

EDA on Olympic Games

January 30, 2024

0.0.1 About Dataset:

0.0.2 Context:

The data contains the Year games played, Country the olympics held,,Athlete's Name, Athlete's playing Discipline ,Athlete's country, Athlete's medals recieved, Athlete's Event played, Athlete's Gender from 1896 - 2012.

0.0.3 Problem Statement:

1. Blend the data from differnet files for Analysis.
2. Clean and Manipulate the data the for Analysis.
3. Give statistics and visualization of the data for Athletes from United States.

0.0.4 Import required libraries and load data

```
[1]: # Import Libraries:  
  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
[2]: # Load Required Data:  
  
summer = pd.read_csv("C:/Users/amitm/Desktop/New folder/Task Impetus/Class/  
↳Python/Case Study/Olympics/summer.csv")  
winter = pd.read_csv("C:/Users/amitm/Desktop/New folder/Task Impetus/Class/  
↳Python/Case Study/Olympics/winter.csv")  
track = pd.read_csv("C:/Users/amitm/Desktop/New folder/Task Impetus/Class/  
↳Python/Case Study/Olympics/data.csv")  
countries = pd.read_csv("C:/Users/amitm/Desktop/New folder/Task Impetus/Class/  
↳Python/Case Study/Olympics/dictionary.csv")
```

0.0.5 Exploring and Cleaning the File Summer:

Summer File includes data of Olympic Games Played in Summer.

Data File includes data regarding Athlete and their details.

Countries File includes data of Countries and their details.

```
[3]: # Describing Summer File:
```

```
type(summer)
```

```
[3]: pandas.core.frame.DataFrame
```

```
[4]: summer.head()
```

```
[4]:
```

	Year	City	Sport	Discipline	Athlete	Country	Gender	\
0	1896	Athens	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	
1	1896	Athens	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	
2	1896	Athens	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	
3	1896	Athens	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	
4	1896	Athens	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	

		Event	Medal
0		100M Freestyle	Gold
1		100M Freestyle	Silver
2	100M Freestyle For Sailors		Bronze
3	100M Freestyle For Sailors		Gold
4	100M Freestyle For Sailors		Silver

```
[5]: # Describing Data File:
```

```
track.head(10)
```

```
[5]:
```

	Rank	Time	Name	Country	Date of Birth	Place	\
0	1	00:01:40.910000	David Rudisha	KEN	1988-12-17	1.0	
1	2	00:01:41.010000	David Rudisha	KEN	1988-12-17	1.0	
2	3	00:01:41.090000	David Rudisha	KEN	1988-12-17	1.0	
3	4	00:01:41.110000	Wilson Kipketer	DEN	1970-12-12	1.0	
4	5	00:01:41.240000	Wilson Kipketer	DEN	1970-12-12	1.0	
5	6	00:01:41.330000	David Rudisha	KEN	1988-12-17	1.0	
6	7	00:01:41.510000	David Rudisha	KEN	1988-12-17	1.0	
7	8	00:01:41.540000	David Rudisha	KEN	1988-12-17	1.0	
8	9	00:01:41.730000	Sebastian Coe	GBR	1956-09-29	1.0	
9	9	00:01:41.730000	Wilson Kipketer	DEN	1970-12-12	1.0	

	City	Date	Gender	Event
0	London	2012-09-08	Men	800 m
1	Rieti	2010-08-29	Men	800 m
2	Berlin	2010-08-22	Men	800 m
3	Köln	1997-08-24	Men	800 m
4	Zürich	1997-08-13	Men	800 m
5	Rieti	2011-10-09	Men	800 m
6	Heusden-Zolder	2010-10-07	Men	800 m

7	Saint-Denis	2012-06-07	Men	800 m
8	Firenze	1981-10-06	Men	800 m
9	Stockholm	1997-07-07	Men	800 m

```
[6]: track.tail(2)
```

```
[6]:
```

	Rank	Time	Name	Country	Date of Birth	Place \
18242	1000	00:03:33.250000	Issac Songok	KEN	1984-04-25	2.0
18243	1000	00:03:33.250000	Bethwell Birgen	KEN	1988-08-06	7.0

	City	Date	Gender	Event
18242	Stockholm	2006-07-25	Men	1500 m
18243	Lausanne	2012-08-23	Men	1500 m

```
[7]: #Describing Countries File:
```

```
countries.head()
```

```
[7]:
```

	Country	Code	Population	GDP per Capita
0	Afghanistan	AFG	32526562.0	594.323081
1	Albania	ALB	2889167.0	3945.217582
2	Algeria	ALG	39666519.0	4206.031232
3	American Samoa*	ASA	55538.0	NaN
4	Andorra	AND	70473.0	NaN

```
[8]: # Details of Summer File:
# Print Total Rows:
```

```
print("Rows,cols",summer.shape)
print("Total no of Rows=",len(summer))
print("Total no of Rows=",summer.shape[0])
```

```
Rows,cols (31165, 9)
Total no of Rows= 31165
Total no of Rows= 31165
```

```
[9]: # Print Total Columns:
```

```
print("Total no of Cols",summer.shape[1])
print("Total no of Cols",len(summer.columns))
```

```
Total no of Cols 9
Total no of Cols 9
```

```
[10]: # Name of the columns:
```

```
summer.columns
```

```
[10]: Index(['Year', 'City', 'Sport', 'Discipline', 'Athlete', 'Country', 'Gender',
          'Event', 'Medal'],
          dtype='object')
```

```
[11]: #Data type of each column:
```

```
summer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31165 entries, 0 to 31164
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            31165 non-null  int64
1   City            31165 non-null  object
2   Sport           31165 non-null  object
3   Discipline       31165 non-null  object
4   Athlete         31165 non-null  object
5   Country         31161 non-null  object
6   Gender          31165 non-null  object
7   Event           31165 non-null  object
8   Medal           31165 non-null  object
dtypes: int64(1), object(8)
memory usage: 2.1+ MB
```

```
[12]: #Statistical information on the data:
```

```
summer.describe()
```

```
[12]:
```

	Year
count	31165.000000
mean	1970.482785
std	33.158416
min	1896.000000
25%	1948.000000
50%	1980.000000
75%	2000.000000
max	2012.000000

```
[13]: summer.describe(include='object')
```

```
[13]:
```

	City	Sport	Discipline	Athlete	Country	Gender	Event	\
count	31165	31165	31165	31165	31161	31165	31165	
unique	22	43	67	22762	147	2	666	
top	London	Aquatics	Athletics	PHELPS, Michael	USA	Men	Football	
freq	3567	4170	3638	22	4585	22746	1497	

Medal

```
count    31165
unique      3
top      Gold
freq     10486
```

```
[14]: summer['Sport'].unique()
```

```
[14]: array(['Aquatics', 'Athletics', 'Cycling', 'Fencing', 'Gymnastics',
        'Shooting', 'Tennis', 'Weightlifting', 'Wrestling', 'Archery',
        'Basque Pelota', 'Cricket', 'Croquet', 'Equestrian', 'Football',
        'Golf', 'Polo', 'Rowing', 'Rugby', 'Sailing', 'Tug of War',
        'Boxing', 'Lacrosse', 'Roque', 'Hockey', 'Jeu de paume', 'Rackets',
        'Skating', 'Water Motorsports', 'Modern Pentathlon', 'Ice Hockey',
        'Basketball', 'Canoe / Kayak', 'Handball', 'Judo', 'Volleyball',
        'Table Tennis', 'Badminton', 'Baseball', 'Softball', 'Taekwondo',
        'Triathlon', 'Canoe'], dtype=object)
```

```
[15]: summer['Medal'].unique()
```

```
[15]: array(['Gold', 'Silver', 'Bronze'], dtype=object)
```

```
[16]: summer['Gender'].unique()
```

```
[16]: array(['Men', 'Women'], dtype=object)
```

```
[17]: summer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31165 entries, 0 to 31164
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            31165 non-null  int64
1   City            31165 non-null  object
2   Sport          31165 non-null  object
3   Discipline      31165 non-null  object
4   Athlete         31165 non-null  object
5   Country         31161 non-null  object
6   Gender          31165 non-null  object
7   Event           31165 non-null  object
8   Medal           31165 non-null  object
dtypes: int64(1), object(8)
memory usage: 2.1+ MB
```

```
[18]: summer.head()
```

```
[18]:   Year  City  Sport Discipline  Athlete Country Gender \
0  1896  Athens  Aquatics  Swimming  HAJOS, Alfred  HUN  Men
```

1	1896	Athens	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men
2	1896	Athens	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men
3	1896	Athens	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men
4	1896	Athens	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men

	Event	Medal
0	100M Freestyle	Gold
1	100M Freestyle	Silver
2	100M Freestyle For Sailors	Bronze
3	100M Freestyle For Sailors	Gold
4	100M Freestyle For Sailors	Silver

[19]: *# Modify the data of Summer File:*

```
summer['Athlete'].str.split(', ').str[::-1].str.join(' ').str.title()
```

```
[19]: 0          Alfred Hajos
      1          Otto Herschmann
      2          Dimitrios Drivas
      3          Ioannis Malokinis
      4          Spiridon Chasapis
      ...
      31160         Damian Janikowski
      31161    Ghasem Gholamreza Rezaei
      31162             Rustam Totrov
      31163         Artur Aleksanyan
      31164             Jimmy Lidberg
      Name: Athlete, Length: 31165, dtype: object
```

```
[20]: summer['Athlete']=summer['Athlete'].str.split(', ').str[::-1].str.join(' ').str.
      ↪title()
      summer['Athlete']
```

```
[20]: 0          Alfred Hajos
      1          Otto Herschmann
      2          Dimitrios Drivas
      3          Ioannis Malokinis
      4          Spiridon Chasapis
      ...
      31160         Damian Janikowski
      31161    Ghasem Gholamreza Rezaei
      31162             Rustam Totrov
      31163         Artur Aleksanyan
      31164             Jimmy Lidberg
      Name: Athlete, Length: 31165, dtype: object
```

[21]: `summer.head()`

```
[21]:
```

	Year	City	Sport	Discipline	Athlete	Country	Gender	\
0	1896	Athens	Aquatics	Swimming	Alfred Hajos	HUN	Men	
1	1896	Athens	Aquatics	Swimming	Otto Herschmann	AUT	Men	
2	1896	Athens	Aquatics	Swimming	Dimitrios Drivas	GRE	Men	
3	1896	Athens	Aquatics	Swimming	Ioannis Malokinis	GRE	Men	
4	1896	Athens	Aquatics	Swimming	Spiridon Chasapis	GRE	Men	

	Event	Medal
0	100M Freestyle	Gold
1	100M Freestyle	Silver
2	100M Freestyle For Sailors	Bronze
3	100M Freestyle For Sailors	Gold
4	100M Freestyle For Sailors	Silver

```
[22]: # Merge and Manipulate Summer and Countries:

newsummer=summer.merge(countries,left_on='Country',right_on='Code',how='left')
```

```
[23]: newsummer.head()
```

```
[23]:
```

	Year	City	Sport	Discipline	Athlete	Country_x	Gender	\
0	1896	Athens	Aquatics	Swimming	Alfred Hajos	HUN	Men	
1	1896	Athens	Aquatics	Swimming	Otto Herschmann	AUT	Men	
2	1896	Athens	Aquatics	Swimming	Dimitrios Drivas	GRE	Men	
3	1896	Athens	Aquatics	Swimming	Ioannis Malokinis	GRE	Men	
4	1896	Athens	Aquatics	Swimming	Spiridon Chasapis	GRE	Men	

	Event	Medal	Country_y	Code	Population	\
0	100M Freestyle	Gold	Hungary	HUN	9844686.0	
1	100M Freestyle	Silver	Austria	AUT	8611088.0	
2	100M Freestyle For Sailors	Bronze	Greece	GRE	10823732.0	
3	100M Freestyle For Sailors	Gold	Greece	GRE	10823732.0	
4	100M Freestyle For Sailors	Silver	Greece	GRE	10823732.0	

	GDP per Capita
0	12363.543460
1	43774.985174
2	18002.230578
3	18002.230578
4	18002.230578

```
[24]: newsummer.head().T #transpose
```

```
[24]:
```

	0	1	2	\
Year	1896	1896	1896	
City	Athens	Athens	Athens	
Sport	Aquatics	Aquatics	Aquatics	

Discipline	Swimming	Swimming	Swimming
Athlete	Alfred Hajos	Otto Herschmann	Dimitrios Drivas
Country_x	HUN	AUT	GRE
Gender	Men	Men	Men
Event	100M Freestyle	100M Freestyle	100M Freestyle For Sailors
Medal	Gold	Silver	Bronze
Country_y	Hungary	Austria	Greece
Code	HUN	AUT	GRE
Population	9844686.0	8611088.0	10823732.0
GDP per Capita	12363.54346	43774.985174	18002.230578

	3	4
Year	1896	1896
City	Athens	Athens
Sport	Aquatics	Aquatics
Discipline	Swimming	Swimming
Athlete	Ioannis Malokinis	Spiridon Chasapis
Country_x	GRE	GRE
Gender	Men	Men
Event	100M Freestyle For Sailors	100M Freestyle For Sailors
Medal	Gold	Silver
Country_y	Greece	Greece
Code	GRE	GRE
Population	10823732.0	10823732.0
GDP per Capita	18002.230578	18002.230578

[25]: *# RENaming The Columns:*

```
summer=newsummer[['Year','City','Sport','Discipline','Athlete','Country_x','Gender','Event'],'M
summer.
columns=['Year','City','Sport','Discipline','Athlete','Code','Gender','Event','Medal','Coun
```

[26]: summer.head()

```
[26]:   Year   City   Sport Discipline      Athlete Code Gender \
0  1896  Athens  Aquatics  Swimming      Alfred Hajos  HUN   Men
1  1896  Athens  Aquatics  Swimming      Otto Herschmann  AUT   Men
2  1896  Athens  Aquatics  Swimming      Dimitrios Drivas  GRE   Men
3  1896  Athens  Aquatics  Swimming      Ioannis Malokinis  GRE   Men
4  1896  Athens  Aquatics  Swimming      Spiridon Chasapis  GRE   Men

      Event  Medal  Country
0      100M Freestyle   Gold  Hungary
1      100M Freestyle  Silver  Austria
2  100M Freestyle For Sailors  Bronze  Greece
3  100M Freestyle For Sailors   Gold  Greece
4  100M Freestyle For Sailors  Silver  Greece
```



```
[27]: summer['Discipline'].unique()
```

```
[27]: array(['Swimming', 'Athletics', 'Cycling Road', 'Cycling Track',  
        'Fencing', 'Artistic G.', 'Shooting', 'Tennis', 'Weightlifting',  
        'Wrestling Gre-R', 'Water polo', 'Archery', 'Basque Pelota',  
        'Cricket', 'Croquet', 'Jumping', 'Football', 'Golf', 'Polo',  
        'Rowing', 'Rugby', 'Sailing', 'Tug of War', 'Diving', 'Boxing',  
        'Lacrosse', 'Roque', 'Wrestling Free.', 'Hockey', 'Jeu de Paume',  
        'Rackets', 'Figure skating', 'Water Motorspor', 'Dressage',  
        'Eventing', 'Modern Pentath.', 'Vaulting', 'Ice Hockey',  
        'Basketball', 'Canoe / Kayak F', 'Handball', 'Judo', 'Volleyball',  
        'Canoe / Kayak S', 'Synchronized S.', 'Rhythmic G.',  
        'Table Tennis', 'Badminton', 'Baseball', 'Mountain Bike',  
        'Softball', 'Beach volley.', 'Trampoline', 'Taekwondo',  
        'Triathlon', 'BMX', 'Marathon swimming', 'Synchronized Swimming',  
        'Water Polo', 'Canoe Slalom', 'Canoe Sprint', 'Cycling BMX',  
        'Gymnastics Artistic', 'Gymnastics Rhythmic', 'Modern Pentathlon',  
        'Beach Volleyball', 'Wrestling Freestyle'], dtype=object)
```

```
[28]: # Check For Null Values:
```

```
summer[summer['Gender']=='Men'].count()
```

```
[28]: Year          22746  
City            22746  
Sport           22746  
Discipline      22746  
Athlete         22746  
Code            22744  
Gender          22746  
Event           22746  
Medal           22746  
Country         19036  
dtype: int64
```

```
[29]: summer.isnull().sum()
```

```
[29]: Year          0  
City            0  
Sport           0  
Discipline      0  
Athlete         0  
Code            4  
Gender          0  
Event           0  
Medal           0  
Country         5423
```

dtype: int64

```
[30]: # Getting Statistics on Gender:
```

```
summer[summer['Gender']=='Men']['Athlete'].count()
```

```
[30]: 22746
```

```
[31]: summer[summer['Gender']=='Women']['Athlete'].count()
```

```
[31]: 8419
```

```
[32]: summer.size
```

```
[32]: 311650
```

```
[33]: summer['Athlete'].count()
```

```
[33]: 31165
```

```
[34]: summer['Athlete'].value_counts()
```

```
[34]: Athlete
Michael Phelps      22
Larisa Latynina     18
Nikolay Andrianov   15
Takashi Ono         13
Edoardo Mangiarotti 13
..
Uddin Zaka          1
Hayat Zafar         1
Rashid Muhammad     1
Muhammad Afzal Manna 1
Jimmy Lidberg        1
Name: count, Length: 22759, dtype: int64
```

```
[35]: summer['Country'].value_counts()
```

```
[35]: Country
United States      4585
United Kingdom     1720
France             1396
Germany            1305
Italy              1296
...
Burundi            1
Senegal            1
Djibouti           1
```

```
Netherlands Antilles*      1
Gabon                       1
Name: count, Length: 129, dtype: int64
```

[36]: *#Repeats of each Athlete:*

```
summer[summer['Country']=='United States']['Athlete'].value_counts()
```

[36]: Athlete

```
Michael Phelps      22
Dara Torres         12
Natalie Coughlin    12
Jenny Thompson      12
Matthew Biondi      11
..
Josiah Henson        1
Jay Thomas Evans     1
William Thomas Smith 1
Richard Carroll Conner 1
Jacob Stephen Varner  1
Name: count, Length: 3148, dtype: int64
```

[37]: *# Show Data stats of Top Athlete:*

```
summer[summer['Athlete']=='Michael Phelps']['Medal'].value_counts()
```

[37]: Medal

```
Gold      18
Bronze     2
Silver     2
Name: count, dtype: int64
```

[38]:

```
summer[(summer['Athlete']=='Michael Phelps') &
        ((summer['Medal']=='Gold') | (summer['Medal']=='Silver') |
         (summer['Medal']=='Bronze'))]
```

[38]:

	Year	City	Sport	Discipline	Athlete	Code	Gender	\
25225	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25253	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25258	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25265	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25277	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25286	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25325	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
25361	2004	Athens	Aquatics	Swimming	Michael Phelps	USA	Men	
27224	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men	
27252	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men	
27258	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men	

27264	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men
27276	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men
27291	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men
27327	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men
27366	2008	Beijing	Aquatics	Swimming	Michael Phelps	USA	Men
29270	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men
29298	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men
29309	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men
29340	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men
29370	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men
29405	2012	London	Aquatics	Swimming	Michael Phelps	USA	Men

		Event	Medal	Country
25225		100M Butterfly	Gold	United States
25253		200M Butterfly	Gold	United States
25258		200M Freestyle	Bronze	United States
25265	200M Individual Medley		Gold	United States
25277	400M Individual Medley		Gold	United States
25286	4X100M Freestyle Relay		Bronze	United States
25325	4X100M Medley Relay		Gold	United States
25361	4X200M Freestyle Relay		Gold	United States
27224		100M Butterfly	Gold	United States
27252		200M Butterfly	Gold	United States
27258		200M Freestyle	Gold	United States
27264	200M Individual Medley		Gold	United States
27276	400M Individual Medley		Gold	United States
27291	4X100M Freestyle Relay		Gold	United States
27327	4X100M Medley Relay		Gold	United States
27366	4X200M Freestyle Relay		Gold	United States
29270		100M Butterfly	Gold	United States
29298		200M Butterfly	Silver	United States
29309		200M Medley	Gold	United States
29340	4X100M Freestyle		Silver	United States
29370	4X100M Medley		Gold	United States
29405	4X200M Freestyle		Gold	United States

[39]: *# Show Top 10 Athletes Men And Top 5 Women Athlete:*

```
summer[summer['Gender']=='Men']['Athlete'].value_counts()[:].index[0:10]
```

[39]: Index(['Michael Phelps', 'Nikolay Andrianov', 'Boris Shakhlin', 'Takashi Ono',
'Edoardo Mangiarotti', 'Paavo Nurmi', 'Alexei Nemov', 'Sawao Kato',
'Matthew Biondi', 'Viktor Ivanovich Chukarin'],
dtype='object', name='Athlete')

[40]: `summer[summer['Gender']=='Women']['Athlete'].value_counts()[:].index[0:5]`

```
[40]: Index(['Larisa Latynina', 'Natalie Coughlin', 'Birgit Fischer',
        'Jenny Thompson', 'Dara Torres'],
        dtype='object', name='Athlete')
```

```
[41]: # Print Highest decorated Athlete Men:

print('The Highest Decorated Male Athlete is:␣
↪',summer[summer['Gender']=='Men']['Athlete'].value_counts()[:1].index[0],
      'with: ',summer[summer['Gender']=='Men']['Athlete'].value_counts()[:1].
↪values[0], ' medals')
```

The Highest Decorated Male Athlete is: Michael Phelps with: 22 medals

```
[42]: # Print Highest decorated Athlete Female:

print('The Highest Decorated Female Athlete is:␣
↪',summer[summer['Gender']=='Women']['Athlete'].value_counts()[:1].index[0],
      'with: ',summer[summer['Gender']=='Women']['Athlete'].value_counts()[:1].
↪values[0], ' medals')
```

The Highest Decorated Female Athlete is: Larisa Latynina with: 18 medals

```
[43]: #Groupby Athletes by Medals:

medals=summer.groupby(['Athlete','Medal'])['Sport'].count().reset_index().
↪sort_values(by='Sport',ascending=False)
medals.head(10)
```

```
[43]:
```

	Athlete	Medal	Sport
17344	Michael Phelps	Gold	18
3517	Carl Lewis	Gold	9
16581	Mark Spitz	Gold	9
19224	Paavo Nurmi	Gold	9
14731	Larisa Latynina	Gold	9
2967	Birgit Fischer	Gold	8
20439	Ray Ewry	Gold	8
16902	Matthew Biondi	Gold	8
21908	Sawao Kato	Gold	8
12071	Jenny Thompson	Gold	8

```
[44]: medals.info()

<class 'pandas.core.frame.DataFrame'>
Index: 26723 entries, 17344 to 13361
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Athlete     26723 non-null  object
1   Medal       26723 non-null  object
```

```

2    Sport    26723 non-null  int64
dtypes: int64(1), object(2)
memory usage: 835.1+ KB

```

[45]: *# Creating Pivot Tables for medals:*

```

medals_ath=medals.pivot(index='Athlete', columns='Medal', values='Sport').
    ↪fillna(0)
medals_ath

```

```

[45]: Medal          Bronze  Gold  Silver
Athlete
- Johnson           1.0   1.0    0.0
... Daumain         1.0   0.0    0.0
A Lam Shin          0.0   0.0    1.0
A. Albert            0.0   1.0    0.0
A. B. Zumelzu        0.0   0.0    1.0
...
Üdön Tersztyanszky   1.0   1.0    1.0
Üdön Tery            0.0   0.0    1.0
Üdön Zombori         0.0   1.0    1.0
Üistein Schirmer     0.0   1.0    0.0
Üsten Üstensen       2.0   0.0    3.0

```

[22759 rows x 3 columns]

```

[46]: medals_ath['Bronze']=medals_ath['Bronze'].astype(int)
medals_ath['Gold']=medals_ath['Gold'].astype(int)
medals_ath['Silver']=medals_ath['Silver'].astype(int)

```

[47]: medals_ath.info()

```

<class 'pandas.core.frame.DataFrame'>
Index: 22759 entries, - Johnson to Üsten Üstensen
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Bronze  22759 non-null    int32
1   Gold    22759 non-null    int32
2   Silver  22759 non-null    int32
dtypes: int32(3)
memory usage: 444.5+ KB

```

[48]: medals_ath.sort_values(by='Gold',ascending=False)

```

[48]: Medal          Bronze  Gold  Silver
Athlete
Michael Phelps           2   18     2

```

Larisa Latynina	4	9	5
Mark Spitz	1	9	1
Paavo Nurmi	0	9	3
Carl Lewis	0	9	1
...
Darius Maskoliunas	1	0	0
Darius Lukminas	1	0	0
Darius Dimavicius	1	0	0
Kim Turner	1	0	0
Østen Østensen	2	0	3

[22759 rows x 3 columns]

```
[49]: medals=summer.groupby(['Athlete','Medal'])['Sport'].count().reset_index().
      ↪sort_values(by=['Sport','Athlete'],ascending=False)
medals.columns=['Athlete','Medal','Count']
medals
```

```
[49]:
```

	Athlete	Medal	Count
17344	Michael Phelps	Gold	18
19224	Paavo Nurmi	Gold	9
16581	Mark Spitz	Gold	9
14731	Larisa Latynina	Gold	9
3517	Carl Lewis	Gold	9
...
4	A. Albert	Gold	1
3	A Lam Shin	Silver	1
2	... Daumain	Bronze	1
0	- Johnson	Bronze	1
1	- Johnson	Gold	1

[26723 rows x 3 columns]

0.0.6 Medals By Country

```
[50]: # Groupby Medals by country:

medals_country=summer.groupby(['Country','Medal'])['Athlete'].count().
      ↪reset_index().sort_values(by='Athlete',ascending=False)
medals_country
```

```
[50]:
```

	Country	Medal	Athlete
283	United States	Gold	2235
284	United States	Silver	1252
282	United States	Bronze	1098
281	United Kingdom	Silver	621
279	United Kingdom	Bronze	553

```

..
264  Trinidad and Tobago    Gold    1
75      Ecuador    Gold    1
166      Luxembourg    Gold    1
132      Israel    Gold    1
72    Dominican Republic  Bronze    1

```

[301 rows x 3 columns]

```
[51]: medals_country.isnull().sum()
```

```

[51]: Country    0
      Medal      0
      Athlete    0
      dtype: int64

```

```
[52]: # Create Pivot Tables and fill null values:
```

```

medals_country=medals_country.pivot(index='Country', columns='Medal',
    ↪values='Athlete').fillna(0)
medals_country

```

```

[52]: Medal      Bronze    Gold    Silver
      Country
Afghanistan      2.0     0.0     0.0
Algeria          8.0     5.0     2.0
Argentina       91.0    69.0    99.0
Armenia          8.0     1.0     2.0
Australia      472.0   312.0   405.0
...
Venezuela       8.0     2.0     2.0
Vietnam         0.0     0.0     2.0
Virgin Islands* 0.0     0.0     1.0
Zambia          1.0     0.0     1.0
Zimbabwe        1.0    18.0     4.0

```

[129 rows x 3 columns]

```
[53]: #Top 10 countries with highest number of medals
```

```

top=medals_country.sort_values(by='Gold',ascending=False)[:10]
top['Bronze']=top['Bronze'].astype(int)
top['Gold']=top['Gold'].astype(int)
top['Silver']=top['Silver'].astype(int)
top

```



```
[53]: Medal          Bronze  Gold  Silver
      Country
United States      1098  2235   1252
United Kingdom      553   546    621
Italy               404   476    416
Germany            475   452    378
Hungary            351   412    316
France             497   408    491
Sweden            328   349    367
Australia          472   312    405
China              221   290    296
Russia            291   239    238
```

```
[54]: # Groupby Countries by highest medal and gender

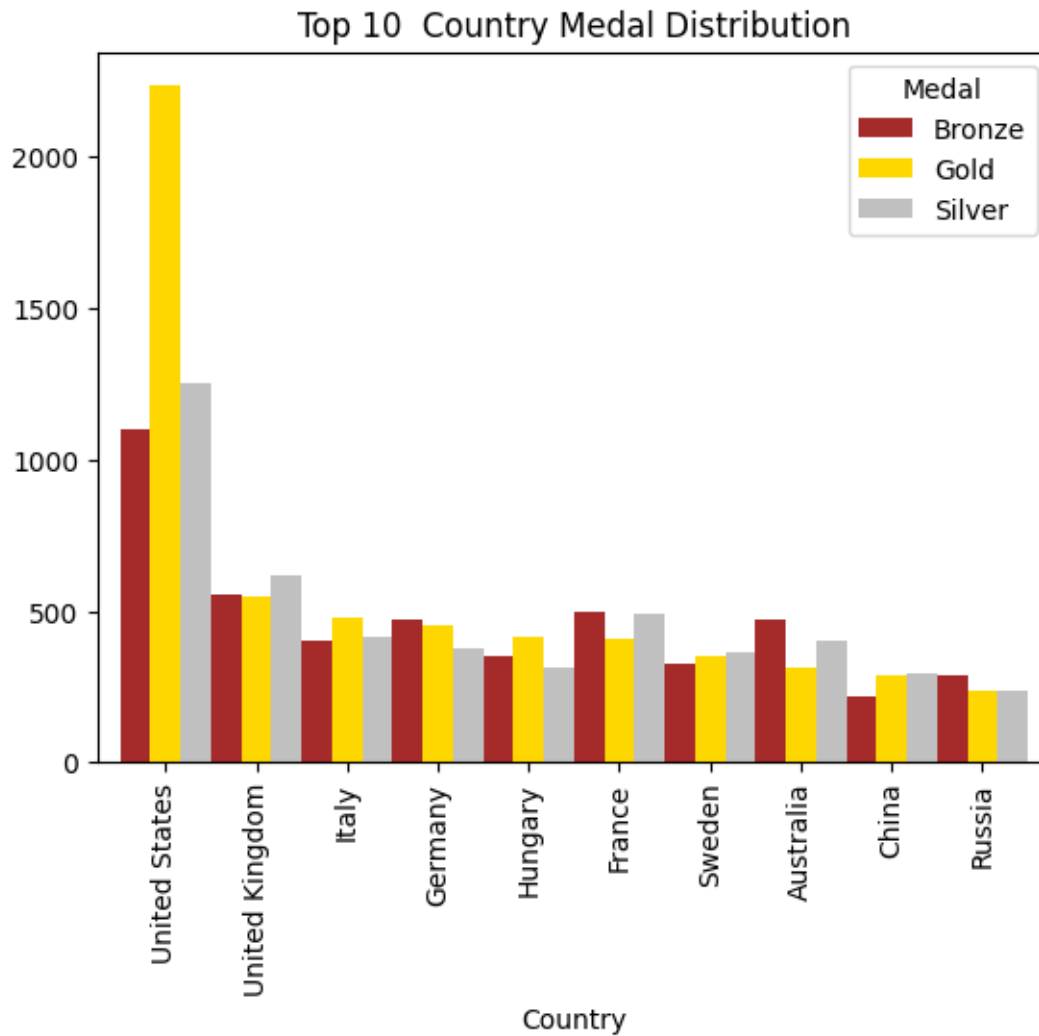
medals_country=summer.groupby(['Country', 'Gender', 'Medal'])['Athlete'].count().
    ↪reset_index().sort_values(by='Athlete',ascending=False)
medals_country
```

```
[54]:          Country Gender  Medal  Athlete
457      United States   Men   Gold    1562
458      United States   Men  Silver     861
456      United States   Men  Bronze     785
460      United States  Women   Gold     673
452      United Kingdom   Men  Silver     512
..          ...      ...      ...      ...
126          Ecuador    Men   Gold         1
127          Ecuador    Men  Silver         1
131             Egypt  Women  Silver         1
132          Eritrea    Men  Bronze         1
306  Netherlands Antilles*  Men  Silver         1
```

[481 rows x 4 columns]

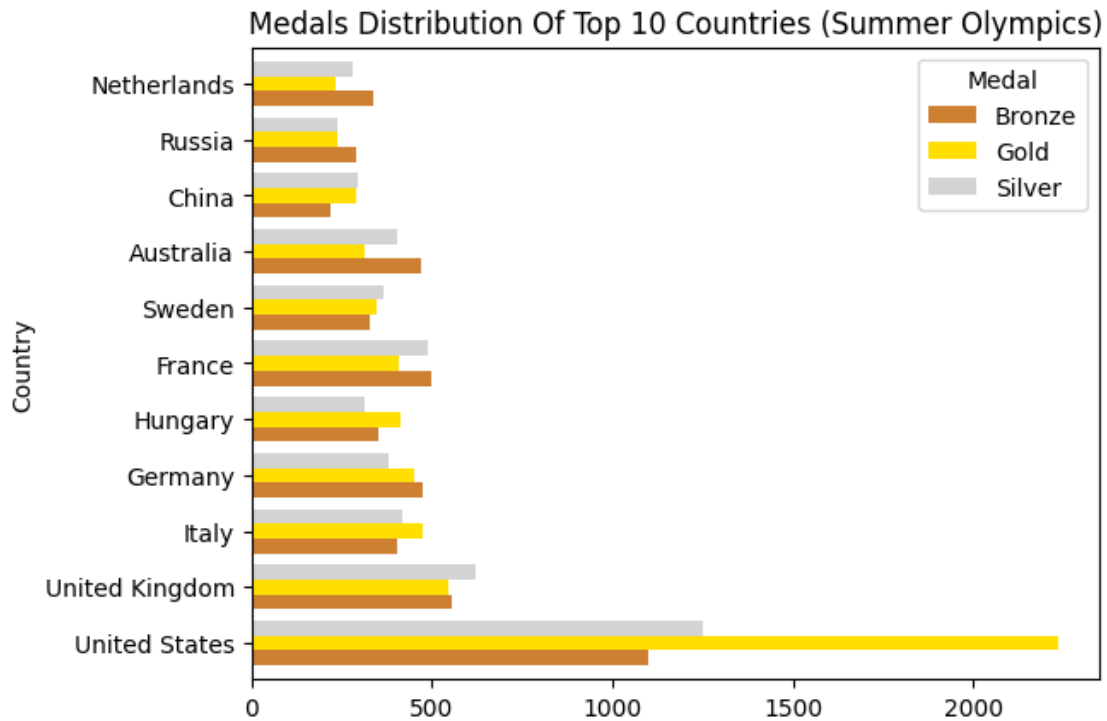
```
[55]: #Plot Graph for Top 10 medal Distribution:

top.plot.bar(width=1,color=['brown','gold','silver'])
plt.title("Top 10 Country Medal Distribution")
plt.show()
```



```
[56]: # Medal Distribution of Top Country:

medals_country=summer.groupby(['Country','Medal'])['Gender'].count().
    ↪reset_index().sort_values(by='Gender',ascending=False)
medals_country=medals_country.pivot(index='Country', columns='Medal', values_
    ↪='Gender').fillna(0)
top=medals_country.sort_values(by='Gold',ascending=False)[:11]
top.plot.barh(width=0.8,color=['#CD7F32','#FFDF00','#D3D3D3'])
plt.title('Medals Distribution Of Top 10 Countries (Summer Olympics)')
plt.show()
```

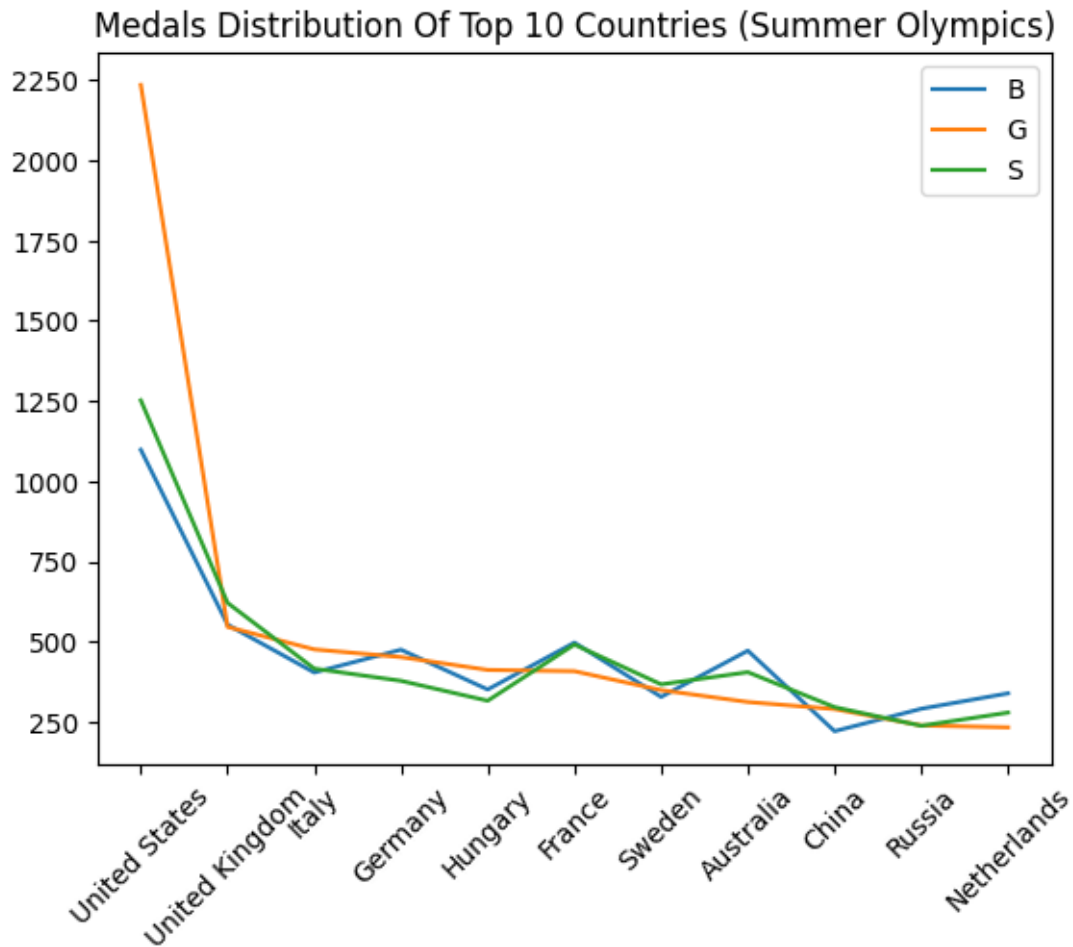


```
[57]: top.index.values
```

```
[57]: array(['United States', 'United Kingdom', 'Italy', 'Germany', 'Hungary',
        'France', 'Sweden', 'Australia', 'China', 'Russia', 'Netherlands'],
        dtype=object)
```

```
[58]: # Medal Distribution of top 10 countries:
```

```
medal=['B','G','S']
plt.plot(top.index.values,top.iloc[:,0:3].values)
plt.xticks(rotation=45)
plt.title('Medals Distribution Of Top 10 Countries (Summer Olympics)')
plt.legend(medal)
plt.show()
```



```
[59]: men=summer[summer['Gender']=='Men']
men=men.groupby(['Athlete','Medal'])['Code'].count().reset_index().
      ↪sort_values(by='Code',ascending=False).fillna(0)
men
```

```
[59]:
```

	Athlete	Medal	Code
12806	Michael Phelps	Gold	18
2675	Carl Lewis	Gold	9
14059	Paavo Nurmi	Gold	9
12271	Mark Spitz	Gold	9
12480	Matthew Biondi	Gold	8
...
7014	Gösta Magnusson	Bronze	1
7013	Gösta Löfgren	Bronze	1
9869	Jonathan Nicholas M. Potter	Gold	1
14380	Pending	Silver	0
2249	Besik Kudukhov	Silver	0

[19739 rows x 3 columns]

```
[60]: men_medals=men.pivot(index='Athlete',columns='Medal', values='Code').fillna(0)
men_medals
```

```
[60]: Medal          Bronze  Gold  Silver
Athlete
- Johnson          1.0    1.0    0.0
... Daumain        1.0    0.0    0.0
A. Albert          0.0    1.0    0.0
A. B. Zumelzu      0.0    0.0    1.0
A. Bögli           0.0    0.0    1.0
...
Ödön Tersztyanszky 1.0    1.0    1.0
Ödön Tery          0.0    0.0    1.0
Ödön Zombori       0.0    1.0    1.0
Öistein Schirmer   0.0    1.0    0.0
Östen Östenssen    2.0    0.0    3.0
```

[16927 rows x 3 columns]

```
[61]: top=men_medals.sort_values(by='Gold',ascending=False)
top=top.head(10)
top
```

```
[61]: Medal          Bronze  Gold  Silver
Athlete
Michael Phelps          2.0   18.0    2.0
Mark Spitz              1.0    9.0    1.0
Paavo Nurmi             0.0    9.0    3.0
Carl Lewis              0.0    9.0    1.0
Sawao Kato              1.0    8.0    3.0
Ray Ewry                0.0    8.0    0.0
Matthew Biondi          1.0    8.0    2.0
Viktor Ivanovich Chukarin 1.0    7.0    3.0
Aladar Gerevich         2.0    7.0    1.0
Nikolay Andrianov       3.0    7.0    5.0
```

```
[62]: #creating a new dataframe for men:

men=summer[summer['Gender']=='Men']
men=men.groupby(['Athlete','Medal'])['Code'].count().reset_index().
    ↪sort_values(by='Code',ascending=False).fillna(0)
men=men[men['Athlete'].isin(summer['Athlete'].value_counts().index[:10])]
men
men=men.pivot(index='Athlete',columns='Medal',values='Code')
```

```
men=men.sort_values(by=['Gold','Silver','Bronze'],ascending=False)
men
```

```
[62]: Medal           Bronze  Gold  Silver
Athlete
Michael Phelps         2    18      2
Sawao Kato              1     8      3
Nikolay Andrianov      3     7      5
Boris Shakhlin         2     7      4
Edoardo Mangiarotti    2     6      5
Takashi Ono            4     5      4
```

```
[63]: men=summer[summer['Gender']=='Men']
men=men.groupby(['Athlete','Medal'])['Code'].count().reset_index().
      ↪sort_values(by='Code',ascending=False)
men
men.columns=['Athlete','Medal','No']
men
```

```
[63]:           Athlete  Medal  No
12806   Michael Phelps   Gold  18
2675      Carl Lewis    Gold   9
14059   Paavo Nurmi     Gold   9
12271    Mark Spitz     Gold   9
12480   Matthew Biondi   Gold   8
...
7014      Gösta Magnusson  Bronze  1
7013      Gösta Löfgren   Bronze  1
9869  Jonathan Nicholas M. Potter   Gold  1
14380           Pending   Silver  0
2249      Besik Kudukhov   Silver  0
```

[19739 rows x 3 columns]

```
[64]: summer['Athlete'].value_counts()
```

```
[64]: Athlete
Michael Phelps      22
Larisa Latynina     18
Nikolay Andrianov   15
Takashi Ono         13
Edoardo Mangiarotti 13
..
Uddin Zaka          1
Hayat Zafar         1
Rashid Muhammad     1
Muhammad Afzal Manna 1
```

```
Jimmy Lidberg          1
Name: count, Length: 22759, dtype: int64
```

```
[65]: men=men[men['Athlete'].isin(summer['Athlete'].value_counts().index[:11])]
men
```

```
[65]:
```

	Athlete	Medal	No
12806	Michael Phelps	Gold	18
16242	Sawao Kato	Gold	8
2385	Boris Shakhlin	Gold	7
13564	Nikolay Andrianov	Gold	7
4386	Edoardo Mangiarotti	Gold	6
773	Alexei Nemov	Bronze	6
13565	Nikolay Andrianov	Silver	5
4387	Edoardo Mangiarotti	Silver	5
17176	Takashi Ono	Gold	5
774	Alexei Nemov	Gold	4
17177	Takashi Ono	Silver	4
17175	Takashi Ono	Bronze	4
2386	Boris Shakhlin	Silver	4
16243	Sawao Kato	Silver	3
13563	Nikolay Andrianov	Bronze	3
4385	Edoardo Mangiarotti	Bronze	2
2384	Boris Shakhlin	Bronze	2
12807	Michael Phelps	Silver	2
12805	Michael Phelps	Bronze	2
775	Alexei Nemov	Silver	2
16241	Sawao Kato	Bronze	1

```
[66]: men['Athlete'].unique()
```

```
[66]: array(['Michael Phelps', 'Sawao Kato', 'Boris Shakhlin',
        'Nikolay Andrianov', 'Edoardo Mangiarotti', 'Alexei Nemov',
        'Takashi Ono'], dtype=object)
```

```
[67]: men[men['Medal'].isin(['Gold', 'Silver', 'Bronze'])]
```

```
[67]:
```

	Athlete	Medal	No
12806	Michael Phelps	Gold	18
16242	Sawao Kato	Gold	8
2385	Boris Shakhlin	Gold	7
13564	Nikolay Andrianov	Gold	7
4386	Edoardo Mangiarotti	Gold	6
773	Alexei Nemov	Bronze	6
13565	Nikolay Andrianov	Silver	5
4387	Edoardo Mangiarotti	Silver	5
17176	Takashi Ono	Gold	5

774	Alexei Nemov	Gold	4
17177	Takashi Ono	Silver	4
17175	Takashi Ono	Bronze	4
2386	Boris Shakhlin	Silver	4
16243	Sawao Kato	Silver	3
13563	Nikolay Andrianov	Bronze	3
4385	Edoardo Mangiarotti	Bronze	2
2384	Boris Shakhlin	Bronze	2
12807	Michael Phelps	Silver	2
12805	Michael Phelps	Bronze	2
775	Alexei Nemov	Silver	2
16241	Sawao Kato	Bronze	1

```
[68]: men=men.pivot(index='Athlete', columns='Medal', values='No').fillna(0)
men
```

```
[68]: Medal          Bronze  Gold  Silver
Athlete
Alexei Nemov          6     4      2
Boris Shakhlin         2     7      4
Edoardo Mangiarotti    2     6      5
Michael Phelps         2    18      2
Nikolay Andrianov      3     7      5
Sawao Kato              1     8      3
Takashi Ono            4     5      4
```

```
[69]: # Create Dataframe for records of only men:

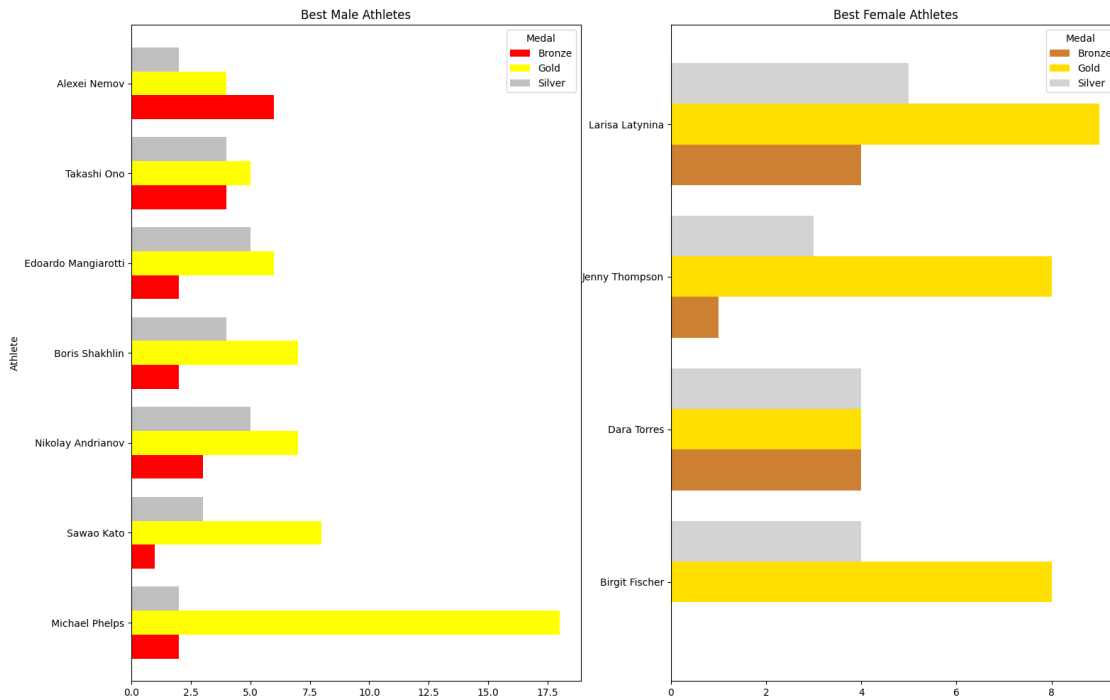
fig,ax=plt.subplots(1,2,figsize=(18,12))
men=summer[summer['Gender']=='Men']
#getting only male atheletes:
men=men.groupby(['Athlete','Medal'])['Code'].count().reset_index().
    ↪sort_values(by='Code',ascending=False)
men=men[men['Athlete'].isin(summer['Athlete'].value_counts().index[:11])]
men
men=men.pivot(index='Athlete',columns='Medal',values='Code')
men=men.sort_values(by=['Gold','Silver','Bronze'],ascending=False)
men.plot.barh(width=0.8,color=['red','yellow','silver'],ax=ax[0])
ax[0].set_title('Best Male Athletes')
ax[0].set_ylabel('Athlete')

# create dataframe for records of only women:

women=summer[summer['Gender']=='Women']
women=women.groupby(['Athlete','Medal'])['Code'].count().reset_index().
    ↪sort_values(by='Code',ascending=False)
#getting only female atheletes:
```



```
women=women[women['Athlete'].isin(summer['Athlete'].value_counts().index[:11])]
women=women.pivot(index='Athlete',columns='Medal', values='Code')
women.plot.barh(width=0.8,color=['#CD7F32','#FFDF00','#D3D3D3'],ax=ax[1])
ax[1].set_title('Best Female Athletes')
ax[1].set_ylabel('')
plt.show()
```



```
[70]: summer['Discipline'].unique()
```

```
[70]: array(['Swimming', 'Athletics', 'Cycling Road', 'Cycling Track',
        'Fencing', 'Artistic G.', 'Shooting', 'Tennis', 'Weightlifting',
        'Wrestling Gre-R', 'Water polo', 'Archery', 'Basque Pelota',
        'Cricket', 'Croquet', 'Jumping', 'Football', 'Golf', 'Polo',
        'Rowing', 'Rugby', 'Sailing', 'Tug of War', 'Diving', 'Boxing',
        'Lacrosse', 'Roque', 'Wrestling Free.', 'Hockey', 'Jeu de Paume',
        'Rackets', 'Figure skating', 'Water Motorspor', 'Dressage',
        'Eventing', 'Modern Pentath.', 'Vaulting', 'Ice Hockey',
        'Basketball', 'Canoe / Kayak F', 'Handball', 'Judo', 'Volleyball',
        'Canoe / Kayak S', 'Synchronized S.', 'Rhythmic G.',
        'Table Tennis', 'Badminton', 'Baseball', 'Mountain Bike',
        'Softball', 'Beach volley.', 'Trampoline', 'Taekwondo',
        'Triathlon', 'BMX', 'Marathon swimming', 'Synchronized Swimming',
        'Water Polo', 'Canoe Slalom', 'Canoe Sprint', 'Cycling BMX',
        'Gymnastics Artistic', 'Gymnastics Rhythmic', 'Modern Pentathlon',
```

```
'Beach Volleyball', 'Wrestling Freestyle'], dtype=object)
```

```
[71]: len(summer['Discipline'].unique())
```

```
[71]: 67
```

```
[72]: summer[(summer['Country']=='Hungary') | (summer['Country']=='Greece') |
↳(summer['Country']=='United States')]
```

```
[72]:
```

	Year	City	Sport	Discipline	Athlete \
0	1896	Athens	Aquatics	Swimming	Alfred Hajos
2	1896	Athens	Aquatics	Swimming	Dimitrios Drivas
3	1896	Athens	Aquatics	Swimming	Ioannis Malokinis
4	1896	Athens	Aquatics	Swimming	Spiridon Chasapis
5	1896	Athens	Aquatics	Swimming	Efstathios Chorophas
...
31125	2012	London	Wrestling	Wrestling Freestyle	Jordan Ernest Burroughs
31127	2012	London	Wrestling	Wrestling Freestyle	Gabor Hatos
31133	2012	London	Wrestling	Wrestling Freestyle	Jacob Stephen Varner
31143	2012	London	Wrestling	Wrestling Freestyle	Peter Modos
31150	2012	London	Wrestling	Wrestling Freestyle	Tamas Lorincz
...
	Code	Gender	Event	Medal	Country
0	HUN	Men	100M Freestyle	Gold	Hungary
2	GRE	Men	100M Freestyle For Sailors	Bronze	Greece
3	GRE	Men	100M Freestyle For Sailors	Gold	Greece
4	GRE	Men	100M Freestyle For Sailors	Silver	Greece
5	GRE	Men	1200M Freestyle	Bronze	Greece
...
31125	USA	Men	Wf 74 KG	Gold	United States
31127	HUN	Men	Wf 74 KG	Bronze	Hungary
31133	USA	Men	Wf 96 KG	Gold	United States
31143	HUN	Men	Wg 55 KG	Bronze	Hungary
31150	HUN	Men	Wg 66 KG	Silver	Hungary

```
[5812 rows x 10 columns]
```

```
[73]: summer[summer['Country'].isin(summer['Country'].value_counts().index)]
```

```
[73]:
```

	Year	City	Sport	Discipline	Athlete \
0	1896	Athens	Aquatics	Swimming	Alfred Hajos
1	1896	Athens	Aquatics	Swimming	Otto Herschmann
2	1896	Athens	Aquatics	Swimming	Dimitrios Drivas
3	1896	Athens	Aquatics	Swimming	Ioannis Malokinis
4	1896	Athens	Aquatics	Swimming	Spiridon Chasapis
...
31160	2012	London	Wrestling	Wrestling Freestyle	Damian Janikowski

31161	2012	London	Wrestling	Wrestling Freestyle	Ghasem Gholamreza Rezaei
31162	2012	London	Wrestling	Wrestling Freestyle	Rustam Totrov
31163	2012	London	Wrestling	Wrestling Freestyle	Artur Aleksanyan
31164	2012	London	Wrestling	Wrestling Freestyle	Jimmy Lidberg

	Code	Gender		Event	Medal	Country
0	HUN	Men		100M Freestyle	Gold	Hungary
1	AUT	Men		100M Freestyle	Silver	Austria
2	GRE	Men	100M Freestyle For Sailors		Bronze	Greece
3	GRE	Men	100M Freestyle For Sailors		Gold	Greece
4	GRE	Men	100M Freestyle For Sailors		Silver	Greece
...
31160	POL	Men		Wg 84 KG	Bronze	Poland
31161	IRI	Men		Wg 96 KG	Gold	Iran
31162	RUS	Men		Wg 96 KG	Silver	Russia
31163	ARM	Men		Wg 96 KG	Bronze	Armenia
31164	SWE	Men		Wg 96 KG	Bronze	Sweden

[25742 rows x 10 columns]

```
[74]: summer[summer['Discipline']=='Wrestling']
```

```
[74]: Empty DataFrame
Columns: [Year, City, Sport, Discipline, Athlete, Code, Gender, Event, Medal,
Country]
Index: []
```

```
[75]: summer.loc[summer['Discipline'].str.
↳contains('Wrestling'),'Discipline']=='Wrestling'
summer
```

```
[75]:
```

	Year	City	Sport	Discipline	Athlete	Code	\
0	1896	Athens	Aquatics	Swimming	Alfred Hajos	HUN	
1	1896	Athens	Aquatics	Swimming	Otto Herschmann	AUT	
2	1896	Athens	Aquatics	Swimming	Dimitrios Drivas	GRE	
3	1896	Athens	Aquatics	Swimming	Ioannis Malokinis	GRE	
4	1896	Athens	Aquatics	Swimming	Spiridon Chasapis	GRE	
...	
31160	2012	London	Wrestling	Wrestling	Damian Janikowski	POL	
31161	2012	London	Wrestling	Wrestling	Ghasem Gholamreza Rezaei	IRI	
31162	2012	London	Wrestling	Wrestling	Rustam Totrov	RUS	
31163	2012	London	Wrestling	Wrestling	Artur Aleksanyan	ARM	
31164	2012	London	Wrestling	Wrestling	Jimmy Lidberg	SWE	

	Gender		Event	Medal	Country
0	Men		100M Freestyle	Gold	Hungary
1	Men		100M Freestyle	Silver	Austria

2	Men	100M Freestyle For Sailors	Bronze	Greece
3	Men	100M Freestyle For Sailors	Gold	Greece
4	Men	100M Freestyle For Sailors	Silver	Greece
...
31160	Men	Wg 84 KG	Bronze	Poland
31161	Men	Wg 96 KG	Gold	Iran
31162	Men	Wg 96 KG	Silver	Russia
31163	Men	Wg 96 KG	Bronze	Armenia
31164	Men	Wg 96 KG	Bronze	Sweden

[31165 rows x 10 columns]

```
[76]: # Create Tables for Country and Discipline:

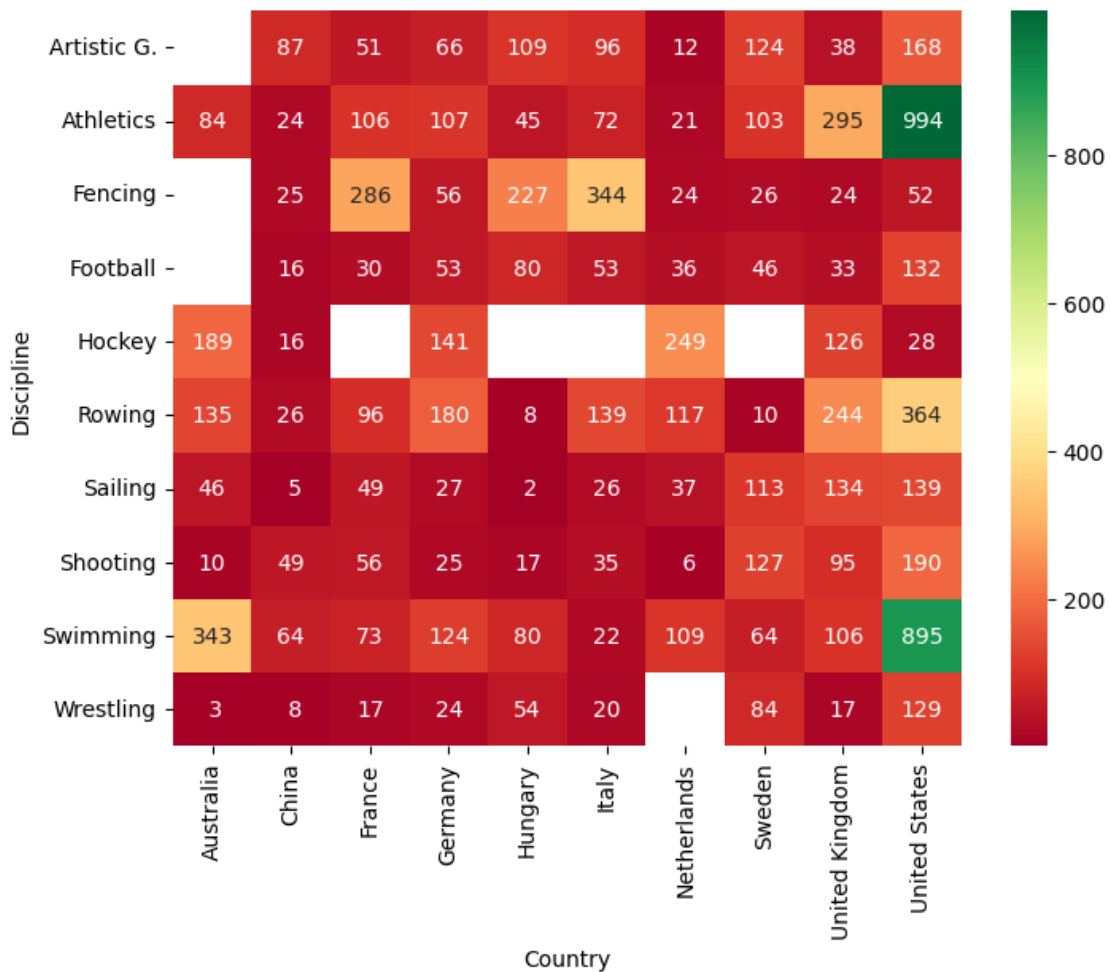
test=summer[summer['Country'].isin(summer['Country'].value_counts()[ :10].index)]
test=test[test['Discipline'].isin(summer['Discipline'].value_counts()[ :10].
    ↪index)]
test=test.groupby(['Country','Discipline'])['Sport'].count().reset_index()
test
test=test.pivot(index='Discipline',columns='Country', values='Sport').fillna(0)
test
```

```
[76]: Country      Australia  China  France  Germany  Hungary  Italy  Netherlands \
Discipline
Artistic G.         0.0    87.0    51.0     66.0    109.0    96.0             12.0
Athletics           84.0    24.0   106.0    107.0     45.0    72.0             21.0
Fencing             0.0    25.0   286.0     56.0    227.0   344.0             24.0
Football            0.0    16.0    30.0     53.0     80.0    53.0             36.0
Hockey             189.0    16.0     0.0    141.0      0.0     0.0            249.0
Rowing              135.0    26.0    96.0    180.0      8.0   139.0            117.0
Sailing             46.0     5.0    49.0     27.0      2.0    26.0             37.0
Shooting            10.0    49.0    56.0     25.0    17.0    35.0              6.0
Swimming            343.0    64.0    73.0    124.0    80.0    22.0            109.0
Wrestling           3.0     8.0    17.0     24.0    54.0    20.0              0.0
```

Country	Sweden	United Kingdom	United States
Discipline			
Artistic G.	124.0	38.0	168.0
Athletics	103.0	295.0	994.0
Fencing	26.0	24.0	52.0
Football	46.0	33.0	132.0
Hockey	0.0	126.0	28.0
Rowing	10.0	244.0	364.0
Sailing	113.0	134.0	139.0
Shooting	127.0	95.0	190.0
Swimming	64.0	106.0	895.0
Wrestling	84.0	17.0	129.0

```
[77]: # Plot Heat Map for country and discipline:
# Correlation between Country , discipline and Number of Athletes playing for
# that discipline:

test=summer[summer['Country'].isin(summer['Country'].value_counts()[ :10].index)]
test=test[test['Discipline'].isin(summer['Discipline'].value_counts()[ :10].
    ↳index)]
test=test.groupby(['Country','Discipline'])['Sport'].count().reset_index()
test=test.pivot(index='Discipline', columns='Country', values='Sport')
sns.heatmap(test,cmap='RdYlGn',annot=True,fmt='2.0f')
fig=plt.gcf()
fig.set_size_inches(8,6)
plt.show()
```



```
[78]: summer[(summer['Country']=='United States') & (summer['Year']==2012)]
```

```
[78]:
```

	Year	City	Sport	Discipline	Athlete \
29216	2012	London	Aquatics	Diving	David Boudia
29232	2012	London	Aquatics	Diving	David Boudia
29233	2012	London	Aquatics	Diving	Nicholas Mccrory
29244	2012	London	Aquatics	Diving	Troy Dumais
29245	2012	London	Aquatics	Diving	Kristian Ipsen
...
31035	2012	London	Volleyball	Volleyball	Logan Tom
31099	2012	London	Wrestling	Wrestling	Clarissa Kyoko Mei Ling Chun
31112	2012	London	Wrestling	Wrestling	Coleman Scott
31125	2012	London	Wrestling	Wrestling	Jordan Ernest Burroughs
31133	2012	London	Wrestling	Wrestling	Jacob Stephen Varner

	Code	Gender	Event	Medal	Country
29216	USA	Men	10M Platform	Gold	United States
29232	USA	Men	Synchronized 10M	Bronze	United States
29233	USA	Men	Synchronized 10M	Bronze	United States
29244	USA	Men	Synchronized 3M	Bronze	United States
29245	USA	Men	Synchronized 3M	Bronze	United States
...
31035	USA	Women	Volleyball	Silver	United States
31099	USA	Women	Wf 48 KG	Bronze	United States
31112	USA	Men	Wf 60 KG	Bronze	United States
31125	USA	Men	Wf 74 KG	Gold	United States
31133	USA	Men	Wf 96 KG	Gold	United States

[250 rows x 10 columns]

```
[79]: # Groupby Medals by year by Country

test1=summer.groupby(['Country','Year'])['Medal'].count().reset_index().
    ↪sort_values(by='Medal',ascending=False).head(10)
test1.head(10)
```

```
[79]:
```

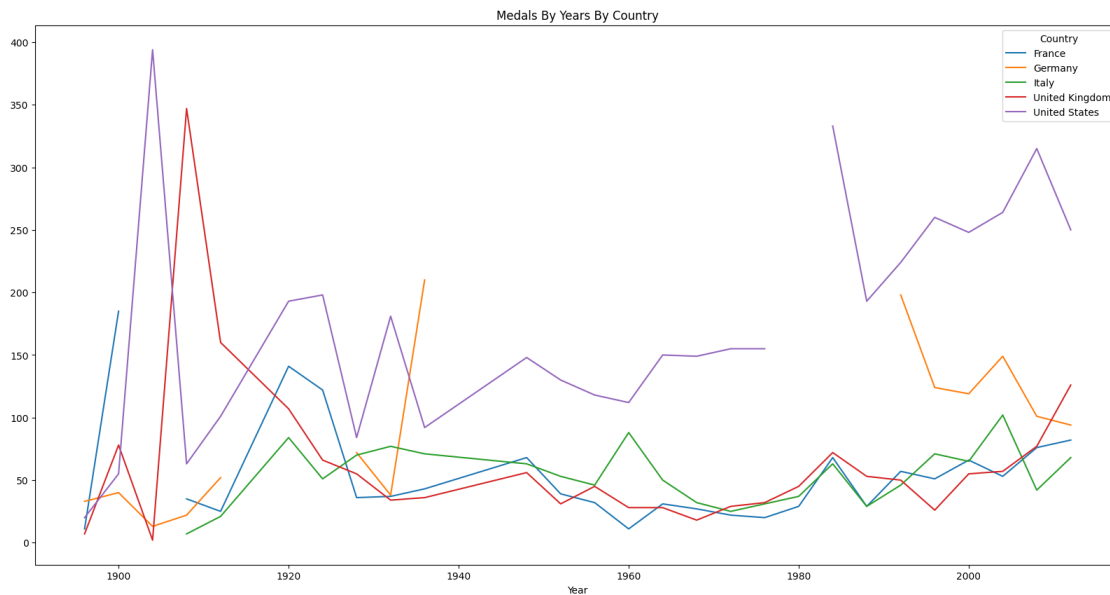
	Country	Year	Medal
1014	United States	1904	394
988	United Kingdom	1908	347
1030	United States	1984	333
1036	United States	2008	315
1035	United States	2004	264
1033	United States	1996	260
1037	United States	2012	250
1034	United States	2000	248
1032	United States	1992	224
368	Germany	1936	210

```
[80]: test1=test1.pivot(index='Year', columns='Country', values='Medal')
test1.head(50)
```

```
[80]: Country  Germany  United Kingdom  United States
Year
1904         NaN          NaN          394.0
1908         NaN        347.0          NaN
1936        210.0          NaN          NaN
1984         NaN          NaN        333.0
1992         NaN          NaN        224.0
1996         NaN          NaN        260.0
2000         NaN          NaN        248.0
2004         NaN          NaN        264.0
2008         NaN          NaN        315.0
2012         NaN          NaN        250.0
```

```
[81]: #Plot Line chart for medals by years by country

test1=summer.groupby(['Country','Year'])['Medal'].count().reset_index()
test1=test1[test1['Country'].isin(summer['Country'].value_counts()[:5].index)]
test1=test1.pivot(index='Year',columns='Country',values='Medal')
test1.plot()
fig=plt.gcf()
fig.set_size_inches(20,10)
plt.title('Medals By Years By Country')
plt.show()
```



1 Winter Game Analysis:

```
[82]: # Check Winter File Data:
```

```
winter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5770 entries, 0 to 5769
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year             5770 non-null   int64
1   City             5770 non-null   object
2   Sport            5770 non-null   object
3   Discipline       5770 non-null   object
4   Athlete          5770 non-null   object
5   Country          5770 non-null   object
6   Gender           5770 non-null   object
7   Event            5770 non-null   object
8   Medal            5770 non-null   object
dtypes: int64(1), object(8)
memory usage: 405.8+ KB
```

```
[83]: winter.describe()
```

```
[83]:          Year
count  5770.000000
mean   1986.350433
std     24.263673
min    1924.000000
25%    1972.000000
50%    1994.000000
75%    2006.000000
max    2014.000000
```

```
[84]: #Print Total Rows:
```

```
print("Rows,cols",winter.shape)
print("Total no of Rows=",len(winter))
print("Total no of Rows=",winter.shape[0])
```

```
Rows,cols (5770, 9)
Total no of Rows= 5770
Total no of Rows= 5770
```

```
[85]: # Print Total Columns:
```

```
print("Total no of Cols",winter.shape[1])
```



```
print("Total no of Cols",len(winter.columns))
```

Total no of Cols 9

Total no of Cols 9

```
[86]: winter['Sport'].unique()
```

```
[86]: array(['Biathlon', 'Bobsleigh', 'Curling', 'Ice Hockey', 'Skating',  
        'Skiing', 'Luge'], dtype=object)
```

```
[87]: winter['Medal'].unique()
```

```
[87]: array(['Bronze', 'Gold', 'Silver'], dtype=object)
```

```
[88]: winter['Gender'].unique()
```

```
[88]: array(['Men', 'Women'], dtype=object)
```

```
[89]: winter.head()
```

```
[89]:
```

	Year	City	Sport	Discipline	Athlete	Country	Gender	\
0	1924	Chamonix	Biathlon	Biathlon	BERTHET, G.	FRA	Men	
1	1924	Chamonix	Biathlon	Biathlon	MANDRILLON, C.	FRA	Men	
2	1924	Chamonix	Biathlon	Biathlon	MANDRILLON, Maurice	FRA	Men	
3	1924	Chamonix	Biathlon	Biathlon	VANDELLE, André	FRA	Men	
4	1924	Chamonix	Biathlon	Biathlon	AUFDENBLATTEN, Adolf	SUI	Men	

	Event	Medal
0	Military Patrol	Bronze
1	Military Patrol	Bronze
2	Military Patrol	Bronze
3	Military Patrol	Bronze
4	Military Patrol	Gold

```
[90]: # Print highest Decorated Male and Female Athlete
```

```
print('The Highest Decorated Male Athlete is:␣  
↪',winter[winter['Gender']=='Men']['Athlete'].value_counts()[:1].  
↪index[0], 'with: ',winter[winter['Gender']=='Men']['Athlete'].value_counts()[:  
↪1].values[0], ' medals')  
print('The Highest Decorated Female Athlete is:␣  
↪',winter[winter['Gender']=='Women']['Athlete'].value_counts()[:1].  
↪index[0], 'with: ',winter[winter['Gender']=='Women']['Athlete'].  
↪value_counts()[:1].values[0], ' medals')
```

The Highest Decorated Male Athlete is: BJOERNDALEN, Ole Einar with: 13 medals

The Highest Decorated Female Athlete is: SMETANINA, Raisa with: 10 medals

```
[91]: # Merge and Manipulate Table:

winter=winter.merge(countries,left_on='Country',right_on='Code',how='left')
winter=winter[['Year','City','Sport','Discipline','Athlete','Country_x','Gender','Event','Medal']]
winter.
↳columns=['Year','City','Sport','Discipline','Athlete','Code','Gender','Event','Medal','Country_x']
```

```
[92]: #GroupBy Sport:

medals=winter.groupby(['Athlete','Medal'])['Sport'].count().reset_index().
↳sort_values(by='Sport',ascending=False)
```

```
[93]: # Drop Duplicates:

medals=medals.drop_duplicates(subset=['Medal'],keep='first')
medals.columns=['Athlete','Medal','Count']
medals
```

```
[93]:
```

	Athlete	Medal	Count
724	DAEHLIE, Björn	Gold	8
2029	KIRVESNIEMI, Harri	Bronze	6
932	EHRIG, Andrea	Silver	5

```
[94]: winter['Country'].unique()
```

```
[94]: array(['France', 'Switzerland', 'Finland', 'Belgium', 'United Kingdom',
        'Sweden', 'Canada', 'United States', 'Austria', 'Norway',
        'Germany', nan, 'Hungary', 'Italy', 'Netherlands', 'Japan',
        'Poland', 'Korea, North', 'Spain', 'Liechtenstein', 'Bulgaria',
        'Korea, South', 'China', 'Luxembourg', 'New Zealand', 'Russia',
        'Ukraine', 'Belarus', 'Australia', 'Slovenia', 'Kazakhstan',
        'Uzbekistan', 'Denmark', 'Czech Republic', 'Croatia', 'Estonia',
        'Latvia', 'Slovakia'], dtype=object)
```

```
[95]: winter
```

```
[95]:
```

	Year	City	Sport	Discipline	Athlete	Code	Gender	\
0	1924	Chamonix	Biathlon	Biathlon	BERTHET, G.	FRA	Men	
1	1924	Chamonix	Biathlon	Biathlon	MANDRILLON, C.	FRA	Men	
2	1924	Chamonix	Biathlon	Biathlon	MANDRILLON, Maurice	FRA	Men	
3	1924	Chamonix	Biathlon	Biathlon	VANDELLE, André	FRA	Men	
4	1924	Chamonix	Biathlon	Biathlon	AUFDENBLATTEN, Adolf	SUI	Men	
...	
5765	2014	Sochi	Skiing	Snowboard	JONES, Jenny	GBR	Women	
5766	2014	Sochi	Skiing	Snowboard	ANDERSON, Jamie	USA	Women	
5767	2014	Sochi	Skiing	Snowboard	MALTAIS, Dominique	CAN	Women	
5768	2014	Sochi	Skiing	Snowboard	SAMKOVA, Eva	CZE	Women	
5769	2014	Sochi	Skiing	Snowboard	TRESPEUCH, Chloe	FRA	Women	

	Event	Medal	Country
0	Military Patrol	Bronze	France
1	Military Patrol	Bronze	France
2	Military Patrol	Bronze	France
3	Military Patrol	Bronze	France
4	Military Patrol	Gold	Switzerland
...
5765	Slopestyle	Bronze	United Kingdom
5766	Slopestyle	Gold	United States
5767	Snowboard Cross	Silver	Canada
5768	Snowboard Cross	Gold	Czech Republic
5769	Snowboard Cross	Bronze	France

[5770 rows x 10 columns]

[96]: *#Groupby countries and medals:*

```
winter_medal=winter.groupby(['Country','Medal'])['Code'].count().reset_index().
    ↪sort_values(by=['Country','Medal'],ascending=False)
winter_medal
```

	Country	Medal	Code
98	Uzbekistan	Gold	1
97	United States	Silver	319
96	United States	Gold	167
95	United States	Bronze	167
94	United Kingdom	Silver	11
..
4	Austria	Gold	79
3	Austria	Bronze	103
2	Australia	Silver	3
1	Australia	Gold	5
0	Australia	Bronze	7

[99 rows x 3 columns]

[97]: `winter_medal=winter.groupby(['Country','Code'])['Medal'].count().reset_index()`
`winter_medal.sort_values(by='Medal',ascending=False).head(10).reset_index()`

	index	Country	Code	Medal
0	35	United States	USA	653
1	5	Canada	CAN	625
2	25	Norway	NOR	457
3	11	Finland	FIN	434
4	31	Sweden	SWE	433
5	13	Germany	GER	360

6	32	Switzerland	SUI	285
7	1	Austria	AUT	280
8	27	Russia	RUS	263
9	15	Italy	ITA	192

```
[98]: medals_map = winter.groupby(['Country', 'Code'])['Medal'].count().reset_index()
medals_map.head()
```

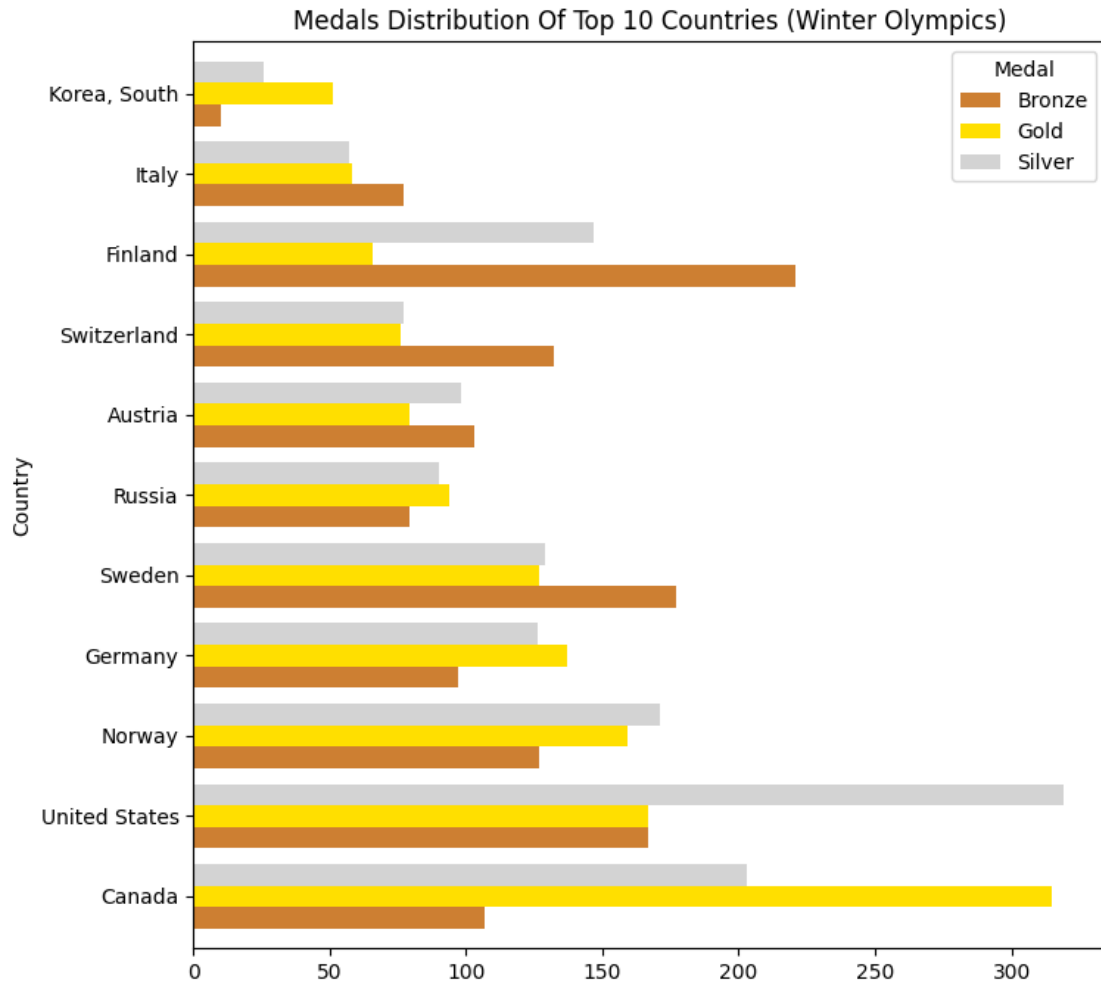
```
[98]:
```

	Country	Code	Medal
0	Australia	AUS	15
1	Austria	AUT	280
2	Belarus	BLR	15
3	Belgium	BEL	13
4	Bulgaria	BUL	6

```
[99]: medals_country=winter.groupby(['Country', 'Medal'])['Gender'].count().
      ↪reset_index().sort_values(by='Gender',ascending=False)
medals_country=medals_country.pivot(index='Country', columns='Medal',
      ↪values='Gender').fillna(0)
top=medals_country.sort_values(by='Gold',ascending=False)[:11]
```

```
[100]: # Plot Medal Distribution for top 10 countries:

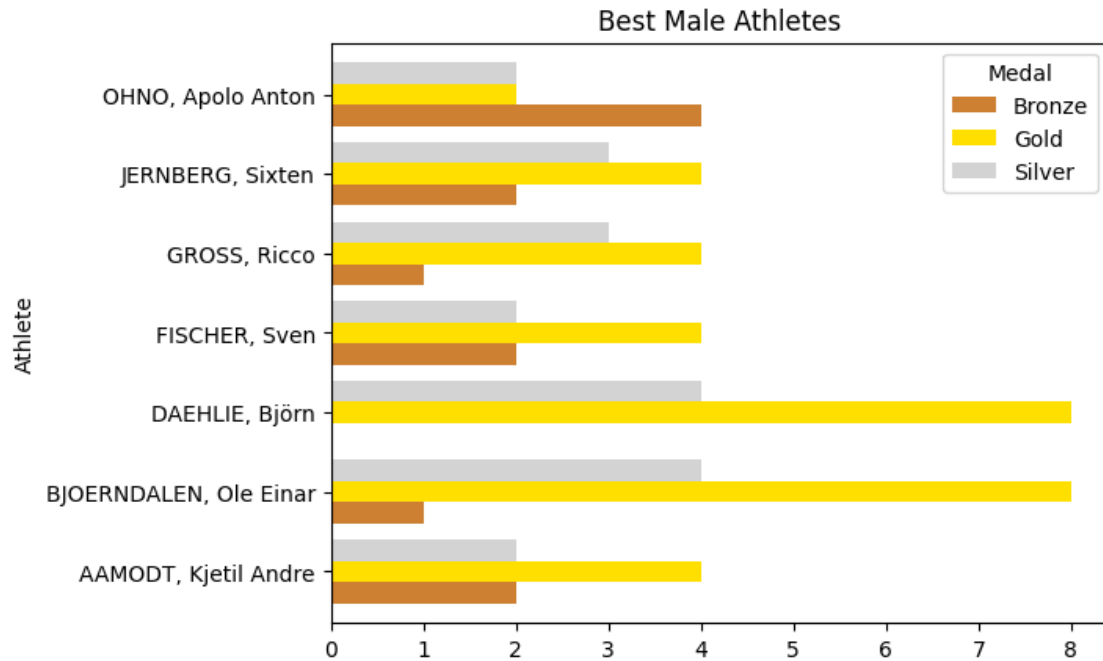
top.plot.barh(width=0.8,color=['#CD7F32', '#FFDF00', '#D3D3D3'])
fig=plt.gcf()
fig.set_size_inches(8,8)
plt.title('Medals Distribution Of Top 10 Countries (Winter Olympics)')
plt.show()
```



```
[101]: # Plot Medal distribution for Top Athlete Male
```

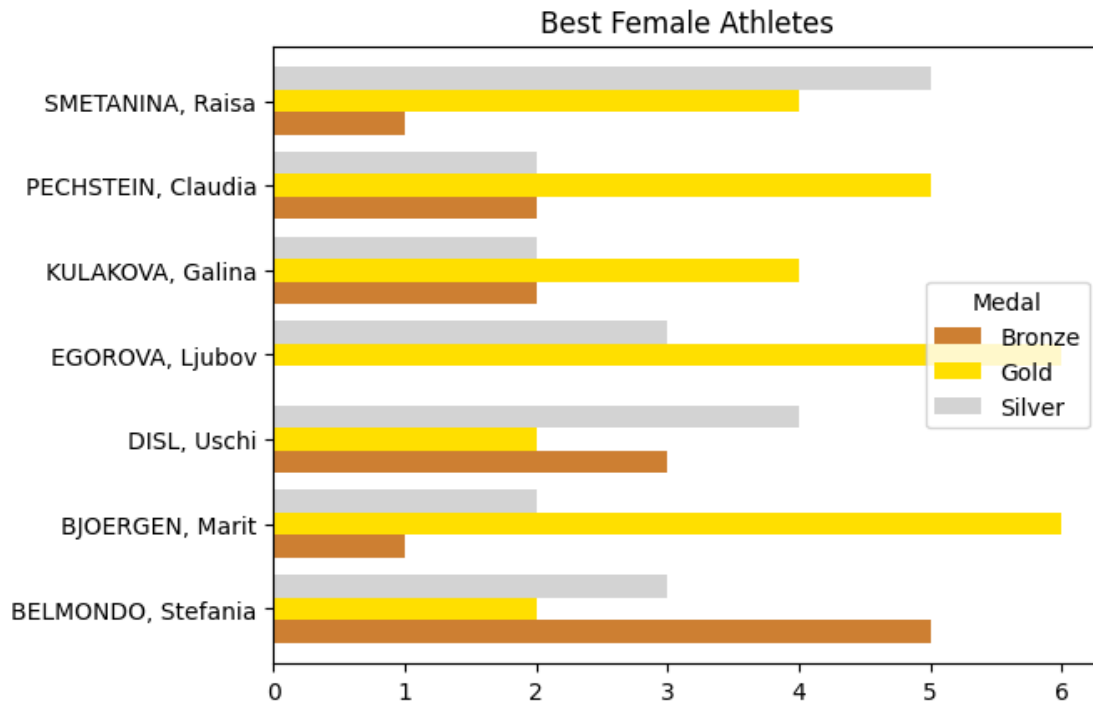
```
fig,ax=plt.figure(figsize=(18,12))
men=winter[winter['Gender']=='Men']
men=men.groupby(['Athlete','Medal'])['Code'].count().reset_index().
    ↪sort_values(by='Code',ascending=False)
men=men[men['Athlete'].isin(winter['Athlete'].value_counts().index[:15])]
men=men.pivot(index='Athlete', columns='Medal', values='Code')
men.plot.barh(width=0.8,color=['#CD7F32','#FFDF00','#D3D3D3'])
plt.title('Best Male Athletes')
plt.ylabel('Athlete')
```

```
[101]: Text(0, 0.5, 'Athlete')
```



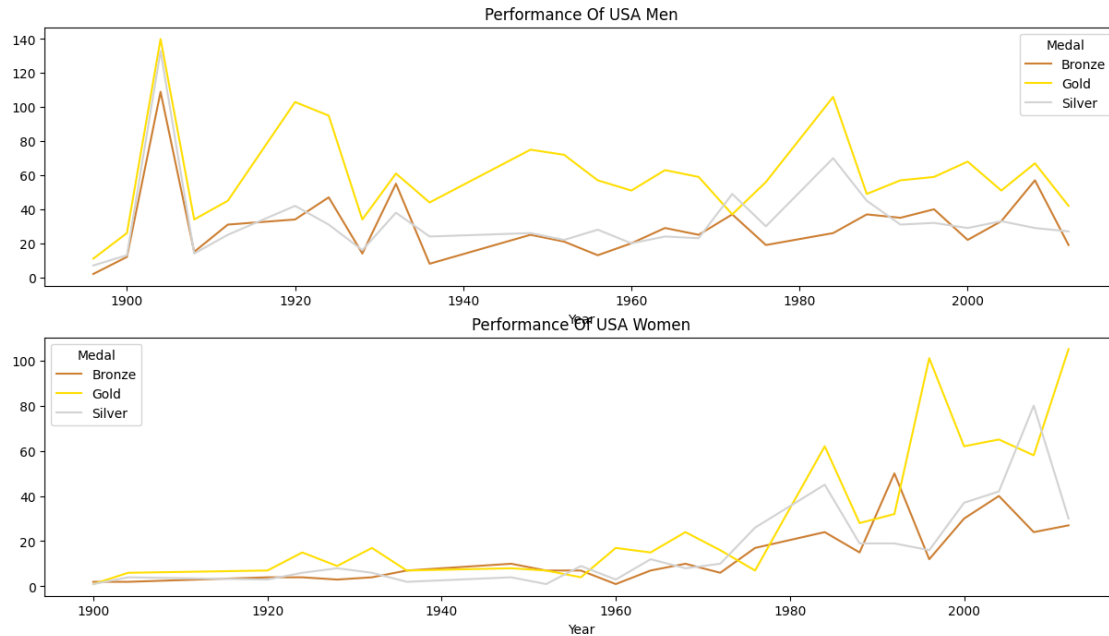
[102]: *# Plot Medal distribution for Top Athlete Female*

```
fig,ax=plt.figure(figsize=(18,12))
women=winter[winter['Gender']=='Women']
women=women.groupby(['Athlete','Medal'])['Code'].count().reset_index().
    ↪sort_values(by='Code',ascending=False)
women=women[women['Athlete'].isin(winter['Athlete'].value_counts().index[:10])]
women=women.pivot(index='Athlete', columns='Medal', values='Code')
women.plot.barh(width=0.8,color=['#CD7F32','#FFDF00','#D3D3D3'])
plt.title('Best Female Athletes')
plt.ylabel('')
plt.show()
```



```
[103]: # Plot performance of Athlete Male and Female for country - United States:

USA_medal_male=summer[(summer['Country']=='United_
↳States')&(summer['Gender']=='Men')]
USA_medal_female=summer[(summer['Country']=='United_
↳States')&(summer['Gender']=='Women')]
fig,ax=plt.subplots(2,figsize=(15,8))
male=USA_medal_male.groupby(['Medal','Year'])['Event'].count().reset_index()
male=male.pivot(index='Year',columns='Medal',values='Event')
male.plot(color=['#CD7F32','#FFDF00','#D3D3D3'],ax=ax[0])
ax[0].set_title('Performance Of USA Men')
female=USA_medal_female.groupby(['Medal','Year'])['Event'].count().reset_index()
female=female.pivot(index='Year',columns='Medal',values='Event')
female.plot(color=['#CD7F32','#FFDF00','#D3D3D3'],ax=ax[1])
ax[1].set_title('Performance Of USA Women')
plt.show()
```



1.0.1 Conclusion:

Exploratory Data Analysis on Olympic Games Data both Summer and Winter: Data cleaning, manipulation of data for further analysis (removing duplicates and counting total medals for top athletes), Creating Groups and pivot tables by medals, Countries, and top athletes (both men and women) for easy analysis, visualising the data acquired from grouping.

Plots for Top 10 Athletes of both Men and women from United States are as shown above.