In [5]:
```
# importing libraraies

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

In [6]:
```
#importing dataset

data = pd.read_csv("dataset_olympics.csv")
```

In [7]:
```
#checking how data looks-like

data.head()
```

Out[7]:

	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	Ba
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	An
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 [8]:
```
#data information

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   ID      70000 non-null  int64  
 1   Name    70000 non-null  object 
 2   Sex     70000 non-null  object 
 3   Age     67268 non-null  float64
 4   Height  53746 non-null  float64
 5   Weight  52899 non-null  float64
 6   Team    70000 non-null  object 
 7   NOC     70000 non-null  object 
 8   Games   70000 non-null  object 
 9   Year    70000 non-null  int64  
 10  Season  70000 non-null  object 
 11  City    70000 non-null  object 
 12  Sport   70000 non-null  object 
 13  Event   70000 non-null  object 
 14  Medal   9690 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 8.0+ MB
```

In [9]:
```
#statistics of data

data.describe()
```

Out[9]:

	ID	Age	Height	Weight	Year
count	70000.000000	67268.000000	53746.000000	52899.000000	70000.000000
mean	18081.846986	25.644645	175.505303	70.900216	1977.766457
std	10235.613253	6.485239	10.384203	14.217489	30.103306
min	1.000000	11.000000	127.000000	25.000000	1896.000000
25%	9325.750000	21.000000	168.000000	61.000000	1960.000000
50%	18032.000000	25.000000	175.000000	70.000000	1984.000000
75%	26978.000000	28.000000	183.000000	79.000000	2002.000000
max	35658.000000	88.000000	223.000000	214.000000	2016.000000

In [10]:
```
#some more insight into data

data.describe(include=["object"])
```

Out[10]:

	Name	Sex	Team	NOC	Games	Season	City	Sport	Event	Meda
count	70000	70000	70000	70000	70000	70000	70000	70000	70000	969(
unique	35556	2	827	226	51	2	42	65	744	3
top	Oksana Aleksandrovna Chusovitina	M	United States	USA	2016 Summer	Summer	London	Athletics	Football Men's Football	Gol(
freq	29	51877	4979	5216	3675	58467	6034	10629	1738	329

In [11]:
```
# checking for missing data

data.isna().sum()
```

Out[11]:
```
ID            0
Name          0
Sex           0
Age        2732
Height    16254
Weight    17101
Team          0
NOC           0
Games         0
Year          0
Season        0
City          0
Sport         0
Event         0
Medal     60310
dtype: int64
```

In [12]:
```
#checking for duplicate

data.duplicated().sum()
```

Out[12]:
```
383
```

In [13]:
```
#removing duplicate

data.drop_duplicates(inplace= True)
```

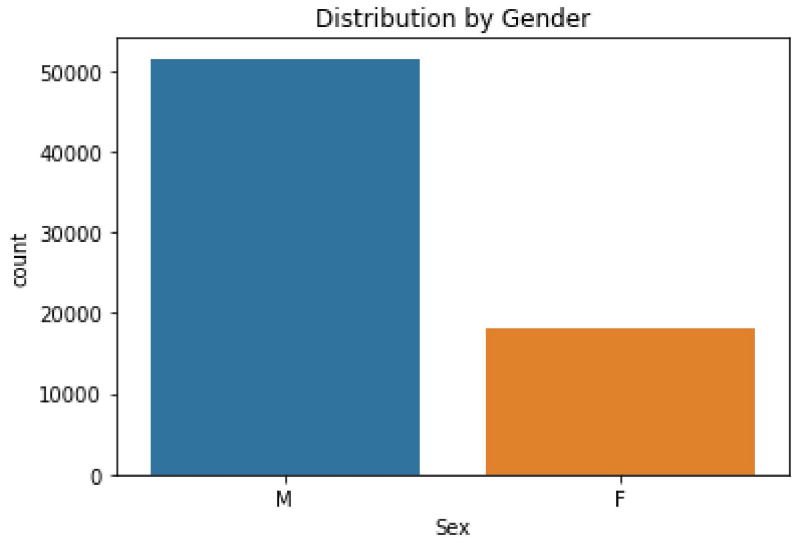
In [14]:
```
#again checking for duplicates,if any

data.duplicated().sum()
```

Out[14]:
```
0
```

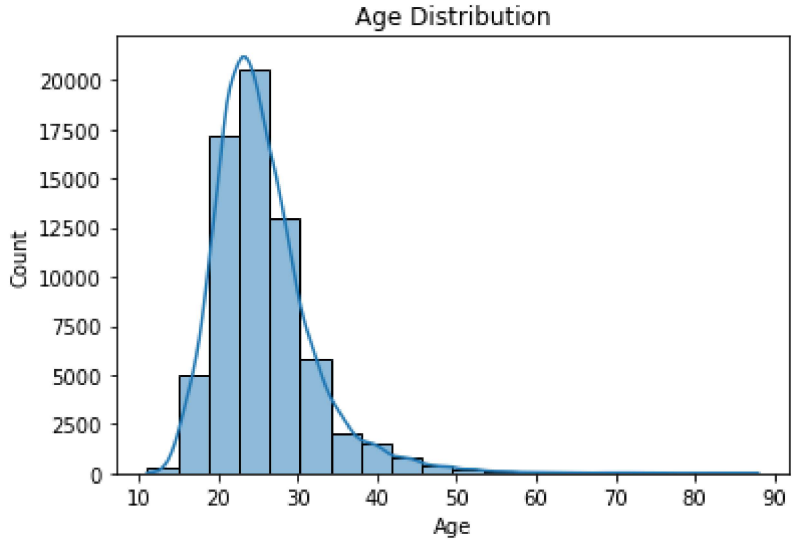
In [15]:
```
#graphical representation of Distribution by Gender

sns.countplot(data= data, x = "Sex")
plt.title("Distribution by Gender")
plt.show()
```



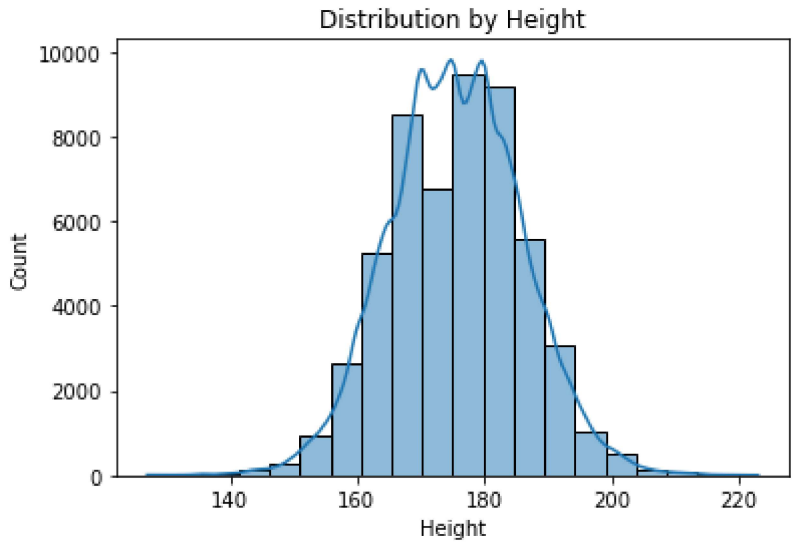
In [16]:
```
#Age distribution

sns.histplot(data=data,x="Age", bins=20,kde=True)
plt.title("Age Distribution")
plt.show()
```



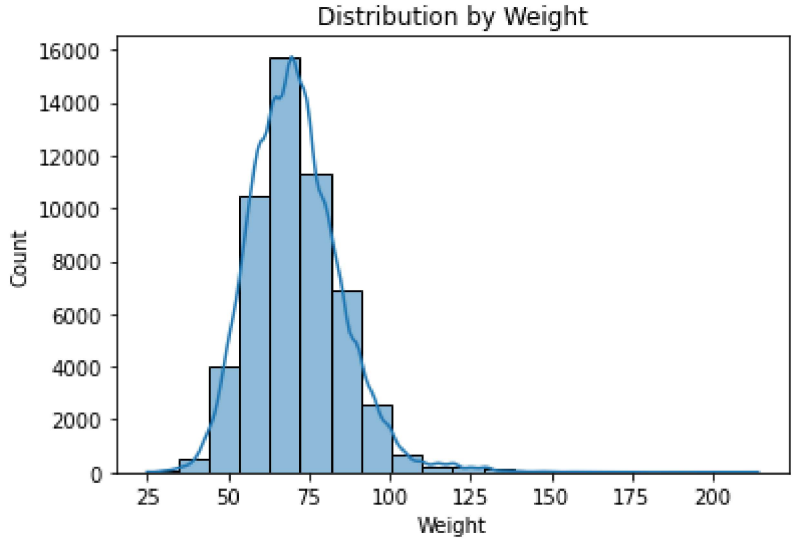
In [17]:
```
#Distribution by height

sns.histplot(data = data,x="Height",bins=20,kde=True)
plt.title("Distribution by Height")
plt.show()
```



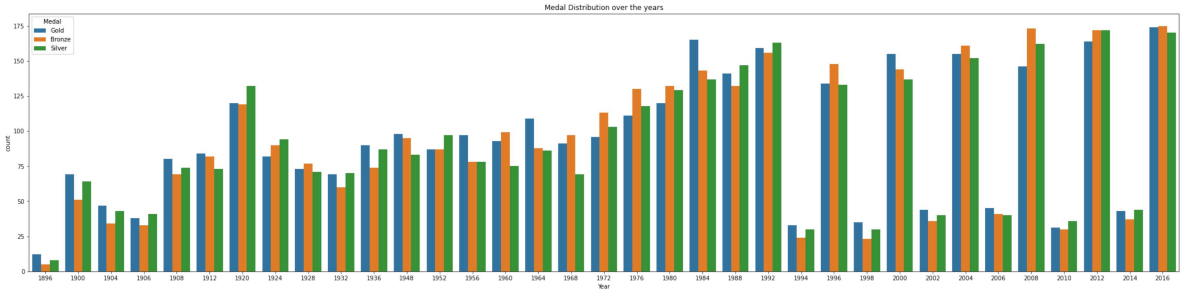
In [18]:
```
#Distribution by weight

sns.histplot(data = data,x="Weight",bins=20,kde=True)
plt.title("Distribution by Weight")
plt.show()
```



In [36]:
```
#Distribution of Medal(Gold,Silver,Bronze) over the years

sns.countplot(data=data,x="Year",hue="Medal")
plt.rcParams["figure.figsize"] = (35,10)
plt.title("Medal Distribution over the years")
plt.show()
```



In [20]:
```
#Average of age of participants over the years

year_avg_age = data.groupby("Year")["Age"].mean()
print(year_avg_age)
```

```
Year
1896    23.029412
1900    29.119883
1904    27.063241
1906    26.989474
1908    27.000000
1912    27.965552
1920    29.241135
1924    28.252267
1928    27.973564
1932    29.606987
1936    27.245665
1948    28.363170
1952    26.273684
1956    26.316156
1960    25.136156
1964    24.852107
1968    24.316722
1972    24.126448
1976    23.656820
1980    23.312364
1984    24.060328
1988    24.257374
1992    24.637827
1994    24.487516
1996    25.338210
1998    25.143860
2000    25.435177
2002    26.029095
2004    25.780111
2006    26.091716
2008    25.685148
2010    26.150776
2012    25.993485
2014    26.082814
2016    26.259592
Name: Age, dtype: float64
```

In [21]:
```
#Maximum height of a participants

sport_median_height = data.groupby("Sport")["Height"].median()
print(sport_median_height.max())
```

```
190.0
```

In [22]:
```
#Maximum weight of a participants

sport_median_weight = data.groupby("Sport")["Weight"].median()
print(sport_median_weight.max())
```

```
95.0
```

In [39]:
```
#Average weight of a participants

sport_median_weight = data.groupby("Sport")["Weight"].median()
print(sport_median_weight.min())
```

```
48.0
```

In [24]:
```
#Most Gold winning countries

country_gold_meadals = data[data["Medal"] == "Gold"].groupby("NOC")["Medal"].
print(country_gold_meadals.max())
```

```
747
```

In [25]:
```
country_gold_meadals[country_gold_meadals==747]
```

Out[25]:
```
NOC
USA    747
Name: Medal, dtype: int64
```

In [26]:
```
#Most Silver winning countries

country_silver_meadals = data[data["Medal"] == "Silver"].groupby("NOC")["Meda
print(country_silver_meadals.max())
```

```
448
```

In [27]:
```
country_silver_meadals[country_silver_meadals==448]
```

Out[27]:
```
NOC
USA    448
Name: Medal, dtype: int64
```

In [28]:
```
#Most Bronze winning countries

country_bronze_meadals = data[data["Medal"] == "Bronze"].groupby("NOC")["Meda
print(country_bronze_meadals.max())
```

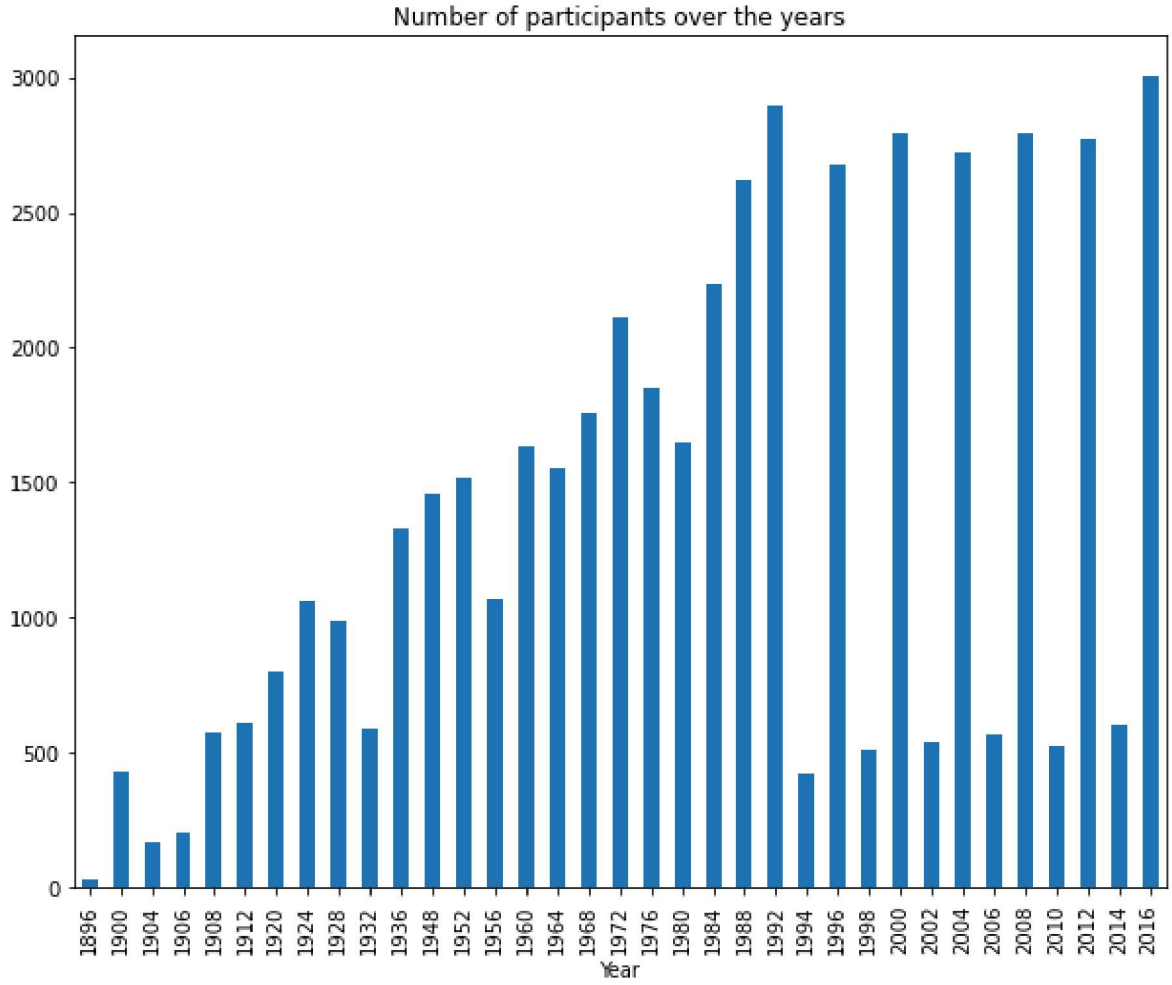
```
366
```

In [29]:
```
country_bronze_meadals[country_bronze_meadals==366]
```

Out[29]:
```
NOC
USA    366
Name: Medal, dtype: int64
```

In [34]:
```
#Number of participants over the years (Graphical Representation)

year_part_count=data.groupby("Year")["ID"].nunique()
year_part_count.plot(kind = "bar")
plt.rcParams["figure.figsize"] = (10,8)
plt.title("Number of participants over the years")
plt.show()
```



In [ ]:

In [ ]: