Hello Folks Myself Nirbhay Tiwari (Data Scientist) Lets Begin Our Olympics Data Analysis Project (Date: 12/20/2023)

```
# importing all necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn

df = pd.read_csv('athlete_events.csv') #Reading  CSV Data

# Reading Top 5 Records From the Athlete Dataset

df.head()
```

```
   ID                      Name Sex   Age  Height  Weight
Team  \
0   1                 A Dijiang   M  24.0   180.0    80.0
China
1   2                  A Lamusi   M  23.0   170.0    60.0
China
2   3       Gunnar Nielsen Aaby   M  24.0     NaN     NaN
Denmark
3   4      Edgar Lindenau Aabye   M  34.0     NaN     NaN
Denmark/Sweden
4   5  Christine Jacoba Aaftink   F  21.0   185.0    82.0
Netherlands

   NOC        Games  Year  Season       City          Sport  \
0  CHN  1992 Summer  1992  Summer  Barcelona     Basketball
1  CHN  2012 Summer  2012  Summer     London           Judo
2  DEN  1920 Summer  1920  Summer  Antwerpen       Football
3  DEN  1900 Summer  1900  Summer      Paris     Tug-Of-War
4  NED  1988 Winter  1988  Winter    Calgary  Speed Skating

                             Event Medal
0       Basketball Men's Basketball   NaN
1      Judo Men's Extra-Lightweight   NaN
2           Football Men's Football   NaN
3       Tug-Of-War Men's Tug-Of-War  Gold
4  Speed Skating Women's 500 metres   NaN
```

```
# Using Info function on whole dataset to get datatype, Null Values
Contained by Each Column in our dataset

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
```

```
# Using Describe to understand the statistics of our dataset

df.describe()
```

```
                  ID            Age         Height         Weight  \
count  271116.000000  261642.000000  210945.000000  208241.000000   
mean    68248.954396      25.556898     175.338970      70.702393   
std     39022.286345       6.393561      10.518462      14.348020   
min         1.000000      10.000000     127.000000      25.000000   
25%     34643.000000      21.000000     168.000000      60.000000   
50%     68205.000000      24.000000     175.000000      70.000000   
75%    102097.250000      28.000000     183.000000      79.000000   
max    135571.000000      97.000000     226.000000     214.000000   

                Year  
count  271116.000000  
mean     1978.378480  
std        29.877632  
min      1896.000000  
25%      1960.000000  
50%      1988.000000  
75%      2002.000000  
max      2016.000000  
```

```
# Now i will be importing another dataset that contains region related data

dfreg = pd.read_csv('datasets_31029_40943_noc_regions.csv')
```

```
dfreg.head()
```

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

```
# Now i am going to merge both of the dataset based on common column
'NOC' in both table, then using head to see top 5 records

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

merge.head()
```

```
   ID                        Name Sex   Age  Height  Weight
Team  \
0   1                   A Dijiang   M  24.0   180.0    80.0
China
1   2                    A Lamusi   M  23.0   170.0    60.0
China
2   3       Gunnar Nielsen Aaby   M  24.0     NaN     NaN
Denmark
3   4      Edgar Lindenau Aabye   M  34.0     NaN     NaN
Denmark/Sweden
4   5  Christine Jacoba Aaftink   F  21.0   185.0    82.0
Netherlands

   NOC        Games  Year  Season       City          Sport  \
0  CHN  1992 Summer  1992  Summer  Barcelona     Basketball
1  CHN  2012 Summer  2012  Summer     London           Judo
2  DEN  1920 Summer  1920  Summer  Antwerpen       Football
3  DEN  1900 Summer  1900  Summer      Paris     Tug-Of-War
4  NED  1988 Winter  1988  Winter    Calgary  Speed Skating

                             Event Medal       region notes
0       Basketball Men's Basketball   NaN        China   NaN
1      Judo Men's Extra-Lightweight   NaN        China   NaN
2           Football Men's Football   NaN      Denmark   NaN
3       Tug-Of-War Men's Tug-Of-War  Gold      Denmark   NaN
4  Speed Skating Women's 500 metres   NaN  Netherlands   NaN
```

```
# Here i am using columns object to read all the columns of our
dataset in list, we can see clearly there's 2 new column added named
region & notes

merge.columns.tolist()
```

```
['ID',
 'Name',
 'Sex',
 'Age',
 'Height',
 'Weight',
 'Team',
 'NOC',
 'Games',
 'Year',
 'Season',
 'City',
 'Sport',
 'Event',
 'Medal',
 'region',
 'notes']
```

```
# Now i will be creating new Dataframe that only contains data of
goldmedalist in it then using head for showing only top 5 rows

goldmedalist = merge[(merge.Medal == 'Gold')]

goldmedalist.head()
```

```
    ID                      Name Sex   Age  Height  Weight
Team  \
3    4     Edgar Lindenau Aabye   M  34.0     NaN     NaN
Denmark/Sweden
42  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland
44  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland
48  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland
60  20       Kjetil Andr Aamodt   M  20.0   176.0    85.0
Norway

    NOC        Games  Year  Season         City          Sport  \
3   DEN  1900 Summer  1900  Summer        Paris     Tug-Of-War
42  FIN  1948 Summer  1948  Summer       London     Gymnastics
44  FIN  1948 Summer  1948  Summer       London     Gymnastics
48  FIN  1948 Summer  1948  Summer       London     Gymnastics
60  NOR  1992 Winter  1992  Winter  Albertville  Alpine Skiing

                                Event Medal   region notes
3        Tug-Of-War Men's Tug-Of-War  Gold  Denmark   NaN
42  Gymnastics Men's Team All-Around  Gold  Finland   NaN
44      Gymnastics Men's Horse Vault  Gold  Finland   NaN
```

```
48  Gymnastics Men's Pommelled Horse  Gold  Finland   NaN
60       Alpine Skiing Men's Super G  Gold   Norway   NaN

# As we can see that index of our goldmedalist is not in sequence at
all it might cause error as we deep dive further more
# So now i am resetting the index of our goldmedalist dataframe and
showing the top 5 records

goldmedalist.reset_index(drop=True, inplace=True)
goldmedalist.head()
```

```
   ID                      Name Sex   Age  Height  Weight
Team  NOC  \
0   4     Edgar Lindenau Aabye   M  34.0     NaN     NaN
Denmark/Sweden  DEN
1  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland  FIN
2  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland  FIN
3  17  Paavo Johannes Aaltonen   M  28.0   175.0    64.0
Finland  FIN
4  20       Kjetil Andr Aamodt   M  20.0   176.0    85.0
Norway  NOR

         Games  Year  Season         City          Sport  \
0  1900 Summer  1900  Summer        Paris     Tug-Of-War
1  1948 Summer  1948  Summer       London     Gymnastics
2  1948 Summer  1948  Summer       London     Gymnastics
3  1948 Summer  1948  Summer       London     Gymnastics
4  1992 Winter  1992  Winter  Albertville  Alpine Skiing

                              Event Medal   region notes
0       Tug-Of-War Men's Tug-Of-War  Gold  Denmark   NaN
1  Gymnastics Men's Team All-Around  Gold  Finland   NaN
2      Gymnastics Men's Horse Vault  Gold  Finland   NaN
3  Gymnastics Men's Pommelled Horse  Gold  Finland   NaN
4       Alpine Skiing Men's Super G  Gold   Norway   NaN
```

```
# Now as we set the index properly, next will check the datatype of
our newly created dataframe name goldmedalist
# to ensure that we dont face any error while we plot graphs

goldmedalist.dtypes
```

```
ID          int64
Name       object
Sex        object
Age       float64
Height    float64
Weight    float64
Team       object
```

```
NOC        object
Games      object
Year        int64
Season     object
City       object
Sport      object
Event      object
Medal      object
region     object
notes      object
dtype: object
```

```
# Checking Count of NA values in Age Column

goldmedalist['Age'].isna().sum()
```

```
148
```

```
# Filling/Replacing NA values with 0

goldmedalist['Age'].fillna(value=0, inplace=True)
```

```
C:\Users\MIPL IT\AppData\Local\Temp\ipykernel_2072\4166540798.py:3:
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
  goldmedalist['Age'].fillna(value=0, inplace=True)
```

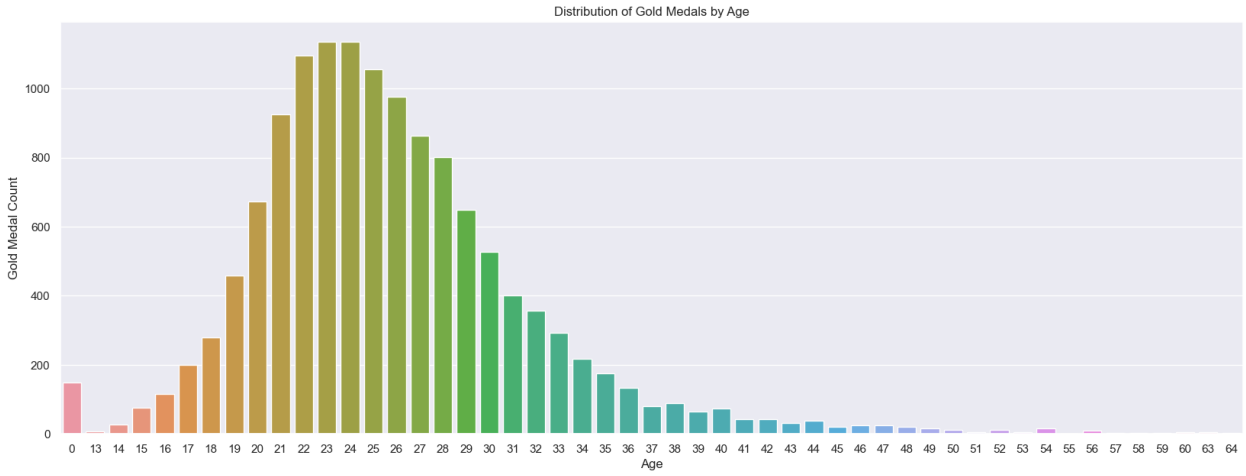
```
# Now Let me quickly convert the Age Column Datatype to int64 from
float64

goldmedalist['Age'] = goldmedalist['Age'].astype('int64')
goldmedalist['Age'].dtypes
```

```
C:\Users\MIPL IT\AppData\Local\Temp\ipykernel_2072\4117185265.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  goldmedalist['Age'] = goldmedalist['Age'].astype('int64')
```

```
dtype('int64')
```

```
# Now i am going to create a graph where i will be showing Gold medals
with respect to age
# For this purpose will pick countplot where i will represent age on
```

```
the X Axis and no of gold medals count on Y Axis

plt.figure(figsize=(20,7))
plt.title('Distribution of Gold Medals by Age')
sns.countplot(data=goldmedalist, x='Age')
plt.ylabel('Gold Medal Count')  # Label for the y-axis
plt.xlabel('Age')  # Label for the x-axis
plt.show()
```




```
# Now Lets Show the number of athletes who has gold medal and their
age is greater than 50 with their other information

goldmedalist50plus = goldmedalist['ID'][(goldmedalist['Age'] >
50)].count()

goldmedalist50plus
```

```
65
```

```
# Filtering Data Using loc method
# goldmedalsports = goldmedalist.loc[goldmedalist['Age'] > 50,
'Sport']

# goldmedalsports

# Lets Filter the Sport's Column where Age is greater than 50

goldmedalistsports = goldmedalist['Sport'][(goldmedalist['Age'] > 50)]

goldmedalistsports
```

```
194         Equestrianism
328               Sailing
581         Equestrianism
582         Equestrianism
1158        Equestrianism
```

```
               ...
12828             Archery
12869            Shooting
12870    Art Competitions
12950             Archery
13130    Art Competitions
Name: Sport, Length: 65, dtype: object
```

```
# Lets reset the index again so that we dont encounter any error..

goldmedalistsports.reset_index(drop=True, inplace=True)
goldmedalistsports.head()
```

```
0    Equestrianism
1          Sailing
2    Equestrianism
3    Equestrianism
4    Equestrianism
Name: Sport, dtype: object
```

```
goldmedalistsports
```

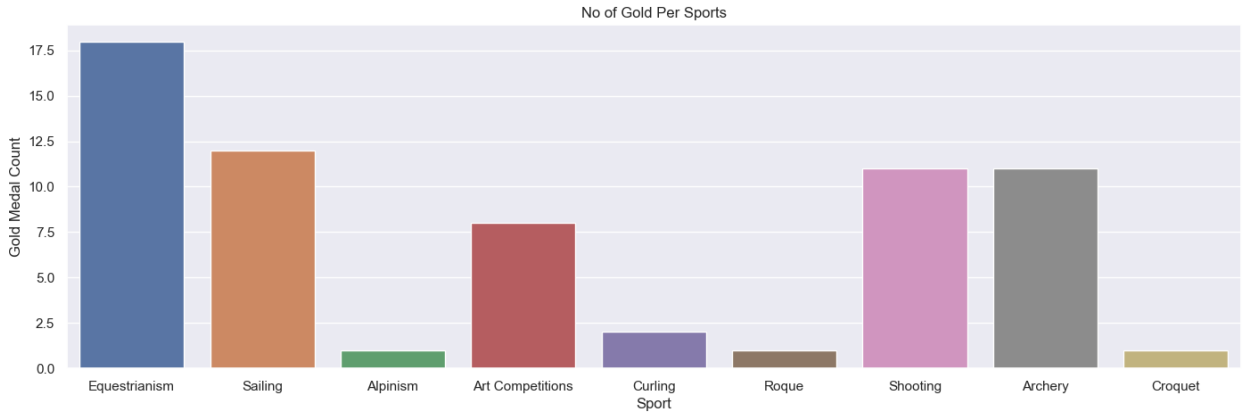
```
0        Equestrianism
1              Sailing
2        Equestrianism
3        Equestrianism
4        Equestrianism
            ...
60             Archery
61            Shooting
62    Art Competitions
63             Archery
64    Art Competitions
Name: Sport, Length: 65, dtype: object
```

```
# Lets Map the Label for our sport column and convert series to
dataframe

goldmedalspersport = pd.DataFrame({'Sport': goldmedalistsports})

goldmedalspersport
```

```
                Sport
0      Equestrianism
1            Sailing
2      Equestrianism
3      Equestrianism
4      Equestrianism
..               ...
60           Archery
61          Shooting
```

```
62  Art Competitions
63           Archery
64  Art Competitions

[65 rows x 1 columns]
```

```
# Now using countplot lets create a graph to show No of Goldmedal hold
by each Sport Category where athlete age is greater than 50

plt.figure(figsize=(17, 5))
plt.title('No of Gold Per Sports')
sns.countplot(data=goldmedalspersport, x='Sport')
plt.ylabel('Gold Medal Count')  # Label for the y-axis
plt.xlabel('Sport')# Label for the x-axis
# plt.xticks(rotation=45)
plt.show()
```




```
merge[['Sex']].value_counts()

Sex
M      196594
F       74522
dtype: int64

# Now lets show the details of women athlete who played sports in
summer

womenparticipation = merge[(merge.Sex == 'F') & (merge.Season ==
'Summer')]

womenparticipation

            ID                                     Name Sex   Age  Height
Weight  \
26           8   Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0
NaN
27           8   Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0
```

```
NaN
32          13                    Minna Maarit Aalto   F  30.0   159.0
55.5
33          13                    Minna Maarit Aalto   F  34.0   159.0
55.5
79          21             Ragnhild Margrethe Aamodt   F  27.0   163.0
NaN
...        ...                                   ...  ..   ...     ...
...
271080  135553  Galina Ivanovna Zybina (-Fyodorova)   F  33.0   168.0
80.0
271099  135560                    Stavroula Zygouri   F  36.0   171.0
63.0
271102  135563             Olesya Nikolayevna Zykina   F  19.0   171.0
64.0
271103  135563             Olesya Nikolayevna Zykina   F  23.0   171.0
64.0
271110  135568                Olga Igorevna Zyuzkova   F  33.0   171.0
69.0

                 Team  NOC        Games  Year  Season
City  \
26       Netherlands  NED  1932 Summer  1932  Summer     Los Angeles

27       Netherlands  NED  1932 Summer  1932  Summer     Los Angeles

32           Finland  FIN  1996 Summer  1996  Summer         Atlanta

33           Finland  FIN  2000 Summer  2000  Summer          Sydney

79            Norway  NOR  2008 Summer  2008  Summer         Beijing

...              ...  ...          ...   ...     ...             ...

271080  Soviet Union  URS  1964 Summer  1964  Summer           Tokyo

271099        Greece  GRE  2004 Summer  2004  Summer          Athina

271102        Russia  RUS  2000 Summer  2000  Summer          Sydney

271103        Russia  RUS  2004 Summer  2004  Summer          Athina

271110       Belarus  BLR  2016 Summer  2016  Summer  Rio de Janeiro


             Sport                                        Event   Medal
\
26       Athletics               Athletics Women's 100 metres     NaN

27       Athletics     Athletics Women's 4 x 100 metres Relay     NaN
```

```
32          Sailing                   Sailing Women's Windsurfer     NaN

33          Sailing                   Sailing Women's Windsurfer     NaN

79         Handball                    Handball Women's Handball    Gold

...             ...                                         ...     ...

271080    Athletics                  Athletics Women's Shot Put  Bronze

271099    Wrestling  Wrestling Women's Middleweight, Freestyle     NaN

271102    Athletics     Athletics Women's 4 x 400 metres Relay  Bronze

271103    Athletics     Athletics Women's 4 x 400 metres Relay  Silver

271110   Basketball               Basketball Women's Basketball     NaN


              region notes
26       Netherlands   NaN
27       Netherlands   NaN
32           Finland   NaN
33           Finland   NaN
79            Norway   NaN
...              ...   ...
271080        Russia   NaN
271099        Greece   NaN
271102        Russia   NaN
271103        Russia   NaN
271110       Belarus   NaN

[59443 rows x 17 columns]
```

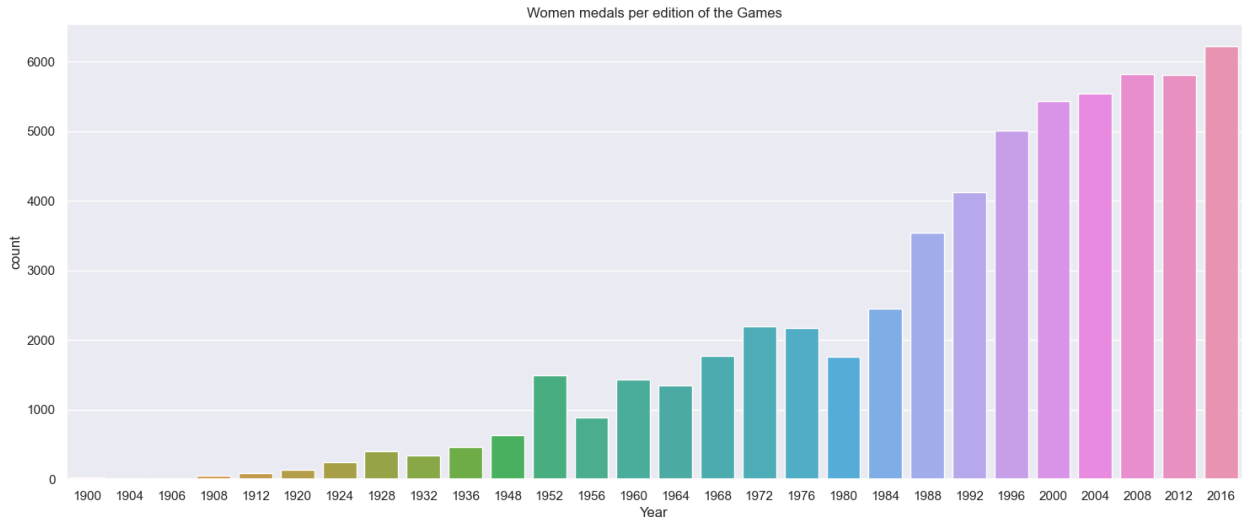
```
# lets reset the index of our dataframe womenparticipation so that we
dont encounter any error

womenparticipation.reset_index(drop=True, inplace=True)

womenparticipation
```

```
           ID                                    Name Sex   Age  Height
Weight  \
0           8   Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0
NaN
1           8   Cornelia "Cor" Aalten (-Strannood)   F  18.0   168.0
NaN
2          13                   Minna Maarit Aalto   F  30.0   159.0
55.5
3          13                   Minna Maarit Aalto   F  34.0   159.0
55.5
```

```
4          21            Ragnhild Margrethe Aamodt   F  27.0   163.0
NaN
...       ...                                  ...  ..   ...     ...
...
59438  135553  Galina Ivanovna Zybina (-Fyodorova)   F  33.0   168.0
80.0
59439  135560                   Stavroula Zygouri   F  36.0   171.0
63.0
59440  135563            Olesya Nikolayevna Zykina   F  19.0   171.0
64.0
59441  135563            Olesya Nikolayevna Zykina   F  23.0   171.0
64.0
59442  135568               Olga Igorevna Zyuzkova   F  33.0   171.0
69.0

               Team  NOC        Games  Year  Season           City  \
0       Netherlands  NED  1932 Summer  1932  Summer     Los Angeles
1       Netherlands  NED  1932 Summer  1932  Summer     Los Angeles
2           Finland  FIN  1996 Summer  1996  Summer         Atlanta
3           Finland  FIN  2000 Summer  2000  Summer          Sydney
4            Norway  NOR  2008 Summer  2008  Summer         Beijing
...             ...  ...          ...   ...     ...             ...
59438  Soviet Union  URS  1964 Summer  1964  Summer           Tokyo
59439        Greece  GRE  2004 Summer  2004  Summer          Athina
59440        Russia  RUS  2000 Summer  2000  Summer          Sydney
59441        Russia  RUS  2004 Summer  2004  Summer          Athina
59442       Belarus  BLR  2016 Summer  2016  Summer  Rio de Janeiro

            Sport                                          Event
Medal  \
0       Athletics                Athletics Women's 100 metres     NaN

1       Athletics     Athletics Women's 4 x 100 metres Relay     NaN

2         Sailing                 Sailing Women's Windsurfer     NaN

3         Sailing                 Sailing Women's Windsurfer     NaN

4        Handball                  Handball Women's Handball    Gold

...           ...                                        ...     ...

59438   Athletics                Athletics Women's Shot Put  Bronze

59439   Wrestling  Wrestling Women's Middleweight, Freestyle     NaN

59440   Athletics     Athletics Women's 4 x 400 metres Relay  Bronze

59441   Athletics     Athletics Women's 4 x 400 metres Relay  Silver
```

```
59442  Basketball                Basketball Women's Basketball     NaN


            region notes
0      Netherlands   NaN
1      Netherlands   NaN
2          Finland   NaN
3          Finland   NaN
4           Norway   NaN
...            ...   ...
59438       Russia   NaN
59439       Greece   NaN
59440       Russia   NaN
59441       Russia   NaN
59442      Belarus   NaN

[59443 rows x 17 columns]
```

```
# Now lets plot the graph to visualise yearwise women participation
growth

sns.set(style="darkgrid")
plt.figure(figsize=(18, 7))
sns.countplot(x='Year', data=womenparticipation)
plt.title('Women medals per edition of the Games')
plt.show()
```

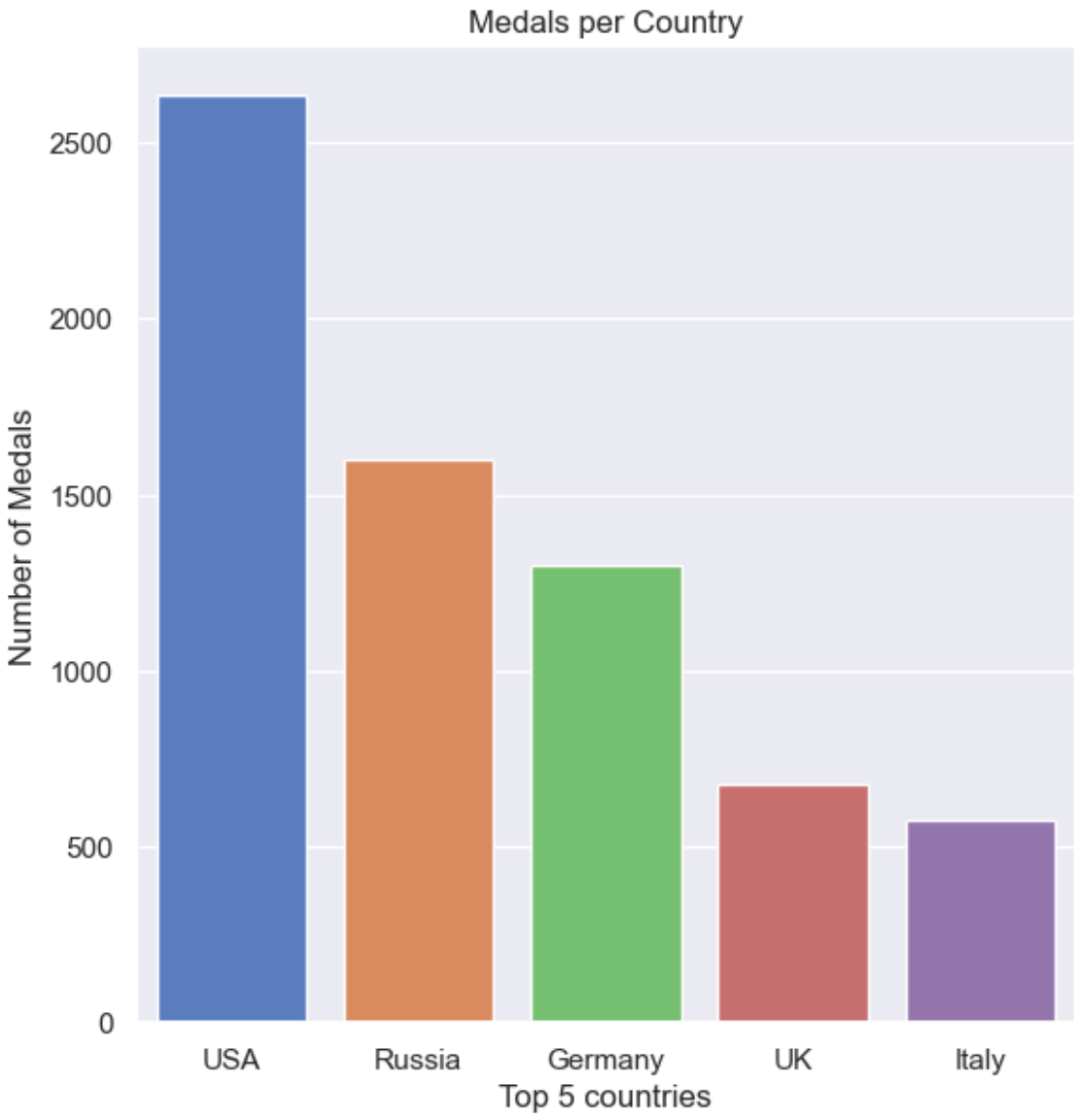



```
# Now lets Filter the Top 5 countries that won medal, later we plot
the data on graph for better presentation

goldbyregion =
goldmedalist.region.value_counts().reset_index(name='Medal').head()
```

```
# Now lets show the no of medals own by each country

goldbyregion
```

	index	Medal
0	USA	2638
1	Russia	1599
2	Germany	1301
3	UK	678
4	Italy	575

```
chart = sns.catplot(x="index", y="Medal", data=goldbyregion,
                    height=6, kind="bar", palette="muted")
chart.despine(left=True)
chart.set_xlabels("Top 5 countries")
chart.set_ylabels("Number of Medals")
plt.title('Medals per Country')
plt.show()
```




```
# Now Lets find out how weight over year have changed for weight
lifters

# lifterovertime = merge[(merge.Sex == 'M') & (merge.Season ==
'Summer')]
# wlovertime = lifterovertime.loc(lifterovertime['Sport'] ==
'Weightlifting')

lifterovertime = merge[(merge['Sex'] == 'M') & (merge['Season'] ==
'Summer')]
wlovertime = lifterovertime[lifterovertime['Sport'] ==
'Weightlifting']

plt.figure(figsize=(18, 5))
sns.pointplot(x='Year', y='Weight', data=wlovertime, palette='Set2')
```

```
plt.title('Weight Change Over Years in Weightlifting')
plt.xlabel('Year')
plt.ylabel('Weight')
plt.show()
```

```
C:\Users\ana\Lib\site-packages\seaborn\algorithms.py:98:
RuntimeWarning: Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
C:\Users\ana\Lib\site-packages\seaborn\algorithms.py:98:
RuntimeWarning: Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
```