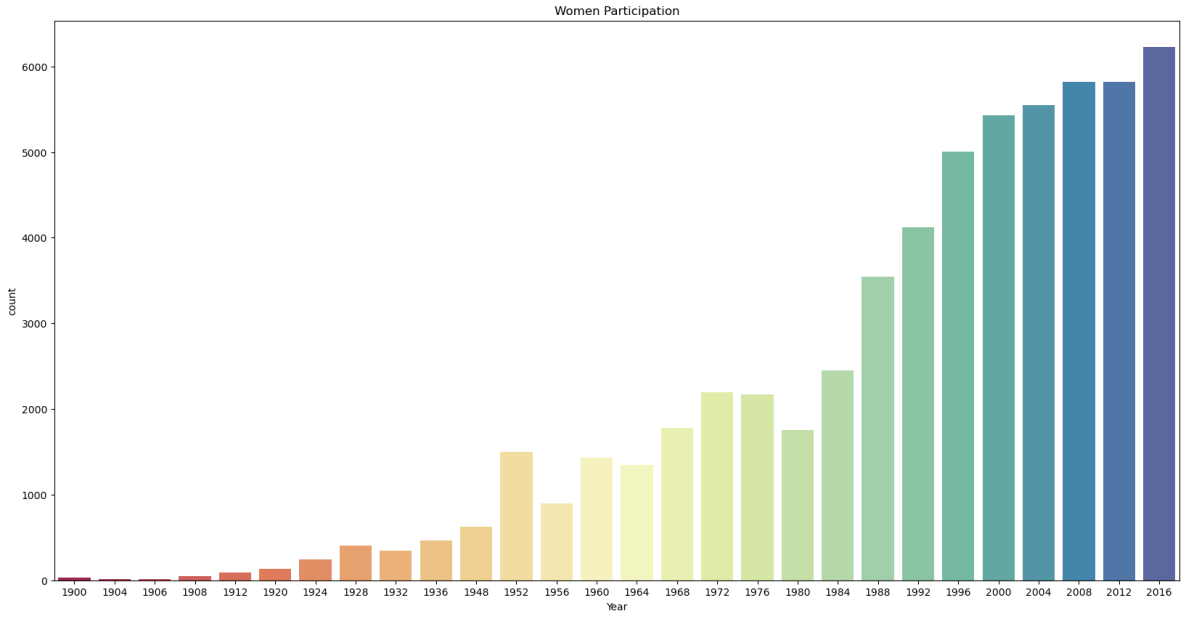
```
# Women Participation in Olympics
women_part = athletes_df[(athletes_df.Sex == 'F') & (athletes_df.Season == 'Summer')]
```

In [27]:

```
plt.figure(figsize=(20,10))
sns.countplot(x='Year',data=women_part, palette='Spectral')
plt.title('Women Participation')
```

Out[27]:

```
Text(0.5, 1.0, 'Women Participation')
```



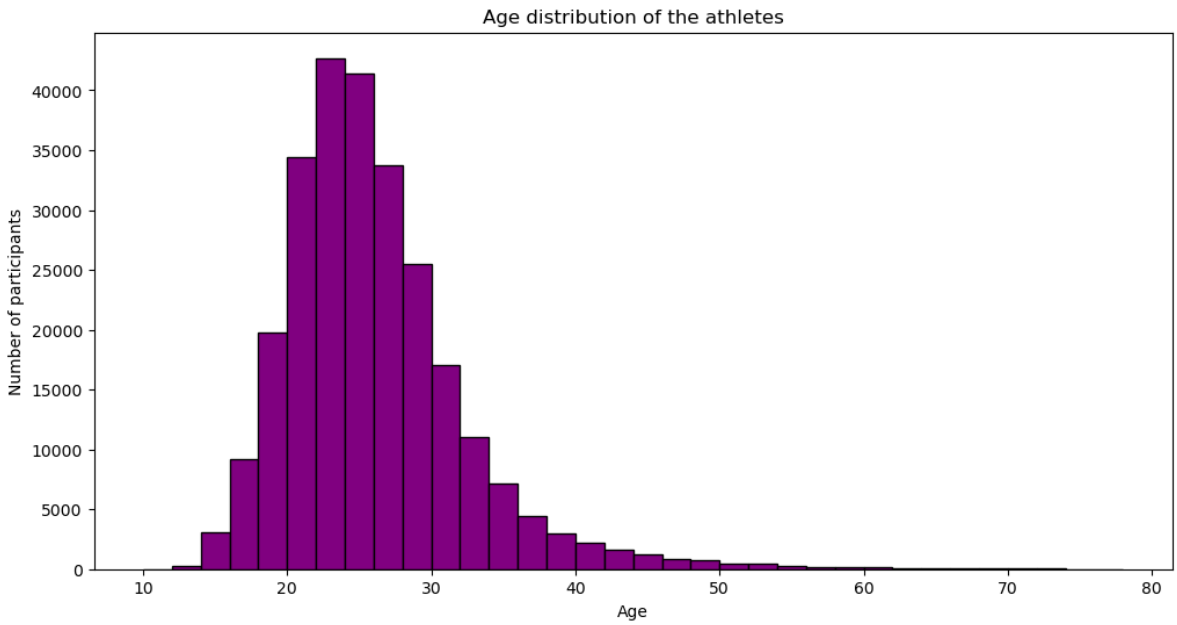
Age,Height and Weight Analysis

In [28]:

```
#age of the distribution
plt.figure(figsize=(12, 6))
plt.title("Age distribution of the athletes")
plt.xlabel('Age')
plt.ylabel('Number of participants')
plt.hist(athletes_df.Age,bins = np.arange(10,80,2),color='purple',edgecolor = 'black')
```

Out[28]:

```
(array([1.4000e+01, 2.2600e+02, 3.0400e+03, 9.2280e+03, 1.9795e+04,
        3.4422e+04, 4.2689e+04, 4.1427e+04, 3.3700e+04, 2.5506e+04,
        1.7047e+04, 1.1046e+04, 7.1180e+03, 4.4560e+03, 3.0170e+03,
        2.1630e+03, 1.6590e+03, 1.2670e+03, 8.3700e+02, 7.6900e+02,
        4.7700e+02, 4.4400e+02, 2.6600e+02, 2.0000e+02, 1.7100e+02,
        1.5600e+02, 1.1800e+02, 1.1400e+02, 5.6000e+01, 8.5000e+01,
        6.1000e+01, 3.2000e+01, 1.6000e+01, 9.0000e+00]),
 array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
        44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76,
        78]),
 <BarContainer object of 34 artists>)
```

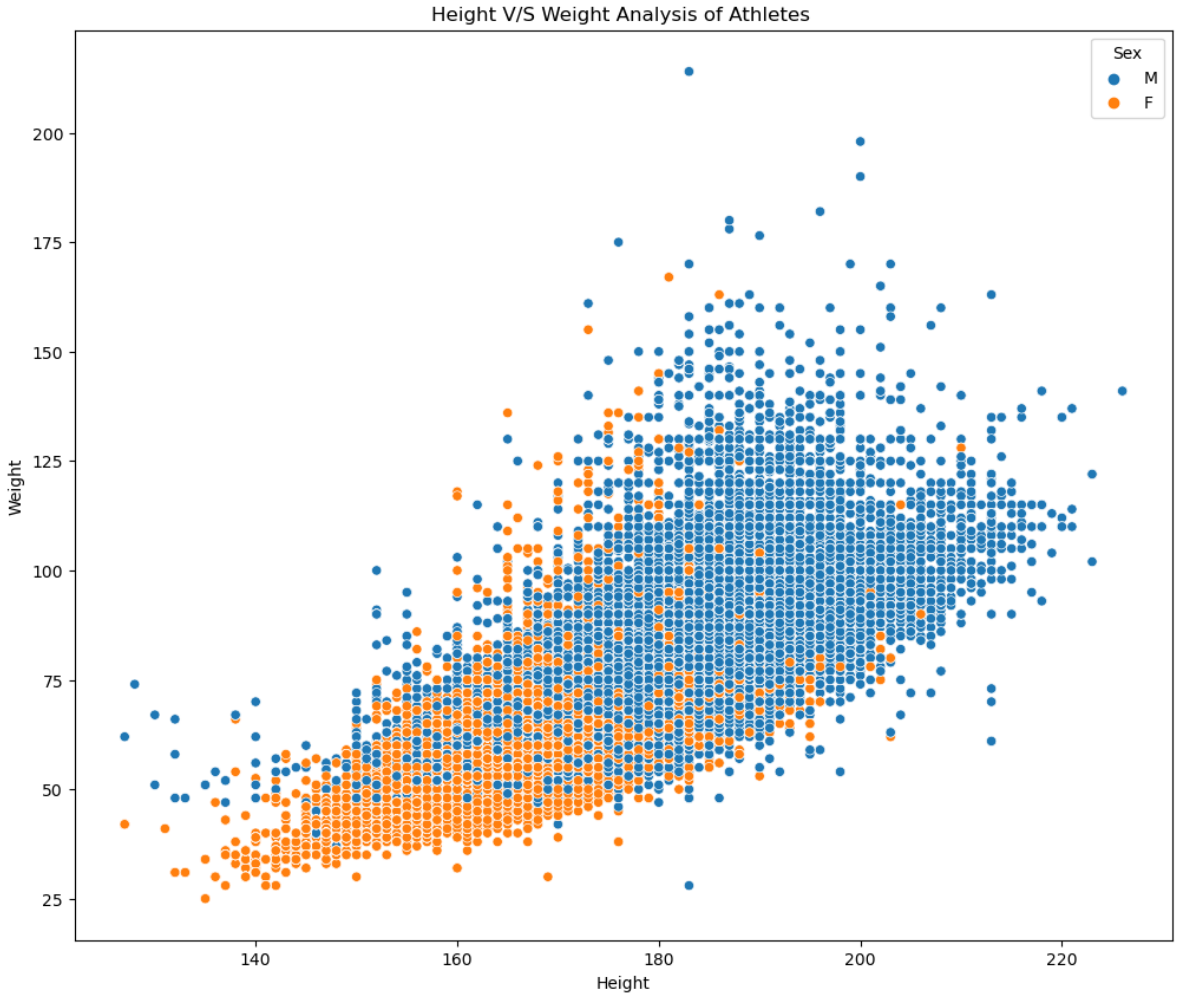


In [29]:

```
height_and_weight = athletes_df[(athletes_df['Height'].notnull()) & (athletes_df['Weight'].
plt.figure(figsize=(12,10))
sns.scatterplot(x="Height",y="Weight",data=height_and_weight,hue="Sex")
plt.title("Height V/S Weight Analysis of Athletes ")
```

Out[29]:

```
Text(0.5, 1.0, 'Height V/S Weight Analysis of Athletes ')
```

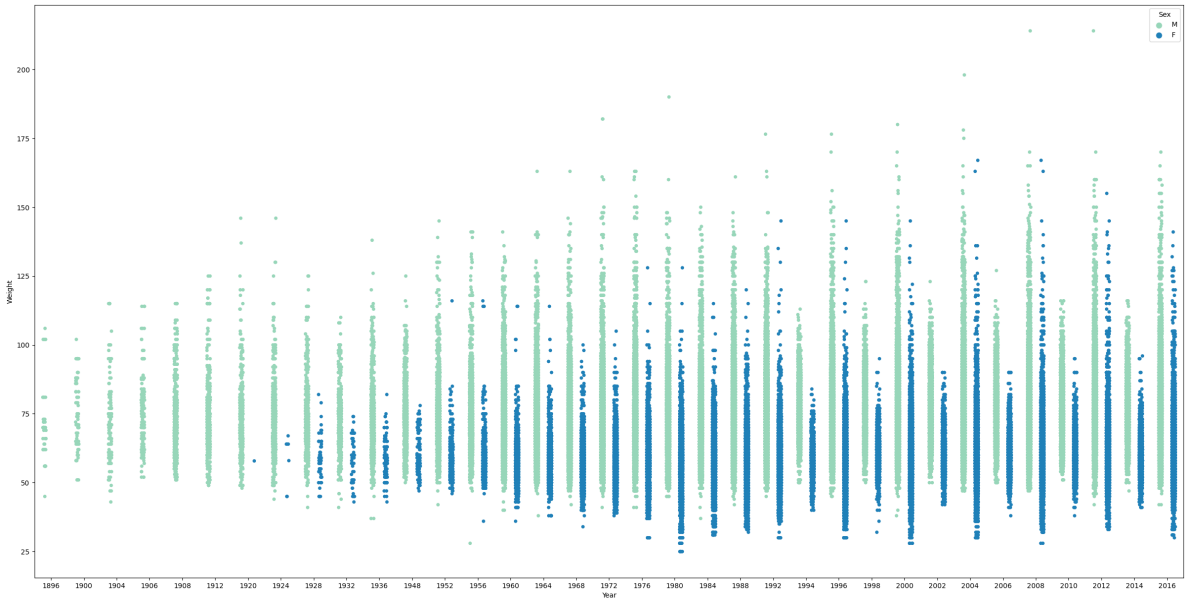


In [30]:

```
weight_df= athletes_df[(athletes_df['Weight'].notnull())]
plt.figure(figsize=(30,15))
sns.stripplot(x="Year",y="Weight",data=weight_df,hue="Sex",dodge=True,palette='YlGnBu')
```

Out[30]:

```
<AxesSubplot:xlabel='Year', ylabel='Weight'>
```

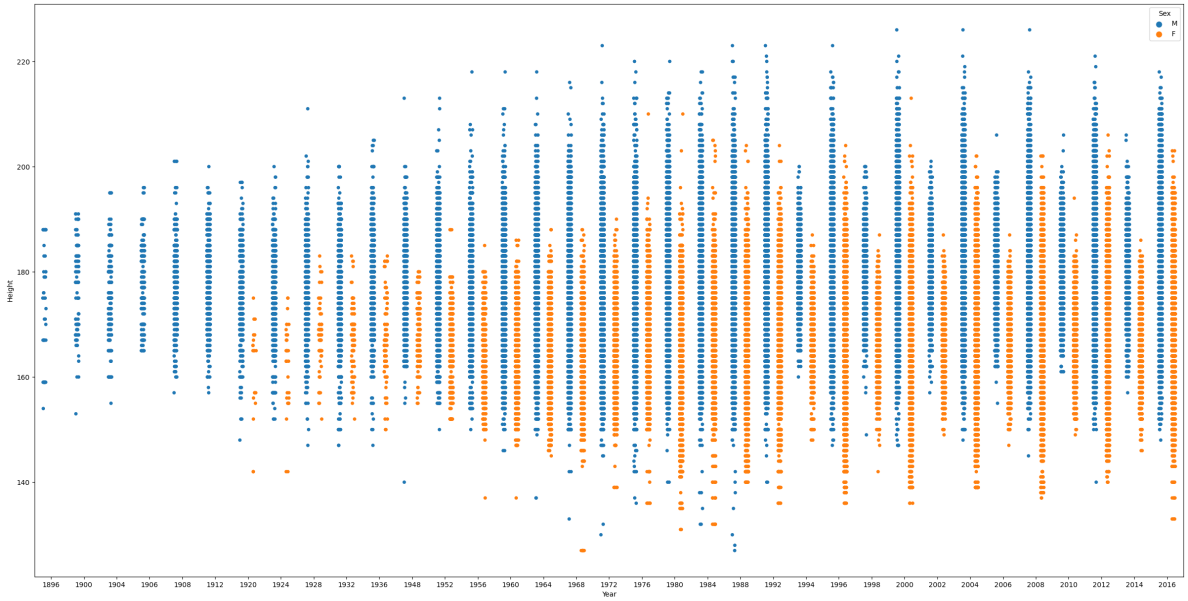


In [31]:

```
height_df= athletes_df[(athletes_df['Height'].notnull())]
plt.figure(figsize=(30,15))
sns.stripplot(x="Year",y="Height",data=height_df,hue="Sex",dodge=True,)
```

Out[31]:

```
<AxesSubplot:xlabel='Year', ylabel='Height'>
```

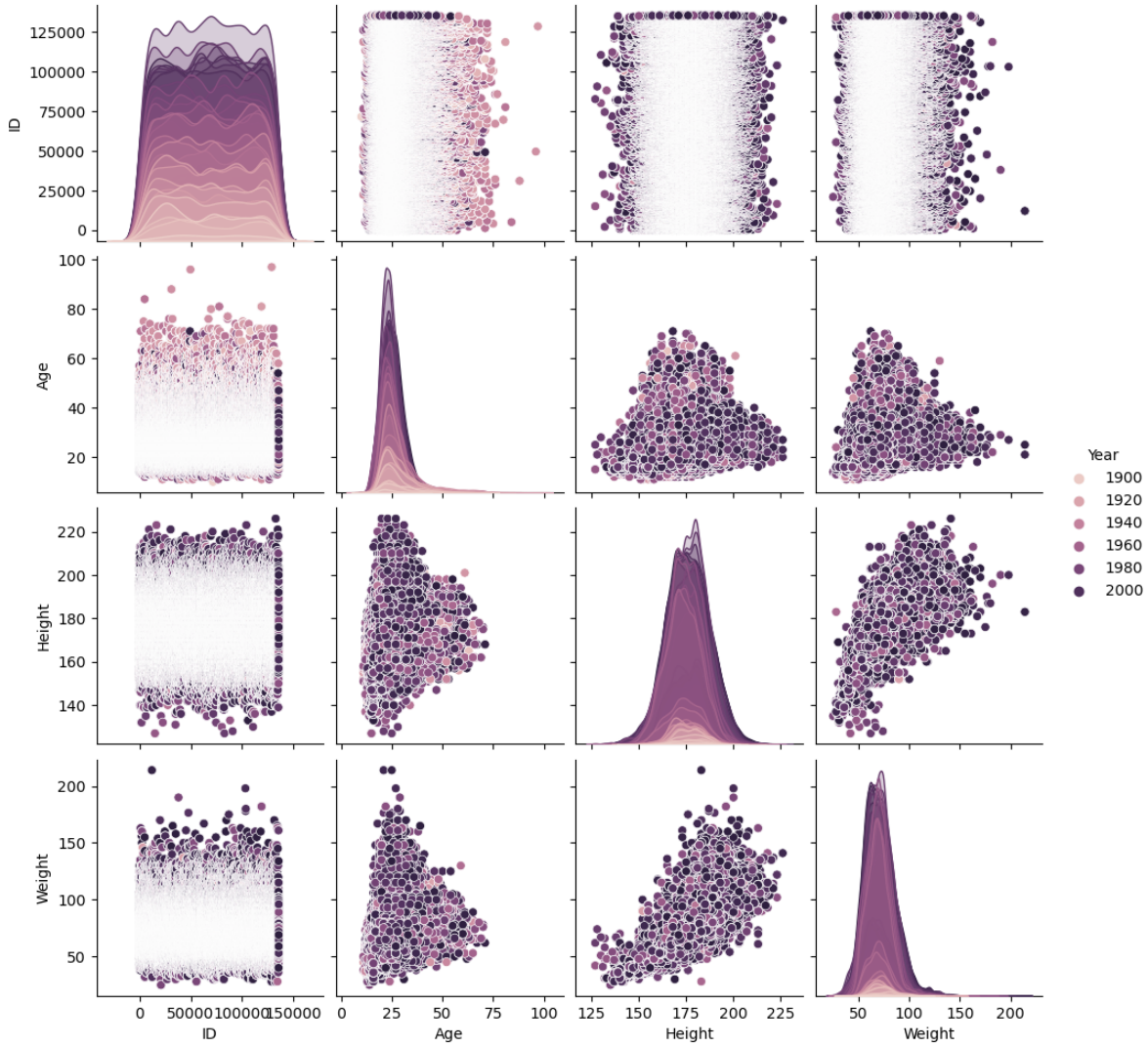


In [32]:

```
sns.pairplot(athletes_df.select_dtypes(["number"]),hue="Year")
```

Out[32]:

```
<seaborn.axisgrid.PairGrid at 0x285551c2250>
```



Medal Analysis

In [62]:

```
#Total no. of medals won by the athletes
medal_counts=athletes_df.Medal.value_counts()
print("Medal Counts : ",medal_counts,sep = "\n")
```

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

In [63]:

```
athletes_df['count']=1
athletes_df
```

Out[63]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	S
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	S
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	S
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	S
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	S
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	
...	...	...	...	...	...	...	...	...	...	...	
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	

271116 rows × 18 columns

In [64]:

```
athletes_df['count']=1
athletes_df.groupby(['Medal','Team']).count()['count']
```

Out[64]:

```
Medal   Team
Bronze  A North American Team      4
        Afghanistan               2
        Algeria                   8
        Ali-Baba II               5
        Amstel Amsterdam          4
                                ...
Silver  West Germany-1           10
        Yugoslavia              167
        Zambia                    1
        Zimbabwe                  4
        Zut                       3
Name: count, Length: 783, dtype: int64
```

In [65]:

```
#Gold Medal athletes
gold_medals=athletes_df[(athletes_df.Medal=='Gold')]
gold_medals
```

Out[65]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900
42	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948
44	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948
48	17	Paavo Johannes Aaltonen	M	28.0	175.0	64.0	Finland	FIN	1948 Summer	1948
60	20	Kjetil Andr Aamodt	M	20.0	176.0	85.0	Norway	NOR	1992 Winter	1992
...	...	...	...	...	...	...	...	...	...	...
270981	135503	Zurab Zviadauri	M	23.0	182.0	90.0	Georgia	GEO	2004 Summer	2004
271009	135520	Julia Zwehl	F	28.0	167.0	60.0	Germany	GER	2004 Summer	2004
271016	135523	Ronald Ferdinand "Ron" Zwerver	M	29.0	200.0	93.0	Netherlands	NED	1996 Summer	1996
271049	135545	Henk Jan Zwolle	M	31.0	197.0	93.0	Netherlands	NED	1996 Summer	1996
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21.0	168.0	80.0	Soviet Union	URS	1952 Summer	1952

13372 rows × 18 columns

Gold medals of athletes above age 50

In [66]:

```
#gold_medals = gold_medals[np.isfinite(gold_medals['Age'])]
gold_medals['ID'][gold_medals['Age'] > 50].count()
```

Out[66]:

```
65
```

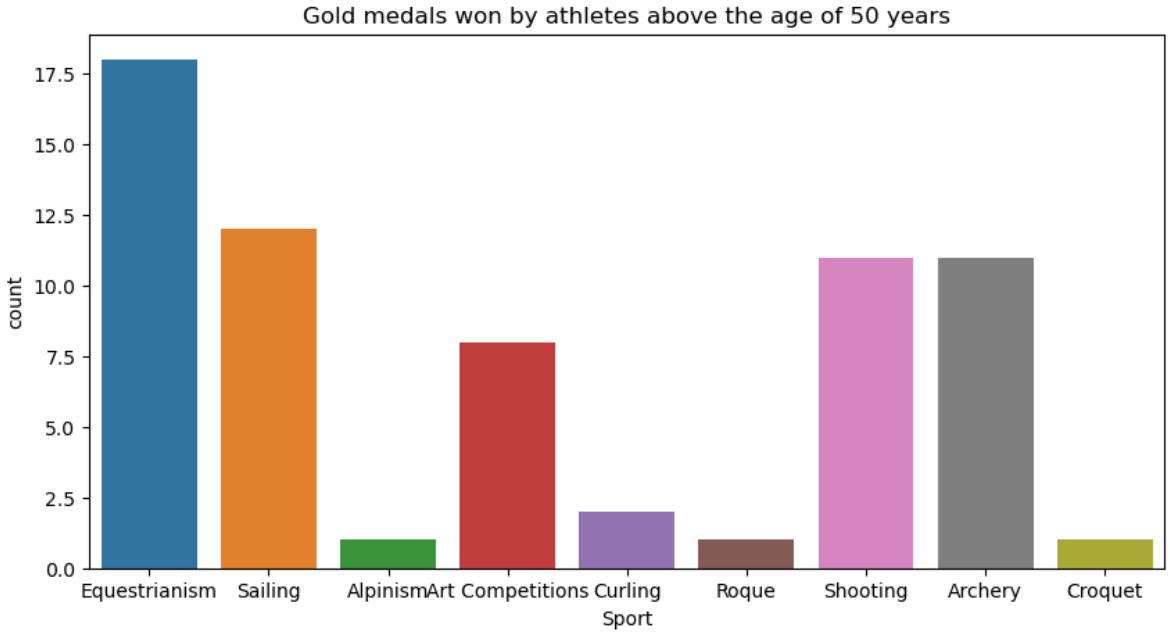
In [67]:

```
#Graph for Gold medals for athletes above 50 years
sport_event=gold_medals['Sport'][gold_medals['Age']>50]
plt.figure(figsize=(10,5))
sns.countplot(sport_event)
plt.title("Gold medals won by athletes above the age of 50 years")
```

```
C:\Users\lenovo\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variable as a keyword arg: x. From version 0.1
2, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretati
on.
  warnings.warn(
```

Out[67]:

```
Text(0.5, 1.0, 'Gold medals won by athletes above the age of 50 years')
```



In [68]:

```
#top 5 countries winning the most number of gold medals
gold_medals.region.value_counts().reset_index(name='Medal').head(5)
```

Out[68]:

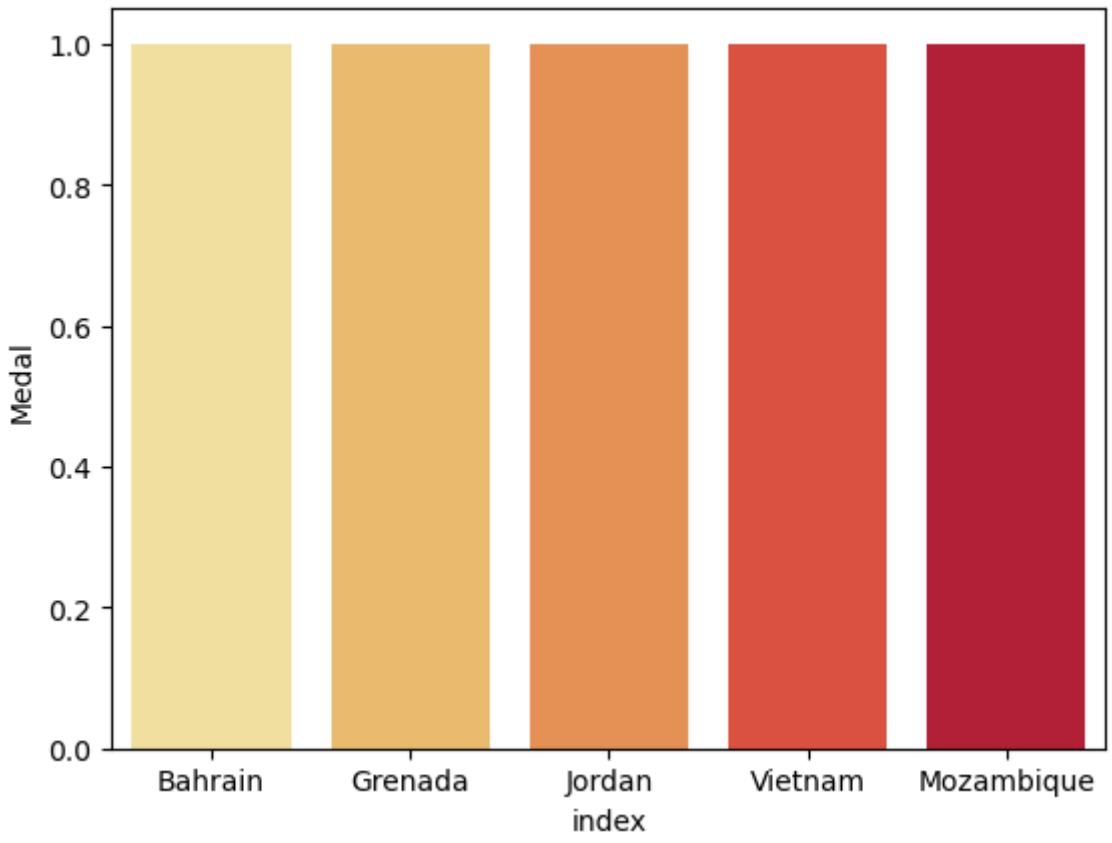
	index	Medal
0	USA	2638
1	Russia	1599
2	Germany	1301
3	UK	678
4	Italy	575

In [69]:

```
total_gold = gold_medals.region.value_counts().reset_index(name='Medal').tail(5)
sns.barplot(x="index",y="Medal",data=total_gold,palette="YlOrRd")
```

Out[69]:

```
<AxesSubplot:xlabel='index', ylabel='Medal'>
```



In [70]:

```
indian_medals=athletes_df[(athletes_df.Medal=='Gold')]
```

In [71]:

```
indian_medals.groupby(['Year']).count().tail(5)
#indian_medals.head(5)
```

Out[71]:

Year	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Season	City	Sport	Event
2008	671	671	671	671	670	664	671	671	671	671	671	671	671
2010	174	174	174	174	174	173	174	174	174	174	174	174	174
2012	632	632	632	632	631	622	632	632	632	632	632	632	632
2014	202	202	202	202	202	190	202	202	202	202	202	202	202
2016	665	665	665	665	664	662	665	665	665	665	665	665	665

In [72]:

```
#india gold_medal
indian_medals_gold = athletes_df[(athletes_df.Medal == 'Gold') & (athletes_df.Team == 'Indi
```

In [73]:

```
gold_india=athletes_df.loc[(athletes_df['Team']=='India') & (athletes_df['Medal']=='Gold')
gold_india.head(2)
```

Out[73]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	C
4732	2699	Shaukat Ali	M	30.0	NaN	NaN	India	IND	1928 Summer	1928	Summer	Amsterd
4736	2703	Syed Mushtaq Ali	M	22.0	165.0	61.0	India	IND	1964 Summer	1964	Summer	Tol

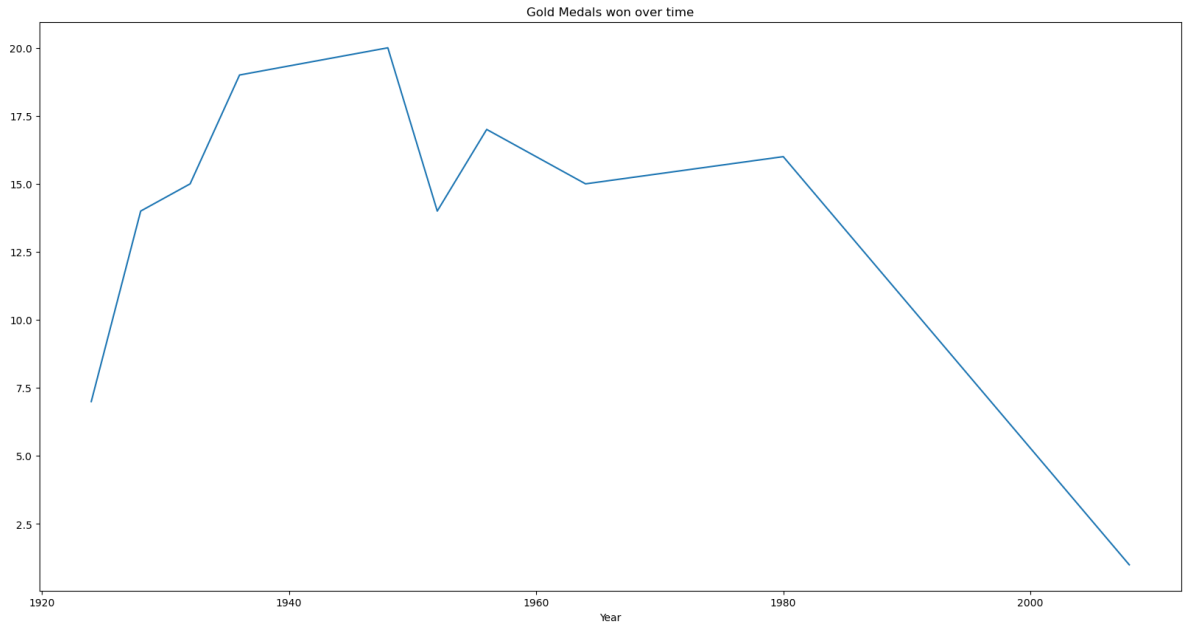
In [74]:

```
part=gold_india.groupby('Year')['Medal'].value_counts()
plt.figure(figsize=(20,10))
part.loc[:,'Gold'].plot()

plt.title("Gold Medals won over time")
```

Out[74]:

```
Text(0.5, 1.0, 'Gold Medals won over time')
```

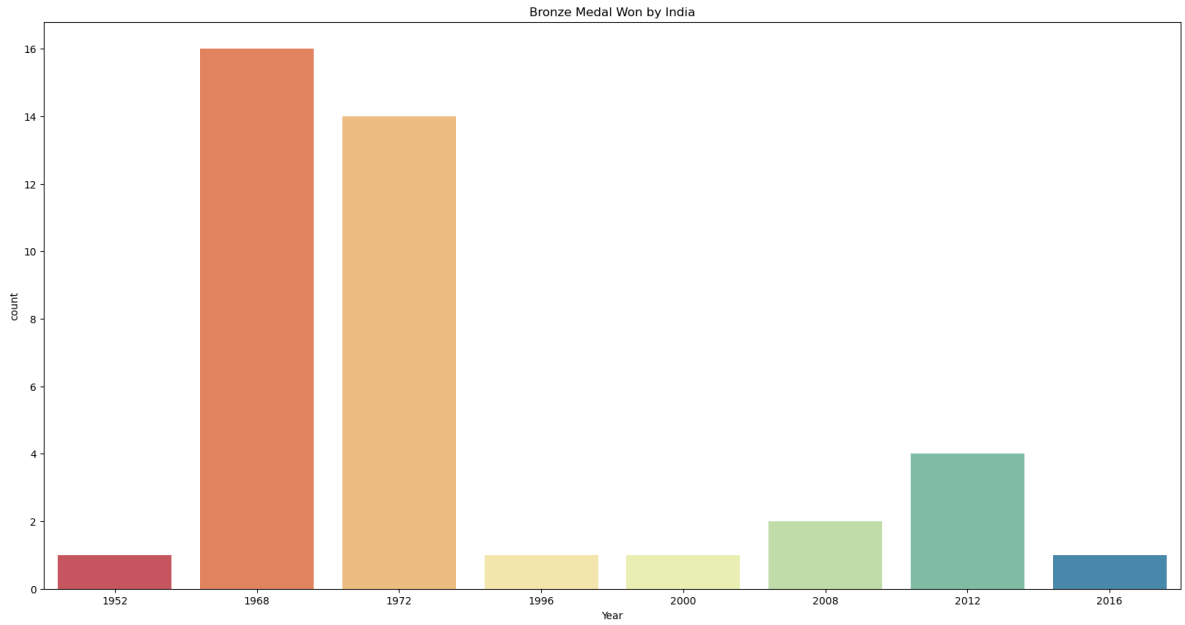


In [75]:

```
bronze_medal = athletes_df[(athletes_df.Medal == 'Bronze') & (athletes_df.Team == 'India')]
plt.figure(figsize=(20,10))
sns.countplot(x='Year',data=bronze_medal, palette='Spectral')
plt.title('Bronze Medal Won by India')
```

Out[75]:

```
Text(0.5, 1.0, 'Bronze Medal Won by India')
```



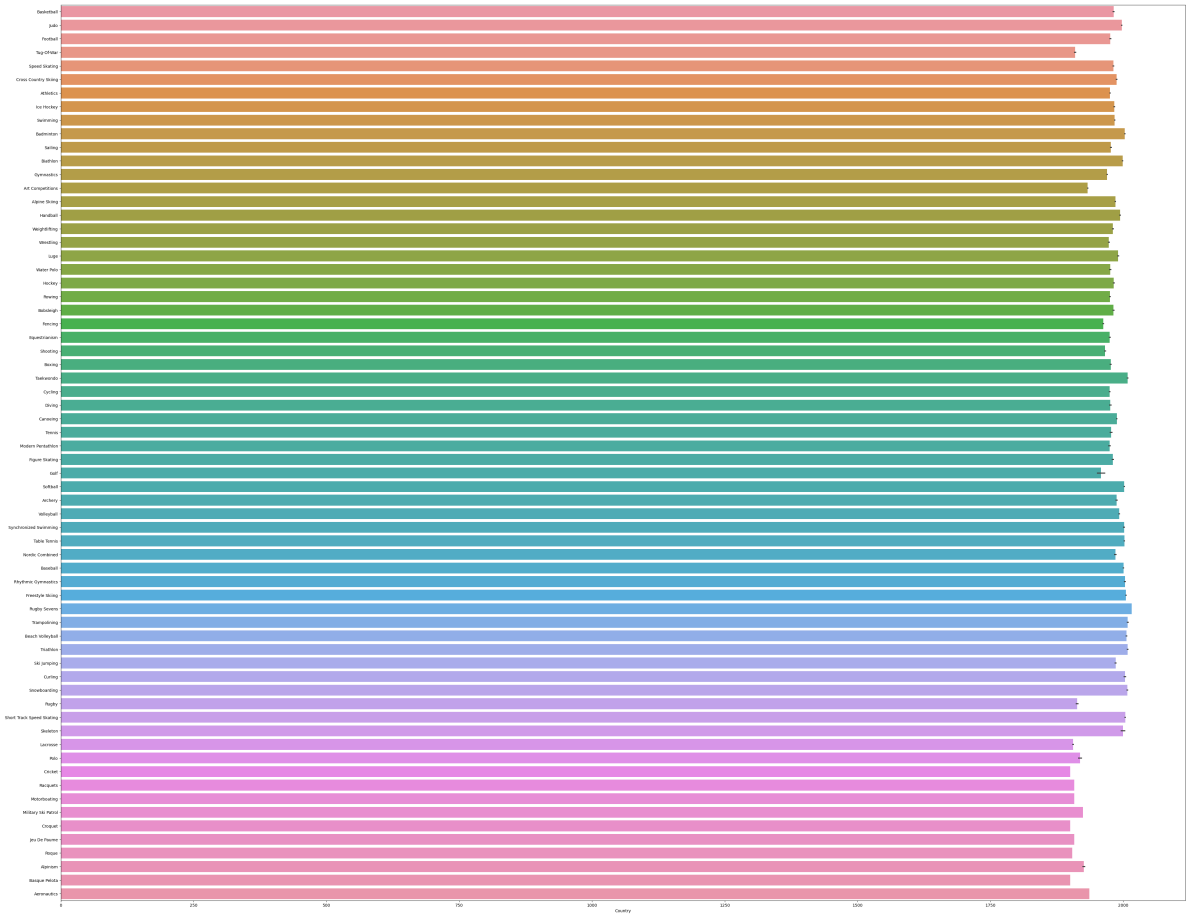
Sport Analysis

In [47]:

```
plt.figure(figsize=(50,40))
sns.barplot(x="Year",y="Sport",data=athletes_df)
plt.ylabel(None);
plt.xlabel("Country")
```

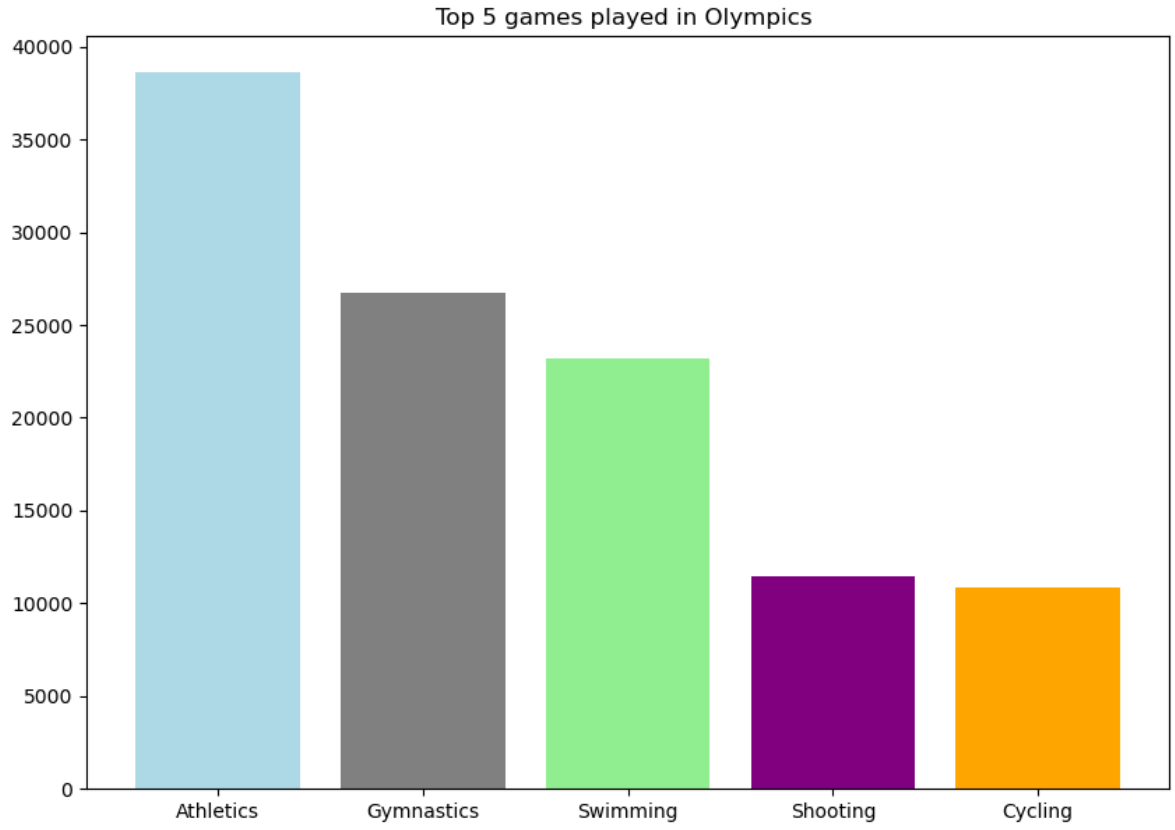
Out[47]:

```
Text(0.5, 0, 'Country')
```



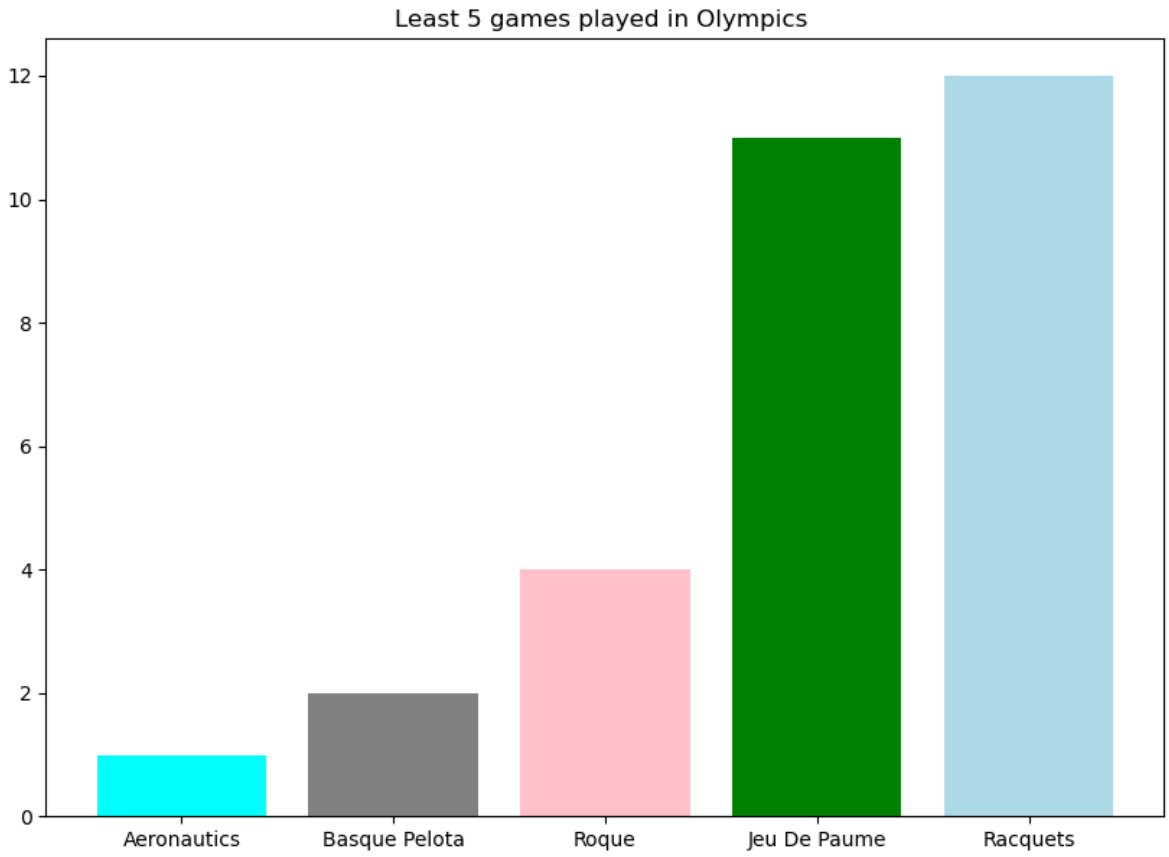
In [48]:

```
#graph for top 5 games played in Olympics
top_5_games = athletes_df.Sport.value_counts().sort_values(ascending=False).head(5)
top_5_games
plt.figure(figsize=(10,7))
bars = plt.bar(top_5_games.index,top_5_games,color=['lightblue', 'grey', 'lightgreen', 'pur
plt.title("Top 5 games played in Olympics ")
plt.show()
```



In [49]:

```
#graph for least 5 games played in Olympics
least_5_games = athletes_df.Sport.value_counts().sort_values(ascending=True).head(5)
least_5_games
plt.figure(figsize=(10,7))
bars = plt.bar(least_5_games.index,least_5_games,color=['cyan', 'grey', 'pink', 'green', 'l
plt.title("Least 5 games played in Olympics ")
plt.show()
```



In [50]:

```
medals = athletes_df.groupby('Name',as_index=False).sum()
medals.head(2)
```

Out[50]:

	Name	ID	Age	Height	Weight	Year	count
0	Gabrielle Marie "Gabby" Adcock (White-)	869	25.0	167.0	0.0	2016	1
1	Eleonora Margarida Josephina Scmitt	215906	32.0	0.0	0.0	3896	2

In [51]:

```
athletes_df_1= pd.concat([athletes_df,pd.get_dummies(athletes_df.Medal)],axis=1)
```

In [52]:

```
athletes_df_1['allmedals'] =athletes_df_1['allmedals'] = athletes_df_1['Bronze'] + athletes
athletes_df_1.head(5)
```

Out[52]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	Spo
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	...	Basketba
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	...	Jud
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	...	Footba
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	...	Tug-O Wa
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	...	Spee Skatin

5 rows × 22 columns

STATS ON INDIA

In [53]:

```
dfindia = athletes_df_1[athletes_df.NOC == 'IND']
```

In [54]:

```
#The years india participated in Olympics
sorted(dfindia.Year.unique())
```

Out[54]:

```
[1900,
 1920,
 1924,
 1928,
 1932,
 1936,
 1948,
 1952,
 1956,
 1960,
 1964,
 1968,
 1972,
 1976,
 1980,
 1984,
 1988,
 1992,
 1996,
 1998,
 2000,
 2002,
 2004,
 2006,
 2008,
 2010,
 2012,
 2014,
 2016]
```

In [55]:

```
indian_medals.groupby(['Year']).count().head(3)
#indian_medals.head(5)
```

Out[55]:

Year	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Season	City	Sport	Event
1896	62	62	62	52	13	13	62	62	62	62	62	62	62
1900	201	201	201	179	27	21	201	201	201	201	201	201	201
1904	173	173	173	160	51	45	173	173	173	173	173	173	173

In [56]:

```
#Number of Medals India Won so far
print("Total number of all Medals India won", dfindia['allmedals'].sum())
```

```
Total number of all Medals India won 197
```

In [57]:

```
#Sports in which India won a Gold
dfindia[dfindia.Gold == 1].Sport.unique()
```

Out[57]:

```
array(['Hockey', 'Shooting', 'Alpinism'], dtype=object)
```

In [58]:

```
#the person who won the first individual Gold for India
dfindia[(dfindia.Gold==1) & (dfindia.Sport == 'Shooting')]
```

Out[58]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	Sport	
22004	11601	Abhinav Bindra	M	25.0	173.0	70.0	India	IND	2008 Summer	2008	...	Shooting	SI A

1 rows × 22 columns

RIO OLMPICS

In [59]:

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

```
2016
```

In [60]:

```
team_names =athletes_df[(athletes_df.Year == max_year) & (athletes_df.Medal == 'Gold')].Tea
team_names.value_counts().head(5)
```

Out[60]:

```
United States    137
Great Britain     64
Russia            50
Germany           47
China             44
Name: Team, dtype: int64
```

In [61]:

```
sns.barplot(x=team_names.value_counts().head(25),y=team_names.value_counts().head(25).index
plt.xlabel('Medals for year 2016(Countrywise)')
plt.ylabel(None)
```

Out[61]:

```
Text(0, 0.5, '')
```