ML Project (OLYMPICS PART)

Name: Ravi Ranjan Kumar

Roll no: 20103122

```python
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df=pd.read_csv('athlete_events.csv')

region_df = pd.read_csv('noc_regions.csv')


df.tail()
```

```
           ID                   Name Sex   Age  Height  Weight      Team
NOC  \
271111  135569          Andrzej ya   M  29.0   179.0    89.0  Poland-1
POL
271112  135570             Piotr ya   M  27.0   176.0    59.0    Poland
POL
271113  135570             Piotr ya   M  27.0   176.0    59.0    Poland
POL
271114  135571  Tomasz Ireneusz ya   M  30.0   185.0    96.0    Poland
POL
271115  135571  Tomasz Ireneusz ya   M  34.0   185.0    96.0    Poland
POL

            Games  Year  Season            City        Sport  \
271111  1976 Winter  1976  Winter       Innsbruck         Luge
271112  2014 Winter  2014  Winter           Sochi  Ski Jumping
271113  2014 Winter  2014  Winter           Sochi  Ski Jumping
271114  1998 Winter  1998  Winter          Nagano    Bobsleigh
271115  2002 Winter  2002  Winter  Salt Lake City    Bobsleigh

                                       Event Medal
271111              Luge Mixed (Men)'s Doubles   NaN
271112  Ski Jumping Men's Large Hill, Individual   NaN
271113       Ski Jumping Men's Large Hill, Team   NaN
271114                   Bobsleigh Men's Four   NaN
271115                   Bobsleigh Men's Four   NaN
```

```python
df.shape
```
```
(271116, 15)
```
```python
df=df[df['Season']=='Summer']
```
```python
df.shape
```
```
(222552, 15)
```
```python
df.tail()
```
```
             ID                         Name  Sex   Age  Height  Weight  \
271106   135565         Fernando scar Zylberberg    M  27.0   168.0    76.0
271107   135566      James Francis "Jim" Zylker    M  21.0   175.0    75.0
271108   135567  Aleksandr Viktorovich Zyuzin    M  24.0   183.0    72.0
271109   135567  Aleksandr Viktorovich Zyuzin    M  28.0   183.0    72.0
271110   135568        Olga Igorevna Zyuzkova    F  33.0   171.0    69.0


                 Team  NOC         Games  Year  Season            City  \
271106      Argentina  ARG   2004 Summer  2004  Summer          Athina
271107  United States  USA   1972 Summer  1972  Summer          Munich
271108         Russia  RUS   2000 Summer  2000  Summer          Sydney
271109         Russia  RUS   2004 Summer  2004  Summer          Athina
271110        Belarus  BLR   2016 Summer  2016  Summer  Rio de Janeiro


             Sport                                 Event Medal
271106      Hockey                   Hockey Men's Hockey   NaN
271107    Football               Football Men's Football   NaN
271108      Rowing  Rowing Men's Lightweight Coxless Fours   NaN
271109      Rowing  Rowing Men's Lightweight Coxless Fours   NaN
271110  Basketball       Basketball Women's Basketball   NaN
```
```python
region_df.tail()
```
```
     NOC   region        notes
225  YEM    Yemen          NaN
226  YMD    Yemen  South Yemen
227  YUG   Serbia   Yugoslavia
```

```
228  ZAM    Zambia           NaN
229  ZIM    Zimbabwe         NaN
```

region_df.shape

(230, 3)

df=df.merge(region_df, on='NOC',how='left')

df.tail()

```
            ID                            Name Sex   Age  Height  Weight
\
222547  135565       Fernando scar Zylberberg   M  27.0   168.0    76.0

222548  135566      James Francis "Jim" Zylker   M  21.0   175.0    75.0

222549  135567  Aleksandr Viktorovich Zyuzin    M  24.0   183.0    72.0

222550  135567  Aleksandr Viktorovich Zyuzin    M  28.0   183.0    72.0

222551  135568       Olga Igorevna Zyuzkova     F  33.0   171.0    69.0


              Team  NOC       Games  Year  Season           City
\
222547     Argentina  ARG  2004 Summer  2004  Summer           Athina

222548  United States  USA  1972 Summer  1972  Summer           Munich

222549        Russia  RUS  2000 Summer  2000  Summer           Sydney

222550        Russia  RUS  2004 Summer  2004  Summer           Athina

222551       Belarus  BLR  2016 Summer  2016  Summer  Rio de Janeiro


             Sport                                 Event Medal
region  \
222547     Hockey                    Hockey Men's Hockey    NaN
Argentina
222548    Football                Football Men's Football    NaN
USA
222549      Rowing  Rowing Men's Lightweight Coxless Fours    NaN
Russia
222550      Rowing  Rowing Men's Lightweight Coxless Fours    NaN
Russia
222551  Basketball      Basketball Women's Basketball        NaN
Belarus
```

```
          notes
222547    NaN
222548    NaN
222549    NaN
222550    NaN
222551    NaN

df['region'].unique()

array(['China', 'Denmark', 'Netherlands', 'Finland', 'Norway',
'Romania',
       'Estonia', 'France', 'Morocco', 'Spain', 'Egypt', 'Iran',
       'Bulgaria', 'Italy', 'Chad', 'Azerbaijan', 'Sudan', 'Russia',
       'Argentina', 'Cuba', 'Belarus', 'Greece', 'Cameroon', 'Turkey',
       'Chile', 'Mexico', 'USA', 'Nicaragua', 'Hungary', 'Nigeria',
       'Algeria', 'Kuwait', 'Bahrain', 'Pakistan', 'Iraq', 'Syria',
       'Lebanon', 'Qatar', 'Malaysia', 'Germany', 'Canada', 'Ireland',
       'Australia', 'South Africa', 'Eritrea', 'Tanzania', 'Jordan',
       'Tunisia', 'Libya', 'Belgium', 'Djibouti', 'Palestine',
'Comoros',
       'Kazakhstan', 'Brunei', 'India', 'Saudi Arabia', 'Maldives',
       'Ethiopia', 'United Arab Emirates', 'Yemen', 'Indonesia',
       'Philippines', nan, 'Uzbekistan', 'Kyrgyzstan', 'Tajikistan',
       'Japan', 'Republic of Congo', 'Switzerland', 'Brazil',
'Monaco',
       'Israel', 'Uruguay', 'Sweden', 'Sri Lanka', 'Armenia',
       'Ivory Coast', 'Kenya', 'Benin', 'UK', 'Ghana', 'Somalia',
'Niger',
       'Mali', 'Afghanistan', 'Poland', 'Costa Rica', 'Panama',
'Georgia',
       'Slovenia', 'Guyana', 'New Zealand', 'Portugal', 'Paraguay',
       'Angola', 'Venezuela', 'Colombia', 'Bangladesh', 'Peru',
       'El Salvador', 'Puerto Rico', 'Uganda', 'Honduras', 'Ecuador',
       'Turkmenistan', 'Mauritius', 'Seychelles', 'Czech Republic',
       'Luxembourg', 'Mauritania', 'Saint Kitts', 'Trinidad',
       'Dominican Republic', 'Saint Vincent', 'Jamaica', 'Liberia',
       'Suriname', 'Nepal', 'Mongolia', 'Austria', 'Palau',
'Lithuania',
       'Togo', 'Namibia', 'Curacao', 'Ukraine', 'Iceland',
       'American Samoa', 'Samoa', 'Rwanda', 'Croatia', 'Dominica',
       'Haiti', 'Malta', 'Cyprus', 'Guinea', 'Belize', 'Thailand',
       'Bermuda', 'Serbia', 'Sierra Leone', 'Papua New Guinea',
       'Individual Olympic Athletes', 'Oman', 'Fiji', 'Vanuatu',
       'Moldova', 'Bahamas', 'Guatemala', 'Latvia',
       'Virgin Islands, British', 'Mozambique', 'Virgin Islands, US',
       'Central African Republic', 'Madagascar', 'Bosnia and
Herzegovina',
       'Guam', 'Cayman Islands', 'Slovakia', 'Barbados', 'Guinea-
Bissau',
       'Timor-Leste', 'Democratic Republic of the Congo', 'Gabon',
       'San Marino', 'Laos', 'Botswana', 'South Korea', 'Cambodia',
```

```
        'North Korea', 'Solomon Islands', 'Senegal', 'Cape Verde',
        'Equatorial Guinea', 'Boliva', 'Antigua', 'Andorra',
'Zimbabwe',
        'Grenada', 'Saint Lucia', 'Micronesia', 'Myanmar', 'Malawi',
        'Zambia', 'Taiwan', 'Sao Tome and Principe', 'Macedonia',
        'Liechtenstein', 'Montenegro', 'Gambia', 'Cook Islands',
'Albania',
        'Swaziland', 'Burkina Faso', 'Burundi', 'Aruba', 'Nauru',
        'Vietnam', 'Bhutan', 'Marshall Islands', 'Kiribati', 'Tonga',
        'Kosovo', 'South Sudan', 'Lesotho'], dtype=object)
```

```python
df.isnull().sum()
```

```
ID              0
Name            0
Sex             0
Age          9189
Height      51857
Weight      53854
Team            0
NOC             0
Games           0
Year            0
Season          0
City            0
Sport           0
Event           0
Medal      188464
region        370
notes      218151
dtype: int64
```

```python
df.duplicated().sum()
```

```
1385
```

```python
df.drop_duplicates(inplace=True)
```

```python
df.duplicated().sum()
```

```
0
```

```python
df['Medal'].value_counts()
```

```
Gold      11456
Bronze    11409
Silver    11212
Name: Medal, dtype: int64
```

```python
df = pd.concat([df,pd.get_dummies(df['Medal'])], axis=1)
```

```python
df.shape
```

```
(221167, 20)

df.tail()
             ID                            Name Sex   Age  Height  Weight
\
222547  135565        Fernando scar Zylberberg   M  27.0   168.0    76.0

222548  135566       James Francis "Jim" Zylker   M  21.0   175.0    75.0

222549  135567  Aleksandr Viktorovich Zyuzin   M  24.0   183.0    72.0

222550  135567  Aleksandr Viktorovich Zyuzin   M  28.0   183.0    72.0

222551  135568         Olga Igorevna Zyuzkova   F  33.0   171.0    69.0


                 Team  NOC         Games  Year  Season            City
\
222547      Argentina  ARG  2004 Summer  2004  Summer          Athina

222548  United States  USA  1972 Summer  1972  Summer          Munich

222549         Russia  RUS  2000 Summer  2000  Summer          Sydney

222550         Russia  RUS  2004 Summer  2004  Summer          Athina

222551        Belarus  BLR  2016 Summer  2016  Summer  Rio de Janeiro


             Sport                                          Event Medal
region  \
222547      Hockey                           Hockey Men's Hockey    NaN
Argentina
222548    Football                       Football Men's Football    NaN
USA
222549      Rowing  Rowing Men's Lightweight Coxless Fours    NaN
Russia
222550      Rowing  Rowing Men's Lightweight Coxless Fours    NaN
Russia
222551  Basketball         Basketball Women's Basketball    NaN
Belarus

        notes  Bronze  Gold  Silver
222547    NaN       0     0       0
222548    NaN       0     0       0
222549    NaN       0     0       0
222550    NaN       0     0       0
222551    NaN       0     0       0
```

```
df.groupby('NOC').sum()
[['Gold','Silver','Bronze']].sort_values('Gold',ascending=False).reset
_index().head(25)
```

```
     NOC    Gold   Silver   Bronze
0    USA   2472.0  1333.0   1197.0
1    URS    832.0   635.0    596.0
2    GBR    635.0   729.0    620.0
3    GER    592.0   538.0    649.0
4    ITA    518.0   474.0    454.0
5    FRA    463.0   567.0    587.0
6    HUN    432.0   328.0    363.0
7    SWE    354.0   396.0    358.0
8    AUS    342.0   452.0    510.0
9    GDR    339.0   277.0    227.0
10   CHN    334.0   317.0    258.0
11   RUS    296.0   278.0    331.0
12   NED    245.0   302.0    371.0
13   JPN    230.0   287.0    333.0
14   NOR    227.0   196.0    167.0
15   DEN    179.0   236.0    177.0
16   KOR    171.0   206.0    175.0
17   CUB    164.0   129.0    116.0
18   ROU    161.0   200.0    290.0
19   CAN    158.0   239.0    344.0
20   FRG    144.0   172.0    188.0
21   FIN    132.0   125.0    217.0
22   IND    131.0    19.0     40.0
23   YUG    130.0   161.0     92.0
24   POL    111.0   185.0    242.0
```

```
df[(df['NOC']=='IND') & (df['Medal']=='Gold')]
```

```
            ID                      Name Sex   Age  Height  Weight
Team   NOC  \
4186       2699           Shaukat Ali   M  30.0     NaN     NaN
India  IND
4190       2703       Syed Mushtaq Ali   M  22.0   165.0    61.0
India  IND
4460       2864    Richard James Allen   M  25.0   172.0     NaN
India  IND
4461       2864    Richard James Allen   M  30.0   172.0     NaN
India  IND
4462       2864    Richard James Allen   M  34.0   172.0     NaN
India  IND
...         ...                    ...  ..   ...     ...     ..
.  ...
191759  117232      Dung Dung Sylvanus   M  31.0   160.0    62.0
India  IND
193964  118553  Carlyle Carrol Tapsell   M  23.0   182.0     NaN
India  IND
```

```
193965  118553   Carlyle Carrol Tapsell   M  27.0    182.0      NaN
India  IND
205949  125560         Max "Maxie" Vaz   M   NaN      NaN      NaN
India  IND
219336  133554     Sayed Muhammad Yusuf   M   NaN      NaN      NaN
India  IND

               Games  Year  Season        City   Sport
Event  \
4186    1928 Summer  1928  Summer    Amsterdam  Hockey  Hockey Men's
Hockey
4190    1964 Summer  1964  Summer        Tokyo  Hockey  Hockey Men's
Hockey
4460    1928 Summer  1928  Summer    Amsterdam  Hockey  Hockey Men's
Hockey
4461    1932 Summer  1932  Summer  Los Angeles  Hockey  Hockey Men's
Hockey
4462    1936 Summer  1936  Summer       Berlin  Hockey  Hockey Men's
Hockey
...             ...   ...     ...          ...     ...
...
191759  1980 Summer  1980  Summer       Moskva  Hockey  Hockey Men's
Hockey
193964  1932 Summer  1932  Summer  Los Angeles  Hockey  Hockey Men's
Hockey
193965  1936 Summer  1936  Summer       Berlin  Hockey  Hockey Men's
Hockey
205949  1948 Summer  1948  Summer       London  Hockey  Hockey Men's
Hockey
219336  1928 Summer  1928  Summer    Amsterdam  Hockey  Hockey Men's
Hockey

        Medal region notes  Bronze  Gold  Silver
4186    Gold  India   NaN       0     1       0
4190    Gold  India   NaN       0     1       0
4460    Gold  India   NaN       0     1       0
4461    Gold  India   NaN       0     1       0
4462    Gold  India   NaN       0     1       0
...      ...    ...   ...     ...   ...     ...
191759  Gold  India   NaN       0     1       0
193964  Gold  India   NaN       0     1       0
193965  Gold  India   NaN       0     1       0
205949  Gold  India   NaN       0     1       0
219336  Gold  India   NaN       0     1       0

[131 rows x 20 columns]

medal_tally =
df.drop_duplicates(subset=['Team','NOC','Games','Year','City','Sport',
'Event','Medal'])
```

```
medal_tally.groupby('NOC').sum()
[['Gold','Silver','Bronze']].sort_values('Gold',ascending=False).reset
_index()
```

```
     NOC   Gold  Silver  Bronze
0    USA  1035.0  802.0   708.0
1    URS   394.0  317.0   294.0
2    GBR   278.0  317.0   300.0
3    GER   235.0  261.0   283.0
4    FRA   234.0  256.0   287.0
..   ...    ...     ...     ...
225  AHO     0.0    1.0     0.0
226  LBR     0.0    0.0     0.0
227  LCA     0.0    0.0     0.0
228  LES     0.0    0.0     0.0
229  LBA     0.0    0.0     0.0

[230 rows x 4 columns]
```

```
medal_tally[medal_tally['NOC']=='IND']
```

```
            ID                                               Name Sex
Age  \
436        281                                    S. Abdul Hamid    M
NaN
437        281                                    S. Abdul Hamid    M
NaN
790        512              Shiny Kurisingal Abraham-Wilson    F
19.0
791        512              Shiny Kurisingal Abraham-Wilson    F
19.0
792        512              Shiny Kurisingal Abraham-Wilson    F
23.0
...        ...                                               ...  ..  ..
.
216942  132143                              Vikas Krishan Yadav    M
24.0
216985  132177                             Mohammad Anas Yahiya    M
21.0
218411  133029  Thyadathuvilla Chandrapillai "T. C." Yohannan    M
29.0
222416  135480                                     Geeta Zutshi    F
23.0
222417  135480                                     Geeta Zutshi    F
27.0

        Height  Weight   Team  NOC      Games  Year  Season
City  \
436        NaN     NaN  India  IND  1928 Summer  1928  Summer
Amsterdam
437        NaN     NaN  India  IND  1928 Summer  1928  Summer
```

```
        Amsterdam
790     167.0   53.0   India   IND   1984 Summer   1984   Summer       Los
Angeles
791     167.0   53.0   India   IND   1984 Summer   1984   Summer       Los
Angeles
792     167.0   53.0   India   IND   1988 Summer   1988   Summer
Seoul
...        ...    ...     ...   ...          ...    ...      ...
...
216942  177.0   69.0   India   IND   2016 Summer   2016   Summer   Rio de
Janeiro
216985  177.0   69.0   India   IND   2016 Summer   2016   Summer   Rio de
Janeiro
218411  174.0   62.0   India   IND   1976 Summer   1976   Summer
Montreal
222416  167.0   51.0   India   IND   1980 Summer   1980   Summer
Moskva
222417  167.0   51.0   India   IND   1984 Summer   1984   Summer       Los
Angeles

            Sport                                      Event Medal region
notes  \
436      Athletics        Athletics Men's 110 metres Hurdles   NaN   India
NaN
437      Athletics        Athletics Men's 400 metres Hurdles   NaN   India
NaN
790      Athletics                 Athletics Women's 800 metres   NaN   India
NaN
791      Athletics   Athletics Women's 4 x 400 metres Relay   NaN   India
NaN
792      Athletics                 Athletics Women's 800 metres   NaN   India
NaN
...         ...                                         ...   ...     ...
...
216942     Boxing                Boxing Men's Middleweight   NaN   India
NaN
216985  Athletics                Athletics Men's 400 metres   NaN   India
NaN
218411  Athletics                  Athletics Men's Long Jump   NaN   India
NaN
222416  Athletics                Athletics Women's 800 metres   NaN   India
NaN
222417  Athletics              Athletics Women's 3,000 metres   NaN   India
NaN

        Bronze   Gold   Silver
436          0      0        0
437          0      0        0
790          0      0        0
791          0      0        0
```

```
792              0     0       0
...            ...   ...     ...
216942           0     0       0
216985           0     0       0
218411           0     0       0
222416           0     0       0
222417           0     0       0

[687 rows x 20 columns]

medal_tally

            ID                                     Name Sex   Age  Height
Weight  \
0            1                               A Dijiang   M  24.0   180.0
80.0
1            2                                A Lamusi   M  23.0   170.0
60.0
2            3                       Gunnar Nielsen Aaby   M  24.0     NaN
NaN
3            4                     Edgar Lindenau Aabye   M  34.0     NaN
NaN
4            8   Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0
NaN
...        ...                                      ...  ..   ...     ...
...
222528  135553  Galina Ivanovna Zybina (-Fyodorova)   F  25.0   168.0
80.0
222530  135553  Galina Ivanovna Zybina (-Fyodorova)   F  33.0   168.0
80.0
222536  135556          Bogusaw Stanisaw Zychowicz   M  19.0   189.0
80.0
222537  135556          Bogusaw Stanisaw Zychowicz   M  19.0   189.0
80.0
222541  135560                 Stavroula Zygouri   F  36.0   171.0
63.0

                   Team  NOC         Games  Year  Season         City  \
0                China  CHN  1992 Summer  1992  Summer     Barcelona
1                China  CHN  2012 Summer  2012  Summer        London
2              Denmark  DEN  1920 Summer  1920  Summer     Antwerpen
3        Denmark/Sweden  DEN  1900 Summer  1900  Summer         Paris
4          Netherlands  NED  1932 Summer  1932  Summer   Los Angeles
...                ...  ...           ...   ...     ...           ...
222528    Soviet Union  URS  1956 Summer  1956  Summer     Melbourne
222530    Soviet Union  URS  1964 Summer  1964  Summer         Tokyo
222536          Poland  POL  1980 Summer  1980  Summer        Moskva
222537          Poland  POL  1980 Summer  1980  Summer        Moskva
222541          Greece  GRE  2004 Summer  2004  Summer        Athina
```

	Sport	Event	Medal
0	Basketball	Basketball Men's Basketball	NaN
1	Judo	Judo Men's Extra-Lightweight	NaN
2	Football	Football Men's Football	NaN
3	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	Athletics	Athletics Women's 100 metres	NaN
...	...	...	...
222528	Athletics	Athletics Women's Shot Put	Silver
222530	Athletics	Athletics Women's Shot Put	Bronze
222536	Swimming	Swimming Men's 100 metres Butterfly	NaN
222537	Swimming	Swimming Men's 200 metres Butterfly	NaN
222541	Wrestling	Wrestling Women's Middleweight, Freestyle	NaN

	region	notes	Bronze	Gold	Silver
0	China	NaN	0	0	0
1	China	NaN	0	0	0
2	Denmark	NaN	0	0	0
3	Denmark	NaN	0	1	0
4	Netherlands	NaN	0	0	0
...	...	...	...	...	...
222528	Russia	NaN	0	0	1
222530	Russia	NaN	1	0	0
222536	Poland	NaN	0	0	0
222537	Poland	NaN	0	0	0
222541	Greece	NaN	0	0	0

[105636 rows x 20 columns]

```python
years=df['Year'].unique().tolist()

years.sort()

years.insert(0,'Overall')

years
```

['Overall',
 1896,

```
 1900,
 1904,
 1906,
 1908,
 1912,
 1920,
 1924,
 1928,
 1932,
 1936,
 1948,
 1952,
 1956,
 1960,
 1964,
 1968,
 1972,
 1976,
 1980,
 1984,
 1988,
 1992,
 1996,
 2000,
 2004,
 2008,
 2012,
 2016]
```

```python
country=df['region'].unique().tolist()
```

```python
country.sort
```

```
<function list.sort(*, key=None, reverse=False)>
```

```python
df['Year'].unique().shape[0]-1
```

```
28
```

```python
df['City'].unique()
```

```
array(['Barcelona', 'London', 'Antwerpen', 'Paris', 'Los Angeles',
       'Helsinki', 'Sydney', 'Atlanta', 'Stockholm', 'Beijing',
       'Rio de Janeiro', 'Athina', 'Mexico City', 'Munich', 'Seoul',
       'Berlin', 'Melbourne', 'Roma', 'Amsterdam', 'Montreal',
'Moskva',
       'Tokyo', 'St. Louis'], dtype=object)
```

```python
df['City'].unique().shape[0]
```

```
23
```

```
famous_sports=df['Sport'].unique()

famous_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)

df['Sport'].unique().shape[0]

52

df['Event'].unique().shape

(651,)

df['Name'].unique().shape

(116122,)

df['region'].unique().shape

(206,)

# participating nations

df.head()

   ID                                Name Sex   Age  Height  Weight  \
0   1                           A Dijiang   M  24.0   180.0    80.0
1   2                            A Lamusi   M  23.0   170.0    60.0
2   3                  Gunnar Nielsen Aaby   M  24.0     NaN     NaN
3   4                Edgar Lindenau Aabye   M  34.0     NaN     NaN
4   8  Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0     NaN

             Team  NOC       Games  Year  Season        City
Sport  \
0           China  CHN  1992 Summer  1992  Summer   Barcelona
Basketball
1           China  CHN  2012 Summer  2012  Summer      London
Judo
2         Denmark  DEN  1920 Summer  1920  Summer   Antwerpen
```

```
                Football
3   Denmark/Sweden  DEN   1900 Summer   1900   Summer          Paris   Tug-
Of-War
4       Netherlands  NED   1932 Summer   1932   Summer  Los Angeles
Athletics

                                Event Medal      region notes  Bronze   Gold
Silver
0    Basketball Men's Basketball    NaN       China   NaN       0      0
0
1  Judo Men's Extra-Lightweight    NaN       China   NaN       0      0
0
2        Football Men's Football    NaN     Denmark   NaN       0      0
0
3   Tug-Of-War Men's Tug-Of-War   Gold     Denmark   NaN       0      1
0
4   Athletics Women's 100 metres    NaN  Netherlands   NaN       0      0
0
```

```python
nations_over_time=df.drop_duplicates(['Year','region'])
['Year'].value_counts().reset_index().sort_values('index')

nations_over_time.rename(columns={'index':'Edition','Year':'No of
Countries'},inplace=True)

import plotly.express as px

fig = px.line(nations_over_time, x='Edition', y='No of Countries')
fig.show()
```

{"data":[{"y":
[12,31,14,20,22,29,29,45,46,47,49,59,67,71,83,93,111,120,91,80,139,156
,168,196,199,200,202,203,204],"hovertemplate":"Edition=%{x}
No of
Countries=%{y}<extra></extra>","mode":"lines","x":
[1896,1900,1904,1906,1908,1912,1920,1924,1928,1932,1936,1948,1952,1956
,1960,1964,1968,1972,1976,1980,1984,1988,1992,1996,2000,2004,2008,2012
,2016],"showlegend":false,"name":"","legendgroup":"","line":
{"dash":"solid","color":"#636efa"},"orientation":"v","xaxis":"x","type
":"scatter","marker":{"symbol":"circle"},"yaxis":"y"}],"config":
{"plotlyServerURL":"https://plot.ly"},"layout":{"legend":
{"tracegroupgap":0},"margin":{"t":60},"xaxis":{"anchor":"y","title":
{"text":"Edition"},"domain":[0,1]},"template":{"data":
{"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"scattermapbox"
:[{"type":"scattermapbox","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],

[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"pie":
[{"automargin":true,"type":"pie"}],"carpet":[{"aaxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"baxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"type":"carpet"}],"ba
r":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"type":"bar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"barpolar":
[{"type":"barpolar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"scatter3d":
[{"line":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d","marker":
{"colorbar":{"outlinewidth":0,"ticks":""}}}],"contour":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"histogram2d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"scatterpolar":
[{"type":"scatterpolar","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"histogram":
[{"type":"histogram","marker":{"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"histogram2dcontour
":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"parcoords":[{"line":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"scatterpolargl":
[{"type":"scatterpolargl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],

[1,"#f0f921"]],"type":"heatmapgl"}],"scattercarpet":
[{"type":"scattercarpet","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"choropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"scatterternary":
[{"type":"scatterternary","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"scatter":[{"fillpattern":
{"solidity":0.2,"fillmode":"overlay","size":10},"type":"scatter"}],"ta
ble":[{"cells":{"fill":{"color":"#EBF0F8"},"line":
{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}],"scattergeo":
[{"type":"scattergeo","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"surface":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"scattergl":
[{"type":"scattergl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}]}],"layout":{"ternary":{"aaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"baxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"caxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"autotypenumbers":"strict","shapedefaults":{"line":
{"color":"#2a3f5f"}},"annotationdefaults":
{"arrowwidth":1,"arrowcolor":"#2a3f5f","arrowhead":0},"coloraxis":
{"colorbar":{"outlinewidth":0,"ticks":""}},"title":{"x":5.0e-
2},"hoverlabel":{"align":"left"},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","scene":{"zaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"xaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"yaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr

idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
},"font":{"color":"#2a3f5f"},"xaxis":{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"polar":{"angularaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"radialaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"plot_bgcolor":"#E5ECF6","geo":
{"subunitcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","sho
wland":true,"showlakes":true,"bgcolor":"white"},"yaxis":
{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"]}},"yaxis":{"anchor":"x","title":
{"text":"No of Countries"},"domain":[0,1]}}}

```python
x = df.drop_duplicates(['Year','Sport','Event'])

import seaborn as sns
plt.figure(figsize=(20,20))
sns.heatmap(x.pivot_table(index='Sport',columns='Year',values='Event',
aggfunc='count').fillna(0).astype('int'),annot=True)
```

<AxesSubplot:xlabel='Year', ylabel='Sport'>
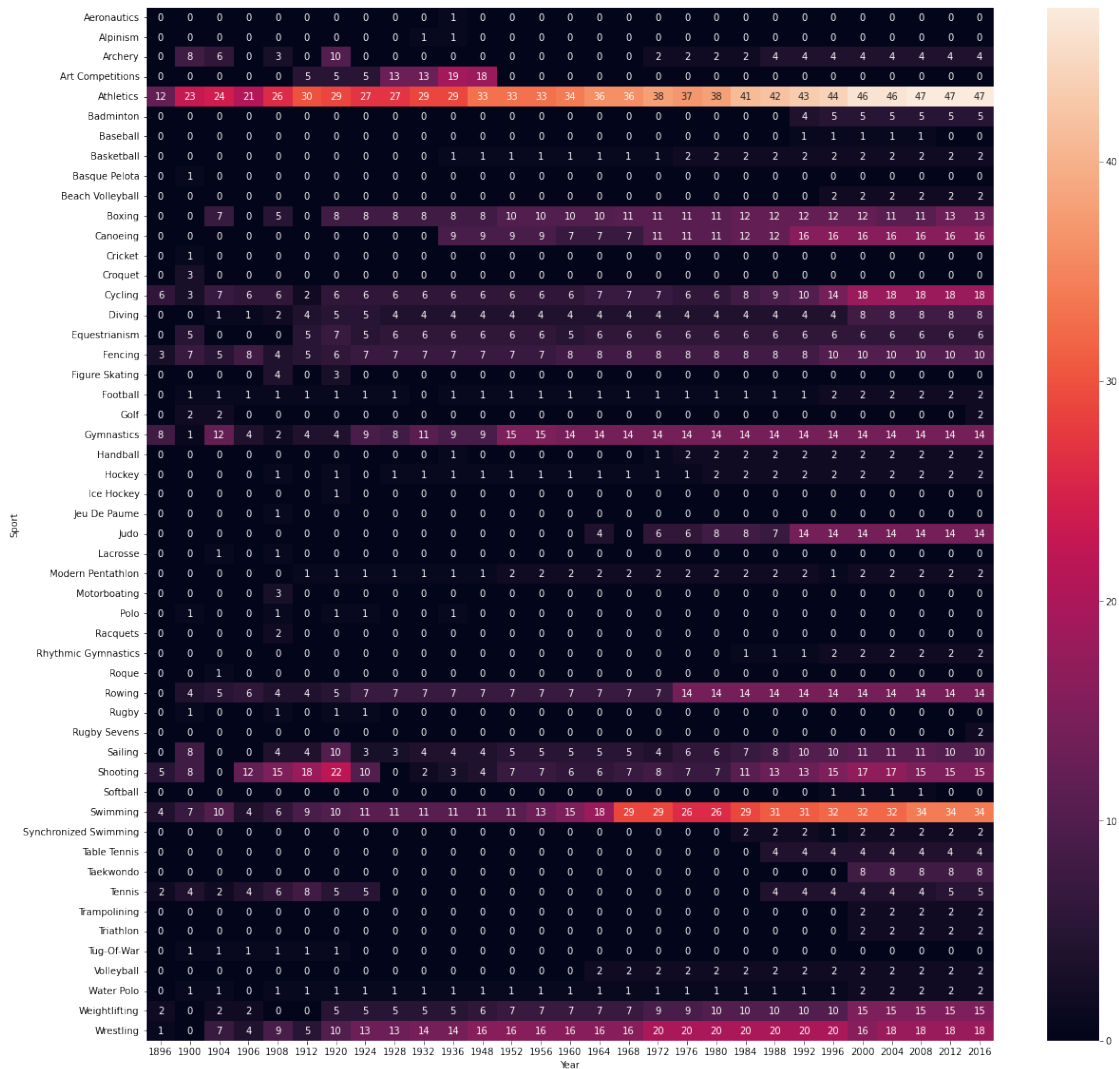
```python
# df.dropna(subset=)

def most_successful(df,sport):
    temp_df=df.dropna(subset=['Medal'])

    if sport != 'Overall':
        temp_df= temp_df[temp_df['Sport']==sport]


x=temp_df['Name'].value_counts().reset_index().head(15).merge(df,left_
on='index',right_on='Name',how='left')
[['index','Name_x','Sport']].drop_duplicates('index')
    x.rename(columns={'index':'Name','Name_x':'Medals'},inplace=True)
    return x


most_successful(df,'Gymnastics')
```

```
                              Name  Medals        Sport
0      Larysa Semenivna Latynina (Diriy-)      18   Gymnastics
19            Nikolay Yefimovich Andrianov      15   Gymnastics
43             Borys Anfiyanovych Shakhlin      13   Gymnastics
67                            Takashi Ono      13   Gymnastics
100                           Sawao Kato      12   Gymnastics
124             Aleksey Yuryevich Nemov      12   Gymnastics
145            Viktor Ivanovych Chukarin      11   Gymnastics
161               Vra slavsk (-Odloilov)      11   Gymnastics
179              gnes Keleti-Srkny (Klein)      10   Gymnastics
193        Vitaly Venediktovich Shcherbo      10   Gymnastics
209     Aleksandr Nikolayevich Dityatin      10   Gymnastics
225         Polina Hryhorivna Astakhova      10   Gymnastics
244                    Akinori Nakayama      10   Gymnastics
260         Nadia Elena Comneci (-Conner)       9   Gymnastics
272          Yury Yevlampiyevich Titov       9   Gymnastics
```

Countary wise analysis

```
temp_df=df.dropna(subset=['Medal'])
```

```
temp_df.drop_duplicates(subset=['Team','NOC','Games','Year','City','Sp
ort','Event','Medal'],inplace=True)
```

```
C:\Users\dell\anaconda3\lib\site-packages\pandas\util\
_decorators.py:311: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
```

```
new_df=temp_df[temp_df['NOC']=='USA']
```

```
final_df=new_df.groupby('Year').count()['Medal'].reset_index()
```

```
final_df
```

```
    Year  Medal
0   1896     19
1   1900     54
2   1904    231
3   1906     23
4   1908     46
5   1912     63
6   1920     95
```

```
7    1924     99
8    1928     56
9    1932    110
10   1936     57
11   1948     84
12   1952     76
13   1956     74
14   1960     71
15   1964     90
16   1968    107
17   1972     94
18   1976     94
19   1984    173
20   1988     94
21   1992    108
22   1996    101
23   2000     91
24   2004    101
25   2008    110
26   2012    103
27   2016    121
```

```
fig = px.line(final_df, x='Year', y='Medal')
fig.show()
```

```
{"data":[{"y":
[19,54,231,23,46,63,95,99,56,110,57,84,76,74,71,90,107,94,94,173,94,10
8,101,91,101,110,103,121],"hovertemplate":"Year=%{x}<br>Medal=%
{y}<extra></extra>","mode":"lines","x":
[1896,1900,1904,1906,1908,1912,1920,1924,1928,1932,1936,1948,1952,1956
,1960,1964,1968,1972,1976,1984,1988,1992,1996,2000,2004,2008,2012,2016
],"showlegend":false,"name":"","legendgroup":"","line":
{"dash":"solid","color":"#636efa"},"orientation":"v","xaxis":"x","type
":"scatter","marker":{"symbol":"circle"},"yaxis":"y"}],"config":
{"plotlyServerURL":"https://plot.ly"},"layout":{"legend":
{"tracegroupgap":0},"margin":{"t":60},"xaxis":{"anchor":"y","title":
{"text":"Year"},"domain":[0,1]},"template":{"data":{"contourcarpet":
[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"scattermapbox"
:[{"type":"scattermapbox","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"pie":
[{"automargin":true,"type":"pie"}],"carpet":[{"aaxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
```

,"startlinecolor":"#2a3f5f","gridcolor":"white"},"baxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"type":"carpet"}],"ba
r":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"type":"bar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"barpolar":
[{"type":"barpolar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"scatter3d":
[{"line":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d","marker":
{"colorbar":{"outlinewidth":0,"ticks":""}}}],"contour":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"histogram2d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"scatterpolar":
[{"type":"scatterpolar","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"histogram":
[{"type":"histogram","marker":{"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"histogram2dcontour
":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"parcoords":[{"line":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"scatterpolargl":
[{"type":"scatterpolargl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"scattercarpet":
[{"type":"scattercarpet","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"choropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"scatterternary":
[{"type":"scatterternary","marker":{"colorbar":

{"outlinewidth":0,"ticks":""}}}],"scatter":[{"fillpattern":
{"solidity":0.2,"fillmode":"overlay","size":10},"type":"scatter"}],"ta
ble":[{"cells":{"fill":{"color":"#EBF0F8"},"line":
{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}],"scattergeo":
[{"type":"scattergeo","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"surface":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"scattergl":
[{"type":"scattergl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}]],"layout":{"ternary":{"aaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"baxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"caxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"autotypenumbers":"strict","shapedefaults":{"line":
{"color":"#2a3f5f"}},"annotationdefaults":
{"arrowwidth":1,"arrowcolor":"#2a3f5f","arrowhead":0},"coloraxis":
{"colorbar":{"outlinewidth":0,"ticks":""}},"title":{"x":5.0e-
2},"hoverlabel":{"align":"left"},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","scene":{"zaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"xaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"yaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
},"font":{"color":"#2a3f5f"},"xaxis":{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"polar":{"angularaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"radialaxis":

{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"plot_bgcolor":"#E5ECF6","geo":
{"subunitcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","sho
wland":true,"showlakes":true,"bgcolor":"white"},"yaxis":
{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"]}},"yaxis":{"anchor":"x","title":
{"text":"Medal"},"domain":[0,1]}}}

```
new_df=temp_df[temp_df['NOC']=='USA']
plt.figure(figsize=(20,20))
sns.heatmap(new_df.pivot_table(index='Sport',columns='Year',values='Me
dal',aggfunc='count').fillna(0),annot=True)
```

<AxesSubplot:xlabel='Year', ylabel='Sport'>
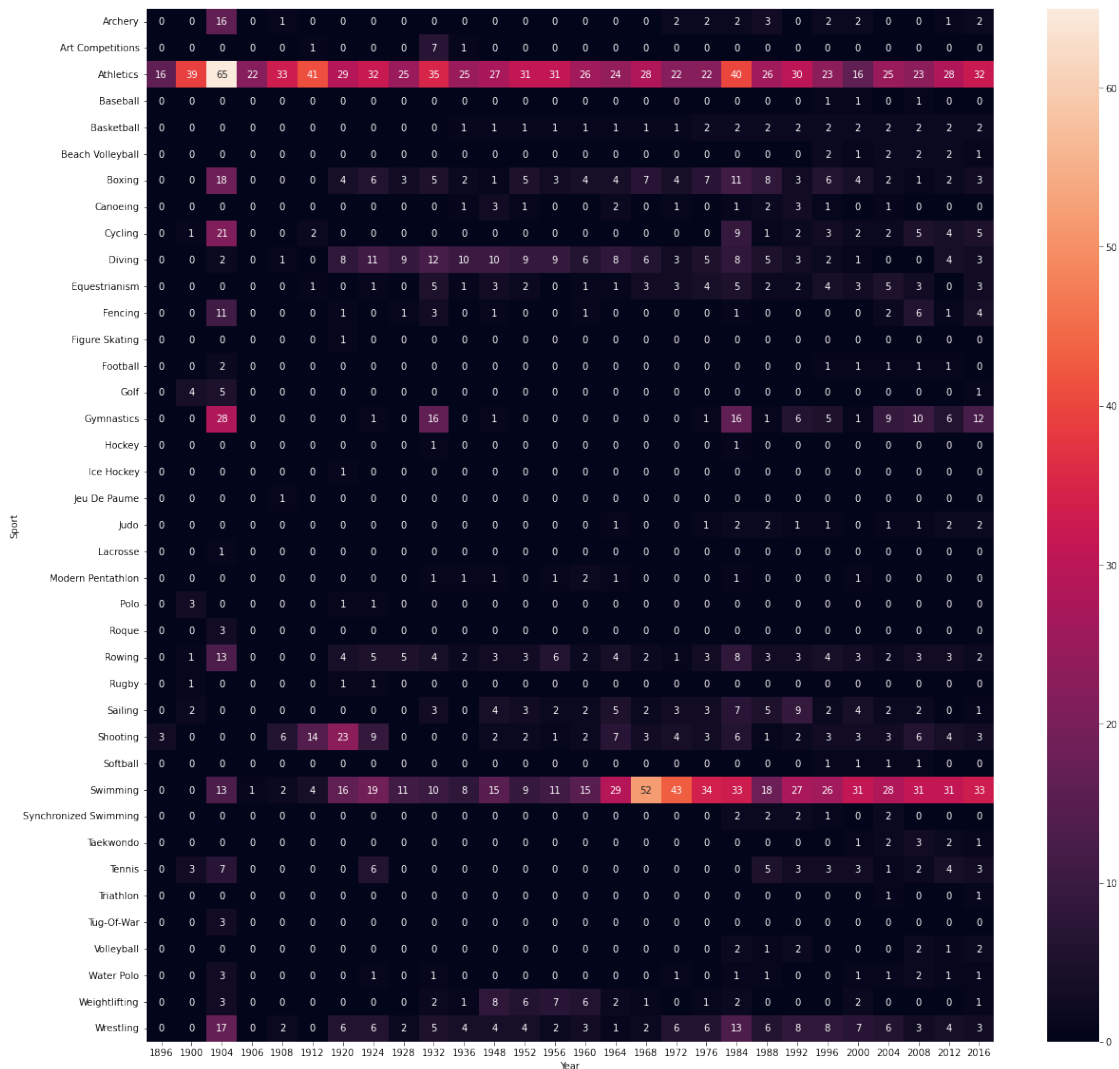
```python
def most_successful(df,country):
    temp_df=df.dropna(subset=['Medal'])

    temp_df= temp_df[temp_df['NOC']==country]


x=temp_df['Name'].value_counts().reset_index().head(15).merge(df,left_
on='index',right_on='Name',how='left')
[['index','Name_x','Sport']].drop_duplicates('index')
    x.rename(columns={'index':'Name','Name_x':'Medals'},inplace=True)
    return x


most_successful(df,'USA')
```

```
                                                       Name   Medals
Sport
0                               Michael Fred Phelps, II      28
Swimming
30                                  Ryan Steven Lochte        12
Swimming
44    Jennifer Elisabeth "Jenny" Thompson (-Cumpelik)      12
Swimming
61                         Natalie Anne Coughlin (-Hall)      12
Swimming
73              Dara Grace Torres (-Hoffman, -Minas)        12
Swimming
86                       Matthew Nicholas "Matt" Biondi      11
Swimming
98                                  Carl Townsend Osburn      11
Shooting
114                                   Mark Andrew Spitz       11
Swimming
126                  Raymond Clarence "Ray" Ewry          10
Athletics
136               Frederick Carlton "Carl" Lewis          10
Athletics
146                                Gary Wayne Hall, Jr.        10
Swimming
156                          Allyson Michelle Felix          9
Athletics
166                          Martin Joseph Sheridan          9
Athletics
181                       Shirley Frances Babashoff          9
Swimming
192                          Nathan Ghar-Jun Adrian          8
Swimming
```

Athelete Analysis

```python
import plotly.figure_factory as ff

athlete_df=df.drop_duplicates(subset=['Name','region'])

fig=ff.create_distplot([athlete_df['Age'].dropna()],['Age
Distribution'],show_hist=False,show_rug=False)
fig.show()
```

{"data":[{"y":[1.0695195716989126e-5,1.3464104361132529e-5,1.7046194382555075e-5,2.189982419088749e-5,2.8659225136091005e-5,3.8184278800250626e-5,5.154367630111859e-5,6.987699964819074e-5,9.423459723133057e-5,1.256432291306863e-4,1.6555176962339117e-4,2.1641132902107955e-4,2.817816521824398e-4,3.6557859835728145e-4,4.70986120072901734e-4,6.00343690800567e-4,7.56809027017176e-4,9.466577140528732e-4,1.17947872048005e-3,1.4646272966631394e-3,1.8061941370278894e-3,2.2018440467008295e-3,2.648702503520038e-3,3.152154259549166e-3,3.727800566964018e-3,4.390764794288225e-3,5.1393325934540384e-3,5.949534915848407e-3,6.79087910112219e-3,7.652903614171944e-3,8.557100035417564e-3,9.538309980630441e-3,1.060978275613107e-2,1.1747273701498686e-2,1.2913379761745055e-2,1.4102163207168292e-2,1.5357648072144368e-2,1.674135703489714e-2,1.8277896946654083e-2,1.9937272617529155e-2,2.1679260972163383e-2,2.3516521130823316e-2,2.552337259459293e-2,2.7768396541796872e-2,3.0235633254609304e-2,3.2825533478708355e-2,3.545068759647512e-2,3.8133145619728664e-2,4.099063392978879e-2,4.410688822193895e-2,4.7413918731563275e-2,5.07225686468365e-2,5.38921142416185e-2,5.697191405506583e-2,6.0155487604027974e-2,6.357468368970523e-2,6.714166924658581e-2,7.061621013404423e-2,7.384415422971917e-2,7.691665671272778e-2,8.00619750891688e-2,8.336043762169881e-2,8.6580325675559171e-2,8.933143552530312e-2,9.14069824050788e-2,9.297113190137198e-2,9.438899873192172e-2,9.586053611462705e-2,9.72301050102457e-2,9.81680515581808e-2,9.853320183153294e-2,9.853283085843231e-2,9.850817339289271e-2,9.857004232776535e-2,9.847088856285695e-2,9.785296856445488e-2,9.66171786667117e-2,9.50313440475289e-2,9.347105536329112e-2,9.20674957601148e-2,9.062380463546685e-2,8.887075479984072e-2,8.677900337509359e-2,8.460012309547023e-2,8.260702881785333e-2,8.082001420016727e-2,7.900646987596978e-2,7.694268706697519e-2,7.465271359761409e-2,7.23770040219579e-2,7.031777599146868e-2,6.843799309963712e-2,6.651983368727078e-2,6.440689243167846e-2,6.216775583818056e-2,6.0016045106067534e-2,5.808511496162052e-2,5.6300212174632214e-2,5.4473841857847656e-2,5.250987208369408e-2,5.049655385243626e-2,4.859900152043319e-2,4.687657915193591e-2,4.522096584433536e-2,4.3473050325543196e-2,4.15880842317232e-2,3.967748895682199e-2,3.789397603285487e-2,3.628831306164639e-2,3.478325534765563e-2,3.3277320036153511e-2,3.175756916158279e-2,3.030366678281366e-2,2.8989774763053946e-2,2.7797108980864835e-2,2.6629804934531404e-2,2.5411786851551845e-2,2.415967565928165e-2,2.2959203268434907e-2,2.18769510125986188e-2,2.0901886020492217e-2,1.997151590124801e-

2,1.9045698797472584e-2,1.8146465965930825e-2,1.7326110907046707e-2,1.660445121148066e-2,1.594479849566012e-2,1.5291586006591305e-2,1.46259692782527334e-2,1.3980218485376271e-2,1.339922295439107e-2,1.2893190557470673e-2,1.2430860836624926e-2,1.197598136182623e-2,1.1524307333093526e-2,1.1101519811735665e-2,1.0727172863303666e-2,1.0385955883854688e-2,1.0037638342779343e-2,9.654472944946157e-3,9.246350558822095e-3,8.848556683238643e-3,8.486695217491284e-3,8.15541599639031e-3,7.830533923532253e-3,7.498516640503182e-3,7.171251701719053e-3,6.872334079681399e-3,6.611932474547973e-3,6.377615707746397e-3,6.149405667997145e-3,5.921193817922326e-3,5.705421203163755e-3,5.5174122171031395e-3,5.357580497788331e-3,5.2107039083804305e-3,5.062016491400441e-3,4.91193765528795e-3,4.773766679982202e-3,4.6577246458164865e-3,4.559131071737905e-3,4.4631887140342925e-3,4.360493249451705e-3,4.256177801808524e-3,4.163147209742441e-3,4.087262479235694e-3,4.020612868062108e-3,3.949707248334145e-3,3.869154217850188e-3,3.7858143713137044e-3,3.7091112012249583e-3,3.6382534259292495e-3,3.5603829206091683e-3,3.4619881342605927e-3,3.34203998607697e-3,3.2137793269373907e-3,3.0939651571998336e-3,2.9907685943438714e-3,2.9016821067698127e-3,2.821529993390503e-3,2.7504581237439664e-3,2.693157952901583e-3,2.6511441729246627e-3,2.6177048317818636e-3,2.581904002469378e-3,2.5377565423120676e-3,2.488836798597664e-3,2.443407220004487e-3,2.405240837790605e-3,2.369730998315861e-3,2.3290446972056983e-3,2.2804474053208e-3,2.2288136791708724e-3,2.180823418949833e-3,2.137257152171733e-3,2.0916543688558613e-3,2.036680714483326e-3,1.971444887978596e-3,1.9022402033994758e-3,1.836247086683982e-3,1.7749331099166956e-3,1.7137887338982347e-3,1.6481729081881215e-3,1.5788933088418562e-3,1.5119508816592384e-3,1.4532674802532038e-3,1.4041750792635208e-3,1.3620050897197858e-3,1.32435124612593e-3,1.29190039223009037e-3,1.2668361472673556e-3,1.2490131955336277e-3,1.234557204734309e-3,1.2187291349552063e-3,1.2000884712735358e-3,1.1815093216633488e-3,1.1669842706804093e-3,1.157625078472681e-3,1.150978914619694e-3,1.1440430316933079e-3,1.1363183439285622e-3,1.1292956378689168e-3,1.1228299412019878e-3,1.112501699359703e-3,1.091321337208117e-3,1.054735780854622e-3,1.004334957412753e-3,9.47493620642173e-4,8.917732013556367e-4,8.417623361769605e-4,7.980577554627734e-4,7.600086905661103e-4,7.27908553910934e-4,7.03088757293709e-4,6.860681754086718e-4,6.756139955742715e-4,6.697742824908813e-4,6.675571977061765e-4,6.691072427218084e-4,6.740171379128625e-4,6.797157732644337e-4,6.820360071168231e-4,6.7781339174628e-4,6.671047210008805e-4,6.528163135212802e-4,6.380523036977392e-4,6.237182719868448e-4,6.085495971953017e-4,5.912159820667173e-4,5.721357209873067e-4,5.531466434423352e-4,5.355514898109151e-4,5.187324843417986e-4,5.008274344833841e-4,4.807526566941025e-4,4.595163407411293e-4,4.396031689793063e-4,4.231992822501008e-4,4.110345215328419e-4,4.027097819324984e-4,3.9769552257260427e-4,3.955989367011836e-4,3.953599630184763e-4,3.9447983533969335e-4,3.895663102888467e-4,3.781684515632474e-4,3.604662628884201e-4,3.3936765213571325e-4,3.18957362841603e-4,3.02612115797675e-4,2.921223618541326e-4,2.8797723121065676e-

4,2.8991310239301403e-4,2.9695039585909966e-4,3.071302870067644e-4,3.177699178982704e-4,3.265364427970109e-4,3.3258096869671276e-4,3.366299250390106e-4,3.398153783002654e-4,3.422817466275105e-4,3.428916309577468e-4,3.402947581935947e-4,3.342684620923492e-4,3.259787707866779e-4,3.1691802496189563e-4,3.0760544669726774e-4,2.973370671716978e-4,2.851799988138591e-4,2.7118040627397035e-4,2.5662489837799423e-4,2.432060127409114e-4,2.3198886836756598e-4,2.2311701720092121e-4,2.1630966213754415e-4,2.1140489988404998e-4,2.0831090107877994e-4,2.065230426731949e-4,2.0491560819439503e-4,2.022176625902974e-4,1.9779085480085994e-4,1.9193110582083926e-4,1.853598839958789e-4,1.7838846512453006e-4,1.705740088527923e-4,1.6120249535741564e-4,1.501402733385492e-4,1.3828402547031373e-4,1.27263550246442418e-4,1.1872448479782195e-4,1.1377155746420854e-4,1.1282900252221936e-4,1.1571756463958046e-4,1.2166416419998955e-4,1.29271411948837e-4,1.3673647855250382e-4,1.4244878397442606e-4,1.45651360815491 94e-4,1.4663261179613673e-4,1.4623664367413564e-4,1.450775529970373e-4,1.4311963713478298e-4,1.3993892016486454e-4,1.3533980341087125e-4,1.2969661498081586e-4,1.237091330795083e-4,1.178572223505566e-4,1.1210802997420054e-4,1.0613125231264927e-4,9.975759735238987e-5,9.320837302746674e-5,8.690317929959587e-5,8.1089410270616 75e-5,7.568213903265026e-5,7.0448329640159 07e-5,6.531701189234899e-5,6.050652607723124e-5,5.63820334787414e-5,5.3224510679021854e-5,5.113462885030346e-5,5.010998442962147e-5,5.015332861161009e-5,5.1282366497609904e-5,5.346868721664153e-5,5.6626223652122 89e-5,6.0682372775585505e-5,6.561017168956807e-5,7.128906399765135e-5,7.725105954127875e-5,8.256787374694042e-5,8.609606213387297e-5,8.700055413738344e-5,8.518877967895674e-5,8.130721247716896e-5,7.62949118298157e-5,7.08479896498726e-5,6.519412974350563e-5,5.927953779792145e-5,5.3123743018405394e-5,4.701128257662906e-5,4.139816610949194e-5,3.667709421058315e-5,3.302237340006569e-5,3.0400283521363065e-5,2.866169309531304e-5,2.7599175358184582e-5,2.6947450296204568e-5,2.6397580315422813e-5,2.5678188110715195e-5,2.4663736700971 76e-5,2.3416101877280074e-5,2.2115186957517234e-5,2.0933751134963484e-5,1.9955735071820058e-5,1.9184069510183733e-5,1.8594638194459476e-5,1.8161169757963482e-5,1.7827292422214007e-5,1.7473547485874727e-5,1.6939302105332685e-5,1.610106534398278e-5,1.4944329595834522e-5,1.3564308346667678e-5,1.2093344128608131e-5,1.0616506914271641e-5,9.14207598961605e-6,7.640864348769599e-6,6.111061146684458e-6,4.612733381196664e-6,3.250196603641951e-6,2.1237486682601772e-6,1.2891798910037753e-6,7.489916508997519e-7,4.733633385117328e-7,4.323844715261896e-7,6.197568864898564e-7,1.055874091456658e-6,1.7680014947309346e-6,2.755336025059481e-6,3.9565079565908095e-6,5.241665436743551e-6,6.442669929723801e-6,7.411865392741817e-6,8.075784092014076e-6,8.446638068716394e-6,8.581211361086214e-6,8.516852788622365e-6,8.234195778349215e-6,7.676256007556472e-6,6.8086574150655 03e-6,5.673710894743718e-6,4.39783913198236e-6,3.148679809442663e-6,2.0726669303667e-6,1.2507732368158764e-6,6.907156553261046e-7,3.4867597063617195e-7,1.6079229197288486e-7,6.771129263552514e-8,2.6032144079417208e-8,9.135926400509786e-9,2.926537084449726e-

9,8.556448087487773e-10,2.2832779722637945e-10,5.5608689976481526e-11,1.236073975920533e-11,2.508442308439314e-12,4.707708219912339e-13,1.2186611767633963e-13,2.7669447204484593e-13,1.473454055497477e-12,7.471478216827513e-12,3.4614779842333886e-11,1.4636481823770927e-10,5.648330914583236e-10,1.989351748582448e-9,6.3945653896509316e-9,1.8759380421419748e-8,5.022662001150807e-8,1.2273188975849808e-7,2.7370888421706035e-7,5.570938604140773e-7,1.0348463096677722e-6,1.7544110068867872e-6,2.714531192729784e-6,3.833242853664009e-6,4.940214862678189e-6,5.810765457860635e-6,6.237762849815752e-6,6.111282753667526e-6,5.464417849873817e-6,4.459266518836327e-6,3.321168906401909e-6,2.2574934653660083e-6,1.4004578603342074e-6,7.929057990086562e-7,4.0971432425069157e-7,1.9321850014367967e-7,8.316187458579358e-8,3.2666893693658944e-8,1.1711146481713285e-8,3.83176611056152e-9,1.144212205236637e-9,3.118330662027976e-10,7.756142135518816e-11,1.7606675429233397e-11,3.647681936806013e-12,6.897069128048956e-13,1.1902019712692847e-13,1.874631151522361e-14,2.7076353918980652e-15,4.698376994282314e-16,9.734103624159363e-16,6.803050078569211e-15,4.5304015121402806e-14,2.755274596938001e-13,1.5293485518151217e-12,7.74743590880234e-12,3.5819593839561584e-11,1.5114599396211134e-10,5.820892979930753e-10,2.0460007282035274e-9,6.563804528091963e-9,1.9220156050411582e-8,5.13735465110003e-8,1.2535859904759865e-7,2.7930966939277833e-7,5.68432299883164e-7,1.0572404170980735e-6,1.7987646663055027e-6,2.8038157177093527e-6,4.014153661008334e-6,5.299817142615461e-6,6.4931597717647329e-6,7.449273844885454e-6,8.098895809425058e-6,8.457555712723502e-6,8.582535341631492e-6,8.509013693506721e-6,8.21503283234182e-6],"mode":"lines","x":[10,10.174,10.348,10.522,10.696,10.87,11.044,11.218,11.392,11.566,11.74,11.914,12.088000000000001,12.262,12.436,12.61,12.783999999999999,12.958,13.132,13.306000000000001,13.48,13.654,13.828,14.001999999999999,14.176,14.35,14.524000000000001,14.698,14.872,15.046,15.219999999999999,15.394,15.568,15.742,15.916,16.09,16.264,16.438,16.612000000000002,16.786,16.96,17.134,17.308,17.482,17.656,17.83,18.003999999999998,18.178,18.352,18.526,18.7,18.874000000000002,19.048000000000002,19.222,19.396,19.57,19.744,19.918,20.092,20.266,20.439999999999998,20.614,20.788,20.962,21.136,21.310000000000002,21.484,21.658,21.832,22.006,22.18,22.354,22.528,22.701999999999998,22.875999999999998,23.05,23.224,23.398,23.572,23.746000000000002,23.92,24.094,24.268,24.442,24.616,24.79,24.964,25.137999999999998,25.311999999999998,25.486,25.66,25.834,26.008,26.182,26.356,26.53,26.704,26.878,27.052,27.226,27.4,27.574,27.748,27.922,28.096,28.27,28.444,28.618,28.792,28.966,29.14,29.314,29.488,29.662,29.836,30.01,30.184,30.358,30.532,30.706,30.88,31.054,31.228,31.402,31.576,31.75,31.924,32.098,32.272,32.446,32.620000000000005,32.794,32.968,33.141999999999996,33.316,33.489999999999995,33.664,33.838,34.012,34.186,34.36,34.534,34.708,34.882000000000005,35.056,35.230000000000004,35.403999999999996,35.578,35.751999999999995,35.926,36.1,36.274,36.448,36.622,36.796,36.97,37.144,37.318,37.492000000000004,37.666,37.84,38.013999999999996,38.188,38.361999999999995,38.536,38.71,38.884,39.058,39.232,39.406,39.58,39.754000000000005,39.928,40.102000000000004,40.275999999999996,40.45,40.623999999999995,40.798,40.972,41.146,41.32,41.494,

41.668,41.842,42.016,42.19,42.364,42.538,42.712,42.886,43.06,43.234,43
.408,43.582,43.756,43.93,44.104,44.278,44.452,44.626,44.8,44.974,45.14
8,45.322,45.496,45.67,45.844,46.018,46.192,46.366,46.54,46.714,46.888,
47.062,47.236,47.41,47.584,47.758,47.932,48.106,48.28,48.454,48.628,48
.802,48.976,49.15,49.324,49.498,49.672,49.846,50.02,50.194,50.368,50.5
42,50.716,50.89,51.064,51.238,51.412,51.586,51.76,51.934,52.108,52.282
,52.456,52.63,52.804,52.978,53.152,53.326,53.5,53.674,53.848,54.022,54
.196,54.37,54.544,54.718,54.892,55.066,55.24,55.414,55.588,55.762,55.9
36,56.11,56.284,56.458,56.632,56.806,56.98,57.154,57.328,57.502,57.676
,57.85,58.024,58.198,58.372,58.546,58.72,58.894,59.068,59.242,59.416,5
9.59,59.764,59.938,60.112,60.286,60.46,60.634,60.808,60.982,61.156,61.
33,61.504,61.678,61.852,62.026,62.2,62.374,62.548,62.722,62.896,63.07,
63.244,63.418,63.592,63.766,63.94,64.114,64.288,64.462,64.636,64.81,64
.98400000000001,65.158,65.332,65.506,65.68,65.854,66.02799999999999,66
.202,66.376,66.55,66.72399999999999,66.898,67.072,67.24600000000001,67
.42,67.594,67.768,67.94200000000001,68.116,68.28999999999999,68.464,68
.638,68.812,68.98599999999999,69.16,69.334,69.50800000000001,69.682,69
.856,70.03,70.20400000000001,70.378,70.55199999999999,70.726,70.9,71.0
74,71.24799999999999,71.422,71.596,71.77000000000001,71.944,72.118,72.
292,72.46600000000001,72.64,72.814,72.988,73.162,73.336,73.50999999999
999,73.684,73.858,74.032,74.206,74.38,74.554,74.728,74.902,75.076,75.2
5,75.424,75.598,75.772,75.946,76.12,76.294,76.468,76.642,76.816,76.99,
77.164,77.338,77.512,77.686,77.86,78.034,78.208,78.382,78.556,78.73,78
.904,79.078,79.252,79.426,79.6,79.774,79.948,80.122,80.296,80.47,80.64
4,80.818,80.992,81.166,81.34,81.514,81.688,81.862,82.036,82.21,82.384,
82.558,82.732,82.906,83.08,83.254,83.428,83.602,83.776,83.95,84.124,84
.298,84.472,84.646,84.82,84.994,85.168,85.342,85.516,85.69,85.864,86.0
38,86.212,86.386,86.56,86.734,86.908,87.082,87.256,87.43,87.604,87.778
,87.952,88.126,88.3,88.474,88.648,88.822,88.996,89.17,89.344,89.518,89
.692,89.866,90.04,90.214,90.388,90.562,90.736,90.91,91.084,91.258,91.4
32,91.606,91.78,91.954,92.128,92.302,92.476,92.65,92.824,92.998,93.172
,93.346,93.52,93.694,93.868,94.042,94.216,94.39,94.564,94.738,94.912,9
5.086,95.26,95.434,95.608,95.782,95.956,96.13,96.304,96.478,96.652,96.
826],"showlegend":true,"name":"Age Distribution","legendgroup":"Age
Distribution","xaxis":"x","type":"scatter","marker":{"color":"rgb(31,
119, 180)"},"yaxis":"y"}],"config":
{"plotlyServerURL":"https://plot.ly"},"layout":
{"hovermode":"closest","barmode":"overlay","legend":
{"traceorder":"reversed"},"xaxis":
{"zeroline":false,"anchor":"y2","domain":[0,1]},"template":{"data":
{"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"scattermapbox"
:[{"type":"scattermapbox","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],

[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"pie":
[{"automargin":true,"type":"pie"}],"carpet":[{"aaxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"baxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"type":"carpet"}],"ba
r":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"type":"bar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"barpolar":
[{"type":"barpolar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"scatter3d":
[{"line":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d","marker":
{"colorbar":{"outlinewidth":0,"ticks":""}}}],"contour":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"histogram2d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"scatterpolar":
[{"type":"scatterpolar","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"histogram":
[{"type":"histogram","marker":{"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"histogram2dcontour
":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"parcoords":[{"line":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"scatterpolargl":
[{"type":"scatterpolargl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"scattercarpet":

[{"type":"scattercarpet","marker":{"colorbar":{"outlinewidth":0,"ticks":""}}}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"scatterternary":[{"type":"scatterternary","marker":{"colorbar":{"outlinewidth":0,"ticks":""}}}],"scatter":[{"fillpattern":{"solidity":0.2,"fillmode":"overlay","size":10},"type":"scatter"}],"table":[{"cells":{"fill":{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":{"color":"white"}},"type":"table"}],"scattergeo":[{"type":"scattergeo","marker":{"colorbar":{"outlinewidth":0,"ticks":""}}}],"surface":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"surface"}],"scattergl":[{"type":"scattergl","marker":{"colorbar":{"outlinewidth":0,"ticks":""}}}]},"layout":{"ternary":{"aaxis":{"linecolor":"white","ticks":"","gridcolor":"white"},"baxis":{"linecolor":"white","ticks":"","gridcolor":"white"},"caxis":{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF6"},"autotypenumbers":"strict","shapedefaults":{"line":{"color":"#2a3f5f"}},"annotationdefaults":{"arrowwidth":1,"arrowcolor":"#2a3f5f","arrowhead":0},"coloraxis":{"colorbar":{"outlinewidth":0,"ticks":""}},"title":{"x":5.0e-2},"hoverlabel":{"align":"left"},"colorscale":{"diverging":[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequentialminus":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequential":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"hovermode":"closest","mapbox":{"style":"light"},"paper_bgcolor":"white","scene":{"zaxis":{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gridwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"},"xaxis":{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gridwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"},"yaxis":{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gridwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}

```
},"font":{"color":"#2a3f5f"},"xaxis":{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"polar":{"angularaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"radialaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"plot_bgcolor":"#E5ECF6","geo":
{"subunitcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","sho
wland":true,"showlakes":true,"bgcolor":"white"},"yaxis":
{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"]}},"yaxis":{"anchor":"free","domain":
[0,1],"position":0}}}
```

```python
x = []
name = []
for sport in famous_sports:
    temp_df = athlete_df[athlete_df['Sport'] == sport]
    x.append(temp_df[temp_df['Medal']=='Gold']['Age'].dropna())
    name.append(sport)


fig=ff.create_distplot(x,name,show_hist=False,show_rug=False)

fig.show()

# Now finish the analyzing dataset
```