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

```
athletes = pd.read_csv('/content/athlete_events.csv')
```

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

```
athletes.head()
```

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

```
regions.head()
```

	NOC	region	notes
0	AFG	Afghanistan	NaN
1	AHO	Curacao	Netherlands Antilles
2	ALB	Albania	NaN
3	ALG	Algeria	NaN
4	AND	Andorra	NaN

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

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Se
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Su
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Su
2	3	Gunnar Nielsen Aabye	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Su

```
athletes_df.shape
```

```
(89283, 17)
```

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

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

```
athletes_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 89283 entries, 0 to 89282
Data columns (total 17 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   ID      89283 non-null  int64  
 1   Name    89283 non-null  object 
 2   Sex     89283 non-null  object 
 3   Age     85938 non-null  float64
 4   Height  68625 non-null  float64
 5   Weight  67567 non-null  float64
 6   Team    89283 non-null  object 
 7   NOC     89283 non-null  object 
 8   Games   89283 non-null  object 
 9   Year    89283 non-null  int64  
 10  Season  89283 non-null  object 
 11  City    89283 non-null  object 
 12  Sport   89283 non-null  object 
 13  Event   89283 non-null  object 
 14  Medal   12600 non-null  object 
 15  Region  89211 non-null  object 
 16  Notes   1564 non-null   object 
dtypes: float64(3), int64(2), object(12)
memory usage: 12.3+ MB
```

```
athletes_df.describe()
```

	ID	Age	Height	Weight	Year
count	89283.000000	85938.000000	68625.000000	67567.000000	89283.000000
mean	22910.426498	25.625672	175.559301	70.935412	1977.741832
std	12979.158701	6.453792	10.392730	14.169830	30.118858
min	1.000000	11.000000	127.000000	25.000000	1896.000000
25%	11749.000000	21.000000	168.000000	61.000000	1960.000000
50%	23002.000000	25.000000	175.000000	70.000000	1984.000000
75%	34252.500000	28.000000	183.000000	79.000000	2002.000000
max	45248.000000	88.000000	223.000000	214.000000	2016.000000

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

```
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
```

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

```
ID            0
Name          0
Sex           0
Age        3345
Height    20658
Weight    21716
Team          0
NOC           0
Games         0
Year          0
Season        0
City          0
Sport         0
Event         0
Medal     76683
Region       72
Notes     87719
dtype: int64
```

```
athletes_df.query('Team =="India"').head(5)
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
505	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	A
506	281	S. Abdul Hamid	M	NaN	NaN	NaN	India	IND	1928 Summer	1928	Summer	A
895	512	Shiny Kurisingal Abraham-Wilson	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	
896	512	Shiny Kurisingal Abraham-	F	19.0	167.0	53.0	India	IND	1984 Summer	1984	Summer	

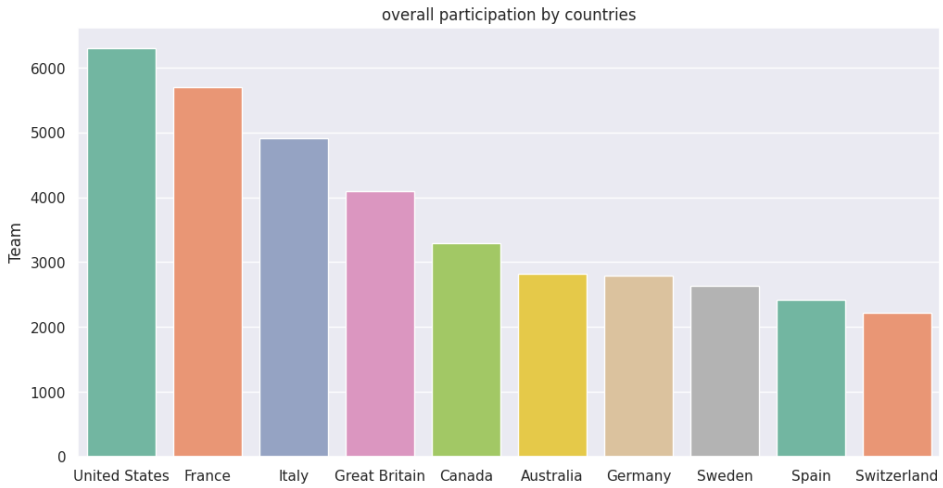
```
athletes_df.query('Team =="Japan"').head(5)
```

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

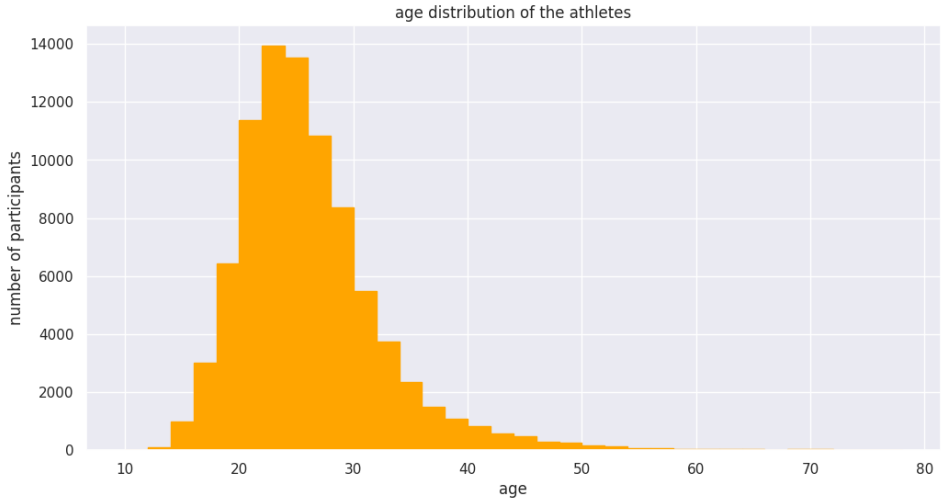
```
top_10_countries = athletes_df.Team.value_counts().sort_values(ascending = False).head(10)
top_10_countries
```

```
United States    6305
France           5695
Italy            4910
Great Britain    4094
Canada           3295
Australia        2820
Germany          2797
Sweden           2634
Spain            2425
Switzerland      2215
Name: Team, dtype: int64
```

```
plt.figure(figsize=(12,6))
#plt.xticks(rotation = 20)
plt.title('overall participation by countries')
sns.barplot(x = top_10_countries.index, y = top_10_countries, palette = "Set2");
```




```
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 = 'orange', edgecolor = 'orange');
```




```
winter_sports = athletes_df[athletes_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_sports = athletes_df[athletes_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',
```

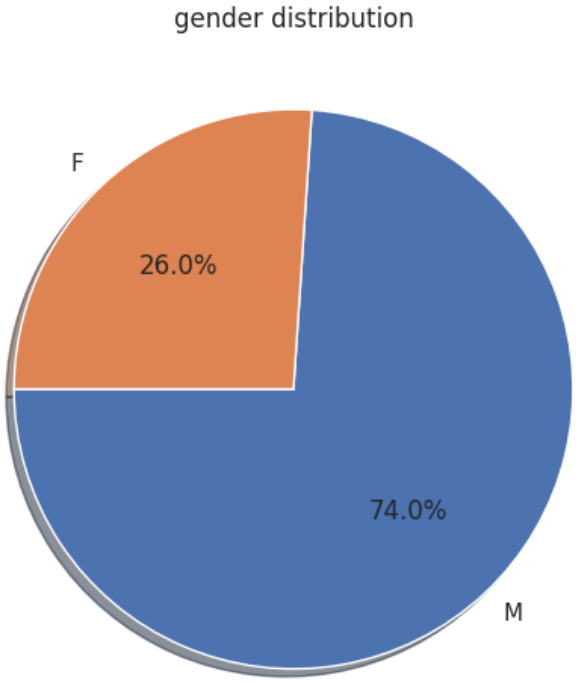
```
       'Shooting', 'Boxing', 'Taekwondo', 'Cycling', 'Diving', 'Canoeing',
       'Tennis', 'Modern Pentathlon', 'Golf', 'Softball', 'Archery',
       'Volleyball', 'Synchronized Swimming', 'Table Tennis', 'Baseball',
       'Rhythmic Gymnastics', 'Rugby Sevens', 'Trampolining',
       'Beach Volleyball', 'Triathlon', 'Rugby', 'Lacrosse', 'Polo',
       'Cricket', 'Ice Hockey', 'Racquets', 'Motorboating', 'Croquet',
       'Figure Skating', 'Jeu De Paume', 'Roque', 'Basque Pelota',
       'Alpinism'], dtype=object)
```

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

```
M    66027
F    23256
Name: Sex, dtype: int64
```

```
plt.figure(figsize=(12,6))
plt.title('gender distribution')
plt.pie(gender_counts, labels = gender_counts.index, autopct = '%1.1f%%', startangle = 180, shadow = True)
```

```
([<matplotlib.patches.Wedge at 0x7f8c31006b30>,
  <matplotlib.patches.Wedge at 0x7f8c30a61ab0>],
 [Text(0.7518041396018317, -0.8029885028302396, 'M'),
  Text(-0.751804214783037, 0.8029884324412533, 'F')],
 [Text(0.4100749852373627, -0.4379937288164943, '74.0%'),
  Text(-0.4100750262452929, 0.43799369042250175, '26.0%')])
```




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

```
Gold      4268
Bronze    4175
Silver    4157
Name: Medal, dtype: int64
```

```
female_participants = athletes_df[(athletes_df.Sex == 'F') & (athletes_df.Season == 'Summer')][['Sex', 'Year']]
female_participants = female_participants.groupby('Year').count().reset_index()
female_participants.head()
```

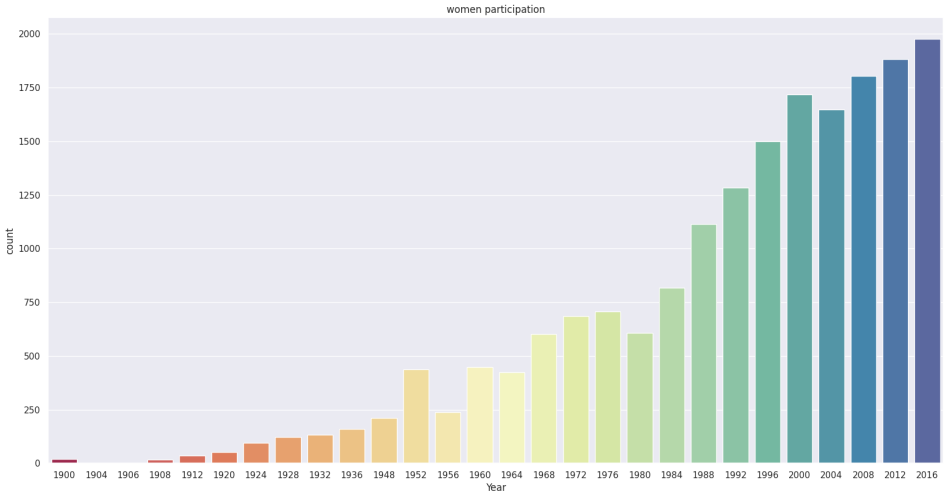
	Year	Sex
0	1900	19

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

	Year	Sex
23	2000	1717
24	2004	1649
25	2008	1803
26	2012	1882
27	2016	1977

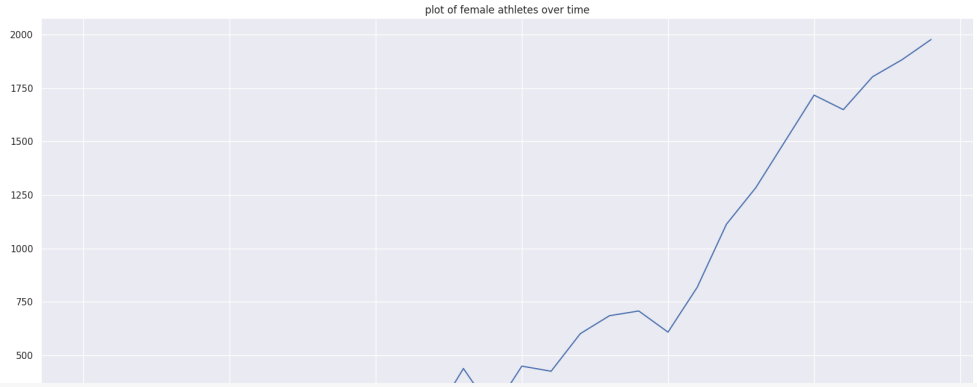
```
womenolympics = athletes_df[(athletes_df.Sex == 'F')&(athletes_df.Season == 'Summer')]
sns.set(style = 'darkgrid')
plt.figure(figsize = (20,10))
sns.countplot(x = 'Year', data = womenolympics, palette = 'Spectral')
plt.title('women participation')
```

```
Text(0.5, 1.0, 'women participation')
```




```
part = womenolympics.groupby('Year')['Sex'].value_counts()
plt.figure(figsize= (20,10))
part.loc[:,'F'].plot()
plt.title('plot of female athletes over time')
```

```
Text(0.5, 1.0, 'plot of female athletes over time')
```




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

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

```
gM = gM[np.isfinite(gM['Age'])]
gM['ID'][gM['Age']>60].count()
```

```
0
```

```
sporting_event = gM['Sport'][gM['Age'] > 60]
plt.figure(figsize=(150,50))
plt.tight_layout()
sns.countplot(data=gM, x='Sport', hue='Age', palette='Set2')

plt.title('gold medal for athletes over 60 year')
```

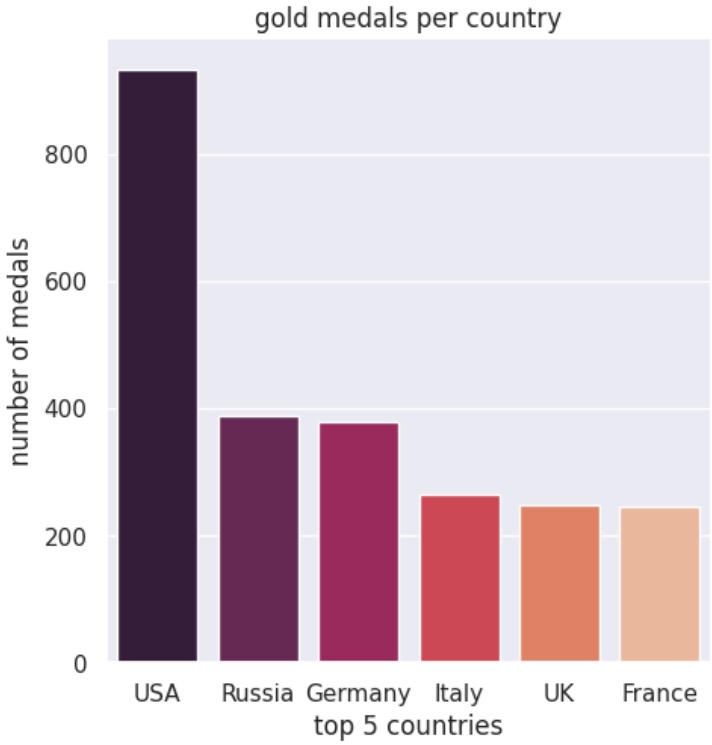
```
Text(0.5, 1.0, 'gold medal for athletes over 60 year')
```

```
gM.Region.value_counts().reset_index(name = 'Medal').head(5)
```

	index	Medal
0	USA	934
1	Russia	389
2	Germany	380
3	Italy	265
4	UK	249

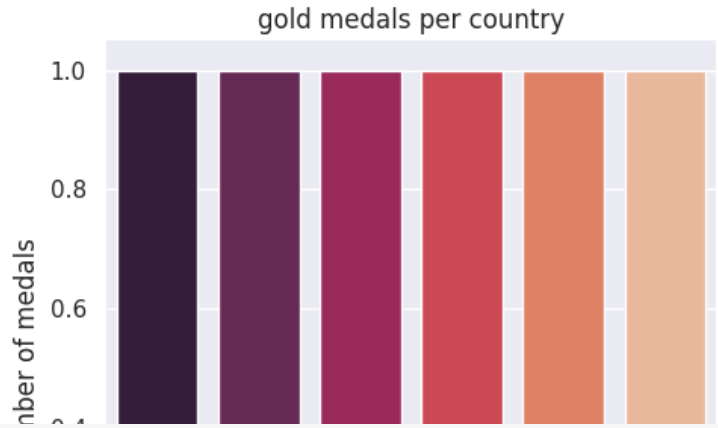
```
totalgM = gM.Region.value_counts().reset_index(name = 'Medal').head(6)
g = sns.catplot(x = "index", y = "Medal", data = totalgM, height = 5, kind = "bar", palette = "rocket" )
g.despine(left = True)
g.set_xlabels("top 5 countries")
g.set_ylabels("number of medals")
plt.title("gold medals per country")
```

```
Text(0.5, 1.0, 'gold medals per country')
```




```
totalgM = gM.Region.value_counts().reset_index(name = 'Medal').tail(6)
g = sns.catplot(x = "index", y = "Medal", data = totalgM, height = 5, kind = "bar", palette = "rocket" )
g.despine(left = True)
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')
```

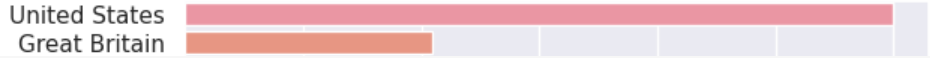



```
max_year = athletes_df.Year.max()
print(max_year)
team_names = athletes_df[(athletes_df.Year == max_year ) & (athletes_df.Medal == 'Gold')].Team
team_names.value_counts().head(10)
```

```
2016
United States    60
Great Britain    21
Brazil           13
Australia        13
Germany          12
Russia           12
China            11
France            6
Jamaica           6
Italy             5
Name: Team, dtype: int64
```

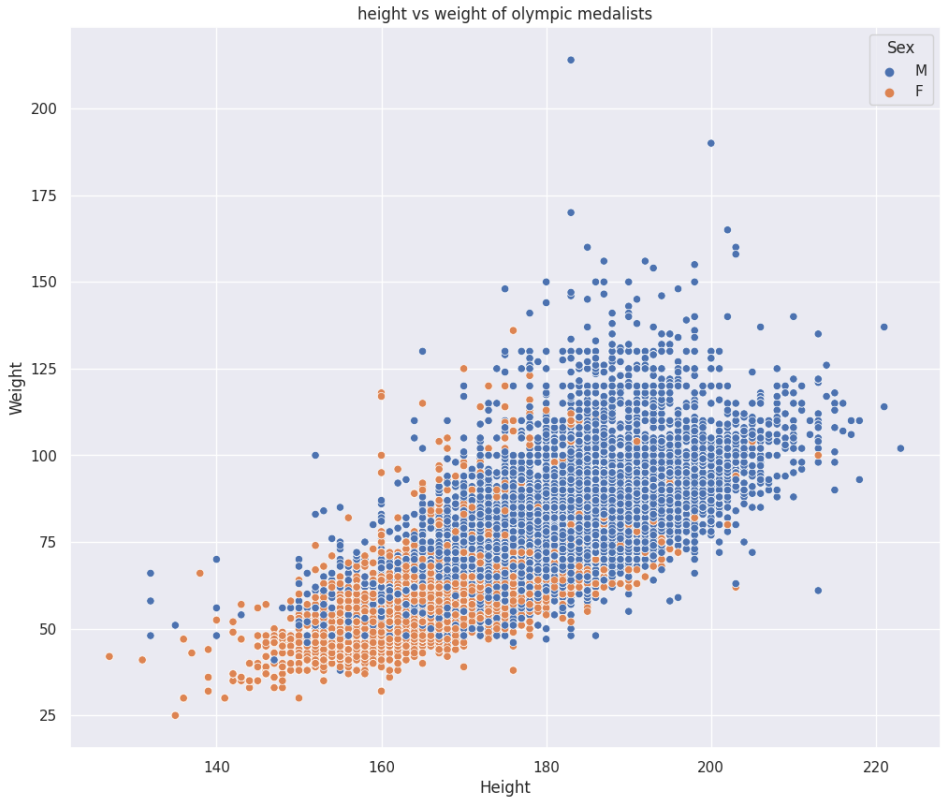
```
team_counts = team_names.value_counts().head(20)

if not team_counts.empty:
    Team = team_counts.min()
    sns.barplot(x=team_counts.values, y=team_counts.index)
    plt.ylabel(None)
    plt.xlabel('country wise medals for the year 2016')
    plt.show()
else:
    print("No data available to generate the bar plot.")
```




```
not_null_medals = athletes_df[(athletes_df['Height'].notnull()) & (athletes_df['Weight'].notnull())]
plt.figure(figsize = (12,10))
axis = sns.scatterplot(x = 'Height', y = 'Weight', data = not_null_medals, hue = 'Sex')
plt.title("height vs weight of olympic medalists")
```

```
Text(0.5, 1.0, 'height vs weight of olympic medalists')
```



✓ 6s completed at 12:29 PM