In [89]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [220…
```
df= pd.read_csv('athlete_events.csv')
region_df = pd.read_csv('noc_regions.csv')
```

In [221…
```
df.head()
```

Out[221]:

	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	Barcel
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Lon
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwer
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	P
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Cal

In [92]:
```
df.tail()
```

Out[92]:

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

In [93]:
```
df.shape
```

Out[93]:
```
(271116, 15)
```

In [225…
```
region_df.rename(columns={'region':'Regions','notes':'Notes'},inplace=True)
```

In [226…
```
df.columns
```

Out[226]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Games',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal'],
      dtype='object')
```

In [227…
```
region_df.columns
```

Out[227]:
```
Index(['NOC', 'Regions', 'Notes'], dtype='object')
```

In [95]:
```
df.head()
```

Out[95]:

	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	Barcel
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	Lon
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwer
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	P
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calg

In [96]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 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 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

In [97]:
```
df.describe()
```

Out[97]:

	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 [98]:
```
df.isna().any()
```

Out[98]:
```
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
dtype: bool
```

In [99]:
```
df.isnull().sum()
```

Out[99]:
```
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
dtype: int64
```

In [100…
```
df.duplicated().sum()
```

Out[100]:
```
1385
```

In [101…
```
df.drop_duplicates(inplace=True)
```

In [102…
```
df.duplicated().sum()
```

Out[102]:
```
0
```

In [231…
```
# join the datasets
df=df.merge(region_df , how = "left" , on = "NOC")
```

In [232…
```
df.query("Team  == 'India' ").head()
```

Out[232]:

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

In [233…
```
df.query("Team  == 'Japan' ").head()
```

Out[233]:

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

In [234…

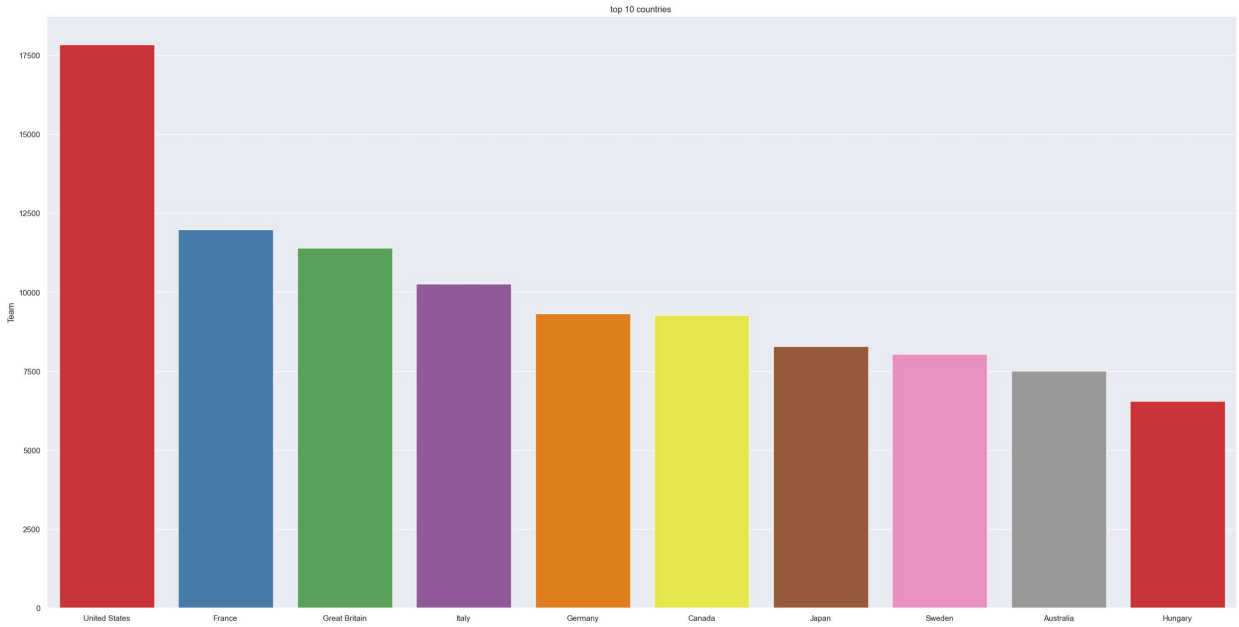
```
Top_10_countreis=df.Team.value_counts().sort_values(ascending=False).head(10)
Top_10_countreis
```

Out[234]:

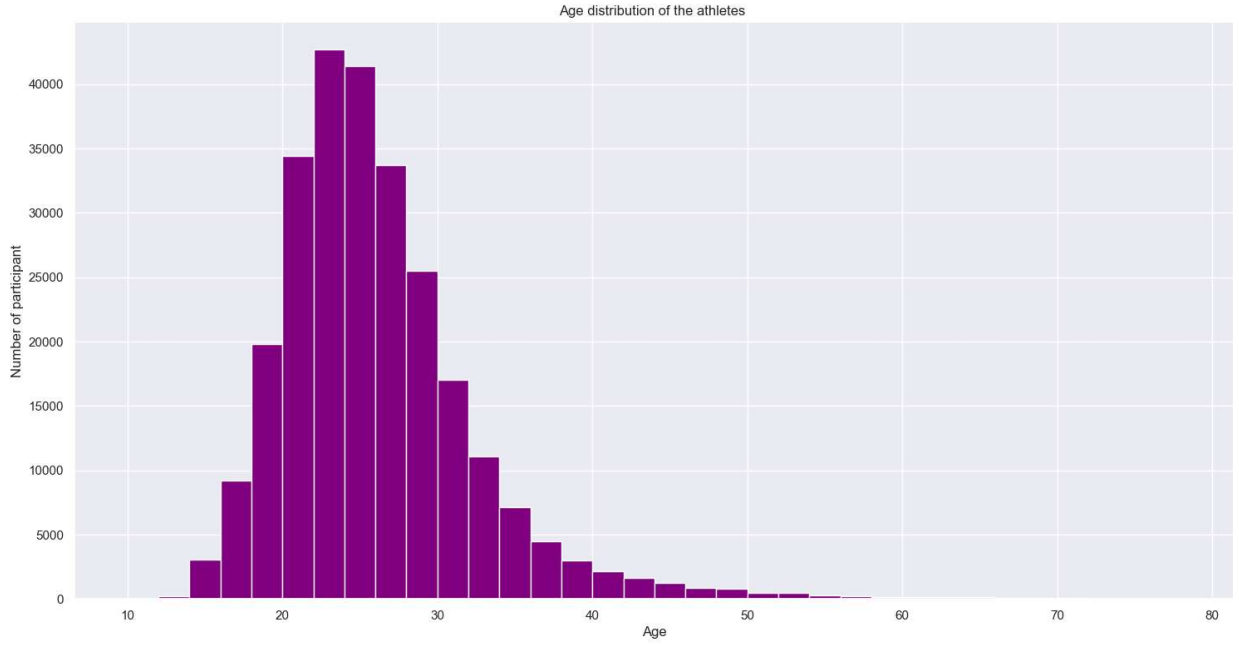
```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Canada            9279
Japan             8289
Sweden            8052
Australia         7513
Hungary           6547
Name: Team, dtype: int64
```

In [246…

```
plt.figure(figsize=(30,15))
plt.title('top 10 countries ')
sns.barplot( x = Top_10_countreis.index , y = Top_10_countreis , palette = 'Set1');
```



In [236…
```
plt.figure(figsize=(18,9))
plt.title('Age distribution of the athletes ')
plt.xlabel('Age')
plt.ylabel('Number of participant')
plt.hist(df.Age , bins = np.arange(10,80,2) , color='purple', edgecolor='white');
```



In [237…
```
winter_sports=df[df.Season=='Winter'].Sport.unique()
winter_sports
```

Out[237]:
```
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 [238…
```
Summer_sports=df[df.Season=='Summer'].Sport.unique()
Summer_sports
```

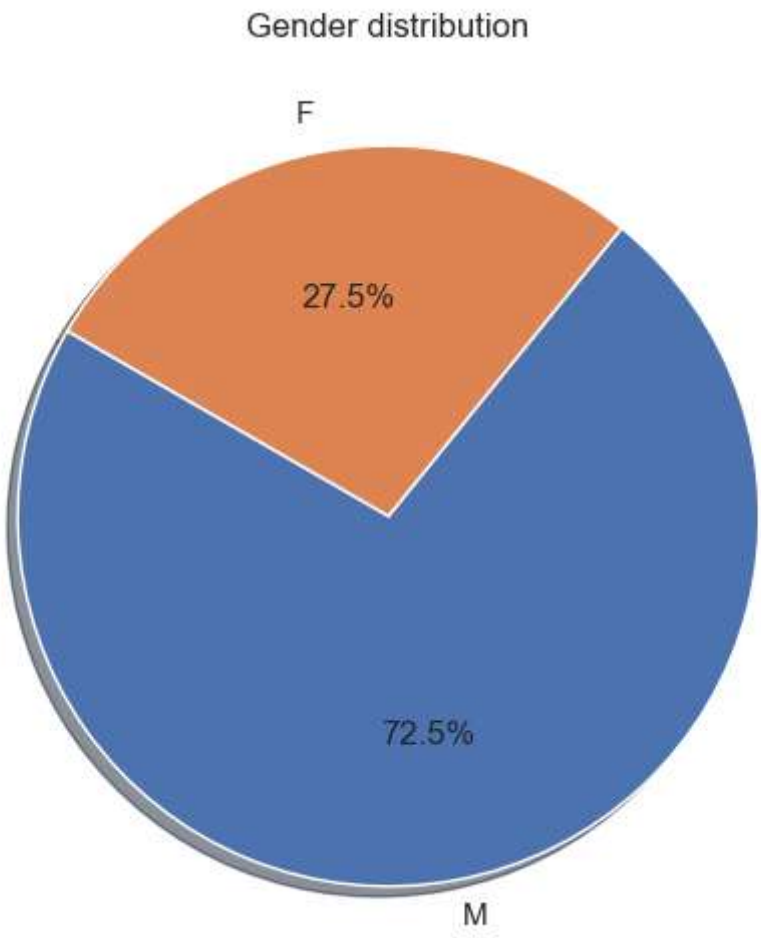
Out[238]:
```
array(['Basketball', 'Judo', 'Football', 'Tug-Of-War', 'Athletics',
       'Swimming', 'Badminton', 'Sailing', 'Gymnastics',
       'Art Competitions', 'Handball', 'Weightlifting', 'Wrestling',
       'Water Polo', 'Hockey', 'Rowing', 'Fencing', 'Equestrianism',
       'Shooting', 'Boxing', '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 [239…
```
# male and female participants
gender_counts = df.Sex.value_counts()
gender_counts
```

Out[239]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [240…
```
plt.figure(figsize=(12,6))
plt.title('Gender distribution')
plt.pie(gender_counts, autopct='%1.1f%%', labels=gender_counts.index, startangle=150,
```

Out[240]:
```
([<matplotlib.patches.Wedge at 0x1e41795fe90>,
  <matplotlib.patches.Wedge at 0x1e41797ae90>],
 [Text(0.20089640434146097, -1.081499253223354, 'M'),
  Text(-0.2008963030841931, 1.081499272032628, 'F')],
 [Text(0.10957985691352415, -0.5899086835763748, '72.5%'),
  Text(-0.10957980168228713, 0.5899086938359788, '27.5%')])
```



In [247…
```
df['Medal'].value_counts()
```

Out[247]:
```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64
```

In [248…
```
female_participants = df[(df.Sex=='F')&(df.Season=='Summer')][['Sex',"Year"]]
female_participants = female_participants.groupby("Year").count().reset_index()
female_participants.head(10)
```

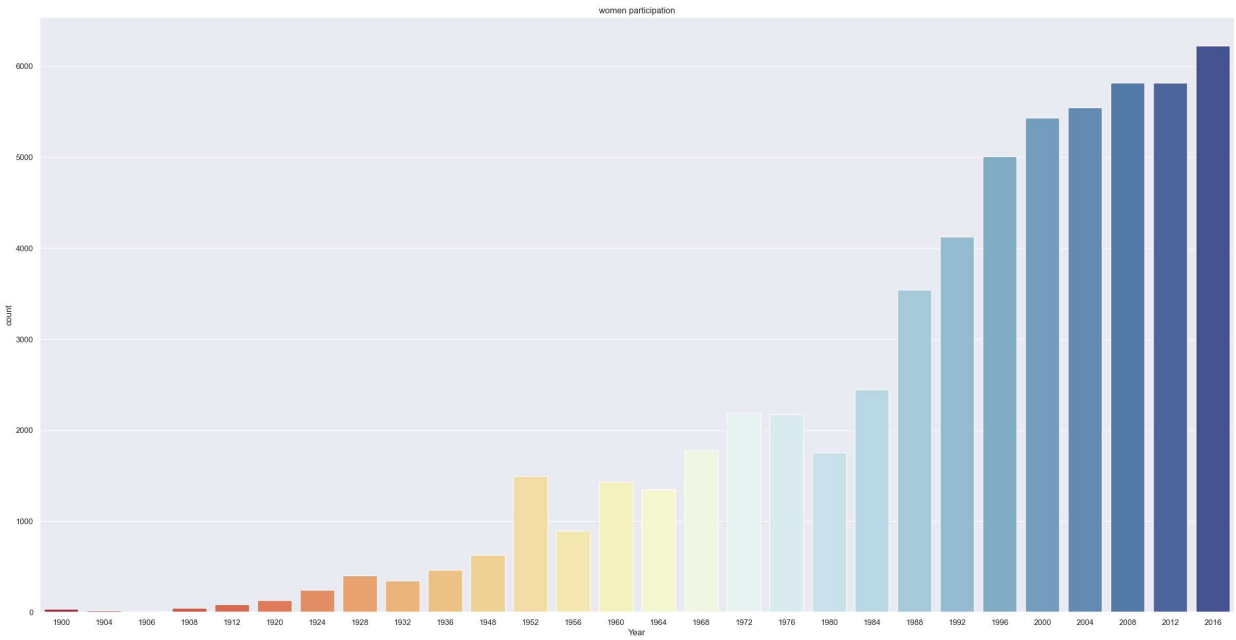
Out[248]:

	Year	Sex
0	1900	33
1	1904	16
2	1906	11
3	1908	47
4	1912	87
5	1920	134
6	1924	244
7	1928	404
8	1932	347
9	1936	468

In [249…
```
Women_Olympics=df[(df.Sex=='F')&(df.Season=='Summer')]
```

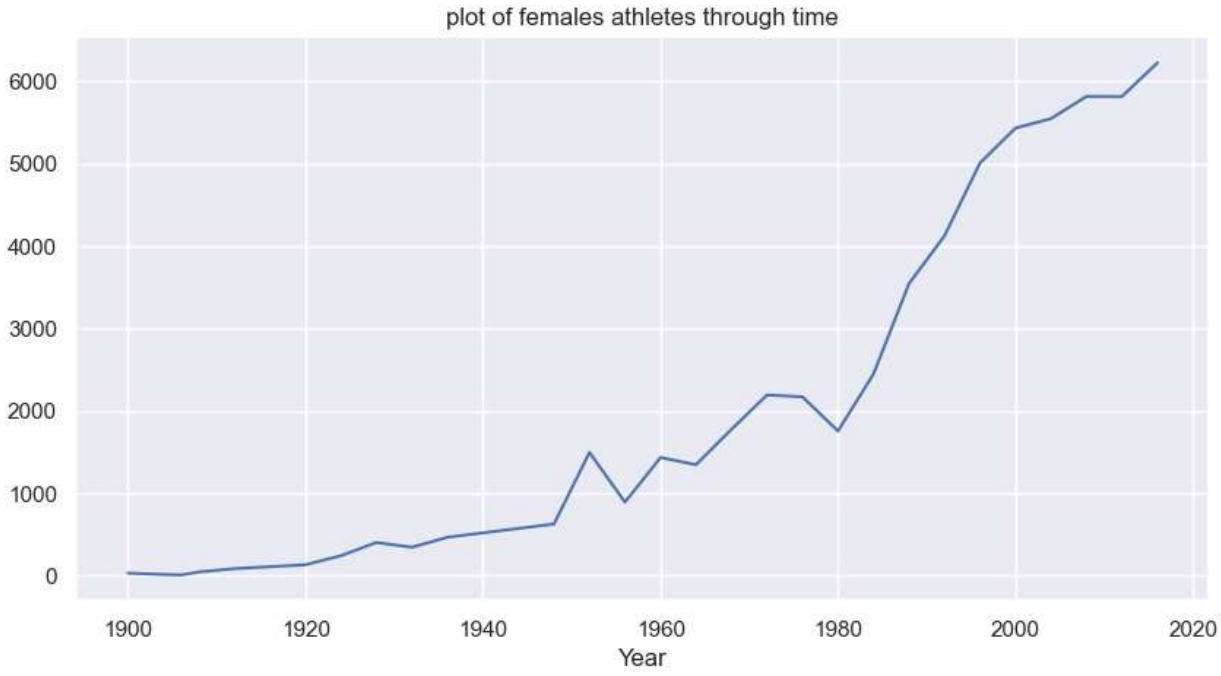
In [250…
```
sns.set(style="darkgrid")
plt.figure(figsize=(30,15))
sns.countplot(x='Year', data=Women_Olympics, palette='RdYlBu')
plt.title('women participation')
```

Out[250]:
```
Text(0.5, 1.0, 'women participation')
```



In [251…
```
part=Women_Olympics.groupby('Year')['Sex'].value_counts()
plt.figure(figsize=(10,5))
part.loc[:,'F'].plot()
plt.title('plot of females athletes through time')
```

Out[251]:
```
Text(0.5, 1.0, 'plot of females athletes through time')
```



In [252…
```
gold_medals=df[df.Medal=='Gold']
gold_medals
```

Out[252]:

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

13372 rows × 17 columns

In [253…
```
# only the value are different from NANS
gold_medals = gold_medals[np.isfinite(gold_medals['Age'])]
```

In [256…
```
# gold medals from each country
gold_medals.Regions.value_counts().reset_index(name='Medal').head(6)
```

Out[256]:

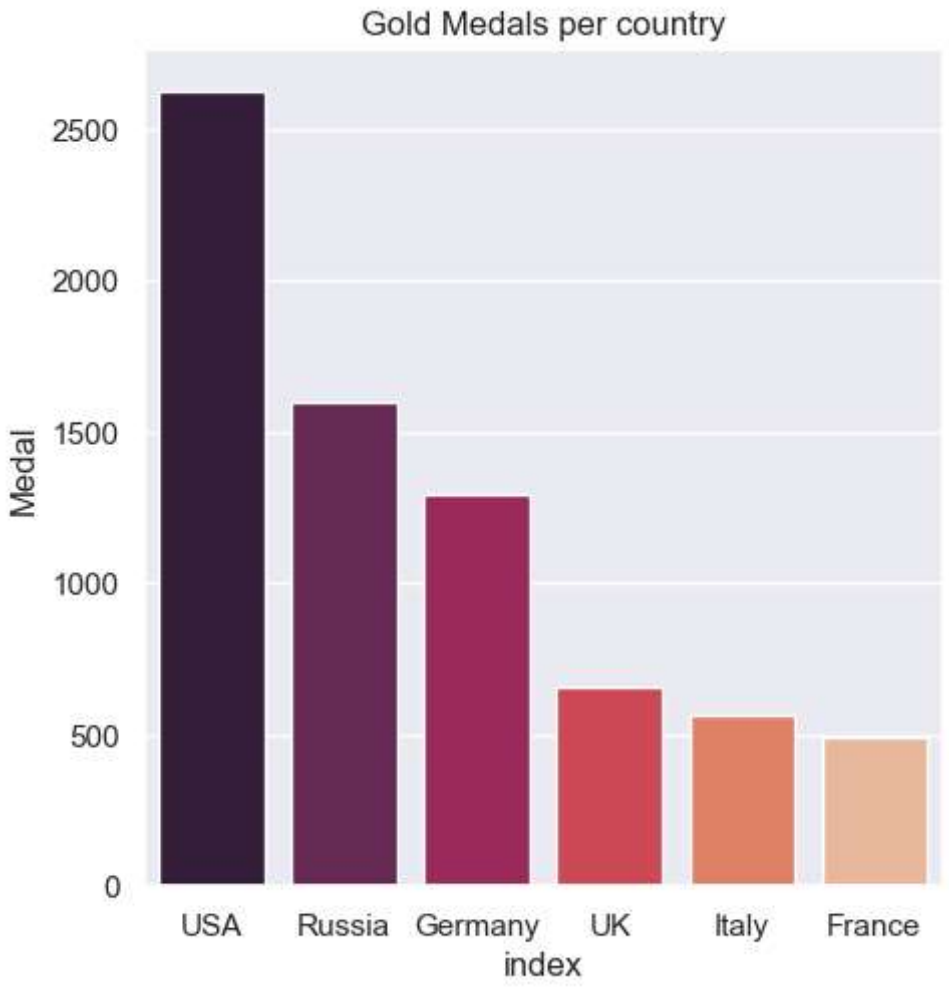
	index	Medal
0	USA	2627
1	Russia	1599
2	Germany	1293
3	UK	657
4	Italy	567
5	France	491

In [274…

```
total_gold_models = gold_medals.Regions.value_counts().reset_index(name='Medal').head(
g=sns.catplot(x='index',y='Medal',data=total_gold_models , height = 5 ,
              kind = "bar" , palette = 'rocket' )
g.despine(left=True)
g.sex_xlabels=("Top 5 countries")
g.sex_ylabels=(" Number of Medals")
plt.title("Gold Medals per country")
```

Out[274]:

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



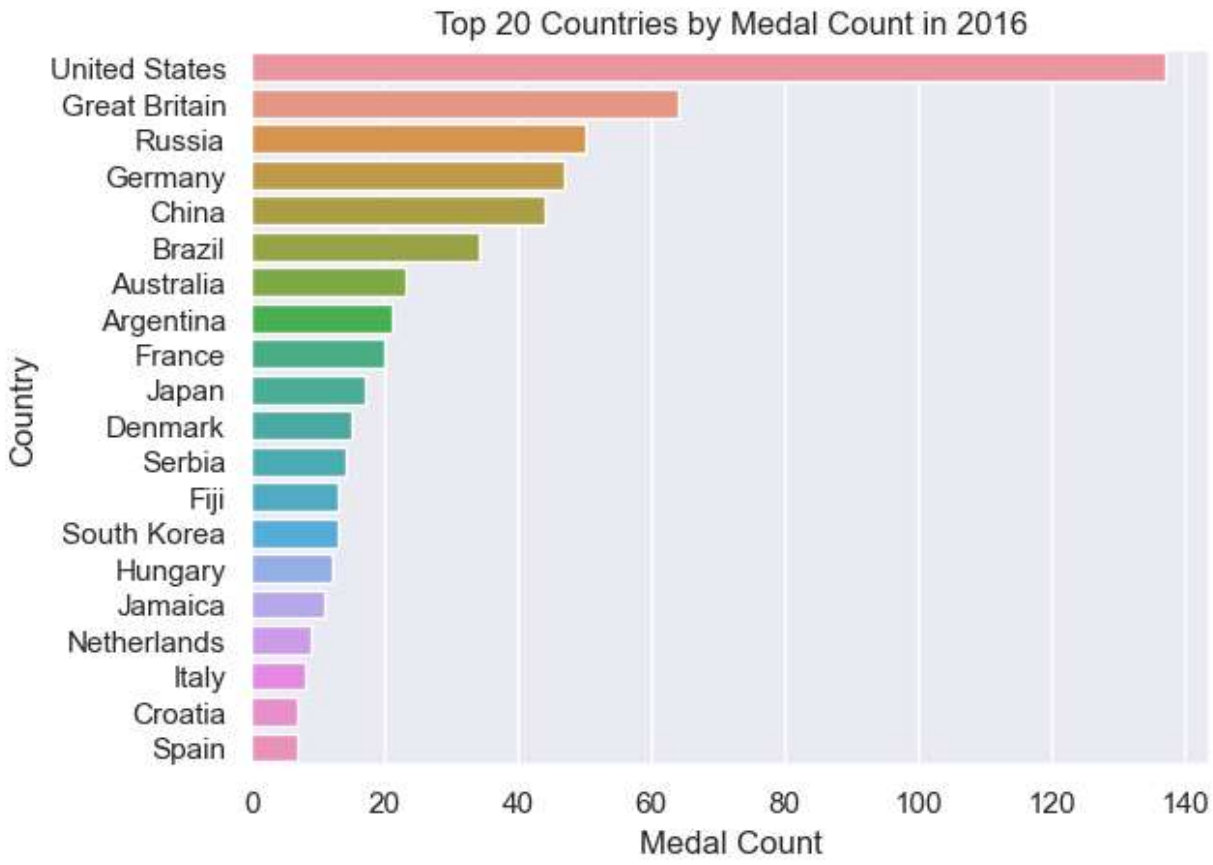
In [276…

```
max_year=df.Year.max()
max_year
team_names=df[(df.Year==max_year)&(df.Medal=="Gold")].Team
team_names.value_counts().head(10)
```

Out[276]:
```
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
```

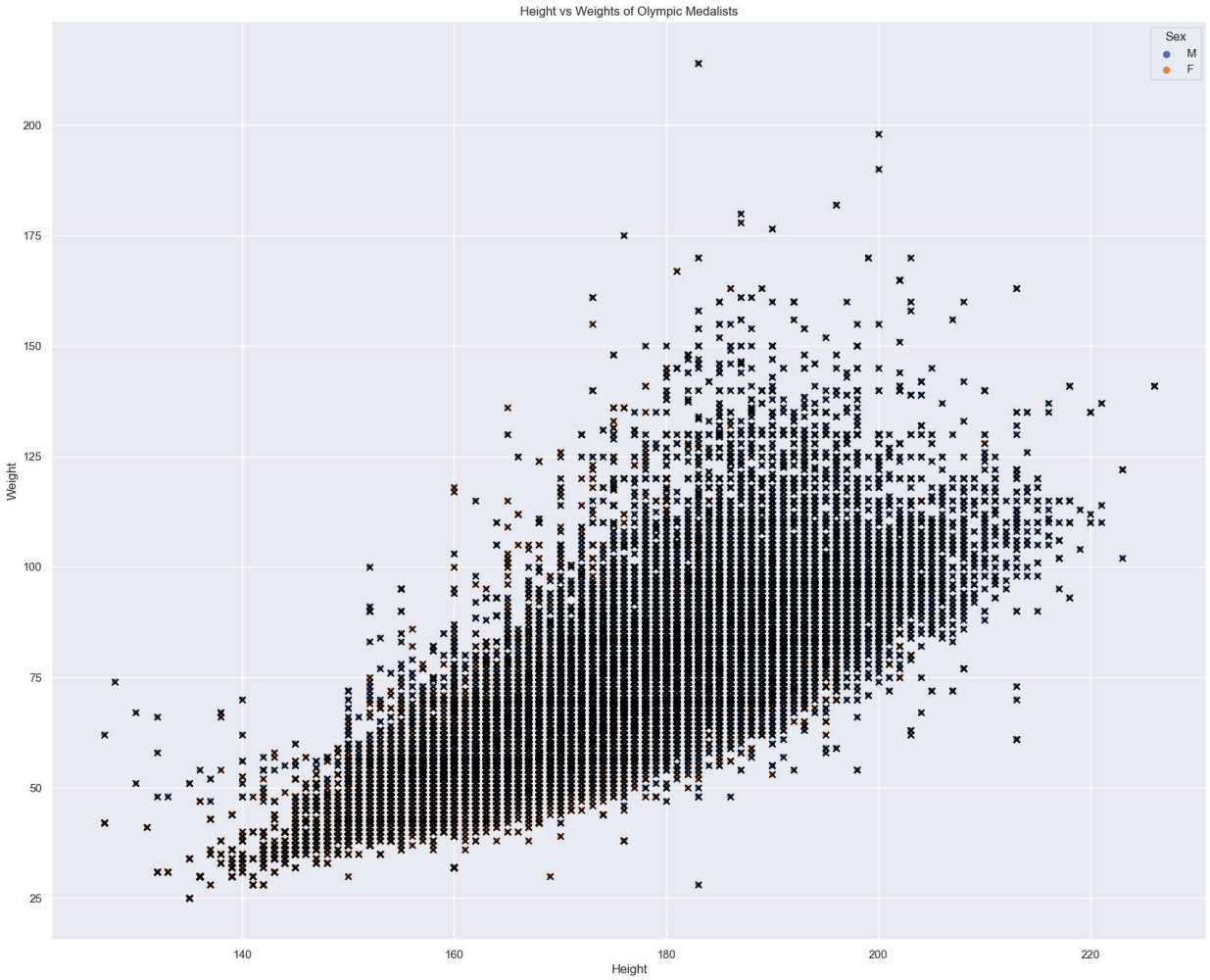
In [291…
```
sns.barplot(x=team_names.value_counts().head(20), y=team_names.value_counts().head(20)
plt.xlabel('Medal Count')
plt.ylabel('Country')
plt.title('Top 20 Countries by Medal Count in 2016')
```

Out[291]:
```
Text(0.5, 1.0, 'Top 20 Countries by Medal Count in 2016')
```



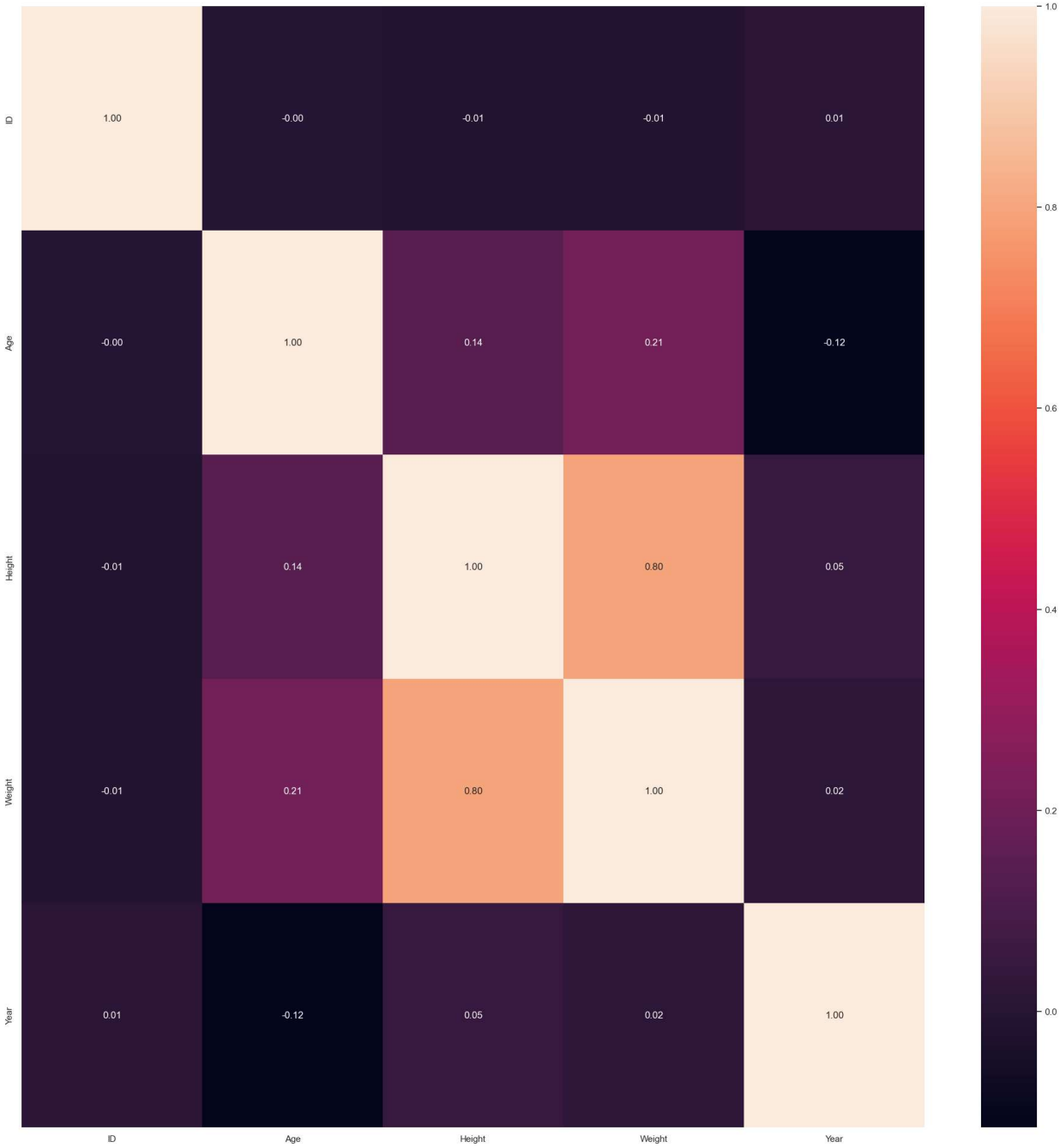
In [300…
```
non_null_medals = df[(df['Height'].notnull()) & (df['Weight'].notnull())]
plt.figure(figsize=(20, 16))
axis = sns.scatterplot(x='Height', y='Weight', data=non_null_medals, hue='Sex')

marker_style = 'x'
axis.scatter(non_null_medals['Height'], non_null_medals['Weight'], marker=marker_style

plt.title('Height vs Weights of Olympic Medalists')
```

Out[300]:
```
Text(0.5, 1.0, 'Height vs Weights of Olympic Medalists')
```



In [302…
```
numeric_columns = df.select_dtypes(include=[np.number])
plt.figure(figsize=(25 ,25))
sns.heatmap(numeric_columns.corr(), annot=True, fmt='.2f')
```

Out[302]:
```
<Axes: >
```



Thanks for getting to this point