Introduction to Data Science

Mini Project

Section : J

Team members:

1. H M Thrupthi – PES1201801987
2. Shaazin Sheikh Shukoor – PES1201801754
3. Ananya Prabhu Angadi – PES1201801433

Dataset Used

120 Years Of Olympic History : Athletes and Results

This is a historical dataset on the modern Olympic Games, including all the Games from Athens 1896 to Rio 2016, documenting all the athletes participating in both the Summer and Winter Olympics over 120 years.

The Olympics are a lens through which to understand global history, including shifting geopolitical power dynamics, women’s empowerment, and the evolving values of society.

Attributes

- ID – Unique number for each athlete
- Name – Athlete's name
- Sex – M or F
- Age – Integer
- Height – In centimeters
- Weight – In kilograms
- Team – Team name
- NOC – National Olympic Committee 3-letter code
- Games – Year and season
- Year – Integer
- Season – Summer or Winter
- City – Host city
- Sport – Sport
- Event – Event
- Medal – Gold, Silver, Bronze, or NA

Features of the Dataset

Name: 120 Years Olympic Events database

Attributes:

```
print(df.columns) #Names of the columns
count_row = df.shape[0]
print(count_row) #Number of rows
```

```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Games',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal'],
      dtype='object')
271116
```

Number of entries = 271116

```
import pandas as pd
import seaborn as sns

df = pd.read_csv('athlete_events.csv')
# df.shape gives (number or rows,number of columns)
df.shape
```

```
(271116, 15)
```

Cleaning the Dataset

Cleaning the dataset

Numerical missing values in 3 columns: Weight, Height and Age

Age	Height	Weight
24.0	180.0	80.0
23.0	170.0	60.0
24.0	NaN	NaN
34.0	NaN	NaN
21.0	185.0	82.0
21.0	185.0	82.0
25.0	185.0	82.0
25.0	185.0	82.0
27.0	185.0	82.0
27.0	185.0	82.0
31.0	188.0	75.0
31.0	188.0	75.0

```python
# Numerical Missing Values
a=df['Height'].isnull().sum()
b=df['Weight'].isnull().sum()
c=df['Weight'].isnull().sum()
print("Missing value in Height column = ",a)
print("Percentage of missing values in Height column=",a/df.shape[0]*100)
print("Missing value in Weight column = ",b)
print("Percentage of missing values in Weight column=",b/df.shape[0]*100)
print("Missing value in Age column = ",b)
print("Percentage of missing values in Age column=",c/df.shape[0]*100)
```

```
Missing value in Height column =  60171
Percentage of missing values in Height column= 22.193821095029435
Missing value in Height column =  62875
Percentage of missing values in Height column= 23.19118015904631
Missing value in Age column =  62875
Percentage of missing values in Age column= 23.19118015904631
```

Before Cleaning

```
print(df)
```

```
           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           5  Christine Jacoba Aaftink   F  21.0   185.0    82.0
5           5  Christine Jacoba Aaftink   F  21.0   185.0    82.0
6           5  Christine Jacoba Aaftink   F  25.0   185.0    82.0
7           5  Christine Jacoba Aaftink   F  25.0   185.0    82.0
8           5  Christine Jacoba Aaftink   F  27.0   185.0    82.0
9           5  Christine Jacoba Aaftink   F  27.0   185.0    82.0
```

After Cleaning

```
#Interpolation of immediate data before and after it (average is taken)
df=df.interpolate()
print(df)
```

```
           ID                      Name Sex   Age  Height  \
0           1                 A Dijiang   M  24.0   180.0
1           2                  A Lamusi   M  23.0   170.0
2           3       Gunnar Nielsen Aaby   M  24.0   175.0
3           4      Edgar Lindenau Aabye   M  34.0   180.0
4           5  Christine Jacoba Aaftink   F  21.0   185.0
5           5  Christine Jacoba Aaftink   F  21.0   185.0
6           5  Christine Jacoba Aaftink   F  25.0   185.0
7           5  Christine Jacoba Aaftink   F  25.0   185.0
8           5  Christine Jacoba Aaftink   F  27.0   185.0
9           5  Christine Jacoba Aaftink   F  27.0   185.0
```

Cleaning the dataset

Categorical missing values in 1 column: City

```
# Categorical Missing Values
d=df['City'].isnull().sum()
print("Missing value in City column = ",d)
print("Percentage of missing values in City column=",d/df.shape[0]*100)
```

```
Missing value in City column =  119
Percentage of missing values in City column= 0.04389265111612741
```

```
In [26]: print(df['City'])
```

```
0        Barcelona
1           London
2        Antwerpen
3            Paris
4          Calgary
5          Calgary
6              NaN
7      Albertville
8      Lillehammer
9      Lillehammer
10     Albertville
11     Albertville
12             NaN
13     Albertville
14     Lillehammer
15     Lillehammer
16     Lillehammer
17     Lillehammer
18     Albertville
```

Before Cleaning

In [26]:
```
print(df['City'])
```

```
0        Barcelona
1           London
2        Antwerpen
3            Paris
4          Calgary
5          Calgary
6              NaN
7      Albertville
8      Lillehammer
9      Lillehammer
10     Albertville
11     Albertville
12             NaN
13     Albertville
14     Lillehammer
15     Lillehammer
16     Lillehammer
17     Lillehammer
18     Albertville
```

After Cleaning

```
# Replacing categorical NaNs with most commonly appearing values
df['City'] = df['City'].fillna(df['City'].value_counts().index[0])
```

```
print(df['City'])
```

```
0        Barcelona
1           London
2        Antwerpen
3            Paris
4          Calgary
5          Calgary
6           London
7      Albertville
8      Lillehammer
9      Lillehammer
10     Albertville
11     Albertville
12          London
13     Albertville
14     Lillehammer
```

Initial Observations

Number of medals awarded

- Over 120 years, only a total of 39783 medals have been awarded. That is, almost 85% of the medals column is empty

```
# Number of medals awarded
d=df['Medal'].isnull().sum()
print("Number of medals awarded = ",df.shape[0]-d)
print("Percentage of empty values in the medals column = ",d/df.shape[0]*100)
```

```
Number of medals awarded = 39783
Percentage of empty values in the medals column = 85.3262072323286
```

- This implies that, out of everyone that participates, only about 15 % of Olympic athletes actually get a medal.

Female vs Male Participation

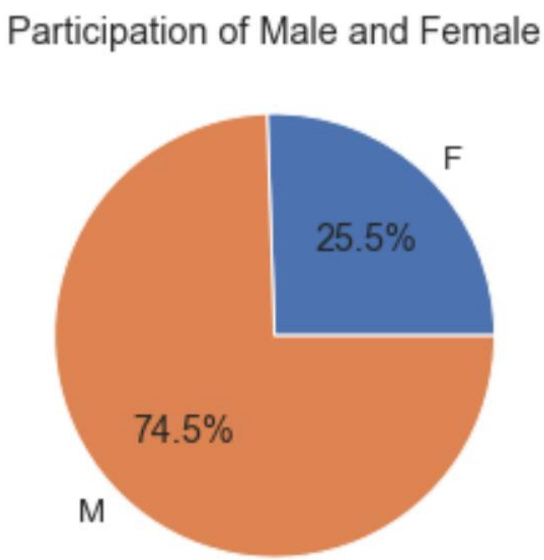
```
print(df['Sex'].value_counts())   #The number of male and female participants
```

```
M    196594
F     74522
Name: Sex, dtype: int64
```

This implies that the male participation is significantly higher than the female participation

Female vs Male Participation

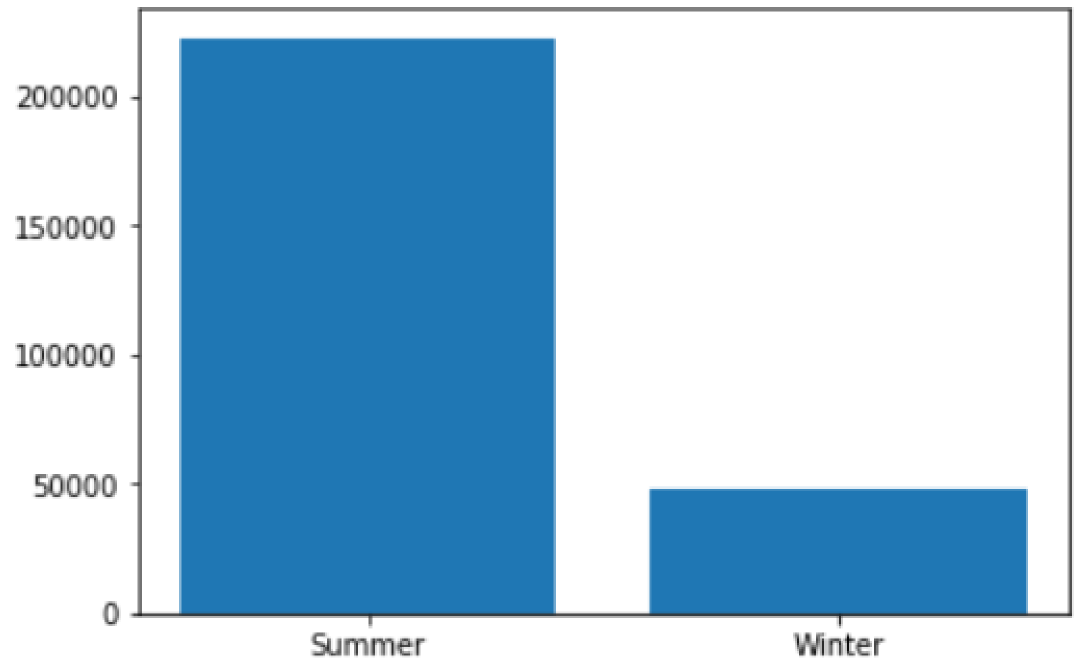
```
var=df.groupby(['Sex']).sum().stack()
temp=var.unstack()
type(temp)
x_list = temp['Age']
label_list = temp.index
plt.axis("equal")
plt.pie(x_list,labels=label_list,autopct="%1.1f%%")
plt.title("Participation of Male and Female")
plt.show()
```



Only about a quarter of the participants are female

Summer Olympics vs Winter Olympics

```
x=["Summer","Winter"]
labels=["Summer","Winter"]
y=[]
count1=0
count2=0
for j in range(0,271116):
    if df['Season'][j]==x[0]:
        count1=count1+1
    if df['Season'][j]==x[1]:
        count2=count2+1
y.append(count1)
y.append(count2)
plt1.bar(x, y, align='center')
plt1.xticks(x,labels)
plt1.yticks(x, y)
plt1.show()
```



This indicates that the Summer Olympics are much more popular than Winter Olympics and attract greater participation.

This could be due to the fact that many countries see harsh winters, thus not allowing them to participate in greater numbers.

Insights

Popular Sports

```
df.Sport.value_counts()
```

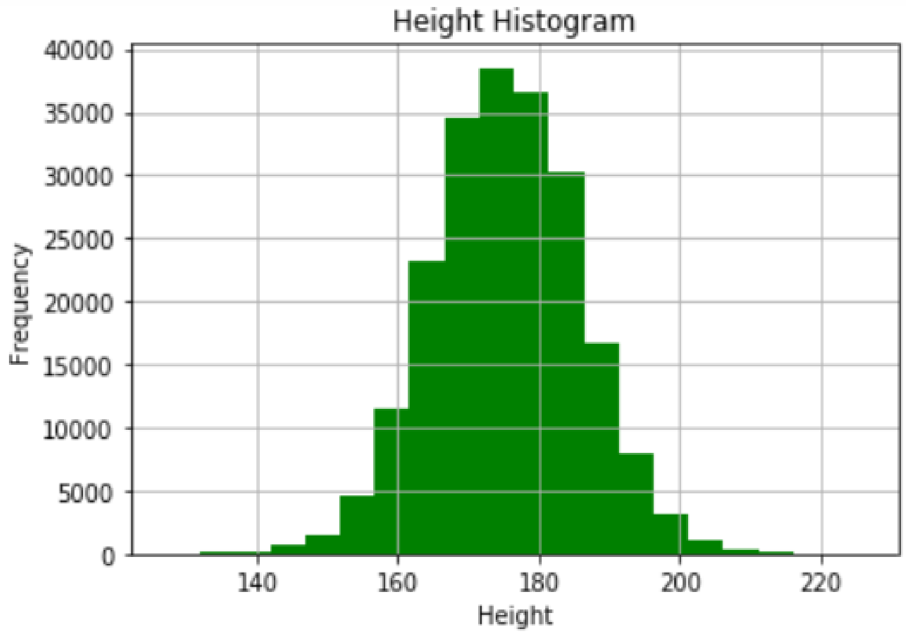
```
Athletics                 38624
Gymnastics                26707
Swimming                  23195
Shooting                  11448
Cycling                   10859
Fencing                   10735
Rowing                    10595
Cross Country Skiing       9133
Alpine Skiing              8829
Wrestling                  7154
Football                   6745
Sailing                    6586
Equestrianism              6344
Canoeing                   6171
Boxing                     6047
Speed Skating              5613
Ice Hockey                 5516
```

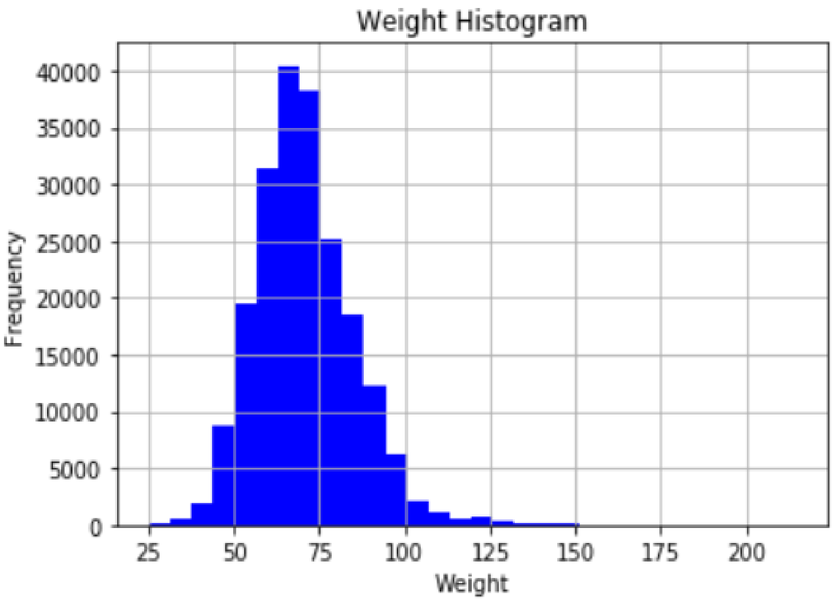
- The most popular sports are athletics, gymnastics and swimming.
- A point to note is that all of these sports have multiple events and participation both individually and as part of a group.

Height, Weight and Age of the Athletes

```
#Plotting Histogram
df['Height'].hist(histtype='stepfilled', color='green',bins=20)
plt.xlabel('Height')
plt.ylabel('Frequency')
plt.title('Height Histogram')
plt.show()
```

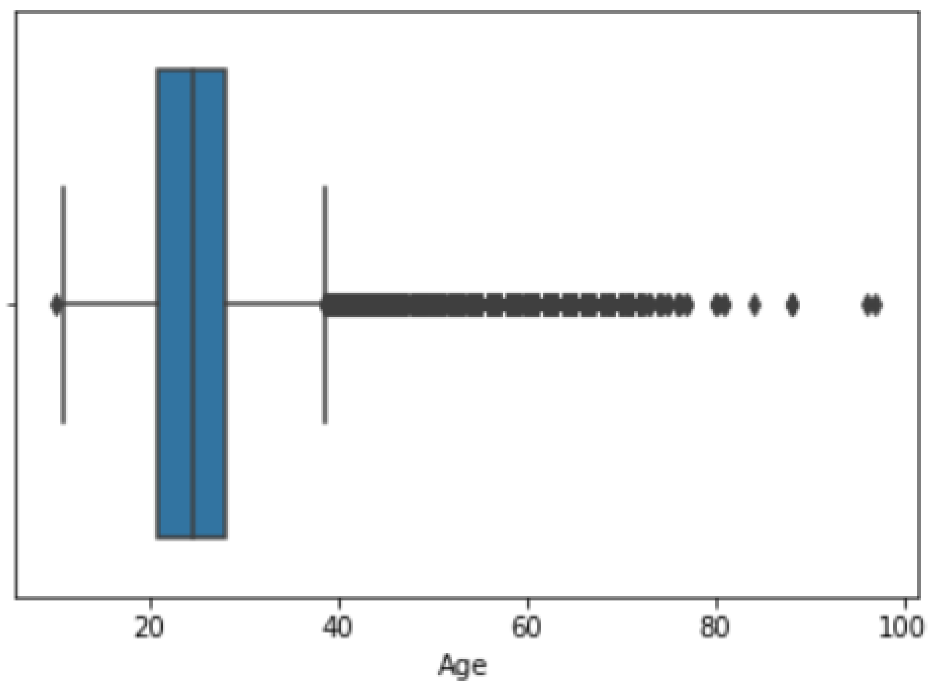
```
df['Weight'].hist(histtype='stepfilled', color='blue',bins=30)
plt1.xlabel('Weight')
plt1.ylabel('Frequency')
plt1.title('Weight Histogram')
plt1.show()
```





Height, Weight and Age of the Athletes

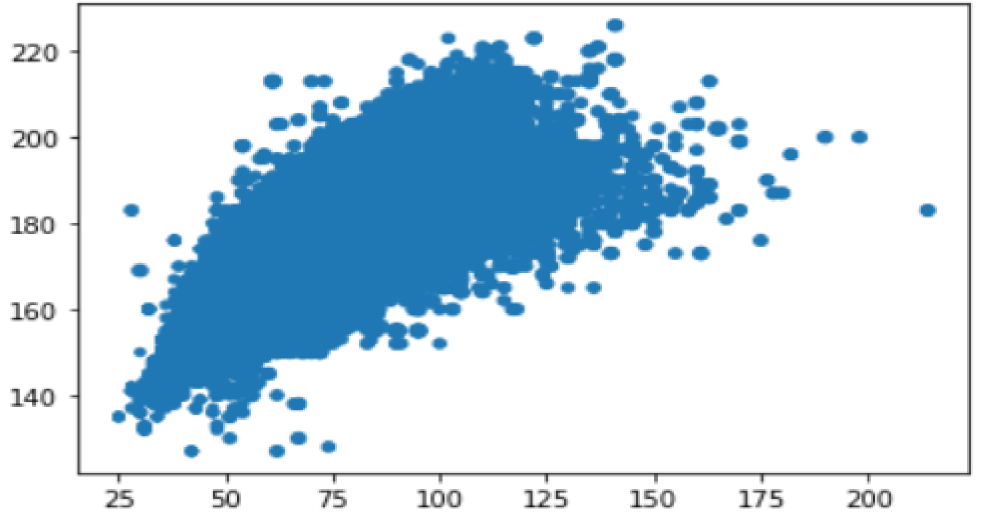
```
sns_plot = sns.boxplot(x=df['Age'])
```



The age is approximately normally distributed with very less standard deviation, albeit the outliers.

Height, Weight and Age of the Athletes

```
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.scatter(df['Weight'],df['Height'], s=df['Age'])
plt.show()
```

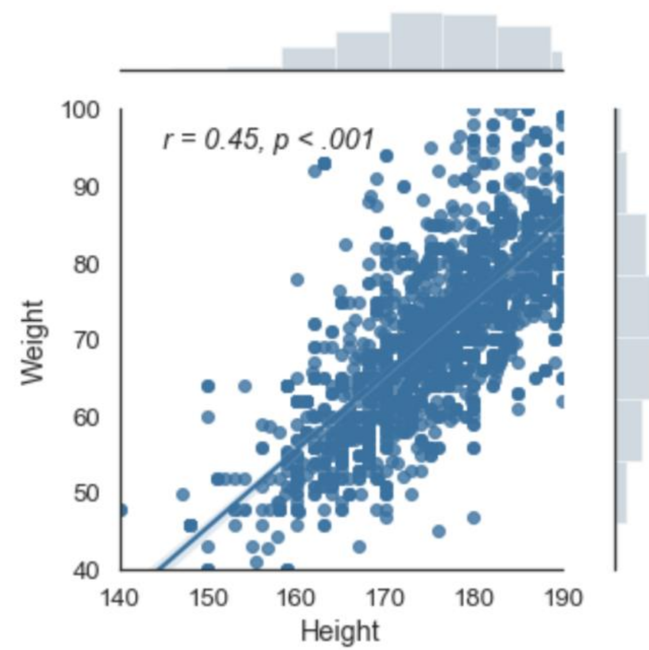


Generally, the height and weight of the athletes are proportional to each other, and age also plays a significant role.

Correlation between height and weight

- Sample size (n) = 2000
- Confidence Interval = 95%
- Correlation Coefficient (r) = 0.738
- Since r value is high, we can conclude that these two attributes are related.

	n	r	CI95%
pearson	2000	0.738	[0.72, 0.76]



Which sports have the heaviest players?

```
male_df = df[df.Sex=='M']#Weights and Statures
sport_weight_height_metrics = male_df.groupby(['Sport'])['Weight','Height'].agg(
  ['min','max','mean'])
sport_weight_height_metrics.Weight.dropna().sort_values('mean', ascending=False)[:5]

#what sports have the heaviest and tallest players, which have the lightest or shortest.
#both height  & weight are heavily dependent on sex,data on the male athletes>female ones,this a
```

	min	max	mean
Sport			
Tug-Of-War	75.0	118.0	95.615385
Basketball	59.0	156.0	91.683529
Rugby Sevens	65.0	113.0	91.006623
Bobsleigh	55.0	145.0	90.387385
Beach Volleyball	62.0	110.0	89.512821
Handball	62.0	132.0	89.387914
Water Polo	61.0	125.0	87.706172
Volleyball	56.0	120.0	86.925926
Baseball	38.0	120.0	85.707792
Ice Hockey	52.0	116.0	83.775593
Rowing	37.0	137.0	83.665663
Judo	52.0	214.0	83.573945
Skeleton	65.0	127.0	82.018349

Heavier players are better suited to games like Tug-Of-War, Basketball and Rugby Sevens

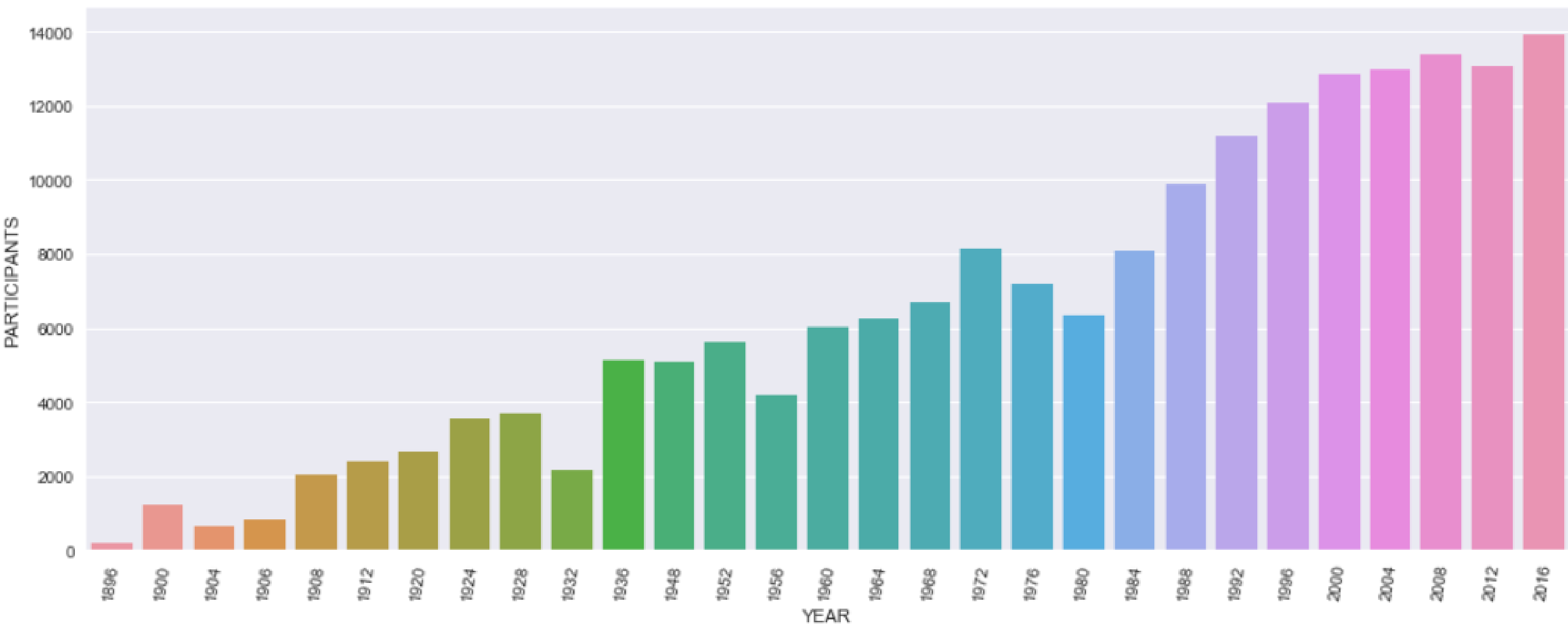
Which sports have the lightest players?

Cycling	48.0	104.0	72.190234
Modern Pentathlon	56.0	91.0	72.068824
Cross Country Skiing	53.0	100.0	71.700832
Table Tennis	50.0	99.0	71.414239
Short Track Speed Skating	51.0	86.0	71.401869
Equestrianism	50.0	100.0	70.924559
Figure Skating	47.0	90.0	69.591644
Triathlon	54.0	82.0	68.803774
Diving	37.0	91.0	67.069378
Nordic Combined	53.0	86.0	66.909560
Trampolining	57.0	84.0	65.837838
Boxing	46.0	140.0	65.296280
Ski Jumping	50.0	85.0	65.245881
Gymnastics	46.0	102.0	63.343605

Lighter players are better suited to games like Gymnastics, Ski Jumping and Boxing

Increase in participation over the years

```
groupedYearID = df.groupby(['Year','ID'],as_index=False).count()[['Year','ID']]  #Group the part
groupedYearID = groupedYearID.groupby('Year',as_index=False).count() #No. of participants every
l = []
for i in [1994,1998,2002,2006,2010,2014]: #The year of winter olympics
    l.append(groupedYearID[groupedYearID.Year == i].index[0]) #Combine winter and summer
for i in l:
    groupedYearID.loc[i,'Year'] = groupedYearID.loc[i,'Year'] +2
groupedYearID = groupedYearID.groupby('Year',as_index=False).sum()
sns.set(rc={'figure.figsize':(16,6)})
plot1 = sns.barplot('Year','ID',data=groupedYearID).set_xticklabels(groupedYearID.Year,rotation=
#plot1.set(xlabel='YEAR',ylabel='Number of people')
plt1.xlabel("YEAR")
plt1.ylabel("PARTICIPANTS")
plt1.show()
```



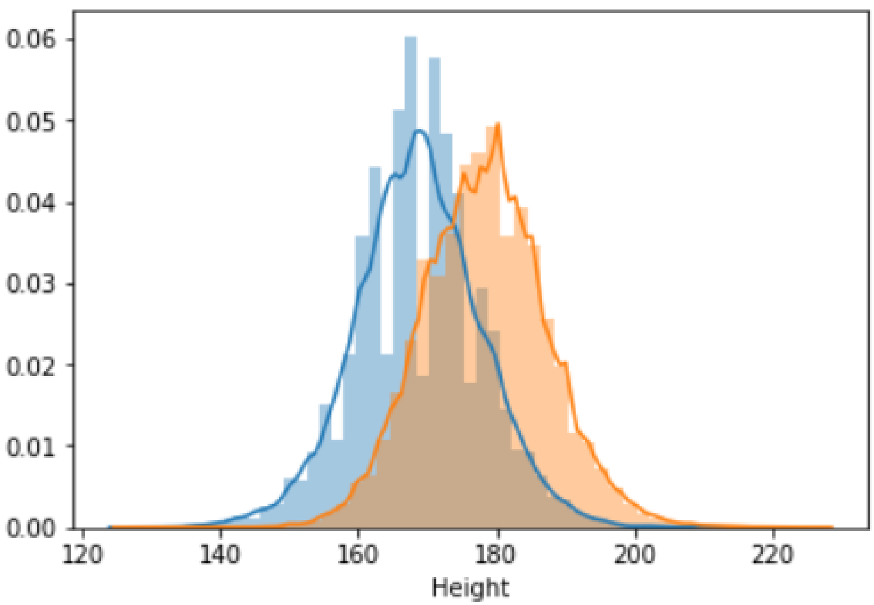
The number of participants has shown a marked increasing over the years, but is now starting to level off

Male vs Female – Height

```
height1=0
height2=0
for i in range(0,271116):
    if df['Sex'][i]=='M':
        height1=height1+df['Height'][i]
    else:
        height2=height2+df['Height'][i]
height1=height1/196594
height2=height2/74522
(sns.distplot(df[df.Sex=='F'].Height),
sns.distplot(df[df.Sex=='M'].Height)
)
print("Average height of male participants = ",height1)
print("Average height of female participants = ",height2)
```

We note that the heights of the male athletes and that of the female athletes are approximately normally distributed

```
Average height of male participants =  178.31555581267386
Average height of female participants =  168.66640214383824
```



Blue: Females
Orange: Males

This implies that the average height of male athletes is slightly higher than that of the female athletes

Diference ~ 10 cms

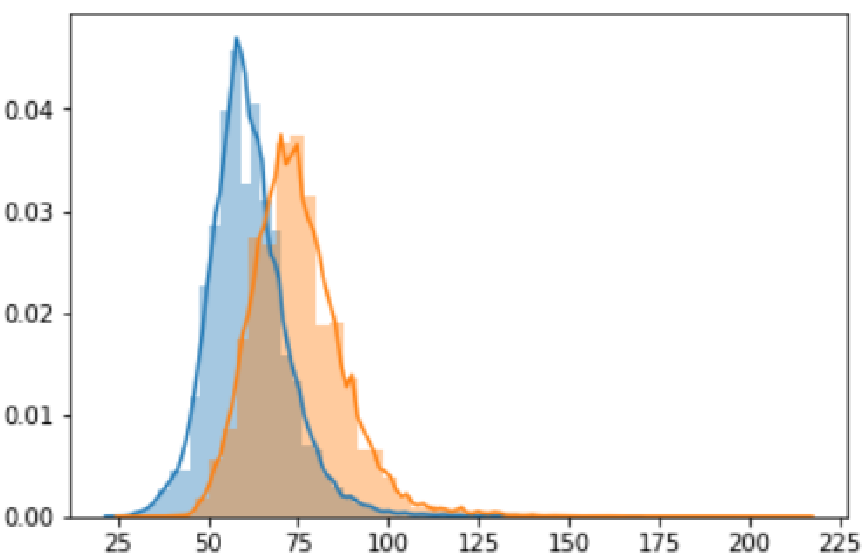
Male vs Female – Weight

```
weight1=0
weight2=0
for i in range(0,271116):
    if df['Sex'][i]=='M':
        weight1=weight1+df['Weight'][i]
    else:
        weight2=weight2+df['Weight'][i]
weight1=weight1/196594
weight2=weight2/74522
(sns.distplot(df[df.Sex=='F'].Weight),
sns.distplot(df[df.Sex=='M'].Weight)
)
print("Average weight of male participants = ",weight1)
print("Average weight of female participants = ",weight2)
```

We note that the weights of the male athletes and that of the female athletes are also normally distributed

```
Average weight of male participants =  74.84730396302415
Average weight of female participants =  61.24981045453986
```

Blue: Females
Orange: Males



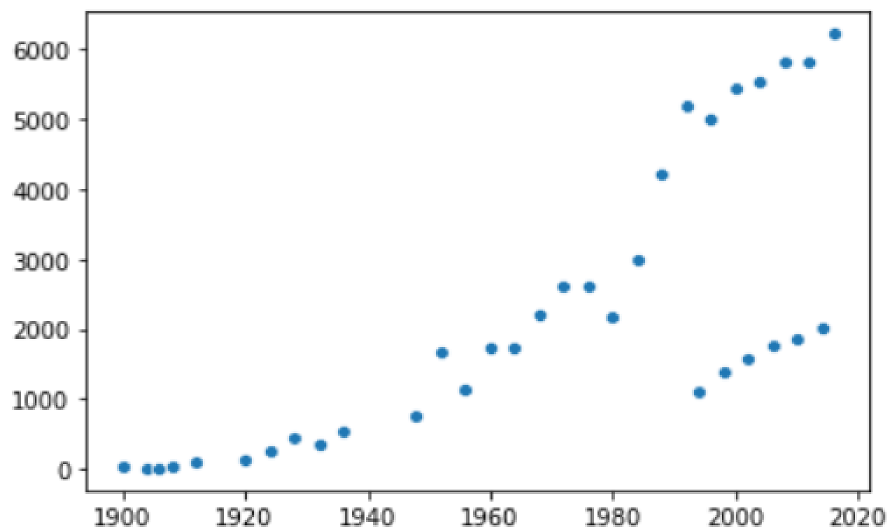
This implies that the average height of male athletes is slightly higher than that of the female athletes

Diference ~ 13 kgs

Increase in female participation over the years

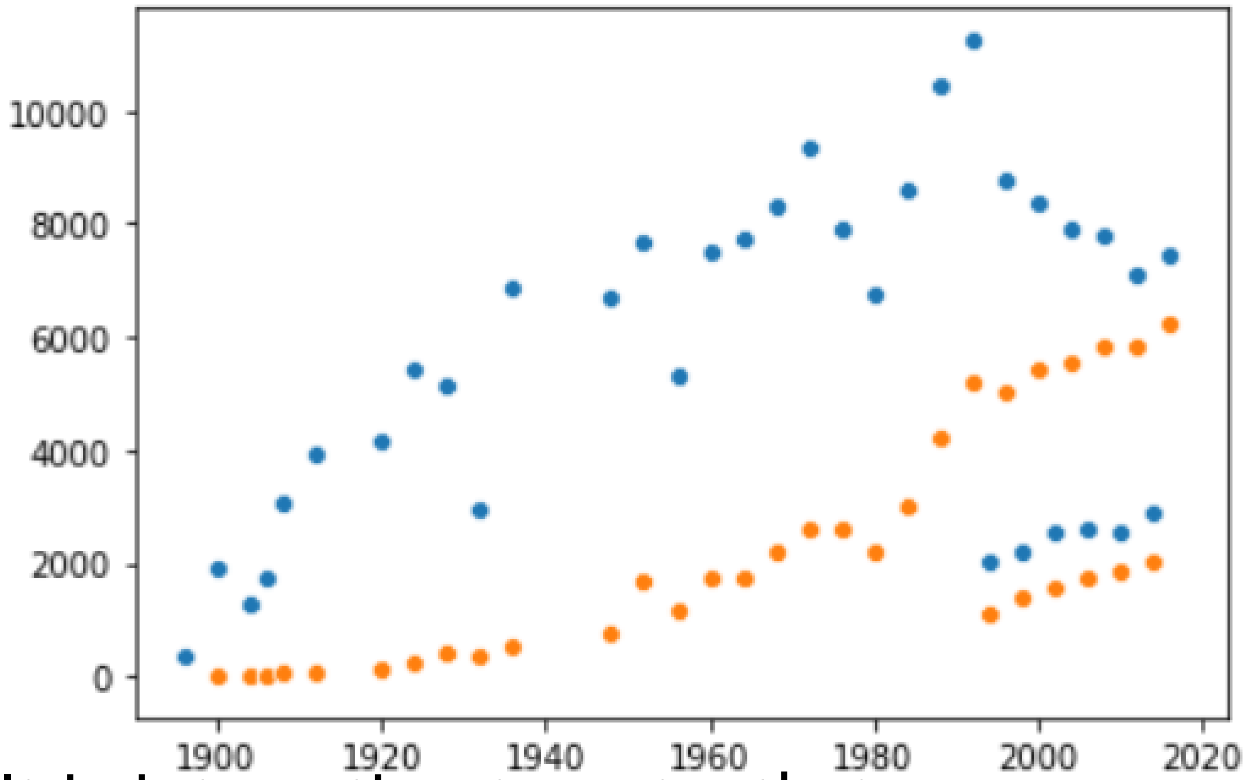
```
female = df[df.Sex=='F']
year_count = female.groupby('Year').agg('count')
years = list(year_count.index)
counts =  list(year_count.Name)
sns.scatterplot(x = years, y = counts)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c2dc7bcb70>
```



The number of women participating in the Olympic Games has shown a steady and marked increase with time, implying that there is greater awareness and greater support for women in sports. This reflects the growth in women empowerment across the world.

Male Participation vs Female Participation

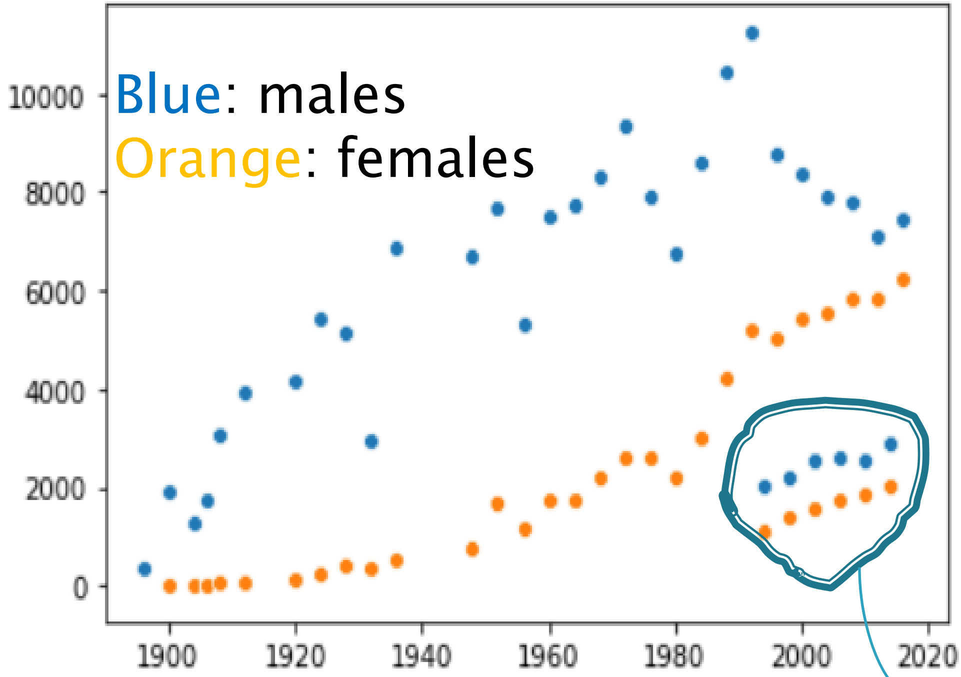


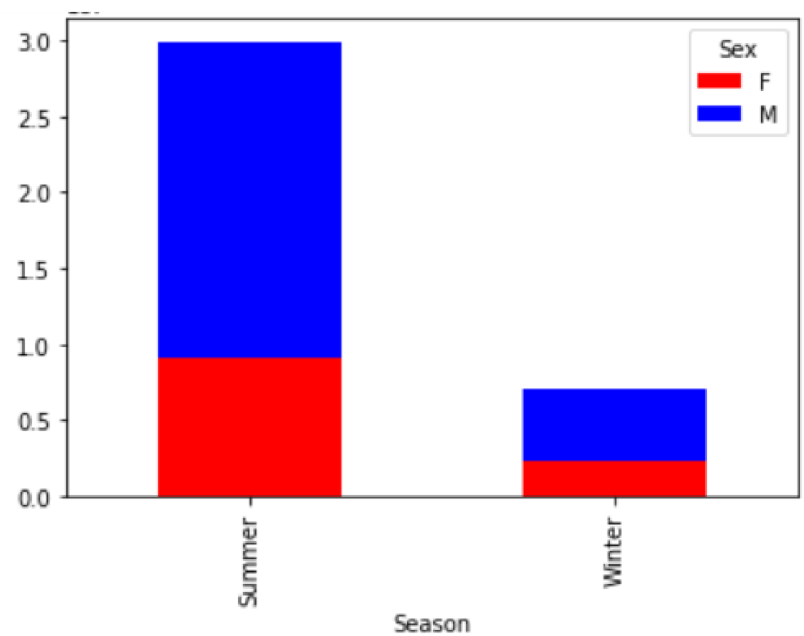
Blue: males
Orange: females

It is interesting to note that women are actually approaching men in sheer numbers!

However, there hasn't been a single year where more females have participated as compared to males.

An important point to note

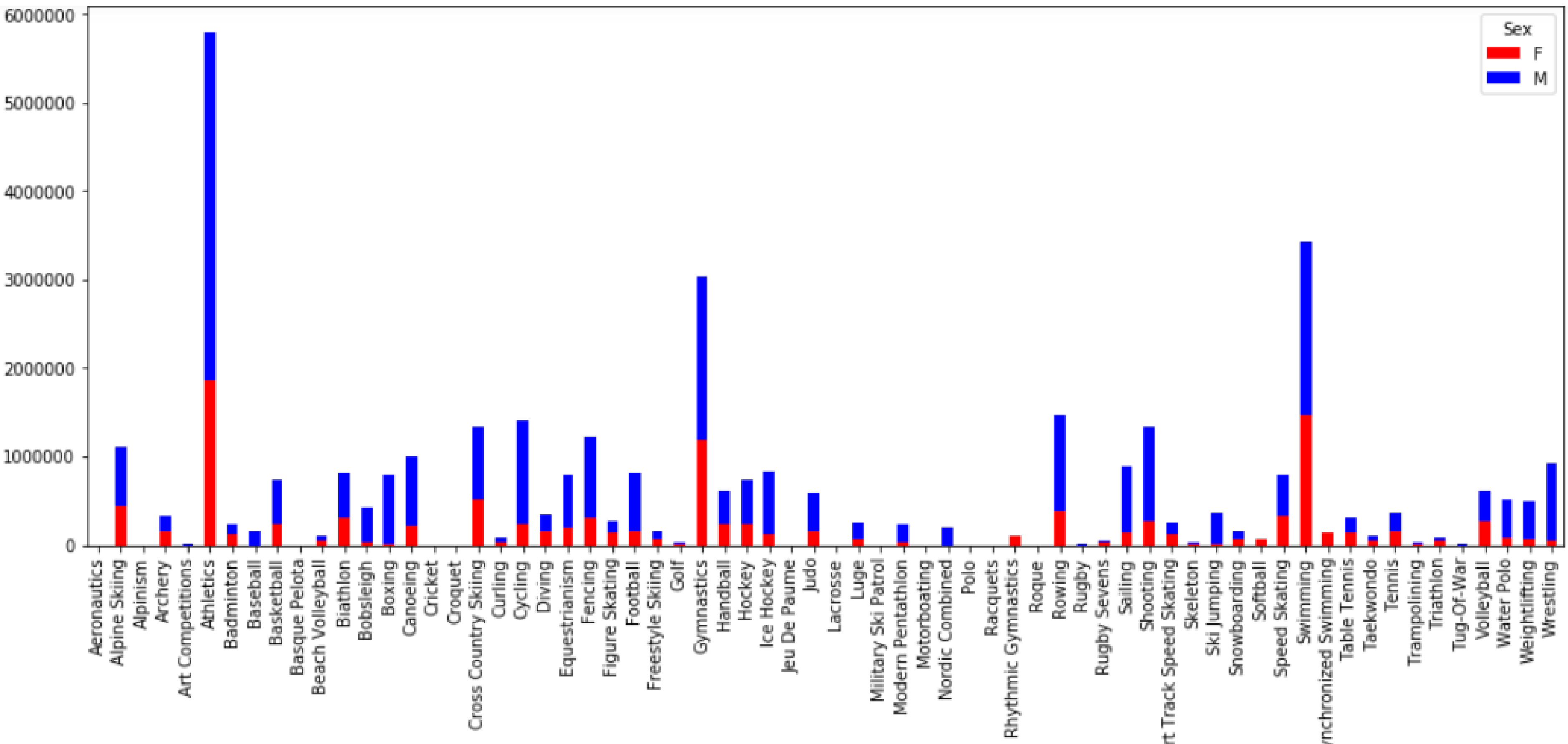




The points towards the lower part of the scatter plot correspond to the Winter Olympics, where the general population is by default low, be it males or females.

These points do not imply a decrease in overall or relative participation.

Male vs Female Participation



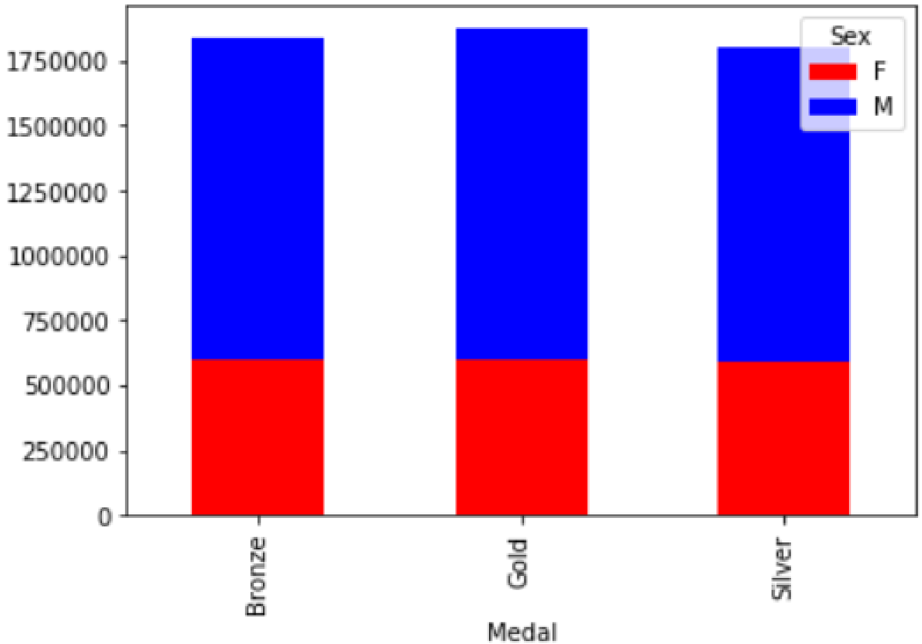
Insights from this graph

- In general, athletics seems to be the most popular sport, followed by (by a long margin) swimming and gymnastics.
- On the other hand, some games like Polo, Cricket and Lacrosse barely show any participation.
- Games like swimming, tennis, table tennis and volleyball have a better women to men participation ratio, as compared to wrestling, weightlifting and baseball, which are completely male dominated.

Male vs Female – Medals

```
var = df.groupby(['Medal','Sex']).Height.sum()
var.unstack().plot(kind='bar',stacked=True,  color=['red','blue'], grid=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2a7b294e0>
```



This implies that male athletes have won a significantly higher number of medals as compared to women, irrespective of the type of medal.

A major reason for why this is so is that, women participation in the Games was very low until recent times, due to heavy discouragement due to socio-economic factors.

Unique participants

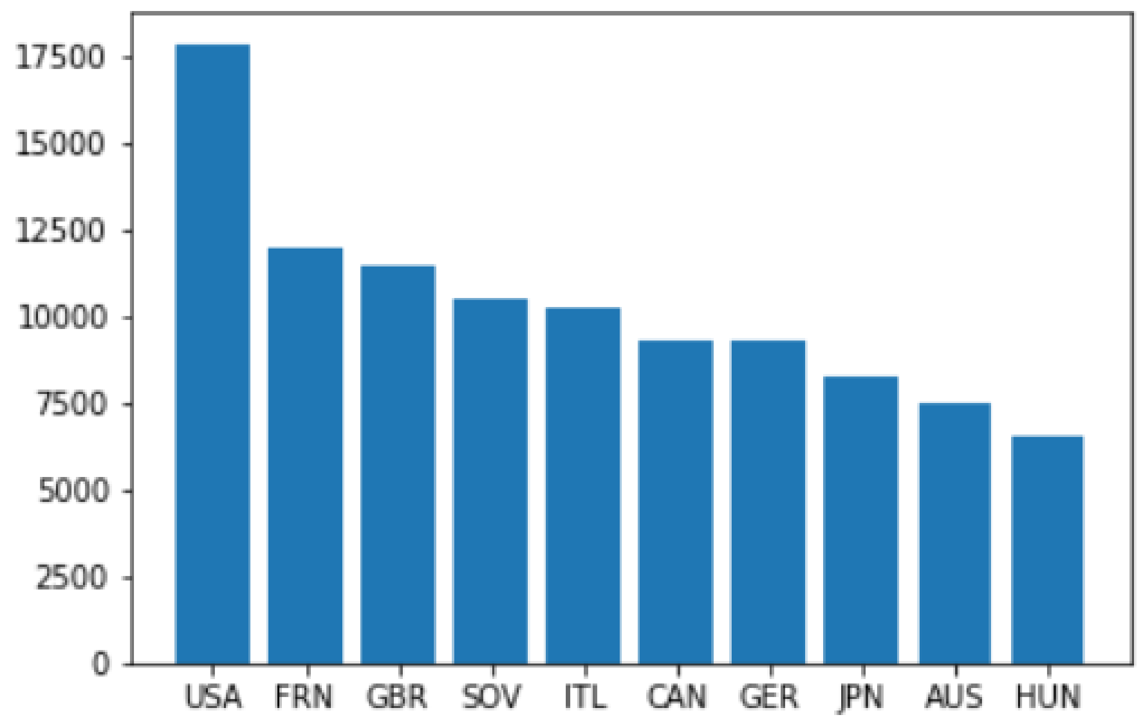
```
total_rows = df.shape[0]
unique_athletes = len(df.Name.unique())
medal_winners = len(df[df.Medal.fillna('None')!='None'].Name.unique())

print("Total athletes = ",total_rows)
print("Total number of unique athletes = ",unique_athletes)
print("Total number of medal winners = ",medal_winners)
```

```
Total athletes =  271116
Total number of unique athletes =  134732
Total number of medal winners =  28202
```

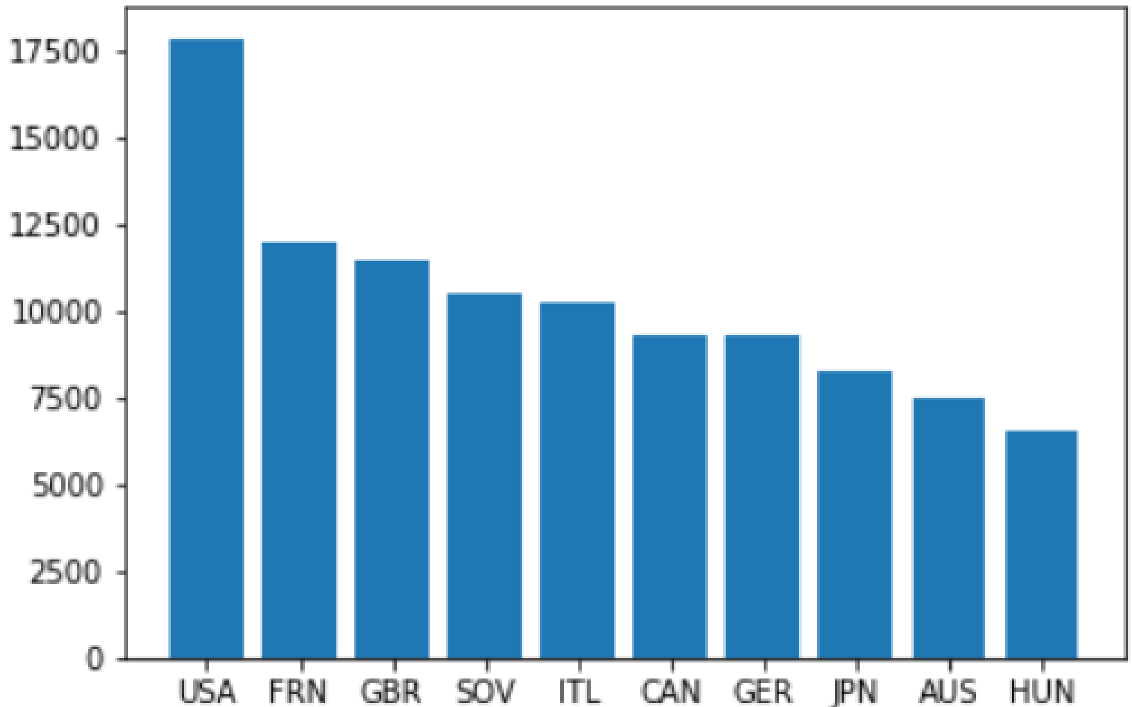
This shows that out of the 271116 athletes, only 134732 are unique participants. That is, most athletes tend to participate in multiple editions of the Games, and in multiple events.

Countries with maximum participation

```
x = ["United States","France","Great Britain", "Soviet Union","Italy","Canada","Germany","Japan"
labels = ["USA", "FRN","GBR","SOV","ITL","CAN","GER","JPN","AUS","HUN"]
y=[]
for i in x:
    count=0
    for j in range(0,271116):
        if df['Team'][j]==i:
            count=count+1
    y.append(count)
for j in range(0,271116):
        if df['Team'][j]=='Russia':
            y[2]=y[2]+1
plt1.bar(x, y, align='center')
plt1.xticks(x,labels)    #optional to set the class names for the bars
plt1.yticks(x, y)     #optional to set the values of y axis
plt1.show()
```



An interesting point to note…



The countries with maximum participation are those which are developed.

This indicates that developed countries have greater resources to spare for betterment of areas like sports, as opposed to developing countries, which are restricted by internal conflicts, unstable governments, poor standard of living and infrastructure.

Medal Count

```
print(df[df.Medal.fillna('None')!='None'].Medal.value_counts())
df[df.Medal.fillna('None')!='None'].shape[0]
```

```
Gold      13372
Bronze    13295
Silver    13116
Name: Medal, dtype: int64

39783
```

Total number of medals awarded

This implies that, the number of gold, silver and bronze medals awarded are approximately equal

Medal Count for each country

```
team_medal_count = df.groupby(['Team','Medal']).Medal.agg('count')
team_medal_count = team_medal_count.reset_index(name='count').sort_values(['count'], ascending=F
team_medal_count.head(10)
```

	Team	Medal	count
726	United States	Gold	2474
727	United States	Silver	1512
725	United States	Bronze	1233
627	Soviet Union	Gold	1058
628	Soviet Union	Silver	716
263	Germany	Gold	679
262	Germany	Bronze	678
626	Soviet Union	Bronze	677
264	Germany	Silver	627
278	Great Britain	Silver	582

Countries with the maximum number of medals:

1)USA (5219 medals)
2)Soviet Union (2451 medals)
3)Germany (1984 medals)

We observe that USA significantly dominates the Olympic Games, with a much higher medal count as compared to every other country.

This may be because the US population is huge, and the country has vast resources to dedicate towards sports.

An interesting observation...

```
def get_country_stats(country):
    return team_medal_count[team_medal_count.Team==country]
```

```
get_country_stats('Soviet Union')
```

	Team	Medal	count
627	Soviet Union	Gold	1058
628	Soviet Union	Silver	716
626	Soviet Union	Bronze	677

- The country with the second highest number of medals won is the Soviet Union, which hasn't been a seperate country for almost 20 years now!
- This means that, even though the Soviet Union does not participate in the games anymore, its record remains, as yet, unbeaten!

On the other hand...

```
df[df.Team=='Croatia'].Year.unique()
```

```
array([2006, 1996, 2000, 1992, 2008, 2012, 2004, 2016, 2014, 2010, 2002,
       1998, 1994], dtype=int64)
```

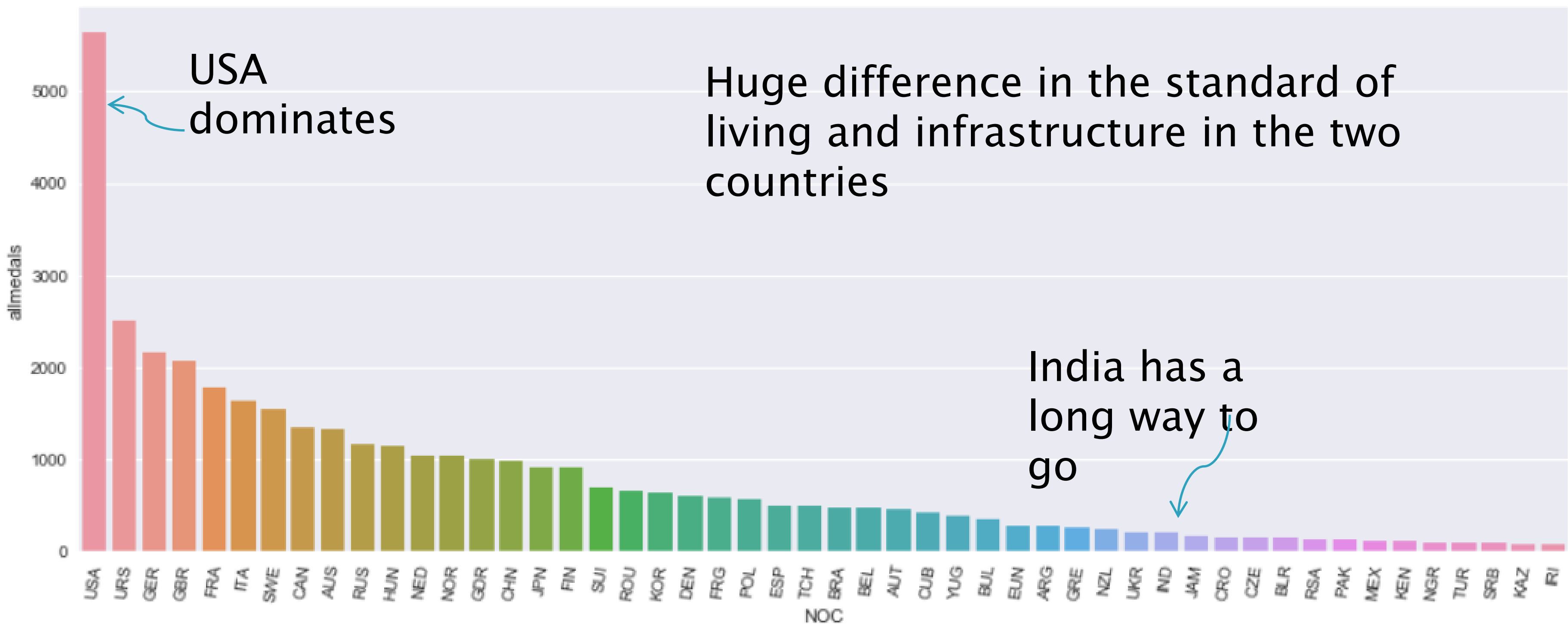
```
get_country_stats('Croatia')
```

	Team	Medal	count
160	Croatia	Gold	58
161	Croatia	Silver	54
159	Croatia	Bronze	37

Croatia, a country which was formed only in 1991, and is relatively new, has already managed to secure more than 149 medals!

Medal Count For Each Country

```
df = pd.concat([df,pd.get_dummies(df.Medal)],axis=1)
df['allmedals'] = df['allmedals'] = df['Bronze'] + df['Gold'] + df['Silver']
groupcountry = df.groupby(by=['NOC'],as_index= False).sum()
top50 = groupcountry.sort_values(by=['allmedals'],ascending = False).head(50)
plot2 = sns.barplot('NOC','allmedals',data=top50).set_xticklabels(top50.NOC,rotation=82)
```



Hypothesis Testing

For the given athletes, we define a hypothesis test for their average height as follows:

Ho : Average height is greater or equal to 175

Ha : Average height is less than 175

Hypothesis Testing

```
#Hypothesis Testing
print("Ho : Average height is less than or equal to 175")
print("Ha : Average height is greater than 175")
mu = 175
df1=df.Height.interpolate()
```

```
Ho : Average height is less than or equal to 175
Ha : Average height is greater than 175
```

Defining the null and the alternative hypothesis

```
sample= pd.DataFrame(df1.sample(n=100))
sample_size = 100
sample
```

Choosing a random sample of size 100

```
sample_mean = sample.Height.mean()
sample_mean
```

```
175.460280952381
```

Sample mean

```
sample_std = sample.Height.std()
sample_std
```

```
10.731843345611225
```

Sample Deviation

Hypothesis Testing

```
alpha = 0.05 #using alpha has 5%
print("z score:")
def z_score(mean,std,size,mu):
    z = (mean-mu)/(std/(size**0.5))
    print("the z score is:",z)
    return z
```

```
z score:
```

Function to compute z-score

```
z = z_score(sample_mean,sample_std,sample_size,mu)
print("one tailed , lower tail")
```

```
the z score is: 0.42889937955191343
one tailed , left tail
```

Z-value of the sample mean

```
p_values = 1-scipy.stats.norm.sf(abs(z)) #one-sided
p_values
```

```
0.6660017741177645
```

p value of the sample mean

Hypothesis Testing

```
if(p_values < alpha):
    print("Null Hypothesis is rejected")
else:
    print("failed to reject Null Hypothesis")
```

```
failed to reject Null Hypothesis
```

Failed to reject the hypothesis, which means that the average height of the athletes may be more than or equal to 175

```
#Confidence interval
a=sample_std
b=sample_size**0.5
c=sample_std/(sample_size**0.5)
d=sample_mean
e=1.645*c
lower=d-e
upper=d+e
upper = sample_mean + (1.645)*(sample_std/(sample_size**0.5))
print("Confidence Interval = (",lower,",",upper,")")
```

```
Confidence Interval = ( 173.69489986488506 , 177.22567632559114 )
```

Confidence Interval

To conclude

- Participation in the Olympics has steadily increased over the past 120 years
- Female participation has also seen a rise
- All the athletes meet certain physical requirements
- Some games are more popular than others
- The Olympics are becoming more and more inclusive as time passes.

Thank you