```
#importing libraries

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


#loading datasets

regions = pd.read_csv('/content/noc_regions.csv')
athlete = pd.read_csv('/content/athlete_events.csv')


athlete.head()
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Baske
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	
2	3	Gunnar Nielsen	M	24.0	NaN	NaN	Denmark	DEN	1920	1920	Summer	Antwerpen	Fo

```
regions.head()
```

	NOC	region	notes
0	AFG	Afghanistan	NaN

#join the dataframes

athlete_df = athlete.merge(regions, how = 'left', on = 'NOC')
athlete_df.head()

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Baske
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	
2	3	Gunnar Nielsen	M	24.0	NaN	NaN	Denmark	DEN	1920	1920	Summer	Antwerpen	Fo

athlete_df.shape

(271116, 17)

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
```

athlete_df.describe()

	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

#total null values in each column

athlete_df.isnull().sum()

```
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 about india

athlete_df.query('Team == "India"').head(10)

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

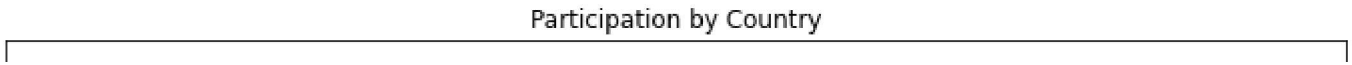
#top 10 participating countries

```
top_10_countries = athlete_df.Team.value_counts().sort_values(ascending = False).head(10)
top_10_countries
```

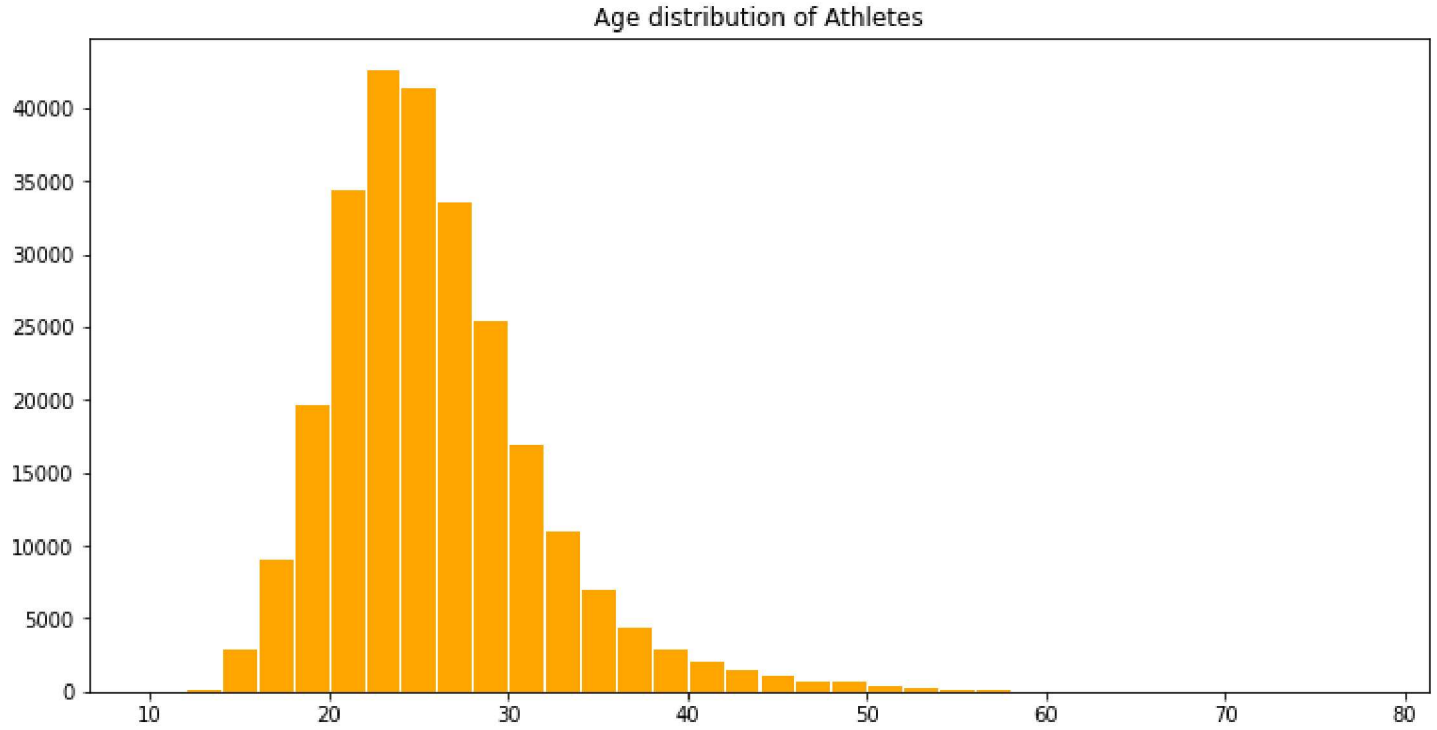
```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Canada            9279
Japan             8289
Sweden            8052
Australia         7513
Hungary           6547
Name: Team, dtype: int64
```

```
#plot for top 10 participating countries

plt.figure(figsize = (12,6))
plt.title('Participation by Country')
sns.barplot(x = top_10_countries.index, y = top_10_countries, palette = 'Set1')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fca563a1350>
```




```
#age distribution of athletes

plt.figure(figsize = (12,6))
plt.title('Age distribution of Athletes')
plt.xlabel = ('Age')
plt.ylabel = ('Number of Athletes')
plt.hist(athlete_df.Age, bins = np.arange(10,80,2), color = 'orange', edgecolor = 'white');
```




```
#winter olympic sports

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

```
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)
```

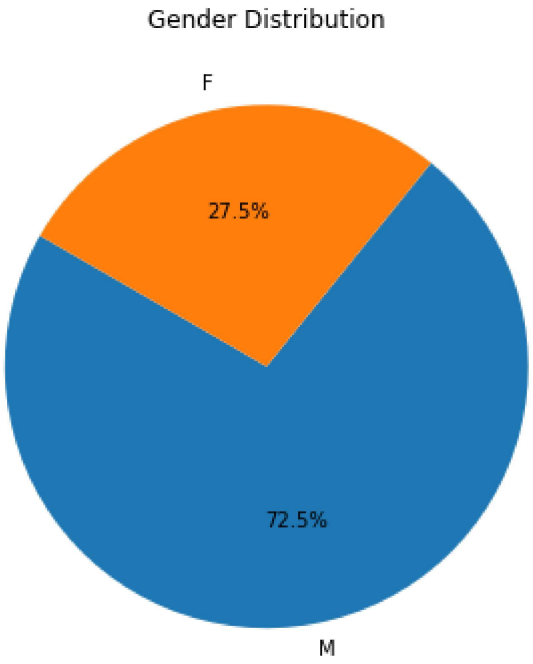
```
#summer olympic sports

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

```
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)
```

```
#male and female athletes

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

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

```
#pie plot for male and female athletes

plt.figure(figsize = (12,6))
plt.title('Gender Distribution')
plt.pie(gender_counts, labels = gender_counts.index, autopct ='%1.1f%%', startangle =150);
```



#total medals

athlete_df.Medal.value_counts()

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

#filtering female athletes

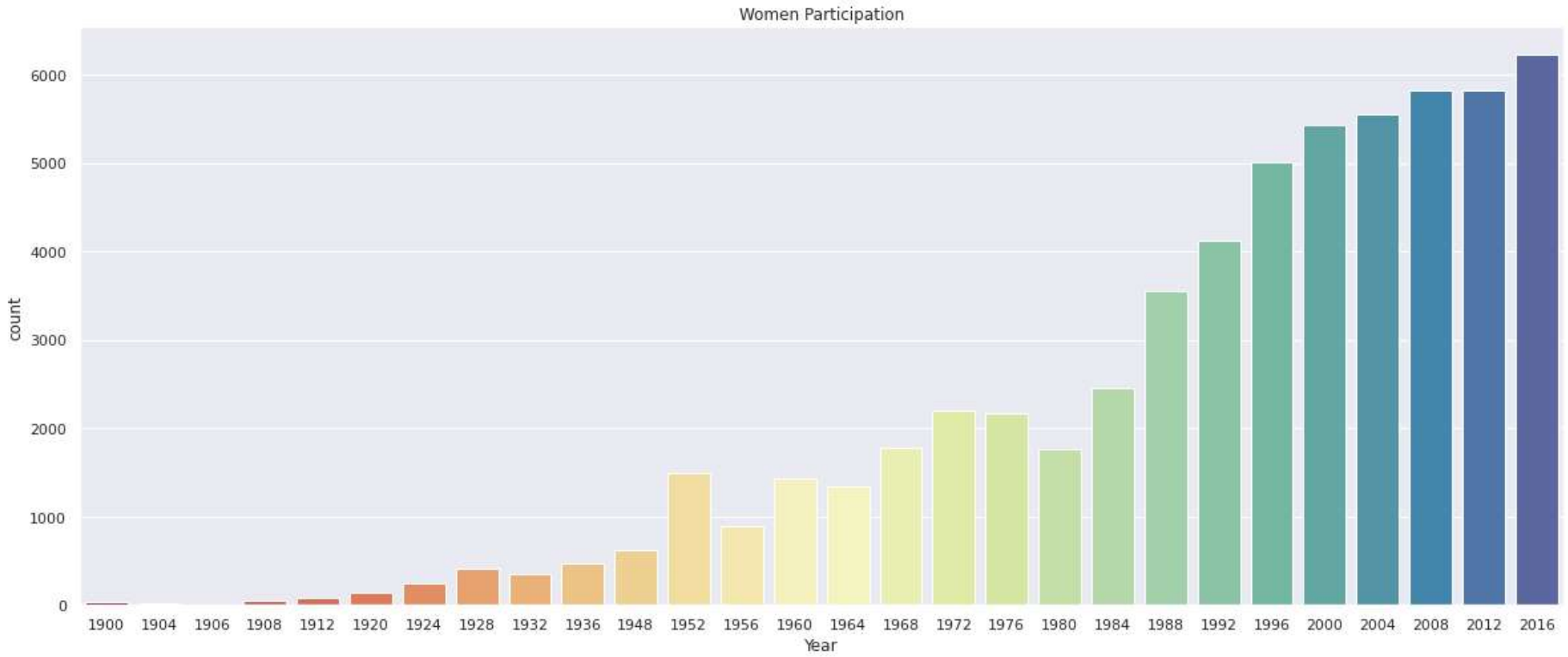
```
women_olympics = athlete_df[(athlete_df.Sex == 'F') & (athlete_df.Season == 'Summer')]
women_olympics
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	
26	8	Cornelia "Cor" Aalten (-Strannood)	F	18.0	168.0	NaN	Netherlands	NED	1932 Summer	1932	Summer	Los Angeles	Athletics	A Wome
27	8	Cornelia "Cor" Aalten (-Strannood)	F	18.0	168.0	NaN	Netherlands	NED	1932 Summer	1932	Summer	Los Angeles	Athletics	A Wom 100
32	13	Minna Maarit Aalto	F	30.0	159.0	55.5	Finland	FIN	1996 Summer	1996	Summer	Atlanta	Sailing	W Wi
33	13	Minna Maarit Aalto	F	34.0	159.0	55.5	Finland	FIN	2000 Summer	2000	Summer	Sydney	Sailing	W Wi
79	21	Ragnhild Margrethe Aamodt	F	27.0	163.0	NaN	Norway	NOR	2008 Summer	2008	Summer	Beijing	Handball	H W H
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
271080	135553	Galina Ivanovna Zybina (-	F	33.0	168.0	80.0	Soviet Union	URS	1964 Summer	1964	Summer	Tokyo	Athletics	A W

#women participation in each olympics

```
sns.set(style = "darkgrid")
plt.figure(figsize = (20,8))
sns.countplot(x = 'Year', data = women_olympics, palette = "Spectral")
plt.title('Women Participation')
```

Text(0.5, 1.0, 'Women Participation')



#top 5 countries with maximum gold medals

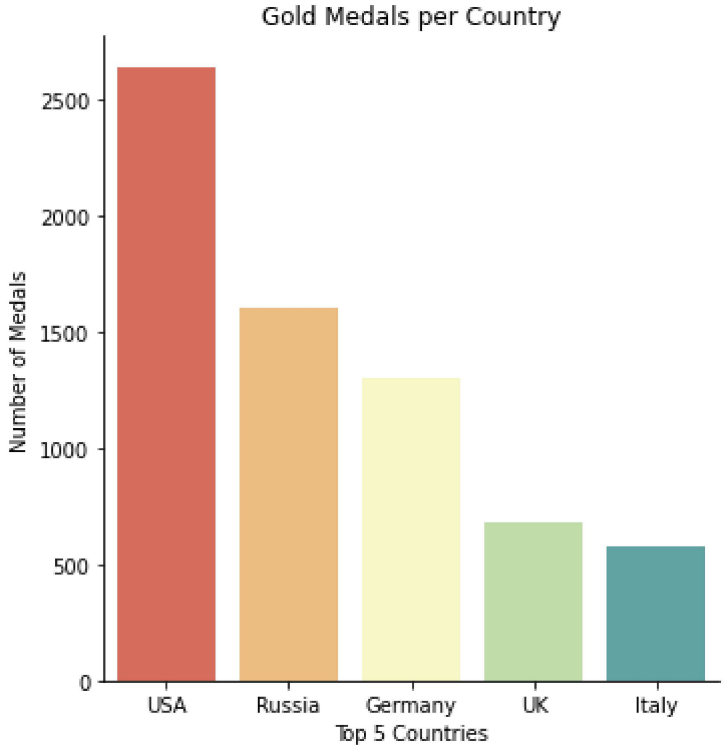
```
gold_medals = athlete_df[(athlete_df.Medal == 'Gold')]
gold_medals.region.value_counts().reset_index(name = 'Medal').head()
```

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

#bar graph for top 5 countries with maximum gold medals

```
total_gold_medals = gold_medals.region.value_counts().reset_index(name = 'Medal').head(5)
g = sns.catplot(x = "index", y = "Medal",data = total_gold_medals, kind = "bar", palette = "Spectral")
g.set_xlabels("Top 5 Countries")
g.set_ylabels("Number of Medals")
plt.title('Gold Medals per Country')
```

```
Text(0.5, 1.0, 'Gold Medals per Country')
```



#recent olympic event year

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

```
2016
```

```
#top 10 countries with highest gold medals at rio olympucs 2016

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

```
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
```

0s completed at 12:07 AM