In [17]:

```python
import pandas as pd
from matplotlib import pyplot as plt
```

In [18]:

```python
data1=pd.read_csv('athletes.csv')
```

In [20]:

```python
data2=pd.read_csv('countries.csv')
```

In [23]:

```python
data3=pd.read_csv('events.csv')
```

In [19]:

```python
data1
```

Out[19]:

	id	name	nationality	sex	dob	height	weight	sport	gold	silver
0	736041664	A Jesus Garcia	ESP	male	10/17/69	1.72	64.0	athletics	0	0
1	532037425	A Lam Shin	KOR	female	9/23/86	1.68	56.0	fencing	0	0
2	435962603	Aaron Brown	CAN	male	5/27/92	1.98	79.0	athletics	0	0
3	521041435	Aaron Cook	MDA	male	1/2/91	1.83	80.0	taekwondo	0	0
4	33922579	Aaron Gate	NZL	male	11/26/90	1.81	71.0	cycling	0	0
...	...	...	...	...	...	...	...	...	...	...
11533	265605954	Zurian Hechavarria	CUB	female	8/10/95	1.64	58.0	athletics	0	0
11534	214461847	Zuzana Hejnova	CZE	female	12/19/86	1.73	63.0	athletics	0	0
11535	88361042	di Xiao	CHN	male	5/14/91	1.85	100.0	wrestling	0	0
11536	900065925	le Quoc Toan Tran	VIE	male	4/5/89	1.60	56.0	weightlifting	0	0
11537	711404576	le Roux Hamman	RSA	male	1/6/92	1.85	70.0	athletics	0	0

11538 rows × 11 columns

In [21]:

```python
data2
```

Out[21]:

	country	code	population	gdp_per_capita
0	Afghanistan	AFG	32526562.0	594.323081
1	Albania	ALB	2889167.0	3945.217582
2	Algeria	ALG	39666519.0	4206.031232
3	American Samoa*	ASA	55538.0	NaN
4	Andorra	AND	70473.0	NaN

	country	code	population	gdp_per_capita
...	...	...	...	...
196	Vietnam	VIE	91703800.0	2111.138024
197	Virgin Islands*	ISV	103574.0	NaN
198	Yemen	YEM	26832215.0	1406.291651
199	Zambia	ZAM	16211767.0	1304.879014
200	Zimbabwe	ZIM	15602751.0	924.143819

201 rows × 4 columns

In [24]:

```
data3
```

Out[24]:

	id	sport	discipline	name	sex	venues
0	701492	aquatics	backstroke	Women's 100m Backstroke	female	Olympic Aquatics Stadium
1	305278	aquatics	backstroke	Women's 200m Backstroke	female	Olympic Aquatics Stadium
2	708010	aquatics	backstroke	Men's 100m Backstroke	male	Olympic Aquatics Stadium
3	729643	aquatics	backstroke	Men's 200m Backstroke	male	Olympic Aquatics Stadium
4	567019	aquatics	breaststroke	Women's 100m Breaststroke	female	Olympic Aquatics Stadium
...	...	...	...	...	...	...
301	755045	wrestling	greco roman	Men's Greco-Roman 59 kg	male	Carioca Arena 2
302	331407	wrestling	greco roman	Men's Greco-Roman 66 kg	male	Carioca Arena 2
303	169856	wrestling	greco roman	Men's Greco-Roman 75 kg	male	Carioca Arena 2
304	858736	wrestling	greco roman	Men's Greco-Roman 85 kg	male	Carioca Arena 2
305	767420	wrestling	greco roman	Men's Greco-Roman 98 kg	male	Carioca Arena 2

306 rows × 6 columns

In [28]:

```
data1.dtypes
```

Out[28]:

```
id                 int64
name              object
nationality       object
sex               object
dob               object
height           float64
weight           float64
sport             object
gold               int64
silver             int64
bronze             int64
dtype: object
```

In [29]:

```
data2.dtypes
```

Out[29]:

```
country            object
```

```
code               object
population        float64
gdp_per_capita    float64
dtype: object
```

In [30]:

```
data3.dtypes
```

Out[30]:

```
id             int64
sport         object
discipline    object
name          object
sex           object
venues        object
dtype: object
```

In [25]:

```
numero_filas=len(data1)
print("Numero de filas:", numero_filas)
```

```
Numero de filas: 11538
```

In [26]:

```
numero_filas=len(data2)
print("Numero de filas:", numero_filas)
```

```
Numero de filas: 201
```

In [27]:

```
numero_filas=len(data3)
print("Numero de filas:", numero_filas)
```

```
Numero de filas: 306
```

In [55]:

```
csv_list=['id', 'name', 'nationality', 'sex', 'dob', 'height', 'weight', 'sport', 'gold
for item in csv_list:
    print(data1[item].describe())
```

```
count    1.153800e+04
mean     4.999885e+08
std      2.908648e+08
min      1.834700e+04
25%      2.450997e+08
50%      5.002011e+08
75%      7.539874e+08
max      9.999878e+08
Name: id, dtype: float64
count           11538
unique          11517
top       Ivan Ivanov
freq                2
Name: name, dtype: object
count     11538
unique      207
top         USA
freq        567
Name: nationality, dtype: object
count     11538
unique        2
top        male
freq       6333
Name: sex, dtype: object
count        11537
unique        5595
top       12/20/90
freq             9
Name: dob, dtype: object
count    11208.000000
mean         1.766282
```

```
std          0.112719
min          1.210000
25%          1.690000
50%          1.760000
75%          1.840000
max          2.210000
Name: height, dtype: float64
count    10879.000000
mean        72.068205
std         16.177334
min         31.000000
25%         60.000000
50%         70.000000
75%         81.000000
max        170.000000
Name: weight, dtype: float64
count          11538
unique            28
top        athletics
freq            2363
Name: sport, dtype: object
count    11538.000000
mean         0.057722
std          0.255910
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          5.000000
Name: gold, dtype: float64
count    11538.000000
mean         0.056769
std          0.239147
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          2.000000
Name: silver, dtype: float64
count    11538.000000
mean         0.061016
std          0.243320
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          2.000000
Name: bronze, dtype: float64
```

In [56]:

```
csv_list=['country', 'code', 'population', 'gdp_per_capita']
for item in csv_list:
    print(data2[item].describe())
```

```
count          201
unique         201
top       Pakistan
freq             1
Name: country, dtype: object
count     201
unique    201
top       CUB
freq        1
Name: code, dtype: object
count    1.960000e+02
mean     3.722825e+07
```

```
std      1.399655e+08
min      1.022200e+04
25%      1.638278e+06
50%      7.450124e+06
75%      2.557454e+07
max      1.371220e+09
Name: population, dtype: float64
count       176.000000
mean      12882.556131
std       17747.141203
min         277.068309
25%        1781.096847
50%        5233.583395
75%       15494.683646
max      101449.968168
Name: gdp_per_capita, dtype: float64
```

In [57]:

```
csv_list=['id', 'sport', 'discipline', 'name', 'sex', 'venues']
for item in csv_list:
    print(data3[item].describe())
```

```
count       306.000000
mean     495682.434641
std      286891.455207
min         617.000000
25%      247029.750000
50%      520445.500000
75%      753803.250000
max      981370.000000
Name: id, dtype: float64
count           306
unique           28
top       athletics
freq             47
Name: sport, dtype: object
count           306
unique           50
top       freestyle
freq             26
Name: discipline, dtype: object
count       306
unique      273
top       Women
freq         11
Name: name, dtype: object
count      306
unique       3
top       male
freq       161
Name: sex, dtype: object
count                306
unique                33
top       Olympic Stadium
freq                  42
Name: venues, dtype: object
```