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

import warnings
warnings.filterwarnings('ignore')
sns.set_style('darkgrid')
```

```
C:\ProgramData\Anaconda3\envs\lib\site-packages\scipy\__init__.py:146: UserWa
rning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of S
ciPy (detected version 1.26.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
df1 = pd.read_csv(r"C:\Users\Sinha Rahul\Downloads\athlete_events.csv (1).zip"
df2 = pd.read_csv(r"C:\Users\Sinha Rahul\Downloads\noc_regions.csv")
```

In [3]:
```
df1.sample(5)
```

Out[3]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Se
185978	93502	Bertrand Pernot du Breuil	M	25.0	NaN	NaN	France	FRA	1952 Summer	1952	Su
7348	4113	Anastasios Andreou	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	Su
261478	130831	Herbert Haydon Wilson	M	33.0	NaN	NaN	Roehampton-1	GBR	1908 Summer	1908	Su
18440	9763	Yana Borysivna Belomoina	F	23.0	164.0	46.0	Ukraine	UKR	2016 Summer	2016	Su
245286	122828	Nigora Tursunkulova	F	17.0	181.0	67.0	Uzbekistan	UZB	2016 Summer	2016	Su

In [4]:
```
df2.sample(5)
```

Out[4]:

	NOC	region	notes
183	SOL	Solomon Islands	NaN
152	NZL	New Zealand	NaN
197	TCH	Czech Republic	NaN
118	LES	Lesotho	NaN
130	MEX	Mexico	NaN

Joins the Dataframes

In [5]:
```
df = df1.merge(df2 , how ='left', on = 'NOC')
df.sample(5)
```

Out[5]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
62417	31968	Khaled El-Farej	M	22.0	155.0	48.0	Syria	SYR	1992 Summer	1992	Summer	B
51250	26346	Paul-Henri Philippe Emmanuel De Le Rue	M	29.0	176.0	82.0	France	FRA	2014 Winter	2014	Winter	
155442	78030	Khaled Meddah	M	26.0	180.0	90.0	Algeria	ALG	2004 Summer	2004	Summer	
237491	119117	Rodman Teltull	M	18.0	170.0	64.0	Palau	PLW	2012 Summer	2012	Summer	
243120	121761	Maria Trebunia (-Strczek)	F	19.0	170.0	68.0	Poland	POL	1976 Winter	1976	Winter	Ir

find out of size of data

In [6]:
```
df.shape
```

Out[6]: (271116, 17)

statistical

In [7]:
```
df.describe()
```

Out[7]:

	ID	Age	Height	Weight	Year
count	271116.000000	261642.000000	210945.000000	208241.000000	271116.000000
mean	68248.954396	25.556898	175.338970	70.702393	1978.378480
std	39022.286345	6.393561	10.518462	14.348020	29.877632
min	1.000000	10.000000	127.000000	25.000000	1896.000000
25%	34643.000000	21.000000	168.000000	60.000000	1960.000000
50%	68205.000000	24.000000	175.000000	70.000000	1988.000000
75%	102097.250000	28.000000	183.000000	79.000000	2002.000000
max	135571.000000	97.000000	226.000000	214.000000	2016.000000

check data type

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 271116 entries, 0 to 271115
Data columns (total 17 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 
 15  region  270746 non-null  object 
 16  notes   5039 non-null    object 
dtypes: float64(3), int64(2), object(12)
memory usage: 37.2+ MB
```

In [9]:
```
df.memory_usage()
```

Out[9]:
```
Index     2168928
ID        2168928
Name      2168928
Sex       2168928
Age       2168928
Height    2168928
Weight    2168928
Team      2168928
NOC       2168928
Games     2168928
Year      2168928
Season    2168928
City      2168928
Sport     2168928
Event     2168928
Medal     2168928
region    2168928
notes     2168928
dtype: int64
```

unirrelevant columns

In [10]:
```
df = df.drop(columns = "notes", axis = 0)
```

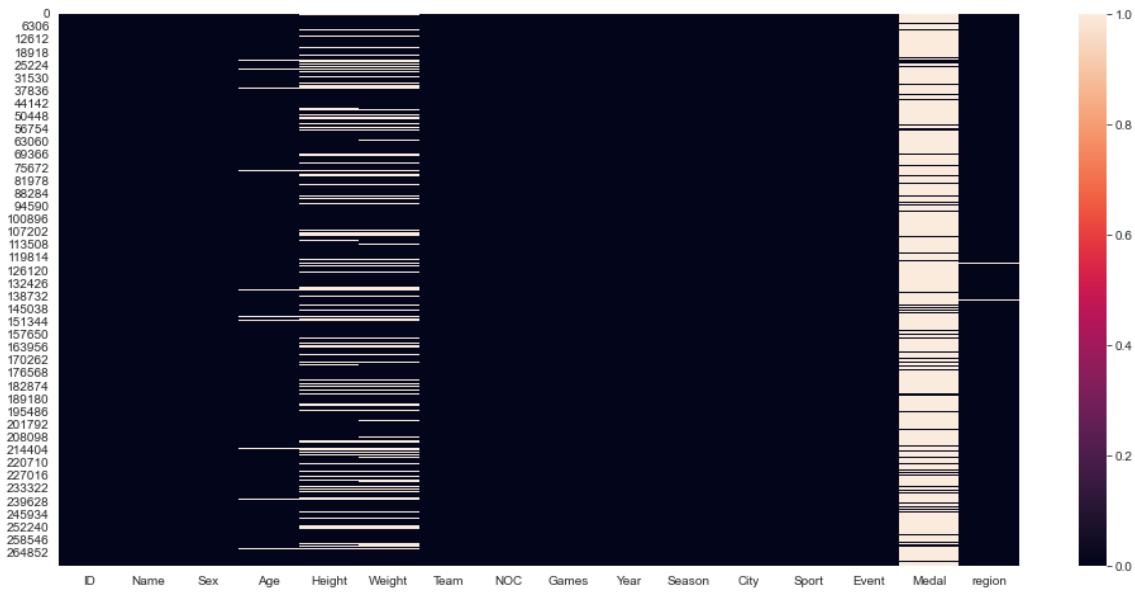
Check the dublicate data and drop them

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
ID             0
Name           0
Sex            0
Age         9474
Height     60171
Weight     62875
Team           0
NOC            0
Games          0
Year           0
Season         0
City           0
Sport          0
Event          0
Medal     231333
region       370
dtype: int64
```

In [12]:
```
plt.figure(figsize=(17,8))
sns.heatmap(df.isnull())
```

Out[12]:
```
<AxesSubplot:>
```



check duplicate data

In [13]:
```
df.duplicated()
```

Out[13]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [14]:
```
df[df.duplicated()]
```

Out[14]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seas
1252	704	Dsir Antoine Acket	M	27.0	NaN	NaN	Belgium	BEL	1932 Summer	1932	Summ
4282	2449	William Truman Aldrich	M	48.0	NaN	NaN	United States	USA	1928 Summer	1928	Summ
4283	2449	William Truman Aldrich	M	48.0	NaN	NaN	United States	USA	1928 Summer	1928	Summ

In [15]:
```
len(df[df.duplicated()])
```

Out[15]: 1385

In [16]:
```
df.drop_duplicates()
```

Out[16]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Se
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Sur
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Sur
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Sur
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Sur
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	W
...	...	...	...	...	...	...	...	...	...	...	
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	W
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	W
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	W
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	W
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	W

269731 rows × 16 columns

In [17]:
```
df.duplicated().sum()
```

Out[17]: 1385

check null value

In [18]:
```
df[['Age','Height','Weight']] = df[['Age','Height','Weight']].fillna(df[['Age'

df.Medal = df.Medal.fillna('None')

df.Age = df.Age.astype(int)
```

In [19]:
```
df
```

Out[19]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Yea
0	1	A Dijiang	M	24	180.00000	80.000000	China	CHN	1992 Summer	199
1	2	A Lamusi	M	23	170.00000	60.000000	China	CHN	2012 Summer	201
2	3	Gunnar Nielsen Aaby	M	24	175.33897	70.702393	Denmark	DEN	1920 Summer	192
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer	190
4	5	Christine Jacoba Aaftink	F	21	185.00000	82.000000	Netherlands	NED	1988 Winter	198
...	...	...	...	...	...	...	...	...	...	.
271111	135569	Andrzej ya	M	29	179.00000	89.000000	Poland-1	POL	1976 Winter	197
271112	135570	Piotr ya	M	27	176.00000	59.000000	Poland	POL	2014 Winter	201
271113	135570	Piotr ya	M	27	176.00000	59.000000	Poland	POL	2014 Winter	201
271114	135571	Tomasz Ireneusz ya	M	30	185.00000	96.000000	Poland	POL	1998 Winter	199
271115	135571	Tomasz Ireneusz ya	M	34	185.00000	96.000000	Poland	POL	2002 Winter	200

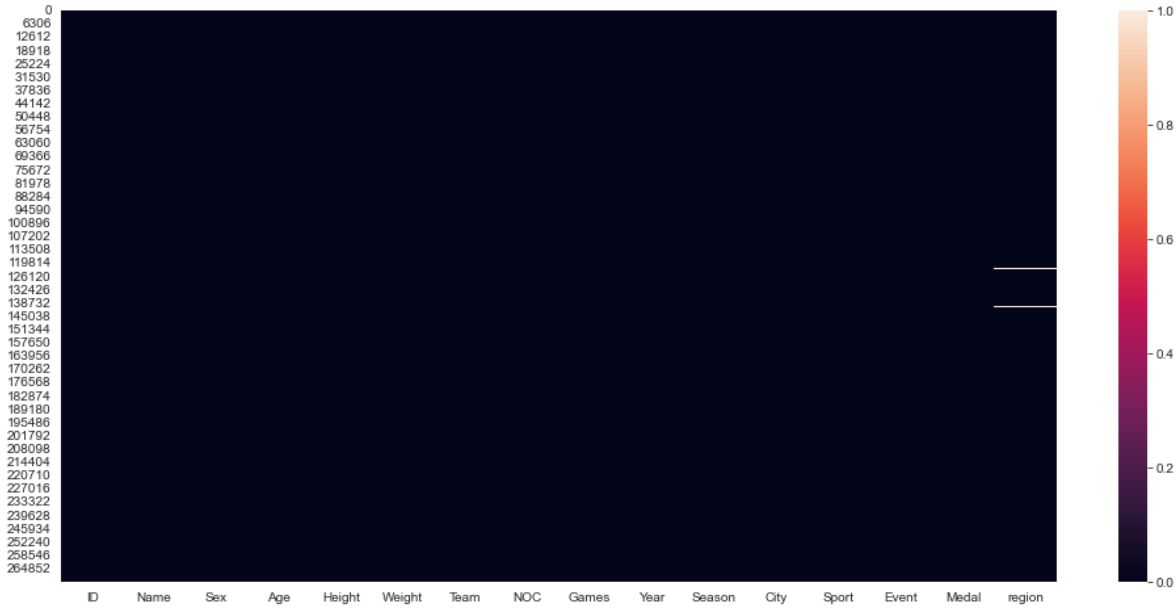
271116 rows × 16 columns

In [20]:
```
df.isnull().sum().sum()
```

Out[20]: 370

In [21]:
```
plt.figure(figsize=(17,8))
sns.heatmap(df.isnull())
```

Out[21]: <AxesSubplot:>



In [22]:
```
df.shape
```

Out[22]: (271116, 16)

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 271116 entries, 0 to 271115
Data columns (total 16 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     271116 non-null  int32
 4   Height  271116 non-null  float64
 5   Weight  271116 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
```

Exploratory Analysis and Visualization

Top countries participating in Olympics

In [24]:
```
df.columns
```

Out[24]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region'],
      dtype='object')
```

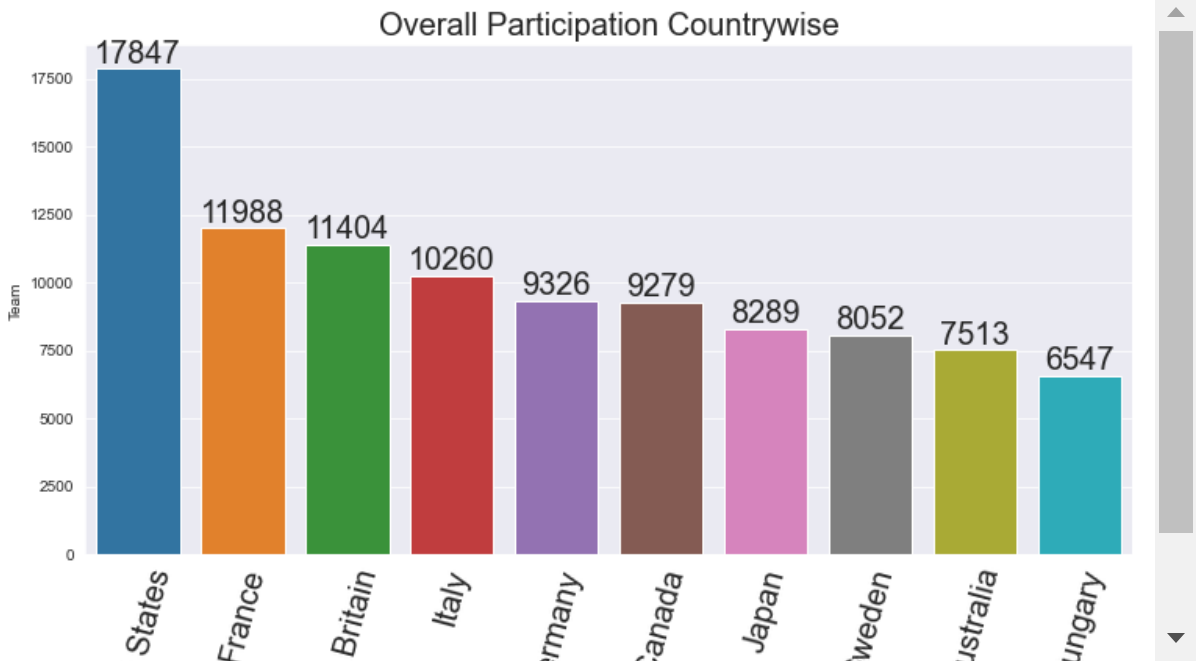
In [25]:
```
df["Team"].nunique()
```

Out[25]:
```
1184
```

In [26]:
```
top_country = df["Team"].value_counts().sort_values(ascending=False).head(10)
top_country
```

Out[26]:
```
United States    17847
France           11988
Great Britain    11404
Italy            10260
Germany           9326
Canada            9279
Japan             8289
Sweden            8052
Australia         7513
Hungary           6547
Name: Team, dtype: int64
```

In [27]:
```
plt.figure(figsize=(12,6))
plt.xticks(rotation=75,fontsize=20)
plt.title('Overall Participation Countrywise',fontsize=20)
barplot = sns.barplot(x=top_country.index, y=top_country)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
```



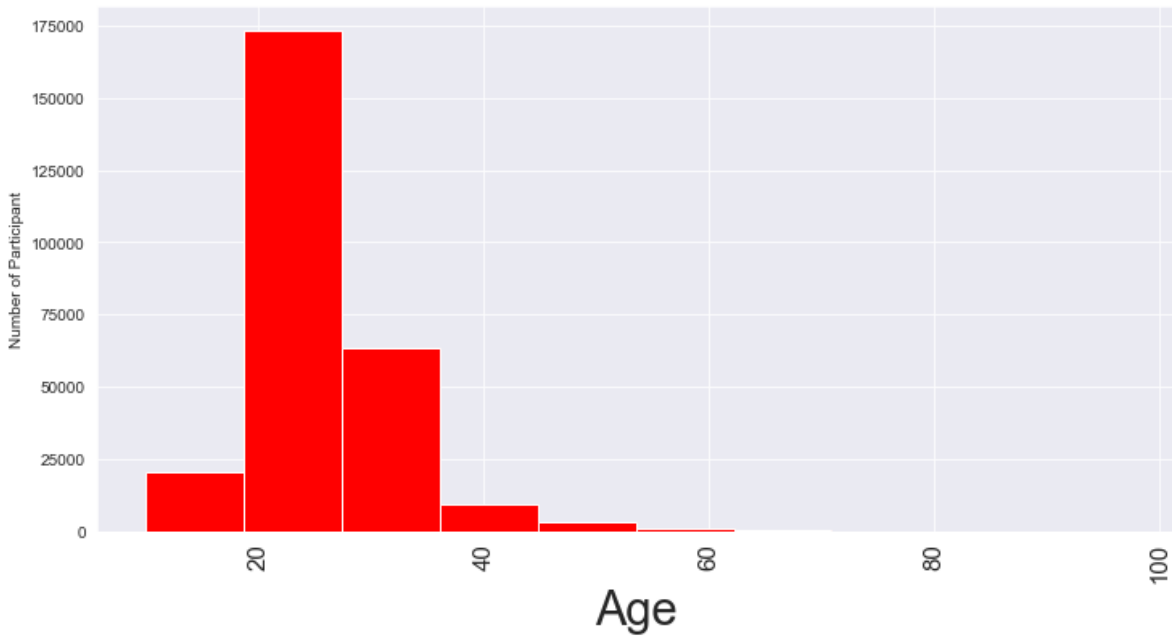
USA has historically won maximum no of medals

Age Distribution

In [28]:
```
df["Age"].value_counts()
```

Out[28]:
```
25    29181
23    21875
24    21720
22    20814
21    19164
      ...
77        2
84        1
10        1
96        1
97        1
Name: Age, Length: 74, dtype: int64
```

In [29]:
```
plt.figure(figsize=(12, 6))
plt.xlabel('Age',fontsize=29)
plt.ylabel('Number of Participant')
plt.hist(df["Age"], color= "r")
plt.xticks(rotation=90,fontsize=15)
plt.show()
```



Gender Distribution

In [30]:
```
df.columns
```

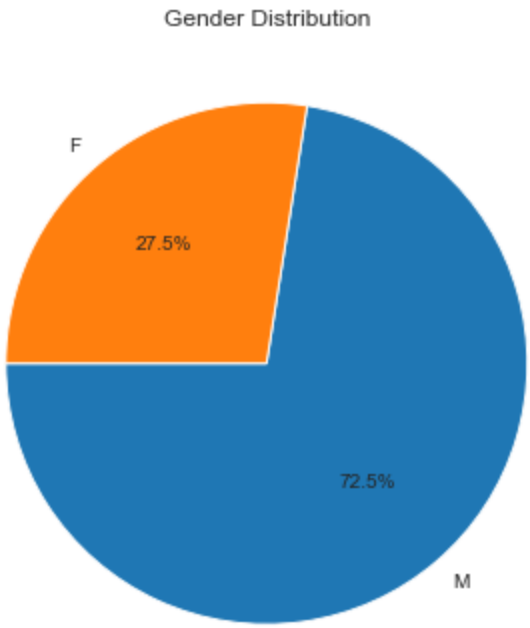
Out[30]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region'],
      dtype='object')
```

In [31]:
```
df["Sex"].value_counts()
```

Out[31]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

To get the percentage

In [32]:
```
plt.figure(figsize=(12,6))
plt.title('Gender Distribution')
plt.pie(df["Sex"].value_counts(), labels=df["Sex"].value_counts().keys(), auto
```



how many team are there participate in this 140 year olympic historty

In [33]:
```
df.columns
```

Out[33]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region'],
      dtype='object')
```

In [34]:
```
df["Team"].nunique()
```

Out[34]:
```
1184
```

how many sports are there participate in this 140 year olympic historty

In [35]:
```
df.columns
```

Out[35]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region'],
      dtype='object')
```

In [36]:
```
df["Sport"].unique()
```

Out[36]:
```
array(['Basketball', 'Judo', 'Football', 'Tug-Of-War', 'Speed Skating',
       'Cross Country Skiing', 'Athletics', 'Ice Hockey', 'Swimming',
       'Badminton', 'Sailing', 'Biathlon', 'Gymnastics',
       'Art Competitions', 'Alpine Skiing', 'Handball', 'Weightlifting',
       'Wrestling', 'Luge', 'Water Polo', 'Hockey', 'Rowing', 'Bobsleigh',
       'Fencing', 'Equestrianism', 'Shooting', 'Boxing', 'Taekwondo',
       'Cycling', 'Diving', 'Canoeing', 'Tennis', 'Modern Pentathlon',
       'Figure Skating', 'Golf', 'Softball', 'Archery', 'Volleyball',
       'Synchronized Swimming', 'Table Tennis', 'Nordic Combined',
       'Baseball', 'Rhythmic Gymnastics', 'Freestyle Skiing',
       'Rugby Sevens', 'Trampolining', 'Beach Volleyball', 'Triathlon',
       'Ski Jumping', 'Curling', 'Snowboarding', 'Rugby',
       'Short Track Speed Skating', 'Skeleton', 'Lacrosse', 'Polo',
       'Cricket', 'Racquets', 'Motorboating', 'Military Ski Patrol',
       'Croquet', 'Jeu De Paume', 'Roque', 'Alpinism', 'Basque Pelota',
       'Aeronautics'], dtype=object)
```

In [37]:
```
df["Sport"].nunique()
```

Out[37]:
```
66
```

how many season are there participate in this 140 year olympic historty

In [38]:
```
df["Season"].unique()
```

Out[38]:
```
array(['Summer', 'Winter'], dtype=object)
```

how many games are there participate in this 140 year olympic historty

In [39]:
```
df["Games"].unique()
```

Out[39]:
```
array(['1992 Summer', '2012 Summer', '1920 Summer', '1900 Summer',
       '1988 Winter', '1992 Winter', '1994 Winter', '1932 Summer',
       '2002 Winter', '1952 Summer', '1980 Winter', '2000 Summer',
       '1996 Summer', '1912 Summer', '1924 Summer', '2014 Winter',
       '1948 Summer', '1998 Winter', '2006 Winter', '2008 Summer',
       '2016 Summer', '2004 Summer', '1960 Winter', '1964 Winter',
       '1984 Winter', '1984 Summer', '1968 Summer', '1972 Summer',
       '1988 Summer', '1936 Summer', '1952 Winter', '1956 Winter',
       '1956 Summer', '1960 Summer', '1928 Summer', '1976 Summer',
       '1980 Summer', '1964 Summer', '2010 Winter', '1968 Winter',
       '1906 Summer', '1972 Winter', '1976 Winter', '1924 Winter',
       '1904 Summer', '1928 Winter', '1908 Summer', '1948 Winter',
       '1932 Winter', '1936 Winter', '1896 Summer'], dtype=object)
```

Young and old athletes detected according to the average age.

In [40]:
```
avarge_age = df["Age"].mean()
avarge_age
```

Out[40]:
```
25.537437849481403
```

In [41]:
```
dff = df["Age"].unique()
dff
```

Out[41]:
```
array([24, 23, 34, 21, 25, 27, 31, 33, 18, 26, 22, 30, 32, 28, 54, 20, 17,
       43, 47, 29, 41, 45, 49, 53, 57, 19, 38, 35, 16, 37, 15, 42, 46, 40,
       36, 14, 39, 48, 52, 44, 55, 50, 71, 63, 51, 58, 13, 60, 75, 65, 56,
       64, 68, 84, 12, 72, 59, 61, 70, 74, 62, 67, 69, 73, 66, 11, 76, 88,
       96, 80, 10, 81, 77, 97])
```

In [42]:
```
df["Age_Level"]=["Old" if item>avarge_age else "Young" for item in df["Age"]]
```

In [43]:
```
df.loc[:10,["Name","Age","Age_Level"]]
```

Out[43]:

	Name	Age	Age_Level
0	A Dijiang	24	Young
1	A Lamusi	23	Young
2	Gunnar Nielsen Aaby	24	Young
3	Edgar Lindenau Aabye	34	Old
4	Christine Jacoba Aaftink	21	Young
5	Christine Jacoba Aaftink	21	Young
6	Christine Jacoba Aaftink	25	Young
7	Christine Jacoba Aaftink	25	Young
8	Christine Jacoba Aaftink	27	Old
9	Christine Jacoba Aaftink	27	Old
10	Per Knut Aaland	31	Old

gold medal winners who are less than 18 years of age

In [44]:
```
df.columns
```

Out[44]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [45]:
```
df["Medal"].unique()
```

Out[45]:
```
array(['None', 'Gold', 'Bronze', 'Silver'], dtype=object)
```

In [46]:
```
df["Medal"].value_counts()
```

Out[46]:
```
None      231333
Gold       13372
Bronze     13295
Silver     13116
Name: Medal, dtype: int64
```

In [47]:
```
((df["Age"]<18) & (df["Medal"]=="Gold"))
```

Out[47]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [48]:
```
df[((df["Age"]<18) & (df["Medal"]=="Gold"))].head(1)
```

Out[48]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
710	404	Louis Grenville "Lou" Abell	M	15	175.33897	70.702393	Vesper Boat Club	USA	1900 Summer	1900	Summer	Paris

silver medal winners who are less than 18 years of age

In [49]:
```
((df["Age"]<18) & (df["Medal"]=="Silver"))
```

Out[49]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [50]:
```
df[((df["Age"]<18) & (df["Medal"]=="Silver"))].head(1)
```

Out[50]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
2587	1449	Thomas Ahrens	M	16	163.0	50.0	Germany	GER	1964 Summer	1964	Summer	Tokyo

bronze medal winners who are less than 18 years of age

In [51]:
```
((df["Age"]<18) & (df["Medal"]=="Bronze"))
```

Out[51]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [52]:
```
df[((df["Age"]<18) & (df["Medal"]=="Bronze"))].head(1)
```

Out[52]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
740	424	Moira Abernethy (-Ford)	F	17	175.33897	70.702393	South Africa	RSA	1956 Summer	1956	Summer	Mell

gold medal winners who are less than 18 years to 25 of age

In [53]:
```
((df["Age"].between(18,25)) & (df["Medal"]=="Gold"))
```

Out[53]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [54]:
```
df[((df["Age"].between(18,25)) & (df["Medal"]=="Gold"))].head(1)
```

Out[54]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sp
60	20	Kjetil Andr Aamodt	M	20	176.0	85.0	Norway	NOR	1992 Winter	1992	Winter	Albertville	Alpi Skii

In [55]:
```
len(df[((df["Age"].between(18,25)) & (df["Medal"]=="Gold"))])
```

Out[55]: 6909

In [56]:
```
len(df[((df["Age"].between(18,25)) & (df["Medal"]=="Silver"))])
```

Out[56]: 6724

In [57]:
```
len(df[((df["Age"].between(18,25)) & (df["Medal"]=="Bronze"))])
```

Out[57]:
```
6983
```

In [58]:
```
len(df[((df["Age"]>=50) & (df["Medal"]=="Gold"))])
```

Out[58]:
```
77
```

In [59]:
```
len(df[((df["Age"]>=50) & (df["Medal"]=="Bronze"))])
```

Out[59]:
```
72
```

In [60]:
```
len(df[((df["Age"]>=50) & (df["Medal"]=="Silver"))])
```

Out[60]:
```
87
```

check the female participants of the years 1896 to 2016

In [61]:
```
df.columns
```

Out[61]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [62]:
```
df3 = df.sort_values(by = "Year")
df3.head(1)
```

Out[62]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seasc
214333	107607	Fritz Richard Gustav Schuft	M	19	175.33897	70.702393	Germany	GER	1896 Summer	1896	Summ

In [63]:
```
df3["Year"].unique()
```

Out[63]:
```
array([1896, 1900, 1904, 1906, 1908, 1912, 1920, 1924, 1928, 1932, 1936,
       1948, 1952, 1956, 1960, 1964, 1968, 1972, 1976, 1980, 1984, 1988,
       1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012,
       2014, 2016], dtype=int64)
```

In [64]:
```
df["Sex"]=="F"
```

Out[64]:
```
0         False
1         False
2         False
3         False
4          True
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Name: Sex, Length: 271116, dtype: bool
```

In [65]:
```
female_participated = df[df["Sex"]=="F"][["Year","Sex"]]
female_participated
```

Out[65]:

	Year	Sex
4	1988	F
5	1988	F
6	1992	F
7	1992	F
8	1994	F
...	...	...
271080	1964	F
271099	2004	F
271102	2000	F
271103	2004	F
271110	2016	F

74522 rows × 2 columns

In [66]:
```
female_participated = female_participated.groupby(["Year"]).count()
```

In [67]:
```
female_participated.sort_values("Sex", ascending=False).head()
```

Out[67]:

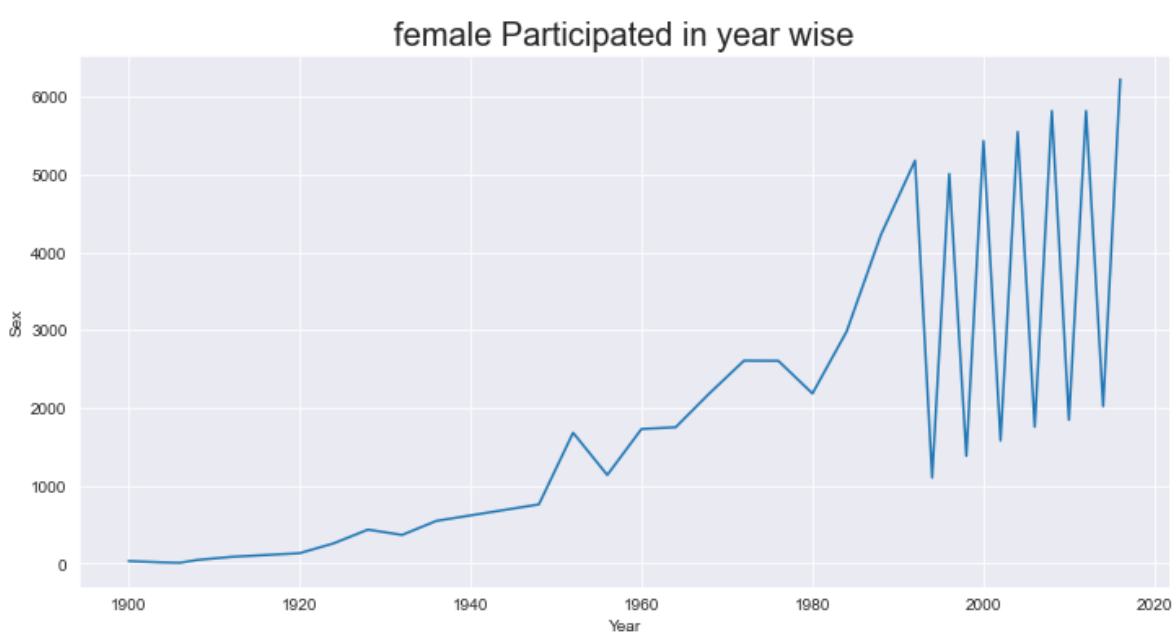
Year	Sex
2016	6223
2008	5816
2012	5815
2004	5546
2000	5431

In [68]:
```
female_participated.sort_values("Sex", ascending=True).head()
```

Out[68]:

Year	Sex
1906	11
1904	16
1900	33
1908	47
1912	87

In [69]:
```
plt.figure(figsize=(12,6))
sns.lineplot(
    data=female_participated, x="Year", y="Sex",  err_style="bars", ci=68
)
plt.title('female Participated in year wise', fontsize=20)
plt.show()
```



check the male participants of the years 1896 to 2016

In [70]:
```
df.columns
```

Out[70]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [71]:
```
df3 = df.sort_values(by = "Year")
df3.head(1)
```

Out[71]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seasc
214333	107607	Fritz Richard Gustav Schuft	M	19	175.33897	70.702393	Germany	GER	1896 Summer	1896	Summ

In [72]:
```
df3["Year"].unique()
```

Out[72]:
```
array([1896, 1900, 1904, 1906, 1908, 1912, 1920, 1924, 1928, 1932, 1936,
       1948, 1952, 1956, 1960, 1964, 1968, 1972, 1976, 1980, 1984, 1988,
       1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012,
       2014, 2016], dtype=int64)
```

In [73]:
```
df["Sex"]=="M"
```

Out[73]:
```
0          True
1          True
2          True
3          True
4         False
          ...
271111     True
271112     True
271113     True
271114     True
271115     True
Name: Sex, Length: 271116, dtype: bool
```

In [74]:
```
Male_participated = df[df["Sex"]=="M"][["Year","Sex"]]
Male_participated
```

Out[74]:

	Year	Sex
0	1992	M
1	2012	M
2	1920	M
3	1900	M
10	1992	M
...	...	...
271111	1976	M
271112	2014	M
271113	2014	M
271114	1998	M
271115	2002	M

196594 rows × 2 columns

In [75]:
```
Male_participated = Male_participated.groupby(["Year"]).count()
```

In [76]:
```
Male_participated
```

Out[76]:

Year	Sex
1896	380
1900	1903
1904	1285
1906	1722
1908	3054
1912	3953
1920	4158
1924	5432
1928	5137
1932	2952
1936	6852
1948	6719
1952	7676
1956	5295
1960	7505
1964	7728
1968	8286
1972	9351
1976	7896
1980	6751
1984	8605
1988	10453
1992	11235
1994	2055
1996	8772
1998	2221
2000	8390
2002	2527
2004	7897
2006	2625
2008	7786
2010	2555
2012	7105
2014	2868

Year	Sex
2016	7465

In [77]:
```
Male_participated.sort_values("Sex", ascending=False).head()
```

Out[77]:

Year	Sex
1992	11235
1988	10453
1972	9351
1996	8772
1984	8605

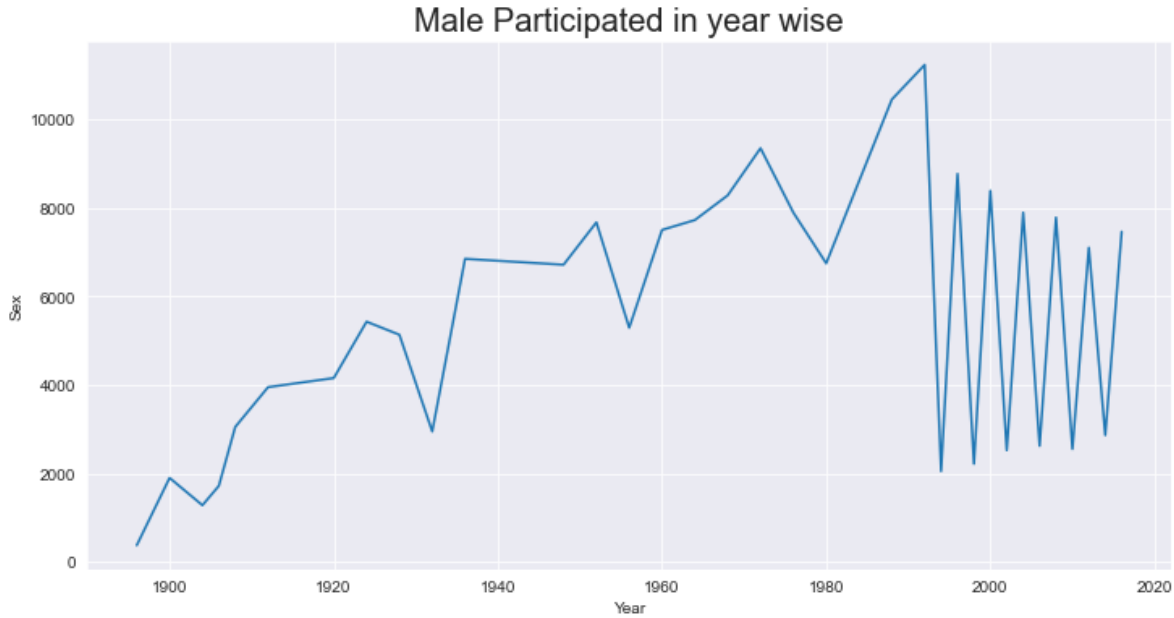
In [78]:
```
Male_participated.sort_values("Sex", ascending=True).head()
```

Out[78]:

Year	Sex
1896	380
1904	1285
1906	1722
1900	1903
1994	2055

In [79]:
```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=Male_participated, x="Year", y="Sex",  err_style="bars", ci=68)
plt.title('Male Participated in year wise', fontsize=20)
plt.show()
```



Participants across seasons

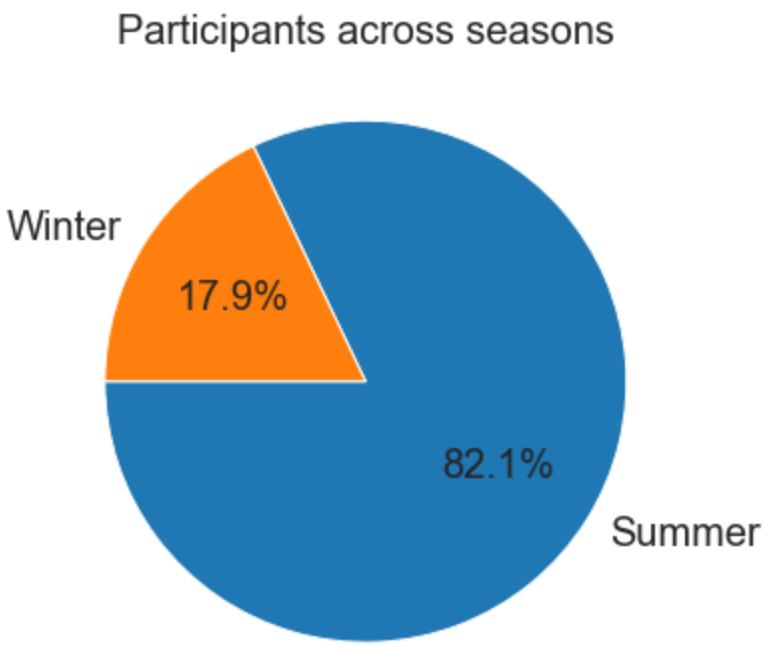
In [80]:
```
df.columns
```

Out[80]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [81]:
```
df["Season"].value_counts()
```

Out[81]:
```
Summer    222552
Winter     48564
Name: Season, dtype: int64
```

In [82]:
```
plt.figure(figsize=(12,6))
plt.title('Participants across seasons', fontsize=20)
plt.pie(df["Season"].value_counts(), labels=df["Season"].value_counts().keys()
       textprops = {"fontsize":20})
plt.show()
```



check the male participants of the years 1896 to 2016 in winter season

In [83]:
```
(df["Sex"]=="M") & (df["Season"]=="Winter")
```

Out[83]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111     True
271112     True
271113     True
271114     True
271115     True
Length: 271116, dtype: bool
```

In [84]:
```
df[(df["Sex"]=="M") & (df["Season"]=="Winter")].head(1)
```

Out[84]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Spo
10	6	Per Knut Aaland	M	31	188.0	75.0	United States	USA	1992 Winter	1992	Winter	Albertville	Cro Count Skii

In [85]:
```
len(df[(df["Sex"]=="M") & (df["Season"]=="Winter")])
```

Out[85]: 33485

In [86]:
```
winter_participated_Male = df[(df["Sex"]=="M") & (df["Season"]=="Winter")][["Y
```

In [87]:
```
winter_participated_Male
```

Out[87]:

	Year	Sex
10	1992	M
11	1992	M
12	1992	M
13	1992	M
14	1994	M
...	...	...
271111	1976	M
271112	2014	M
271113	2014	M
271114	1998	M
271115	2002	M

33485 rows × 2 columns

In [88]:
```
winter_participated_Male = winter_participated_Male.groupby("Year").count()
```

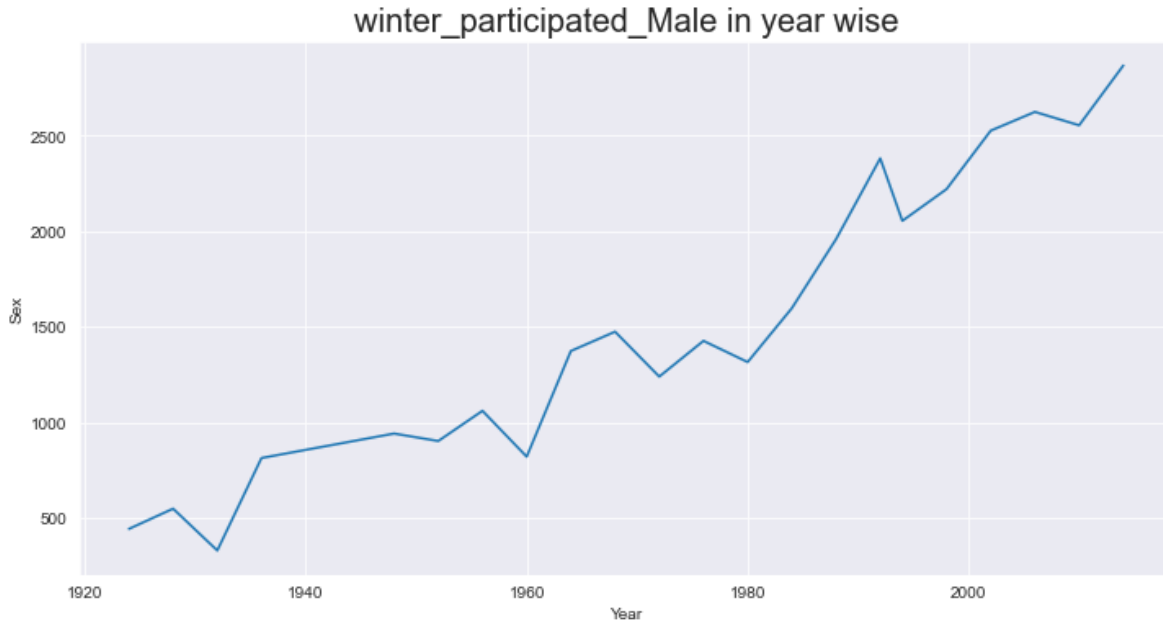
In [89]:
```
winter_participated_Male
```

Out[89]:

Year	Sex
1924	443
1928	549
1932	330
1936	814
1948	942
1952	903
1956	1061
1960	821
1964	1374
1968	1475
1972	1240
1976	1427
1980	1316
1984	1598
1988	1959
1992	2382
1994	2055
1998	2221
2002	2527
2006	2625
2010	2555
2014	2868

In [90]:
```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=winter_participated_Male, x="Year", y="Sex",  err_style="bars", ci=68
plt.title('winter_participated_Male in year wise', fontsize=20)
plt.show()
```



check the female participants of the years 1896 to 2016 in winter season

In [91]:
```
(df["Sex"]=="F") & (df["Season"]=="Winter")
```

Out[91]:
```
0         False
1         False
2         False
3         False
4          True
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [92]:
```
df[(df["Sex"]=="F") & (df["Season"]=="Winter")].head(1)
```

Out[92]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
4	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary

In [93]:
```
len(df[(df["Sex"]=="F") & (df["Season"]=="Winter")])
```

Out[93]: 15079

In [94]:
```
winter_participated_female = df[(df["Sex"]=="F") & (df["Season"]=="Winter")][[
```

In [95]:
```
winter_participated_female
```

Out[95]:

	Year	Sex
4	1988	F
5	1988	F
6	1992	F
7	1992	F
8	1994	F
...	...	...
271061	2002	F
271062	2002	F
271063	2002	F
271064	2002	F
271065	2002	F

15079 rows × 2 columns

In [96]:
```
winter_participated_female = winter_participated_female.groupby("Year").count(
```

In [97]:
```
winter_participated_female
```

Out[97]:

Year	Sex
1924	17
1928	33
1932	22
1936	81
1948	133
1952	185
1956	246
1960	295
1964	404
1968	416
1972	415
1976	434
1980	430
1984	536
1988	680
1992	1054
1994	1105
1998	1384
2002	1582
2006	1757
2010	1847
2014	2023

In [98]:
```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=winter_participated_female, x="Year", y="Sex",  err_style="bars", ci=
plt.title('winter_participated_female in year wise', fontsize=20)
plt.show()
```

check the male participants of the years 1896 to 2016 in summer season

In [99]:
```
(df["Sex"]=="M") & (df["Season"]=="Summer")
```

Out[99]:
```
0          True
1          True
2          True
3          True
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [100]:
```
df[(df["Sex"]=="M") & (df["Season"]=="Summer")].head(1)
```

Out[100]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
0	1	A Dijiang	M	24	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Baske

In [101]:
```
len(df[(df["Sex"]=="M") & (df["Season"]=="Summer")])
```

Out[101]: 163109

In [102]:
```
summer_participated_male = df[(df["Sex"]=="M") & (df["Season"]=="Summer")][["Y
```

In [103]:
```
summer_participated_male
```

Out[103]:

	Year	Sex
0	1992	M
1	2012	M
2	1920	M
3	1900	M
29	1952	M
...	...	...
271105	2000	M
271106	2004	M
271107	1972	M
271108	2000	M
271109	2004	M

163109 rows × 2 columns

In [104]:
```
summer_participated_male = summer_participated_male.groupby("Year").count()
```

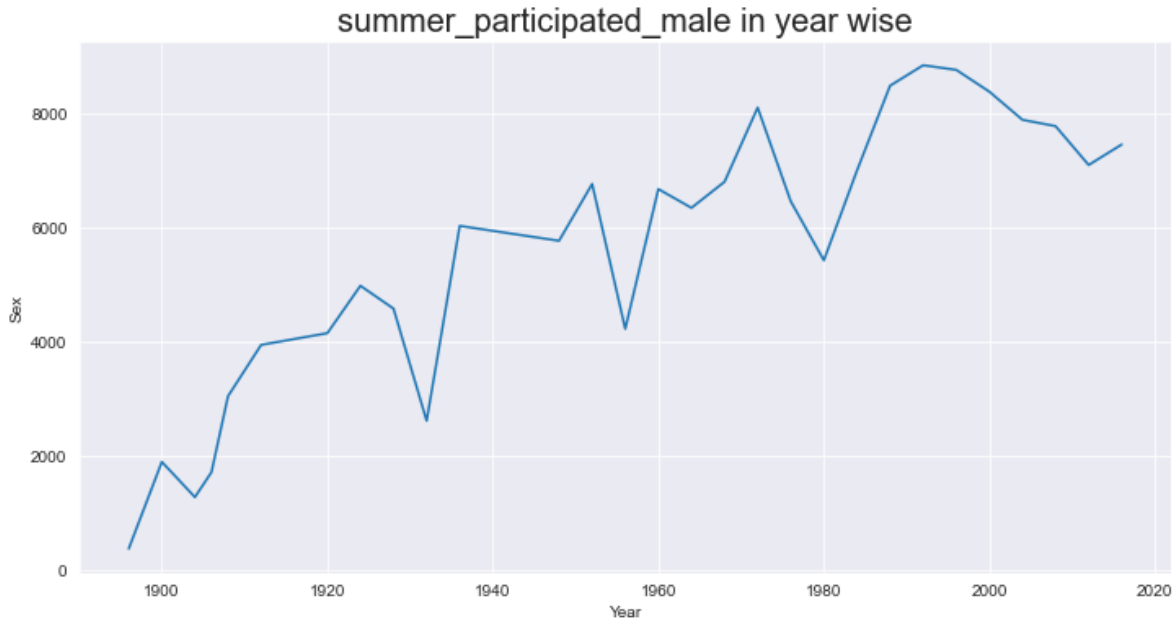
In [105]:
```
summer_participated_male
```

Out[105]:

Year	Sex
1896	380
1900	1903
1904	1285
1906	1722
1908	3054
1912	3953
1920	4158
1924	4989
1928	4588
1932	2622
1936	6038
1948	5777
1952	6773
1956	4234
1960	6684
1964	6354
1968	6811
1972	8111
1976	6469
1980	5435
1984	7007
1988	8494
1992	8853
1996	8772
2000	8390
2004	7897
2008	7786
2012	7105
2016	7465

In [106]:
```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=summer_participated_male, x="Year", y="Sex",  err_style="bars",ci=68)
plt.title('summer_participated_male in year wise', fontsize=20)
plt.show()
```



check the female participants of the years 1896 to 2016 in summer season

In [107]:
```
(df["Sex"]=="F") & (df["Season"]=="Summer")
```

Out[107]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [108]:
```
df[(df["Sex"]=="F") & (df["Season"]=="Summer")].head(1)
```

Out[108]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
26	8	Cornelia "Cor" Aalten (-Strannood)	F	18	168.0	70.702393	Netherlands	NED	1932 Summer	1932	Summer	Ang

In [109]:
```
len(df[(df["Sex"]=="F") & (df["Season"]=="Summer")])
```

Out[109]: 59443

In [110]:
```
summer_participated_female = df[(df["Sex"]=="F") & (df["Season"]=="Summer")][[
```

In [111]:
```
summer_participated_female
```

Out[111]:

	Year	Sex
26	1932	F
27	1932	F
32	1996	F
33	2000	F
79	2008	F
...	...	...
271080	1964	F
271099	2004	F
271102	2000	F
271103	2004	F
271110	2016	F

59443 rows × 2 columns

In [112]:
```
summer_participated_female = summer_participated_female.groupby("Year").count(
```

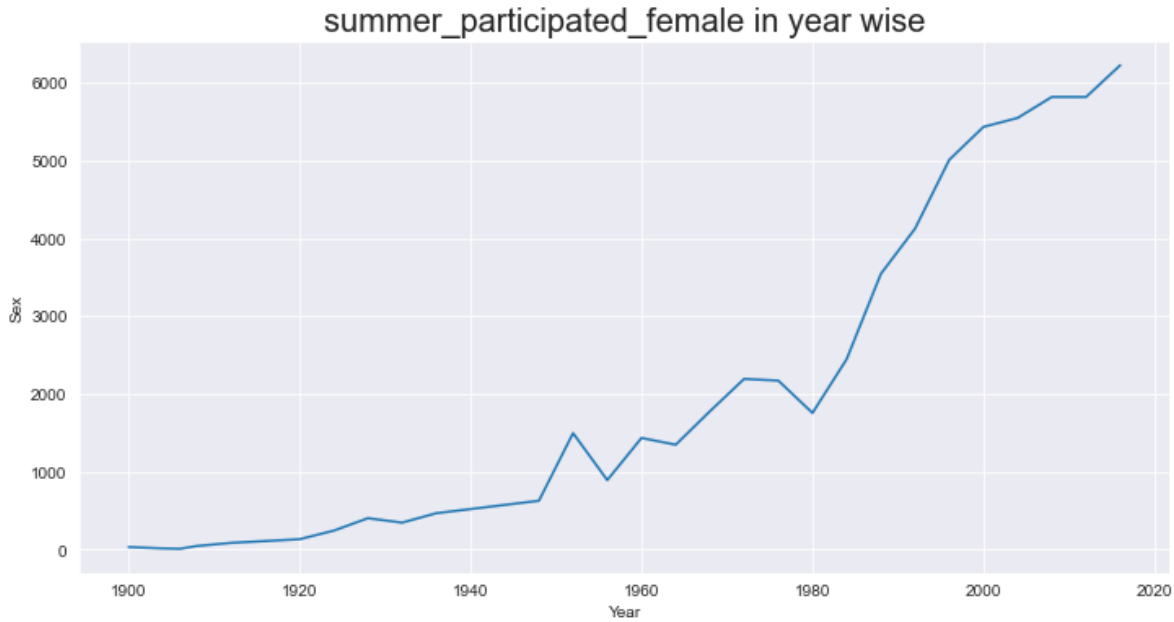
In [113]:
```
summer_participated_female
```

Out[113]:

Year	Sex
1900	33
1904	16
1906	11
1908	47
1912	87
1920	134
1924	244
1928	404
1932	347
1936	468
1948	628
1952	1497
1956	893
1960	1435
1964	1348
1968	1777
1972	2193
1976	2172
1980	1756
1984	2447
1988	3543
1992	4124
1996	5008
2000	5431
2004	5546
2008	5816
2012	5815
2016	6223

In [114]:
```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=summer_participated_female, x="Year", y="Sex",  err_style="bars", ci=
plt.title('summer_participated_female in year wise', fontsize=20)
plt.show()
```



Which countries WON the maximum Gold Medals in last held Olympic competitions

In [115]:
```
(df["Year"]==2016) & (df["Medal"]=="Gold")
```

Out[115]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [116]:
```
(df[(df["Year"]==2016) & (df["Medal"]=="Gold")])
```

Out[116]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seas
1080	608	Ahmad Abughaush	M	20	178.0	68.0	Jordan	JOR	2016 Summer	2016	Sumn
1495	832	Nicola Virginia Adams	F	33	164.0	51.0	Great Britain	GBR	2016 Summer	2016	Sumn
1849	1017	Nathan Ghar-Jun Adrian	M	27	198.0	100.0	United States	USA	2016 Summer	2016	Sumn
1850	1017	Nathan Ghar-Jun Adrian	M	27	198.0	100.0	United States	USA	2016 Summer	2016	Sumn
2453	1374	Tontowi Ahmad	M	29	179.0	72.0	Indonesia-1	INA	2016 Summer	2016	Sumn
...	...	...	...	...	...	...	...	...	...	...	
269277	134740	Zheng Shuyin	F	22	188.0	75.0	China	CHN	2016 Summer	2016	Sumn
269312	134759	Viktoriya Yuryevna Zhilinskayte	F	27	188.0	80.0	Russia	RUS	2016 Summer	2016	Sumn
269368	134783	Zhong Tianshi	F	25	168.0	62.0	China	CHN	2016 Summer	2016	Sumn
269511	134857	Zhu Ting	F	21	198.0	78.0	China	CHN	2016 Summer	2016	Sumn
270281	135205	Shakhobiddin Shokirovich Zoirov	M	23	169.0	52.0	Uzbekistan	UZB	2016 Summer	2016	Sumn

665 rows × 17 columns

In [117]:
```
team = df[(df["Year"]==2016) & (df["Medal"]=="Gold")][["Team"]]
```

In [118]:
```
team
```

Out[118]:

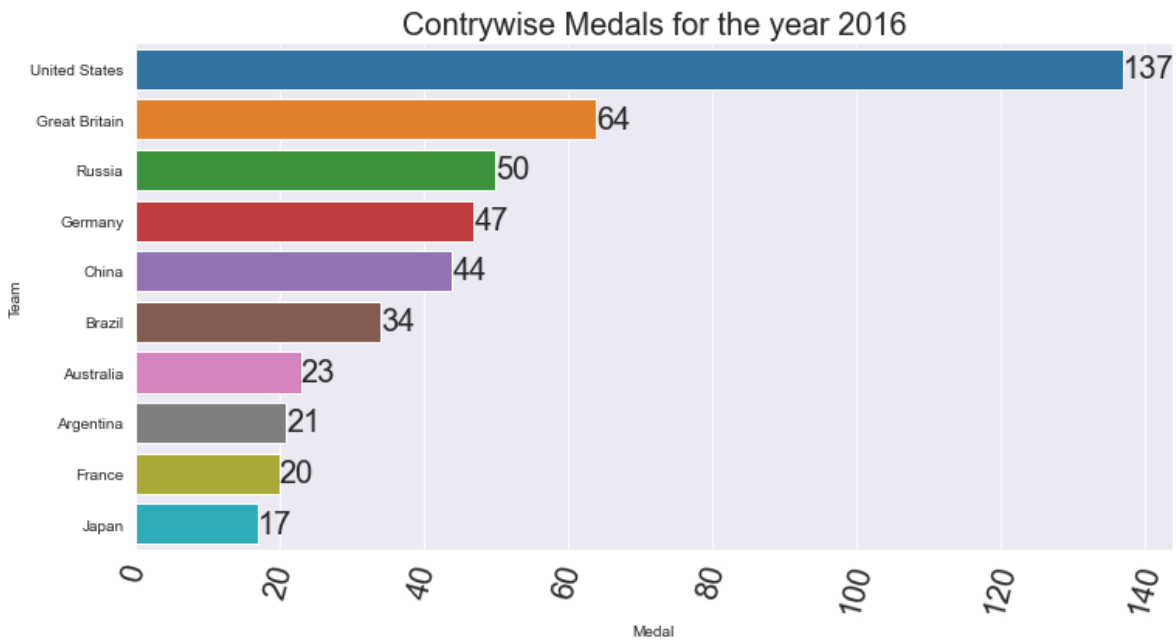
	Team
1080	Jordan
1495	Great Britain
1849	United States
1850	United States
2453	Indonesia-1
...	...
269277	China
269312	Russia
269368	China
269511	China
270281	Uzbekistan

In [119]:
```
data = team.value_counts().head(10).reset_index().rename(columns = {0:"Medal"}
data
```

Out[119]:

	Team	Medal
0	United States	137
1	Great Britain	64
2	Russia	50
3	Germany	47
4	China	44
5	Brazil	34
6	Australia	23
7	Argentina	21
8	France	20
9	Japan	17

In [120]:
```
plt.figure(figsize=(12,6))
plt.xticks(rotation=75,fontsize=20)
plt.title("Contrywise Medals for the year 2016",fontsize=20)
barplot = sns.barplot(x="Medal", y="Team", data = data)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
```



which sport fetch the most gold medals in last olympic held

In [121]:
```
(df["Year"]==2016) & (df["Medal"]=="Gold")
```

Out[121]:
```
0         False
1         False
2         False
3         False
4         False
          ...
271111    False
271112    False
271113    False
271114    False
271115    False
Length: 271116, dtype: bool
```

In [122]:
```
df[(df["Year"]==2016) & (df["Medal"]=="Gold")]
```

Out[122]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seas
1080	608	Ahmad Abughaush	M	20	178.0	68.0	Jordan	JOR	2016 Summer	2016	Sumn
1495	832	Nicola Virginia Adams	F	33	164.0	51.0	Great Britain	GBR	2016 Summer	2016	Sumn
1849	1017	Nathan Ghar-Jun Adrian	M	27	198.0	100.0	United States	USA	2016 Summer	2016	Sumn
1850	1017	Nathan Ghar-Jun Adrian	M	27	198.0	100.0	United States	USA	2016 Summer	2016	Sumn
2453	1374	Tontowi Ahmad	M	29	179.0	72.0	Indonesia-1	INA	2016 Summer	2016	Sumn
...	...	...	...	...	...	...	...	...	...	...	
269277	134740	Zheng Shuyin	F	22	188.0	75.0	China	CHN	2016 Summer	2016	Sumn
269312	134759	Viktoriya Yuryevna Zhilinskayte	F	27	188.0	80.0	Russia	RUS	2016 Summer	2016	Sumn
269368	134783	Zhong Tianshi	F	25	168.0	62.0	China	CHN	2016 Summer	2016	Sumn
269511	134857	Zhu Ting	F	21	198.0	78.0	China	CHN	2016 Summer	2016	Sumn
270281	135205	Shakhobiddin Shokirovich Zoirov	M	23	169.0	52.0	Uzbekistan	UZB	2016 Summer	2016	Sumn

665 rows × 17 columns

In [123]:
```
sports = df[(df["Year"]==2016) & (df["Medal"]=="Gold")][["Sport"]]
```

In [124]:
```
sports
```

Out[124]:

	Sport
1080	Taekwondo
1495	Boxing
1849	Swimming
1850	Swimming
2453	Badminton
...	...
269277	Taekwondo
269312	Handball
269368	Cycling
269511	Volleyball
270281	Boxing

665 rows × 1 columns

In [125]:
```
data1 = sports.value_counts().head(10).reset_index()
data1
```

Out[125]:

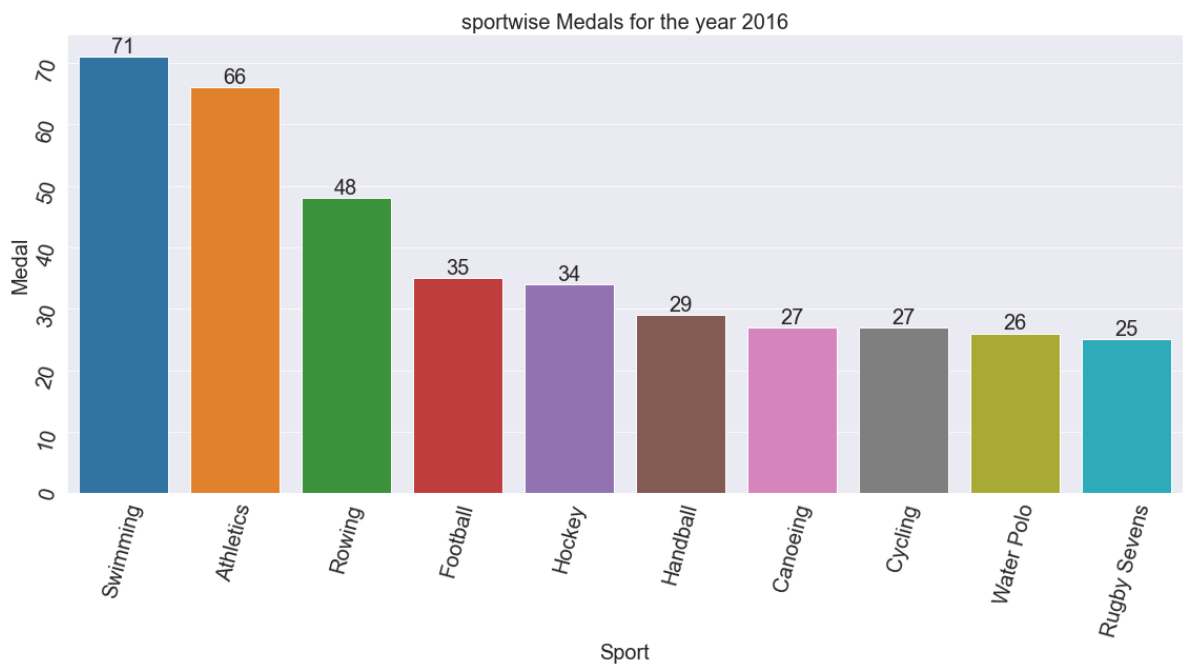
	Sport	0
0	Swimming	71
1	Athletics	66
2	Rowing	48
3	Football	35
4	Hockey	34
5	Handball	29
6	Canoeing	27
7	Cycling	27
8	Water Polo	26
9	Rugby Sevens	25

In [126]:
```
data1 = sports.value_counts().head(10).reset_index().rename(columns = {0:"Meda
data1
```

Out[126]:

	Sport	Medal
0	Swimming	71
1	Athletics	66
2	Rowing	48
3	Football	35
4	Hockey	34
5	Handball	29
6	Canoeing	27
7	Cycling	27
8	Water Polo	26
9	Rugby Sevens	25

In [127]:
```
plt.figure(figsize=(19,8))
barplot = sns.barplot(x = "Sport",y = "Medal", data = data1)
plt.xlabel("Sport",fontsize=20)
plt.ylabel("Medal",fontsize=20)
plt.xticks(rotation=75,fontsize=20)
plt.yticks(rotation=75,fontsize=20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.title("sportwise Medals for the year 2016",fontsize=20)
plt.show()
```



In [128]:
```
sports = df[(df["Year"]==2016) & (df["Medal"]=="Gold")][["Sport", "Team"]]
```

In [129]:
```
sports.value_counts().head(10).reset_index().rename(columns = {0:"Medal"})
```

Out[129]:

	Sport	Team	Medal
0	Swimming	United States	48
1	Athletics	United States	27
2	Basketball	United States	24
3	Football	Germany	18
4	Hockey	Argentina	18
5	Football	Brazil	17
6	Hockey	Great Britain	16
7	Rowing	Great Britain	15
8	Handball	Russia	15
9	Cycling	Great Britain	14

sports and event across winter and summer Olympics

In [130]:
```
df.columns
```

Out[130]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [131]:
```
DF = df[df["Season"] == "Summer"]
DF.head(1)
```

Out[131]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	S
0	1	A Dijiang	M	24	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Baske

In [132]:
```
DF["Sport"].nunique()
```

Out[132]:
```
52
```

In [133]:
```
df["Event"].nunique()
```

Out[133]: 765

In [134]:
```
DF1 = df[df["Season"]=="Winter"]
DF1.head()
```

Out[134]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City
4	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary
5	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary
6	5	Christine Jacoba Aaftink	F	25	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville
7	5	Christine Jacoba Aaftink	F	25	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville
8	5	Christine Jacoba Aaftink	F	27	185.0	82.0	Netherlands	NED	1994 Winter	1994	Winter	Lillehamme

In [135]:
```
DF1["Sport"].nunique()
```

Out[135]: 17

In [136]:
```
DF1["Event"].nunique()
```

Out[136]: 119

countries winning maximum medals per year

In [137]:
```
df.columns
```

Out[137]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [138]:
```
data2 = df["Medal"]!="None"
```

In [139]:
```
data2 = df[df["Medal"]!="None"]
```

In [140]:
```
data2
```

Out[140]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer
37	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
38	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
40	16	Juhamatti Tapio Aaltonen	M	28	184.00000	85.000000	Finland	FIN	2014 Winter
41	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer
...	...	...	...	...	...	...	...	...	...
271078	135553	Galina Ivanovna Zybina (-Fyodorova)	F	25	168.00000	80.000000	Soviet Union	URS	1956 Summer
271080	135553	Galina Ivanovna Zybina (-Fyodorova)	F	33	168.00000	80.000000	Soviet Union	URS	1964 Summer
271082	135554	Bogusaw Zych	M	28	182.00000	82.000000	Poland	POL	1980 Summer
271102	135563	Olesya Nikolayevna Zykina	F	19	171.00000	64.000000	Russia	RUS	2000 Summer
271103	135563	Olesya Nikolayevna Zykina	F	23	171.00000	64.000000	Russia	RUS	2004 Summer

39783 rows × 17 columns

In [141]:
```
Data1 =data2.groupby(['Team','Year'])['Medal'].count().reset_index()
```

In [142]:
```
Data1
```

Out[142]:

	Team	Year	Medal
0	A North American Team	1900	4
1	Afghanistan	2008	1
2	Afghanistan	2012	1
3	Algeria	1984	2
4	Algeria	1992	2
...	...	...	...
2007	Zambia	1996	1
2008	Zimbabwe	1980	15
2009	Zimbabwe	2004	3
2010	Zimbabwe	2008	4
2011	Zut	1908	3

2012 rows × 3 columns

In [143]:
```
data4 = Data1.sort_values(by = "Medal",ascending=False).head(10)
```

In [144]:
```
data4.sort_values(by = "Year")
```

Out[144]:

	Team	Year	Medal
1611	Soviet Union	1976	336
1612	Soviet Union	1980	488
512	East Germany	1980	287
1874	United States	1984	359
1614	Soviet Union	1988	352
1847	Unified Team	1992	271
1878	United States	1996	255
1882	United States	2004	259
1884	United States	2008	309
1888	United States	2016	256

Which sport is having most number of Gold Medals so far? (Top 5)

In [145]:
```
df["Medal"].value_counts()
```

Out[145]:
```
None      231333
Gold       13372
Bronze     13295
Silver     13116
Name: Medal, dtype: int64
```

In [146]:
```
Most_number_medal = df[["Sport","Medal"]]
Most_number_medal = Most_number_medal.value_counts("Sport").reset_index(name="
print(f"SPORTS HAVING MOST NUMBER OF MEDALS:\n")
Most_number_medal
```

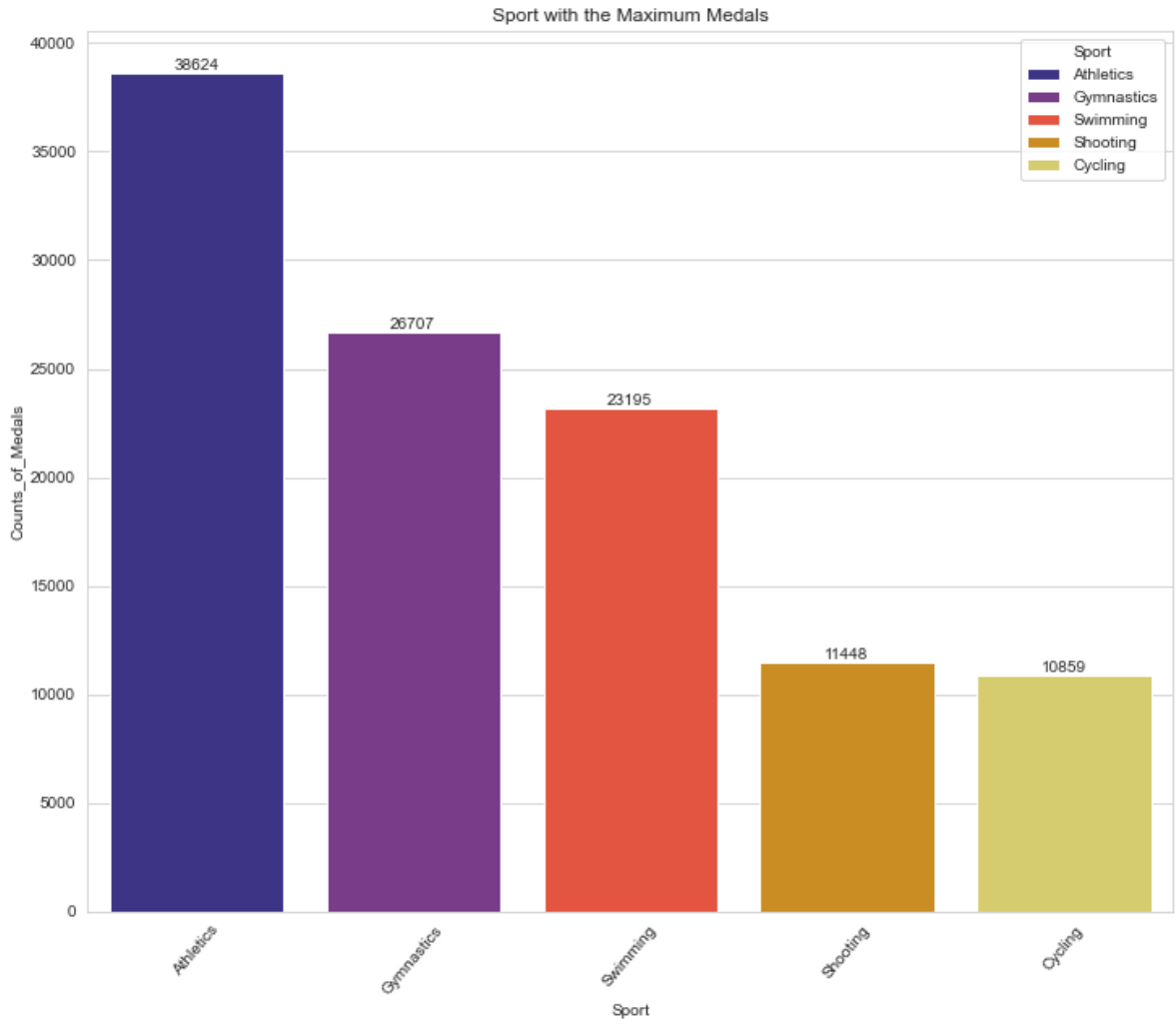
```
SPORTS HAVING MOST NUMBER OF MEDALS:
```

Out[146]:

	Sport	Counts_of_Medals
0	Athletics	38624
1	Gymnastics	26707
2	Swimming	23195
3	Shooting	11448
4	Cycling	10859

In [147]:
```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("Sport with the Maximum Medals")
barplot = sns.barplot(data=Most_number_medal, x="Sport", y="Counts_of_Medals",
barplot.set_xticklabels(rotation=50, labels=Most_number_medal.Sport)
plt.ylabel("Counts_of_Medals")
for bars in barplot.containers:
    barplot.bar_label(bars)
plt.show()
```



Which player has won most number of medals? (Top 5)

In [148]:
```
df.columns
```

Out[148]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [149]:
```
Individual_medal = df[["Name","Medal"]]
Individual_medal['Name'].unique()
Individual_medal = Individual_medal.value_counts("Name").reset_index(name="Cou
print(f"PLAYERS WHO WON MOST NUMBER OF MEDALS:\n")
Individual_medal
```

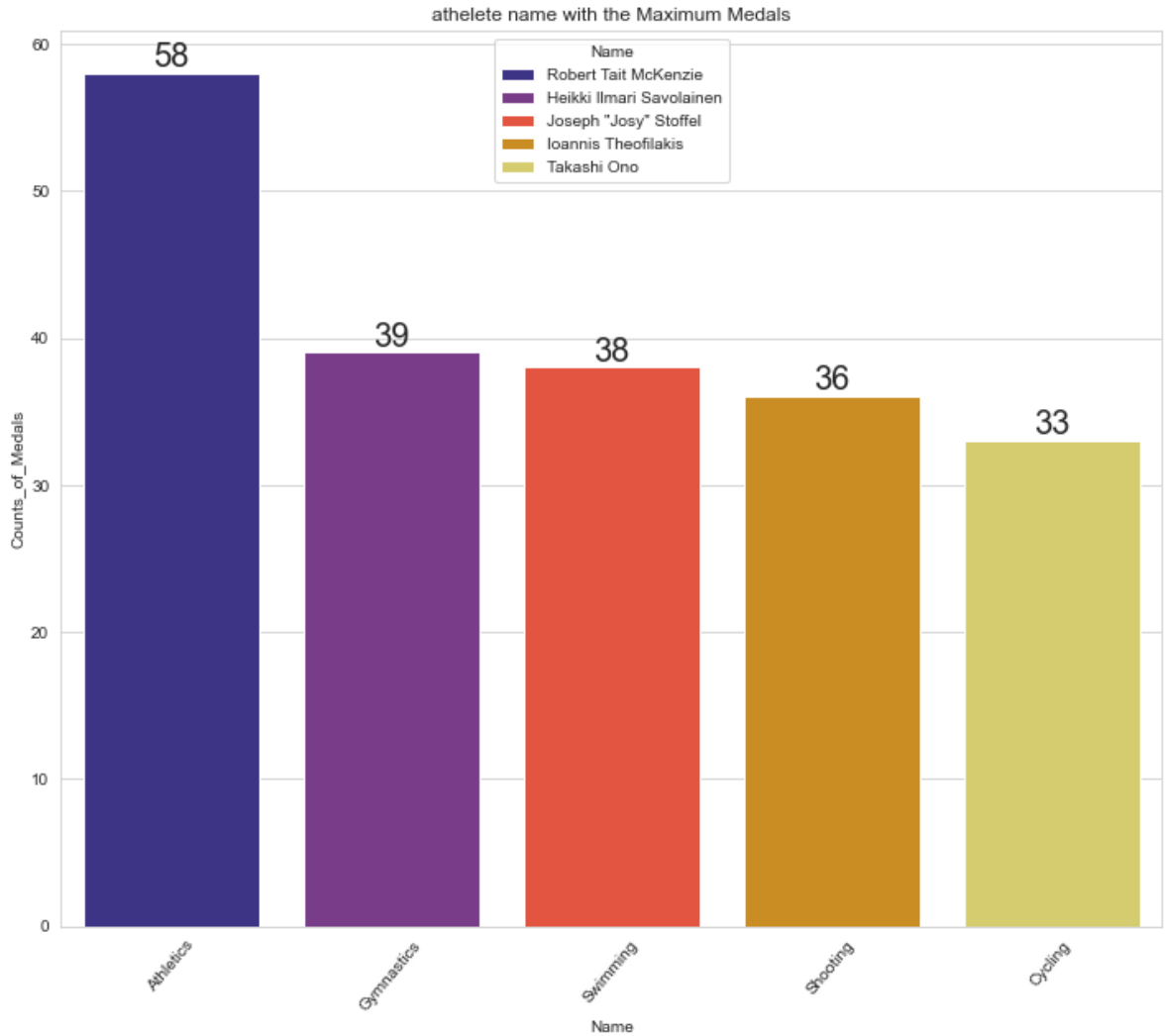
```
PLAYERS WHO WON MOST NUMBER OF MEDALS:
```

Out[149]:

	Name	Counts_of_Medals
0	Robert Tait McKenzie	58
1	Heikki Ilmari Savolainen	39
2	Joseph "Josy" Stoffel	38
3	Ioannis Theofilakis	36
4	Takashi Ono	33

In [150]:
```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("athelete name with the Maximum Medals")
barplot = sns.barplot(data=Individual_medal, x="Name", y="Counts_of_Medals", h
barplot.set_xticklabels(rotation=50, labels=Most_number_medal.Sport)
plt.ylabel("Counts_of_Medals")
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



Which player has won most number Gold Medals of medals? (Top 5)

In [151]:
```
Gold_medal = df[(df["Medal"] == "Gold")][["Name", "Medal"]]
```

In [152]:
```
Gold_medal.head()
```

Out[152]:

	Name	Medal
3	Edgar Lindenau Aabye	Gold
42	Paavo Johannes Aaltonen	Gold
44	Paavo Johannes Aaltonen	Gold
48	Paavo Johannes Aaltonen	Gold
60	Kjetil Andr Aamodt	Gold

In [153]:
```
Gold_medal = Gold_medal.value_counts(["Name"]).reset_index(name = "Counts_of_G
```

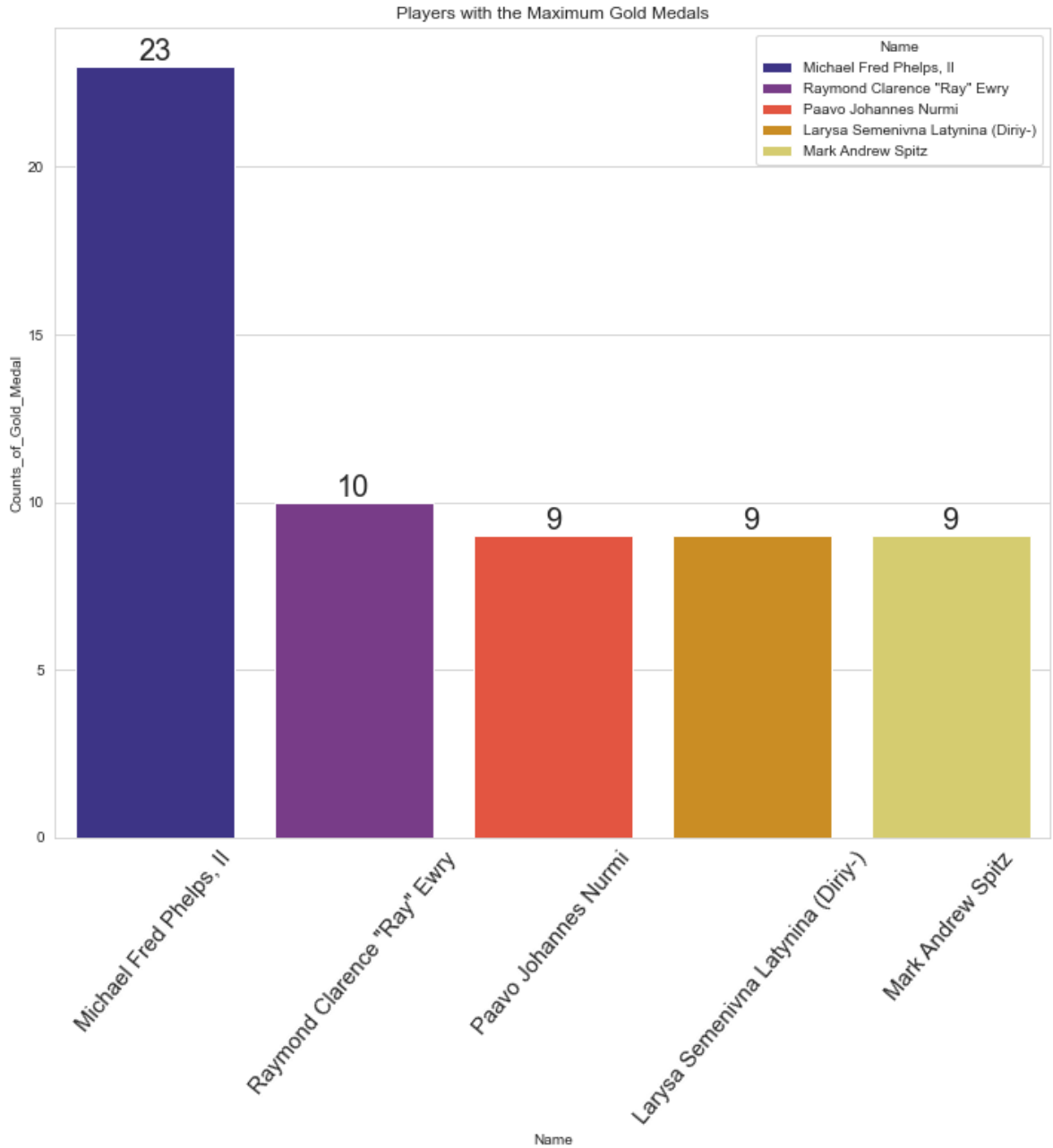
In [154]:
```
Gold_medal
```

Out[154]:

	Name	Counts_of_Gold_Medal
0	Michael Fred Phelps, II	23
1	Raymond Clarence "Ray" Ewry	10
2	Paavo Johannes Nurmi	9
3	Larysa Semenivna Latynina (Diriy-)	9
4	Mark Andrew Spitz	9

In [155]:
```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("Players with the Maximum Gold Medals")
barplot = sns.barplot(data=Gold_medal, x="Name", y="Counts_of_Gold_Medal", hue
barplot.set_xticklabels(rotation=50, labels=Gold_medal.Name, fontsize = 15)
plt.ylabel("Counts_of_Gold_Medal")
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



In which year India won first Gold Medal in Summer Olympics?

In [156]:
```
gold_dataset= df[(df["Medal"]=="Gold") & (df["Team"]=="India") & (df["Season"]
gold_dataset
```

Out[156]:

	Year	Team	Sport
4732	1928	India	Hockey
4736	1964	India	Hockey
5032	1928	India	Hockey
5033	1932	India	Hockey
5034	1936	India	Hockey
...	...	...	...
233749	1980	India	Hockey
236445	1932	India	Hockey
236446	1936	India	Hockey
250689	1948	India	Hockey
266934	1928	India	Hockey

131 rows × 3 columns

In [157]:
```
India_goldmedals=gold_dataset.sort_values(by="Year", ascending=True)
print(f"INDIA WON FIRST GOLD MEDAL IN SUMMER OLYMPICS:\n")
India_goldmedals.head(1)
```

```
INDIA WON FIRST GOLD MEDAL IN SUMMER OLYMPICS:
```

Out[157]:

	Year	Team	Sport
4732	1928	India	Hockey

In which year India won first Gold Medal in winter Olympics?

In [158]:
```
gold_dataset= df[(df["Medal"]=="Gold") & (df["Team"]=="India") & (df["Season"]
gold_dataset.head()
```

Out[158]:

	Year	Team	Sport
218235	1924	India	Alpinism
218239	1924	India	Alpinism
218240	1924	India	Alpinism
218241	1924	India	Alpinism
218242	1924	India	Alpinism

In [159]:
```
India_goldmedals=gold_dataset.sort_values(by="Year", ascending=True)
print(f"INDIA WON FIRST GOLD MEDAL IN SUMMER OLYMPICS:\n")
India_goldmedals.head(1)
```

```
INDIA WON FIRST GOLD MEDAL IN SUMMER OLYMPICS:
```

Out[159]:

	Year	Team	Sport
218235	1924	India	Alpinism

find total number of medal per sport by india

In [160]:
```
gold_dataset
```

Out[160]:

	Year	Team	Sport
218235	1924	India	Alpinism
218239	1924	India	Alpinism
218240	1924	India	Alpinism
218241	1924	India	Alpinism
218242	1924	India	Alpinism
218243	1924	India	Alpinism
218244	1924	India	Alpinism

In [161]:
```
total_number_of_medal = df[(df["Team"]=="India") & (df["Medal"]!="None")]
```

In [162]:
```
total_number_of_medal.head(1)
```

Out[162]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
4732	2699	Shaukat Ali	M	30	175.33897	70.702393	India	IND	1928 Summer	1928	Summer	Am

In [163]:
```
total_number_of_medal_sport = total_number_of_medal.value_counts("Sport").rese
```

In [164]:
```
total_number_of_medal_sport
```

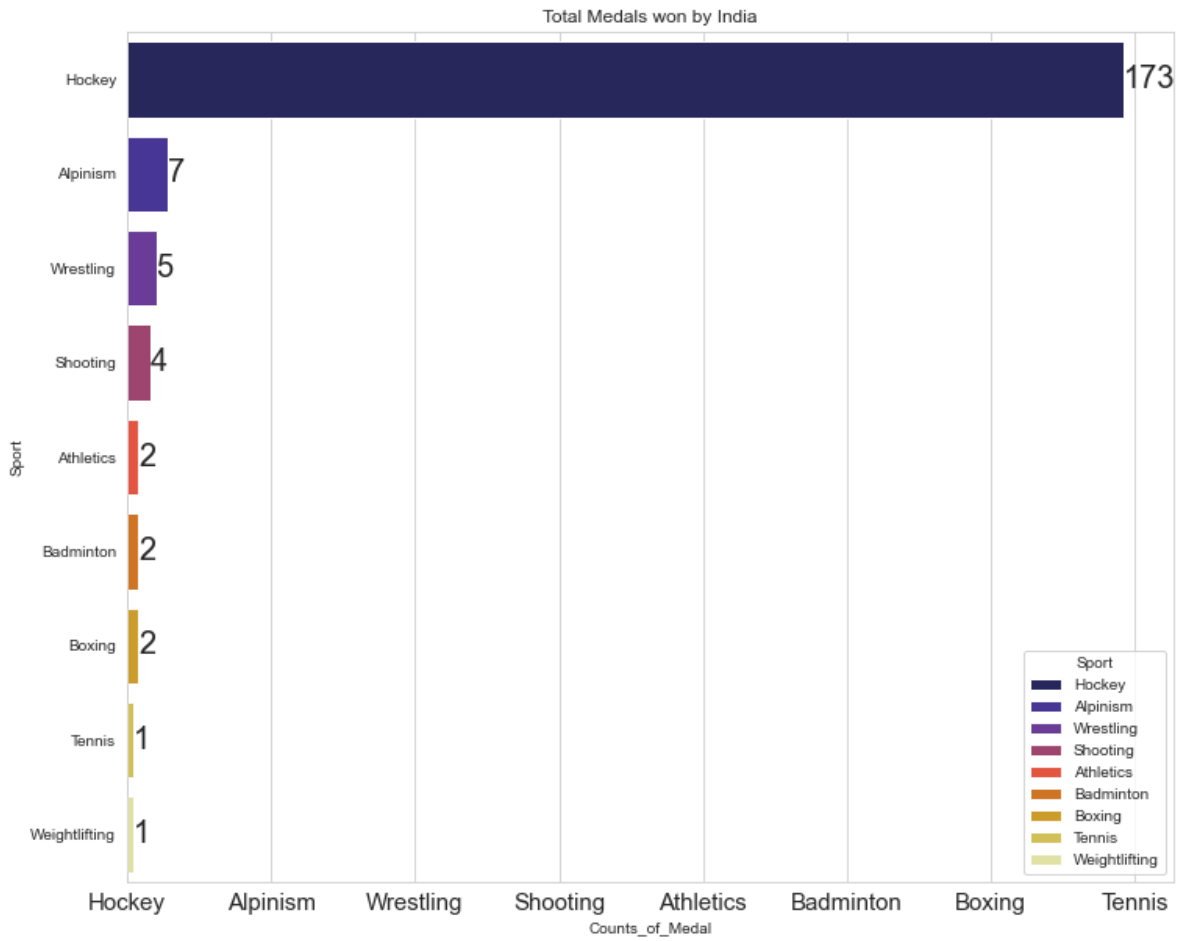
Out[164]:

	Sport	Counts_of_Medal
0	Hockey	173
1	Alpinism	7
2	Wrestling	5
3	Shooting	4
4	Athletics	2
5	Badminton	2
6	Boxing	2
7	Tennis	1
8	Weightlifting	1

In [165]:

```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("Total Medals won by India")
barplot = sns.barplot(data=total_number_of_medal_sport, y="Sport", x="Counts_o
barplot.set_xticklabels(rotation=0, labels=total_number_of_medal_sport.Sport,f
plt.xlabel("Counts_of_Medal")
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



In [166]:

```
total_number_of_medal = df[(df["Team"]=="India") & (df["Medal"]=="Gold")]
total_number_of_medal_sport = total_number_of_medal.value_counts("Sport").rese
total_number_of_medal_sport
```

Out[166]:

	Sport	Counts_of_Medal
0	Hockey	130
1	Alpinism	7
2	Shooting	1

Which event is most popular in terms on number of players? (Top 5)

In [167]:

```
Number_of_players = df[["Event","Name"]]
Number_of_players = Number_of_players.value_counts("Event").reset_index(name="
print(f"MOST POPULAR EVENT IN TERMS ON NUMBER OF PLAYERS:\n")
Number_of_players
```

```
MOST POPULAR EVENT IN TERMS ON NUMBER OF PLAYERS:
```

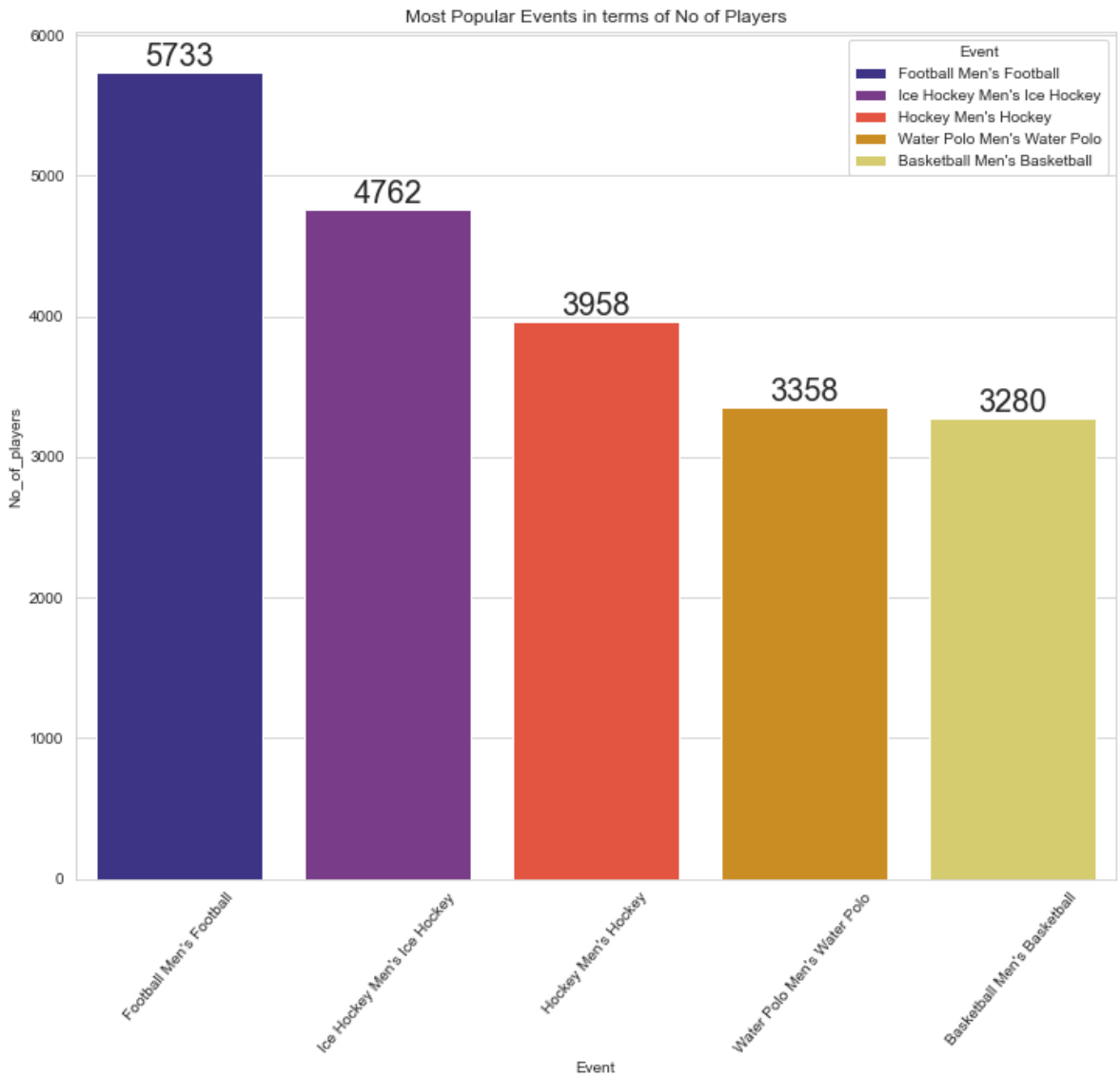
Out[167]:

	Event	No_of_players
0	Football Men's Football	5733
1	Ice Hockey Men's Ice Hockey	4762
2	Hockey Men's Hockey	3958
3	Water Polo Men's Water Polo	3358
4	Basketball Men's Basketball	3280

In [168]:

```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("Most Popular Events in terms of No of Players")
barplot = sns.barplot(data=Number_of_players, x="Event", y="No_of_players", hu
barplot.set_xticklabels(rotation=50, labels=Number_of_players.Event)
plt.ylabel("No_of_players")
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



Which sport is having most female Gold Medalists? (Top 5)

In [169]:
```
Female_gold_medalists = df[(df["Medal"] == "Gold") & (df["Sex"] == "F")][["Spo
Female_gold_medalists = Female_gold_medalists.value_counts(["Sport"]).reset_in
print(f"\nFEMALE GOLD MEDALISTS IN SUMMER OLYMPICS:\n")
Female_gold_medalists
```

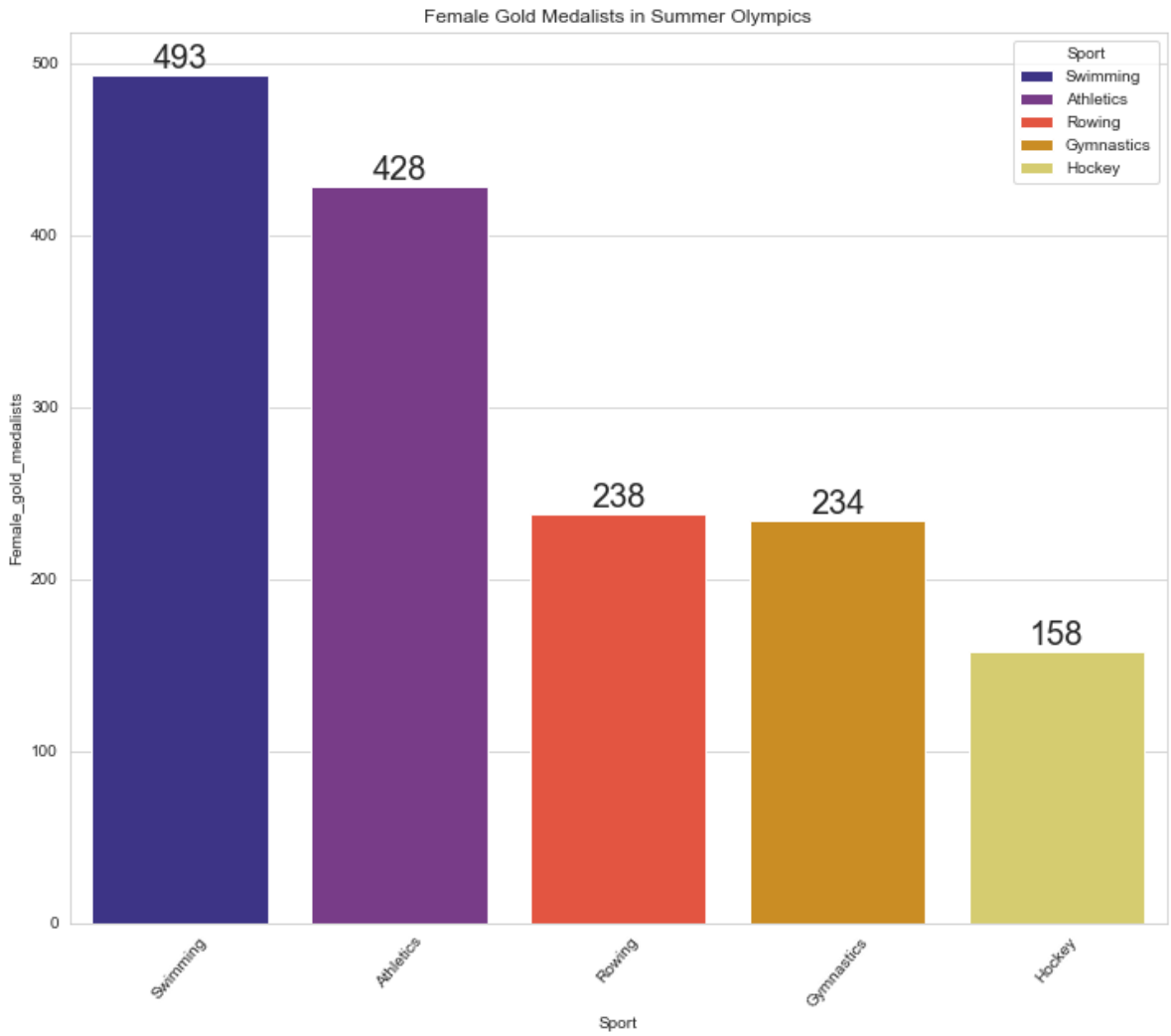
```
FEMALE GOLD MEDALISTS IN SUMMER OLYMPICS:
```

Out[169]:

	Sport	Female_gold_medalists
0	Swimming	493
1	Athletics	428
2	Rowing	238
3	Gymnastics	234
4	Hockey	158

In [170]:
```
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 10)

plt.title("Female Gold Medalists in Summer Olympics")
barplot = sns.barplot(data=Female_gold_medalists, x="Sport", y="Female_gold_me
barplot.set_xticklabels(rotation=50, labels=Female_gold_medalists.Sport)
plt.ylabel("Female_gold_medalists")
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



In [171]:
```
Gender_by_Medals = df[df["Medal"]!="None"]
```

In [172]:
```
Gender_by_Medals
```

Out[172]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer
37	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
38	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
40	16	Juhamatti Tapio Aaltonen	M	28	184.00000	85.000000	Finland	FIN	2014 Winter
41	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer
...	...	...	...	...	...	...	...	...	...
271078	135553	Galina Ivanovna Zybina (-Fyodorova)	F	25	168.00000	80.000000	Soviet Union	URS	1956 Summer
271080	135553	Galina Ivanovna Zybina (-Fyodorova)	F	33	168.00000	80.000000	Soviet Union	URS	1964 Summer
271082	135554	Bogusaw Zych	M	28	182.00000	82.000000	Poland	POL	1980 Summer
271102	135563	Olesya Nikolayevna Zykina	F	19	171.00000	64.000000	Russia	RUS	2000 Summer
271103	135563	Olesya Nikolayevna Zykina	F	23	171.00000	64.000000	Russia	RUS	2004 Summer

39783 rows × 17 columns

In [173]:
```
Gender_by_Medals = pd.crosstab(Gender_by_Medals["Sex"],Gender_by_Medals["Medal
Gender_by_Medals
```

Out[173]:

Medal	Bronze	Gold	Silver
Sex			
F	3771	3747	3735
M	9524	9625	9381

In [174]:
```
india_by_Medals = df[(df["Team"]=="India") & (df["Medal"]!="None")]
india_by_Medals = pd.crosstab(india_by_Medals["Team"],india_by_Medals["Medal"]
india_by_Medals
```

Out[174]:

Medal	Bronze	Gold	Silver
Team			
India	40	138	19

sport wise with maximum medals

In [175]:
```
test1=df.groupby(['Team','Year'])['Medal'].count().reset_index()
```

In [176]:
```
test1.sort_values(by = "Medal",ascending=False)
```

Out[176]:

	Team	Year	Medal
4841	United States	1992	936
4840	United States	1988	886
4803	Unified Team	1992	832
4843	United States	1996	827
4821	United States	1904	823
...	...	...	...
3640	Pirouette-31	1900	1
1682	Gabon	1972	1
4159	Solos Carex	2008	1
3644	Plume-patte-32	1900	1
3609	Pettit-Poucet-6	1900	1

5170 rows × 3 columns

In [177]:
```
df.columns
```

Out[177]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [178]:
```
df2 = df[df["Medal"]!="None"]
```

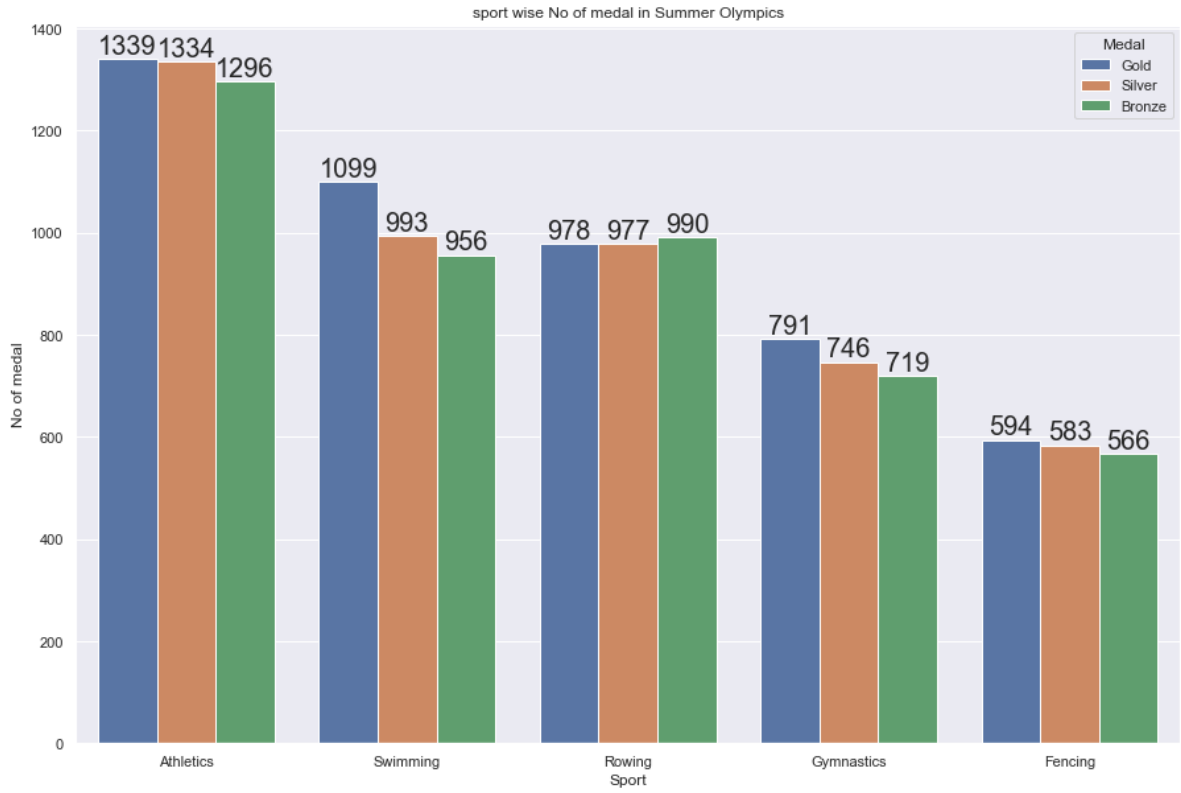
In [179]:
```
df3 = df2.groupby(['Sport','Medal'])[['Medal']].value_counts().sort_values(asc
```

In [180]:
```
df3
```

Out[180]:

	Sport	Medal	No of medal
0	Athletics	Gold	1339
1	Athletics	Silver	1334
2	Athletics	Bronze	1296
3	Swimming	Gold	1099
4	Swimming	Silver	993
5	Rowing	Bronze	990
6	Rowing	Gold	978
7	Rowing	Silver	977
8	Swimming	Bronze	956
9	Gymnastics	Gold	791
10	Gymnastics	Silver	746
11	Gymnastics	Bronze	719
12	Fencing	Gold	594
13	Fencing	Silver	583
14	Fencing	Bronze	566

In [181]:
```
plt.figure(figsize=(15,10))
sns.set_theme(style="darkgrid")
barplot = sns.barplot(x = "Sport",y = "No of medal", hue=df3["Medal"], data =
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.title("sport wise No of medal in Summer Olympics")
plt.show()
```



What is the number of Sports held per year in Summer Olypics, since 1896?

In [182]:
```
data3 = df[(df["Season"]=="Summer") & (df["Medal"]!="None")]
data3
```

Out[182]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer
37	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
38	15	Arvo Ossian Aaltonen	M	30	175.33897	70.702393	Finland	FIN	1920 Summer
41	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer
42	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer
...	...	...	...	...	...	...	...	...	...
271078	135553	Galina Ivanovna Zybina (-Fyodorova)	F	25	168.00000	80.000000	Soviet Union	URS	1956 Summer
271080	135553	Galina Ivanovna Zybina (-Fyodorova)	F	33	168.00000	80.000000	Soviet Union	URS	1964 Summer
271082	135554	Bogusaw Zych	M	28	182.00000	82.000000	Poland	POL	1980 Summer
271102	135563	Olesya Nikolayevna Zykina	F	19	171.00000	64.000000	Russia	RUS	2000 Summer
271103	135563	Olesya Nikolayevna Zykina	F	23	171.00000	64.000000	Russia	RUS	2004 Summer

34088 rows × 17 columns

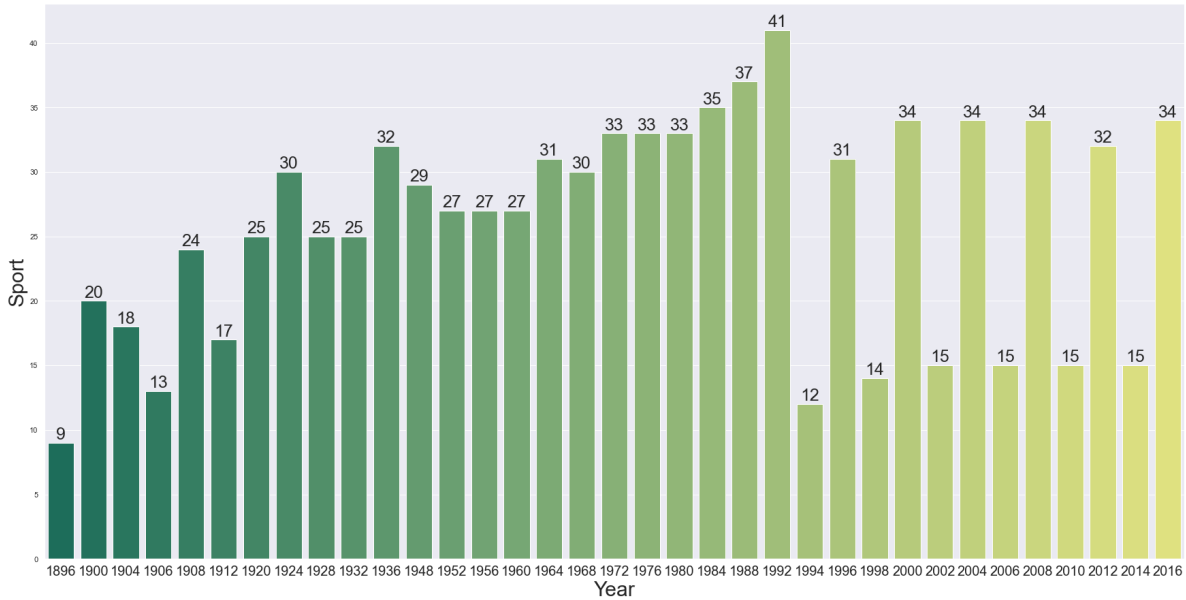
In [183]:
```
df_sports = df.groupby("Year")["Sport"].nunique().sort_values(ascending=False)
```

In [184]:
```
df_sports
```

Out[184]:

	Year	Sport
0	1992	41
1	1988	37
2	1984	35
3	2016	34
4	2008	34
5	2004	34
6	2000	34
7	1976	33
8	1980	33
9	1972	33
10	1936	32
11	2012	32
12	1964	31
13	1996	31
14	1968	30
15	1924	30
16	1948	29
17	1952	27
18	1956	27
19	1960	27
20	1932	25
21	1928	25
22	1920	25
23	1908	24
24	1900	20
25	1904	18
26	1912	17
27	2002	15
28	2006	15
29	2010	15
30	2014	15
31	1998	14
32	1906	13
33	1994	12
34	1896	9

In [185]:
```
fig = plt.figure(figsize=(30,15))
barplot = sns.barplot(x = "Year", y = "Sport", data=df_sports, palette="summer
plt.xlabel("Year", fontsize = 30)
plt.ylabel("Sport", fontsize = 30)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 25)
plt.xticks(fontsize = 20)
plt.show()
```



How many cities hosted the Summer Olympics games, since 1896?

In [186]:
```
df_medal_holders = df[(df.Medal !='None') & (df.Season == 'Summer')]

df_medal_holders_above50 = df_medal_holders[df.Age >= 50]

df_medal_holders_above50.head(10)
```

Out[186]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seas
3680	2112	Abdullah Al-Rashidi	M	52	183.00000	83.000000	Individual Olympic Athletes	IOA	2016 Summer	2016	Summ
5077	2894	Derek Swithin Allhusen	M	54	180.00000	71.000000	Great Britain	GBR	1968 Summer	1968	Summ
5078	2894	Derek Swithin Allhusen	M	54	180.00000	71.000000	Great Britain	GBR	1968 Summer	1968	Summ
7961	4404	Johan August Anker	M	56	175.33897	70.702393	Norna	NOR	1928 Summer	1928	Summ
13393	7272	Nikolaus "Klaus" Balkenhol	M	52	178.00000	83.000000	Germany	GER	1992 Summer	1992	Summ
13394	7272	Nikolaus "Klaus" Balkenhol	M	52	178.00000	83.000000	Germany	GER	1992 Summer	1992	Summ
13396	7272	Nikolaus "Klaus" Balkenhol	M	56	178.00000	83.000000	Germany	GER	1996 Summer	1996	Summ
14364	7744	Ernest Barberolle	M	58	156.00000	56.000000	France	FRA	1920 Summer	1920	Summ
17552	9349	Ludger Beerbaum	M	52	190.00000	85.000000	Germany	GER	2016 Summer	2016	Summ
21999	11599	Rudolf Georg Binding	M	60	175.33897	70.702393	Germany	GER	1928 Summer	1928	Summ

In [187]:
```
df.columns
```

Out[187]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [188]:
```
data4 = df[df["Season"]=="Summer"]
```

In [189]:
```
data4
```

Out[189]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Y
0	1	A Dijiang	M	24	180.00000	80.000000	China	CHN	1992 Summer	19
1	2	A Lamusi	M	23	170.00000	60.000000	China	CHN	2012 Summer	20
2	3	Gunnar Nielsen Aaby	M	24	175.33897	70.702393	Denmark	DEN	1920 Summer	19
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer	19
26	8	Cornelia "Cor" Aalten (-Strannood)	F	18	168.00000	70.702393	Netherlands	NED	1932 Summer	19
...	...	...	...	...	...	...	...	...	...	
271106	135565	Fernando scar Zylberberg	M	27	168.00000	76.000000	Argentina	ARG	2004 Summer	20
271107	135566	James Francis "Jim" Zylker	M	21	175.00000	75.000000	United States	USA	1972 Summer	19
271108	135567	Aleksandr Viktorovich Zyuzin	M	24	183.00000	72.000000	Russia	RUS	2000 Summer	20
271109	135567	Aleksandr Viktorovich Zyuzin	M	28	183.00000	72.000000	Russia	RUS	2004 Summer	20
271110	135568	Olga Igorevna Zyuzkova	F	33	171.00000	69.000000	Belarus	BLR	2016 Summer	20

222552 rows × 17 columns

In [190]:
```
data5 = data4["City"].nunique()
```

In [191]:
```
data5
```

Out[191]: 23

How many cities in which Olympics was conducted more than once?

In [192]:
```
data4.groupby(["City"])["Year"].nunique().sort_values(ascending=False).reset_i
```

Out[192]:

	City	Conduct no of year
0	Athina	3
1	London	3
2	Stockholm	2
3	Los Angeles	2
4	Paris	2
5	Amsterdam	1
6	Moskva	1
7	Sydney	1
8	St. Louis	1
9	Seoul	1
10	Roma	1
11	Rio de Janeiro	1
12	Munich	1
13	Mexico City	1
14	Montreal	1
15	Antwerpen	1
16	Melbourne	1
17	Helsinki	1
18	Berlin	1
19	Beijing	1
20	Barcelona	1
21	Atlanta	1
22	Tokyo	1

How many cities hosted the winter Olympics games, since 1896?

In [193]:
```
data6 = df[df["Season"]=="Winter"]
```

In [194]:
```
data6
```

Out[194]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season
4	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter
5	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter
6	5	Christine Jacoba Aaftink	F	25	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter
7	5	Christine Jacoba Aaftink	F	25	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter
8	5	Christine Jacoba Aaftink	F	27	185.0	82.0	Netherlands	NED	1994 Winter	1994	Winter
...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter
271112	135570	Piotr ya	M	27	176.0	59.0	Poland	POL	2014 Winter	2014	Winter
271113	135570	Piotr ya	M	27	176.0	59.0	Poland	POL	2014 Winter	2014	Winter
271114	135571	Tomasz Ireneusz ya	M	30	185.0	96.0	Poland	POL	1998 Winter	1998	Winter
271115	135571	Tomasz Ireneusz ya	M	34	185.0	96.0	Poland	POL	2002 Winter	2002	Winter

48564 rows × 17 columns

In [195]:
```
data7 = data6["City"].nunique()
```

In [196]:
```
data7
```

Out[196]: 19

How many cities in which Olympics was conducted more than once?

In [197]:
```
data6.groupby(["City"])["Year"].nunique().sort_values(ascending=False).reset_i
```

Out[197]:

	City	Conduct no of year
0	Innsbruck	2
1	Lake Placid	2
2	Sankt Moritz	2
3	Albertville	1
4	Salt Lake City	1
5	Torino	1
6	Squaw Valley	1
7	Sochi	1
8	Sarajevo	1
9	Sapporo	1
10	Nagano	1
11	Oslo	1
12	Calgary	1
13	Lillehammer	1
14	Grenoble	1
15	Garmisch-Partenkirchen	1
16	Cortina d'Ampezzo	1
17	Chamonix	1
18	Vancouver	1

How many medals were won by women in Summer Olympics?

In [198]:
```
df.columns
```

Out[198]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [199]:
```
Dt = df[(df["Sex"]=="F") & (df["Season"]=="Summer") & (df["Medal"]!="None")]
```

In [200]:
```
Dt
```

Out[200]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year
79	21	Ragnhild Margrethe Aamodt	F	27	163.00000	70.702393	Norway	NOR	2008 Summer	2008
91	29	Willemien Aardenburg	F	22	175.33897	70.702393	Netherlands	NED	1988 Summer	1988
105	37	Ann Kristin Aarnes	F	23	182.00000	64.000000	Norway	NOR	1996 Summer	1996
161	65	Patimat Abakarova	F	21	165.00000	49.000000	Azerbaijan	AZE	2016 Summer	2016
163	67	Mariya Vasilyevna Abakumova (-Tarabina)	F	22	179.00000	80.000000	Russia	RUS	2008 Summer	2008
...	...	...	...	...	...	...	...	...	...	...
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21	168.00000	80.000000	Soviet Union	URS	1952 Summer	1952
271078	135553	Galina Ivanovna Zybina (-Fyodorova)	F	25	168.00000	80.000000	Soviet Union	URS	1956 Summer	1956
271080	135553	Galina Ivanovna Zybina (-Fyodorova)	F	33	168.00000	80.000000	Soviet Union	URS	1964 Summer	1964
271102	135563	Olesya Nikolayevna Zykina	F	19	171.00000	64.000000	Russia	RUS	2000 Summer	2000
271103	135563	Olesya Nikolayevna Zykina	F	23	171.00000	64.000000	Russia	RUS	2004 Summer	2004

9442 rows × 17 columns

In [201]:
```
Dt1 = Dt.value_counts("Medal")
```

In [202]:
```
Dt1
```

Out[202]:
```
Medal
Bronze    3174
Gold      3140
Silver    3128
dtype: int64
```

How many medals were won by women in winter Olympics?

In [203]:
```
Dt = df[(df["Sex"]=="F") & (df["Season"]=="Winter") & (df["Medal"]!="None")]
```

In [204]:
```
Dt
```

Out[204]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season
974	548	Yekaterina Konstantinovna Abramova	F	23	167.0	65.0	Russia	RUS	2006 Winter	2006	Winter
1210	680	Rita Nikolayevna Achkina	F	30	168.0	63.0	Soviet Union	URS	1968 Winter	1968	Winter
1519	840	Victoria "Vicki" Adams	F	24	164.0	69.0	Great Britain	GBR	2014 Winter	2014	Winter
2126	1181	Meghan Christina Agosta (-Marciano)	F	18	168.0	67.0	Canada	CAN	2006 Winter	2006	Winter
2127	1181	Meghan Christina Agosta (-Marciano)	F	23	168.0	67.0	Canada	CAN	2010 Winter	2010	Winter
...	...	...	...	...	...	...	...	...	...	...	...
269977	135066	Katia Zini	F	24	164.0	57.0	Italy	ITA	2006 Winter	2006	Winter
269984	135067	Mara Zini	F	26	169.0	53.0	Italy	ITA	2006 Winter	2006	Winter
270174	135163	Luiza Zotkowska	F	23	160.0	52.0	Poland	POL	2010 Winter	2010	Winter
270178	135163	Luiza Zotkowska	F	27	160.0	52.0	Poland	POL	2014 Winter	2014	Winter
270480	135290	Vera Zozua (-Pyatnizin)	F	24	157.0	69.0	Soviet Union	URS	1980 Winter	1980	Winter

1811 rows × 17 columns

In [205]:
```
Dt1 = Dt.value_counts("Medal")
```

In [206]:
```
Dt1
```

Out[206]:
```
Medal
Gold      607
Silver    607
Bronze    597
dtype: int64
```

Top 10 Women Athletes - Gold Medals

In [207]:
```
data7 = df[(df["Sex"]=="F") & (df["Medal"]=="Gold")]
```

In [208]:
```
data7
```

Out[208]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year
79	21	Ragnhild Margrethe Aamodt	F	27	163.00000	70.702393	Norway	NOR	2008 Summer	2008
283	150	Margaret Ives Abbott (-Dunne)	F	23	175.33897	70.702393	United States	USA	1900 Summer	1900
1494	832	Nicola Virginia Adams	F	29	164.00000	51.000000	Great Britain	GBR	2012 Summer	2012
1495	832	Nicola Virginia Adams	F	33	164.00000	51.000000	Great Britain	GBR	2016 Summer	2016
1527	846	Valerie Kasanita Adams-Vili (-Price)	F	23	193.00000	120.000000	New Zealand	NZL	2008 Summer	2008
...	...	...	...	...	...	...	...	...	...	...
270588	135331	Olha Valentynivna Zubareva	F	22	182.00000	90.000000	Soviet Union	URS	1980 Summer	1980
270934	135488	Nataliya Vladimirovna Zuyeva	F	19	176.00000	62.000000	Russia	RUS	2008 Summer	2008
270970	135501	Ellina Aleksandrovna Zvereva (Kisheyeva-)	F	39	183.00000	100.000000	Belarus	BLR	2000 Summer	2000
271009	135520	Julia Zwehl	F	28	167.00000	60.000000	Germany	GER	2004 Summer	2004
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21	168.00000	80.000000	Soviet Union	URS	1952 Summer	1952

3747 rows × 17 columns

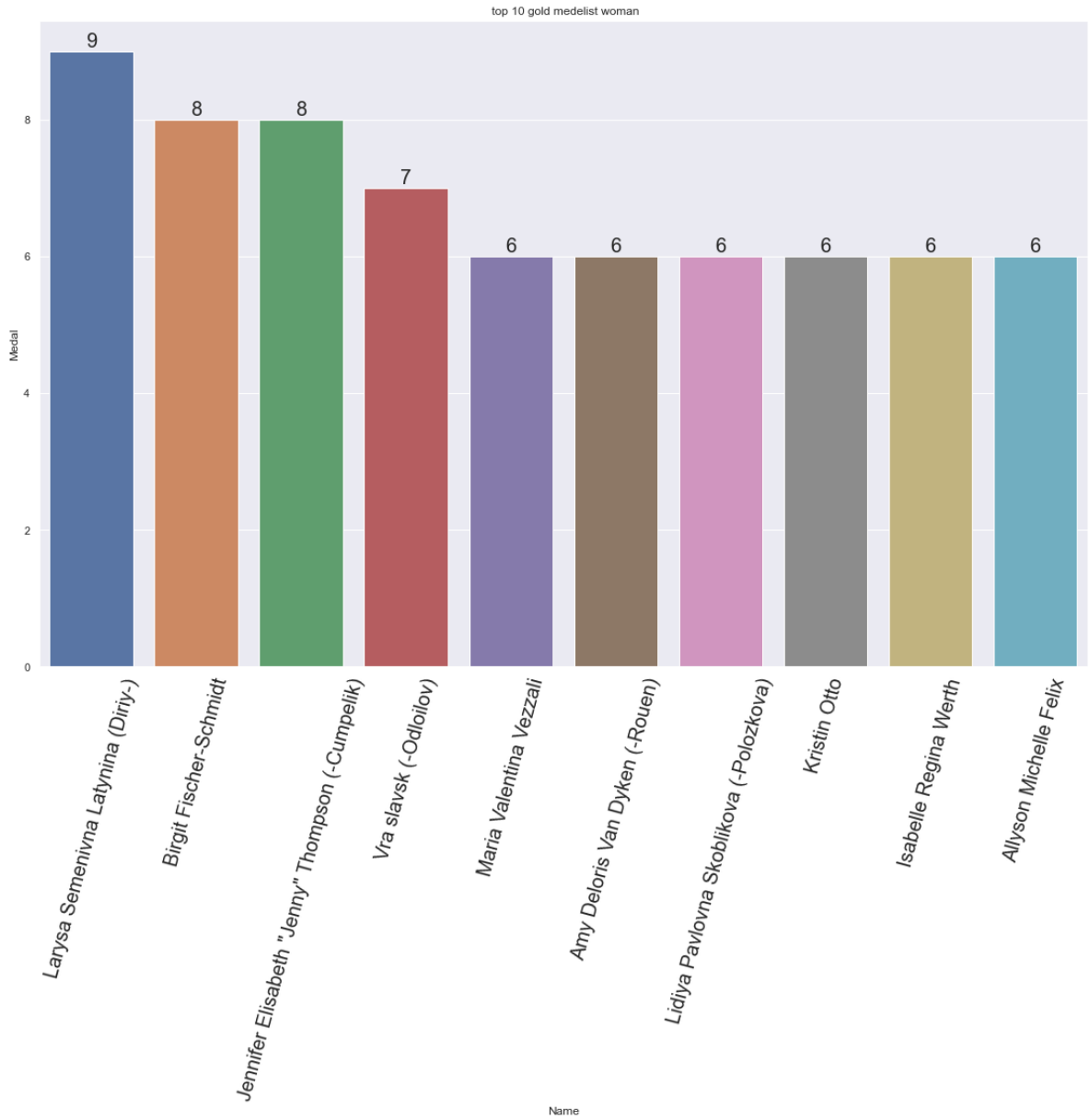
In [209]:
```
data8 = data7.groupby(['Name'])['Medal'].count().sort_values(ascending=False).
```

In [210]:
```
data8
```

Out[210]:

	Name	Medal
0	Larysa Semenivna Latynina (Diriy-)	9
1	Birgit Fischer-Schmidt	8
2	Jennifer Elisabeth "Jenny" Thompson (-Cumpelik)	8
3	Vra slavsk (-Odloilov)	7
4	Maria Valentina Vezzali	6
5	Amy Deloris Van Dyken (-Rouen)	6
6	Lidiya Pavlovna Skoblikova (-Polozkova)	6
7	Kristin Otto	6
8	Isabelle Regina Werth	6
9	Allyson Michelle Felix	6

In [211]:
```
plt.figure(figsize=(19,12))
barplot = sns.barplot(x = "Name",y = "Medal", data = data8 )
plt.xticks(rotation=75,fontsize=20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.title(" top 10 gold medelist woman")
plt.show()
```



Top 10 Women Athletes - Total Medals

In [212]:
```
df.columns
```

Out[212]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [213]:
```
data9 = df[(df["Sex"]=="F") & (df["Medal"]!="None")]
```

In [214]:
```
data9
```

Out[214]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year
79	21	Ragnhild Margrethe Aamodt	F	27	163.00000	70.702393	Norway	NOR	2008 Summer	2008
91	29	Willemien Aardenburg	F	22	175.33897	70.702393	Netherlands	NED	1988 Summer	1988
105	37	Ann Kristin Aarnes	F	23	182.00000	64.000000	Norway	NOR	1996 Summer	1996
161	65	Patimat Abakarova	F	21	165.00000	49.000000	Azerbaijan	AZE	2016 Summer	2016
163	67	Mariya Vasilyevna Abakumova (-Tarabina)	F	22	179.00000	80.000000	Russia	RUS	2008 Summer	2008
...	...	...	...	...	...	...	...	...	...	...
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21	168.00000	80.000000	Soviet Union	URS	1952 Summer	1952
271078	135553	Galina Ivanovna Zybina (-Fyodorova)	F	25	168.00000	80.000000	Soviet Union	URS	1956 Summer	1956
271080	135553	Galina Ivanovna Zybina (-Fyodorova)	F	33	168.00000	80.000000	Soviet Union	URS	1964 Summer	1964
271102	135563	Olesya Nikolayevna Zykina	F	19	171.00000	64.000000	Russia	RUS	2000 Summer	2000
271103	135563	Olesya Nikolayevna Zykina	F	23	171.00000	64.000000	Russia	RUS	2004 Summer	2004

11253 rows × 17 columns

In [215]:
```
data10 = data9.groupby(["Name"])["Medal"].count().reset_index()
```

In [216]:
```
data11 = data10.sort_values(by = "Medal", ascending=False)[:10]
```

In [217]:
```
data11
```

Out[217]:

	Name	Medal
3672	Larysa Semenivna Latynina (Diriy-)	18
2896	Jennifer Elisabeth "Jenny" Thompson (-Cumpelik)	12
5054	Natalie Anne Coughlin (-Hall)	12
1393	Dara Grace Torres (-Hoffman, -Minas)	12
772	Birgit Fischer-Schmidt	12
6893	Vra slavsk (-Odloilov)	11
2001	Franziska van Almsick	10
5636	Raisa Petrovna Smetanina	10
6218	Stefania Belmondo	10
2726	Isabelle Regina Werth	10

How many athlete participated in each sport in Summer Olympics?

In [218]:
```
dff1 = df[df["Season"]=="Summer"].groupby("Sport")["Name"].nunique().reset_ind
```

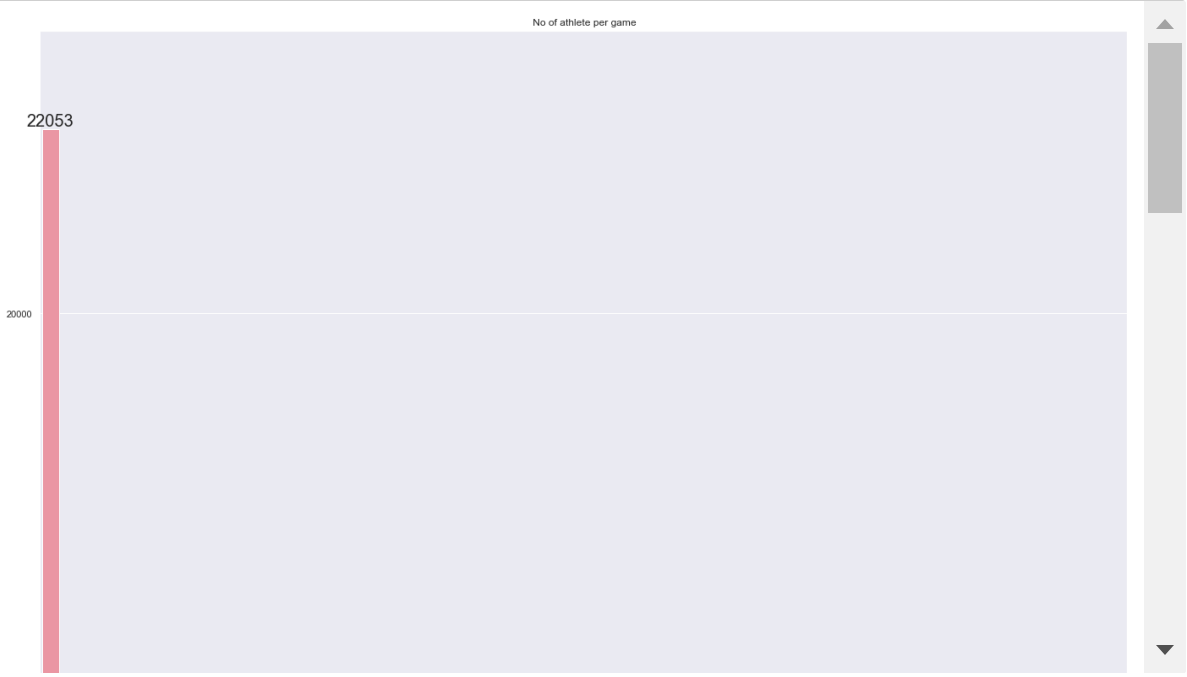
In [219]:
```
dff1
```

Out[219]:

	Sport	No of athlete per game
4	Athletics	22053
40	Swimming	8761
34	Rowing	7684
19	Football	6161
14	Cycling	5819
10	Boxing	5254
51	Wrestling	4987
38	Shooting	4879
37	Sailing	4480
21	Gymnastics	4132
17	Fencing	4118
23	Hockey	3813
7	Basketball	3412
11	Canoeing	3204
50	Weightlifting	2880
26	Judo	2721
22	Handball	2699
49	Water Polo	2599
48	Volleyball	2502
16	Equestrianism	2345
3	Art Competitions	1814
15	Diving	1466
44	Tennis	1246
2	Archery	1113
28	Modern Pentathlon	864
5	Badminton	811
6	Baseball	761
42	Table Tennis	749
32	Rhythmic Gymnastics	567
41	Synchronized Swimming	549
43	Taekwondo	470
9	Beach Volleyball	383
39	Softball	367
46	Triathlon	355
36	Rugby Sevens	299
20	Golf	218

	Sport	No of athlete per game
47	Tug-Of-War	160
35	Rugby	155
45	Trampolining	93
30	Polo	87
27	Lacrosse	60
24	Ice Hockey	60
18	Figure Skating	45
12	Cricket	24
29	Motorboating	14
25	Jeu De Paume	11
13	Croquet	10
31	Racquets	7
1	Alpinism	4
33	Roque	4
8	Basque Pelota	2
0	Aeronautics	1

In [220]:
```
plt.figure(figsize=(23,45))
barplot = sns.barplot(x = "Sport",y = "No of athlete per game", data = dff1 )
plt.xticks(rotation=90,fontsize=20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.title(" No of athlete per game")
plt.show()
```



In [221]:
```
df[df["Season"]=="Summer"].groupby(["Year","Sport"])["Name"].count().reset_ind
```

Out[221]:

	Year	Sport	No of athlete per game
639	2016	Athletics	2508
505	2000	Athletics	2468
234	1952	Gymnastics	2391
474	1996	Athletics	2386
607	2012	Athletics	2278
573	2008	Athletics	2244
539	2004	Athletics	2175
418	1988	Athletics	2062
445	1992	Athletics	2054
595	2008	Swimming	1749

Top 5 countries to win most Gold medals across all seasons collectively and particular country USA

In [222]:
```
data11 = df[(df['Medal'] == 'Gold')][['Team']].value_counts().reset_index(name
```

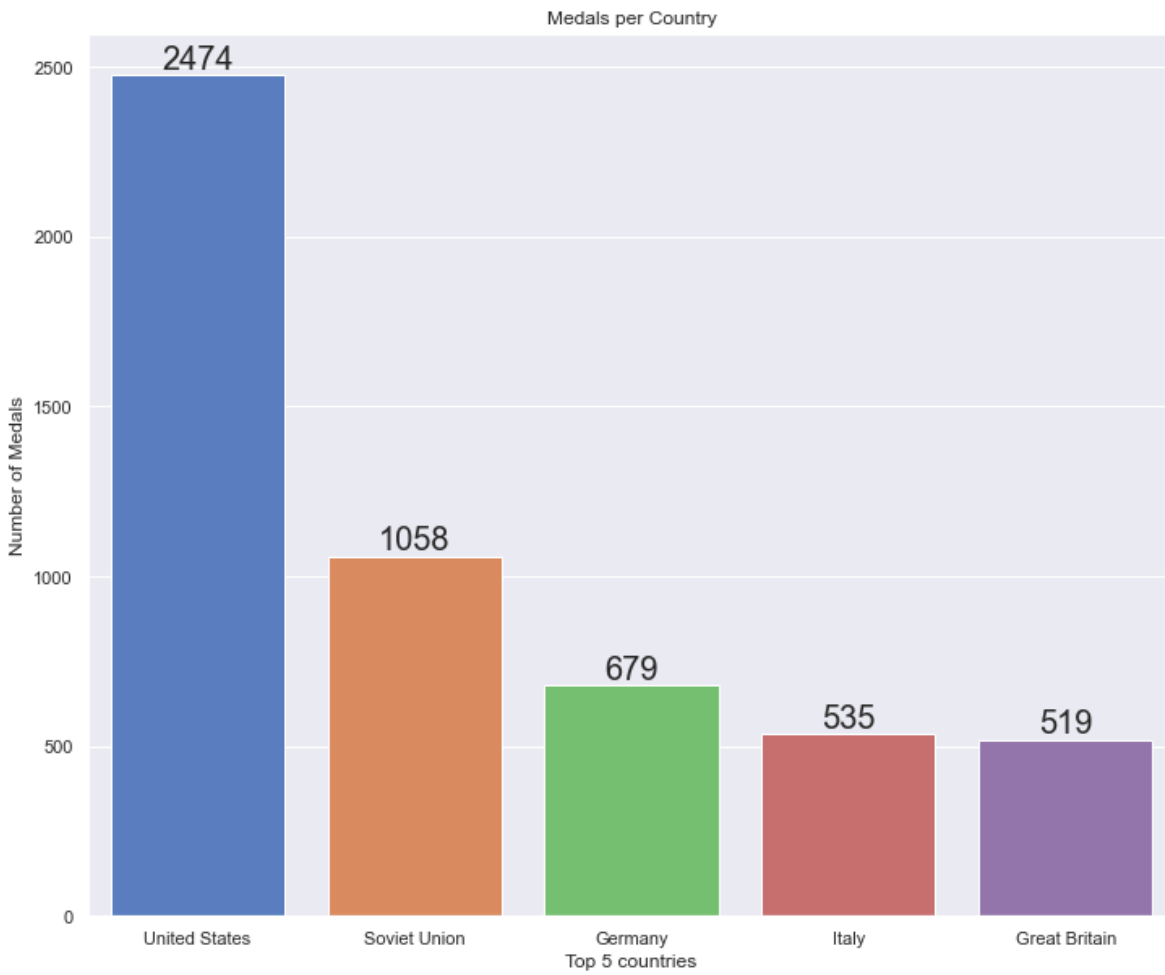
In [223]:
```
data11
```

Out[223]:

	Team	no of gold Medal
0	United States	2474
1	Soviet Union	1058
2	Germany	679
3	Italy	535
4	Great Britain	519

In [224]:
```
g = sns.barplot(x="Team", y="no of gold Medal", data=data11,
                palette="muted")
plt.xlabel("Top 5 countries")
plt.ylabel("Number of Medals")
for bars in g.containers:
    g.bar_label(bars,fontsize = 20)
plt.title('Medals per Country')
```

Out[224]:
```
Text(0.5, 1.0, 'Medals per Country')
```



In [225]:
```
data12 = df[(df['Medal'] == 'Gold')][['Team',"Year"]].value_counts().reset_ind
```

In [226]:
```
data12
```

Out[226]:

	Team	Year	no of gold Medal
0	Soviet Union	1980	201
1	United States	1984	190
2	Soviet Union	1988	168
3	United States	1996	157
4	Soviet Union	1976	148
...	...	...	...
1189	Jordan	2016	1
1190	Japan	2014	1
1191	Japan	2006	1
1192	Japan	1952	1
1193	Zimbabwe	2008	1

1194 rows × 3 columns

In [227]:
```
data12[data12["Team"]=="United States"]
```

Out[227]:

	Team	Year	no of gold Medal
1	United States	1984	190
3	United States	1996	157
6	United States	2012	139
7	United States	2016	137
8	United States	2000	128
11	United States	2008	121
12	United States	2004	115
13	United States	1920	111
15	United States	1968	100
16	United States	1960	97
17	United States	1964	96
22	United States	1992	92
23	United States	1924	92
25	United States	1988	89
28	United States	1948	82
29	United States	1952	77
30	United States	1976	73
32	United States	1972	72
33	United States	1932	71
38	United States	1904	65
41	United States	1956	61
46	United States	1936	51
54	United States	1928	48
59	United States	1912	46
87	United States	1908	34
121	United States	1998	25
134	United States	1980	24
188	United States	1900	18
276	United States	1906	12
299	United States	1896	11
338	United States	2002	9
341	United States	2006	9
384	United States	2014	8
388	United States	2010	8
477	United States	1994	6

In [228]:
```
data12[data12["Team"]=="India"]
```

Out[228]:

	Team	Year	no of gold Medal
162	India	1948	20
175	India	1936	19
193	India	1956	17
210	India	1980	16
217	India	1964	15
229	India	1932	15
249	India	1928	14
253	India	1952	14
398	India	1924	7
1161	India	2008	1

Top 5 countries to win most Gold medals in 2012

In [229]:
```
df.columns
```

Out[229]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

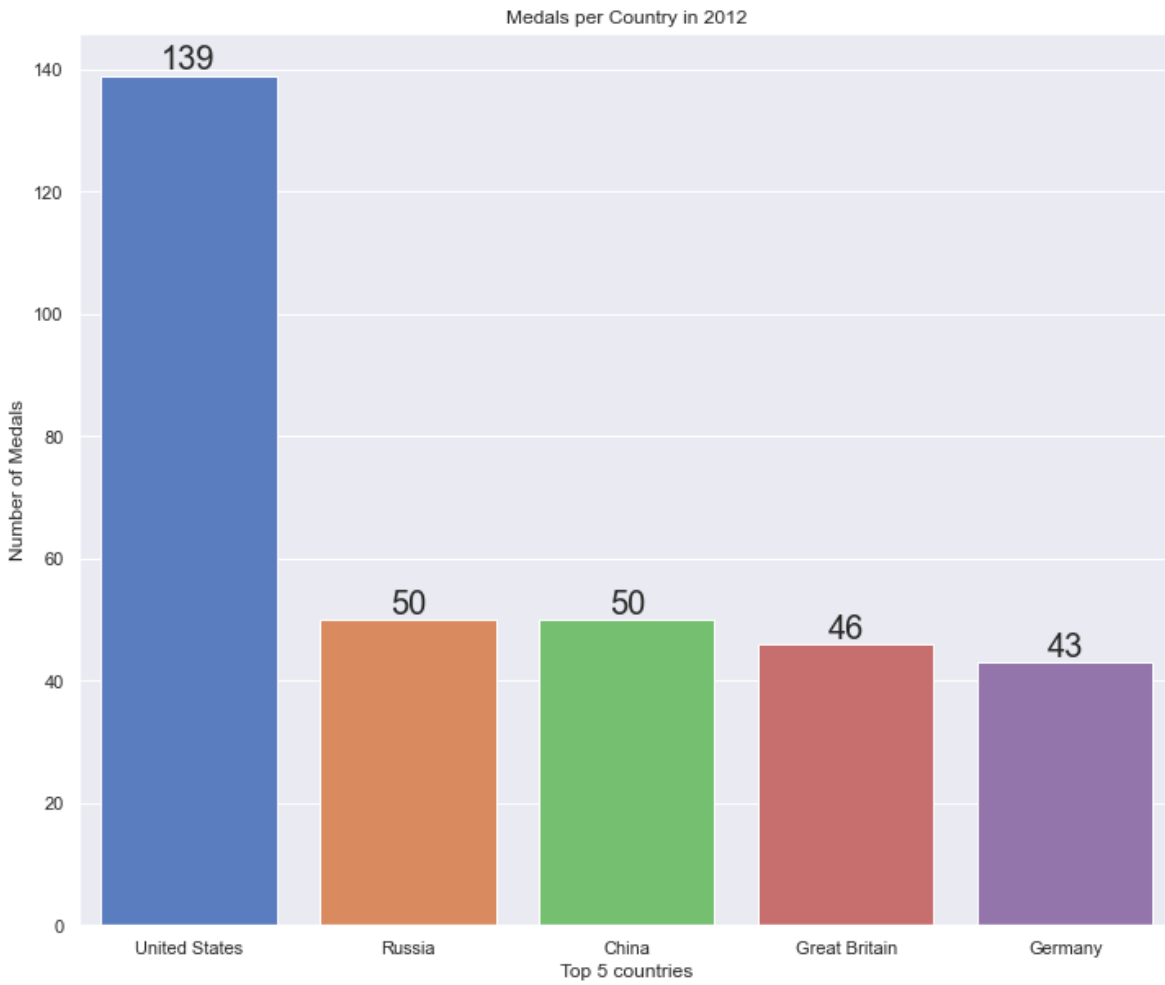
In [230]:
```
data13 = df[(df["Medal"]=="Gold") & (df["Year"]==2012)]["Team"].value_counts()
data13
```

Out[230]:

	index	No of gold medal
0	United States	139
1	Russia	50
2	China	50
3	Great Britain	46
4	Germany	43

In [231]:
```
g = sns.barplot(x="index", y="No of gold medal", data=data13,
                palette="muted")
plt.xlabel("Top 5 countries")
plt.ylabel("Number of Medals")
for bars in g.containers:
    g.bar_label(bars,fontsize = 20)
plt.title('Medals per Country in 2012')
```

Out[231]:
```
Text(0.5, 1.0, 'Medals per Country in 2012')
```



event with the greatest number of Gold Medals

In [232]:
```
df.columns
```

Out[232]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [233]:
```
data14 = df[df["Medal"]=="Gold"]
```

In [234]:
```
data14
```

Out[234]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Y
3	4	Edgar Lindenau Aabye	M	34	175.33897	70.702393	Denmark/Sweden	DEN	1900 Summer	1
42	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer	1
44	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer	1
48	17	Paavo Johannes Aaltonen	M	28	175.00000	64.000000	Finland	FIN	1948 Summer	1
60	20	Kjetil Andr Aamodt	M	20	176.00000	85.000000	Norway	NOR	1992 Winter	1
...	...	...	...	...	...	...	...	...	...	
270981	135503	Zurab Zviadauri	M	23	182.00000	90.000000	Georgia	GEO	2004 Summer	2
271009	135520	Julia Zwehl	F	28	167.00000	60.000000	Germany	GER	2004 Summer	2
271016	135523	Ronald Ferdinand "Ron" Zwerver	M	29	200.00000	93.000000	Netherlands	NED	1996 Summer	1
271049	135545	Henk Jan Zwolle	M	31	197.00000	93.000000	Netherlands	NED	1996 Summer	1
271076	135553	Galina Ivanovna Zybina (-Fyodorova)	F	21	168.00000	80.000000	Soviet Union	URS	1952 Summer	1

13372 rows × 17 columns

In [235]:
```
data15 = data14.groupby("Event")["Medal"].count().reset_index()
```

In [236]:
```
data16 = data15.sort_values(by = "Medal",ascending=False)[:5]
```

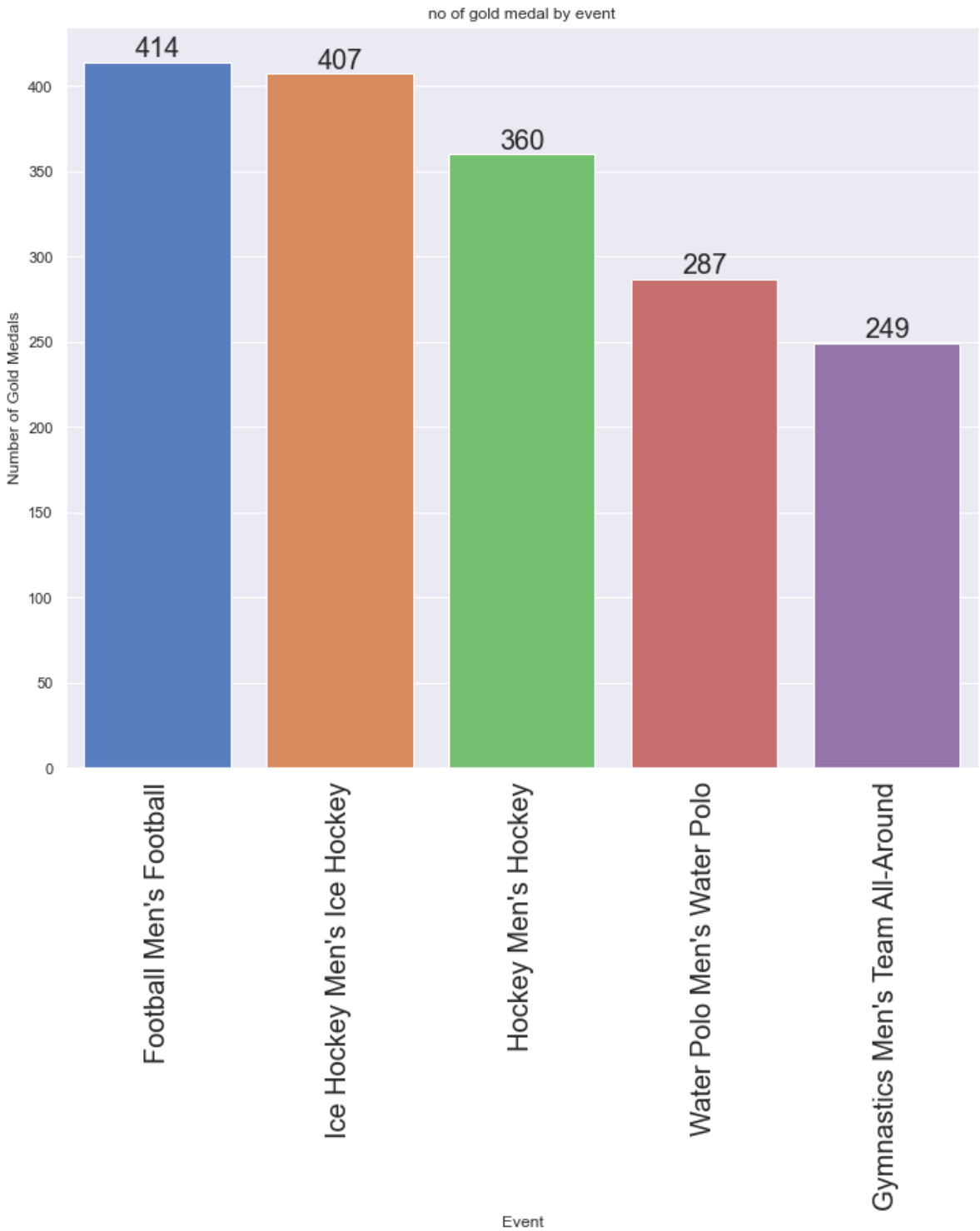
In [237]:
```
data16
```

Out[237]:

	Event	Medal
336	Football Men's Football	414
382	Ice Hockey Men's Ice Hockey	407
380	Hockey Men's Hockey	360
698	Water Polo Men's Water Polo	287
367	Gymnastics Men's Team All-Around	249

In [238]:
```
barplot = sns.barplot(x="Event", y="Medal", data=data16,
          palette="muted")
plt.xlabel("Event")
plt.ylabel("Number of Gold Medals")
plt.xticks(rotation=90,fontsize=20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.title('no of gold medal by event')
```

Out[238]: Text(0.5, 1.0, 'no of gold medal by event')



How many athlete participate in each Summer Olympic?

In [239]:
```
df.columns
```

Out[239]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

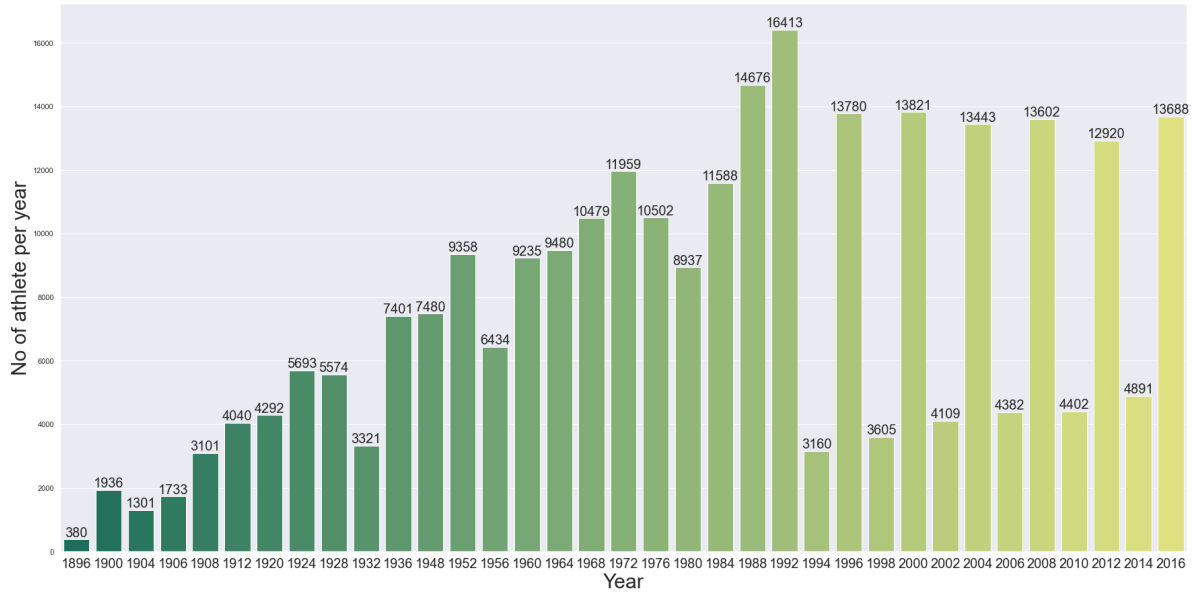
In [240]:
```
data17 = df.groupby("Year")["Name"].count().reset_index(name = "No of athlete
```

In [241]:
```
data17
```

Out[241]:

	Year	No of athlete per year
0	1896	380
1	1900	1936
2	1904	1301
3	1906	1733
4	1908	3101
5	1912	4040
6	1920	4292
7	1924	5693
8	1928	5574
9	1932	3321
10	1936	7401
11	1948	7480
12	1952	9358
13	1956	6434
14	1960	9235
15	1964	9480
16	1968	10479
17	1972	11959
18	1976	10502
19	1980	8937
20	1984	11588
21	1988	14676
22	1992	16413
23	1994	3160
24	1996	13780
25	1998	3605
26	2000	13821
27	2002	4109
28	2004	13443
29	2006	4382
30	2008	13602
31	2010	4402
32	2012	12920
33	2014	4891
34	2016	13688

In [242]:
```
fig = plt.figure(figsize=(30,15))
barplot = sns.barplot(x = "Year", y = "No of athlete per year", data=data17, p
plt.xlabel("Year", fontsize = 30)
plt.ylabel("No of athlete per year", fontsize = 30)
plt.xticks(fontsize = 20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



Top 10 countries with total medals in Summer Olympic games

In [243]:
```
data18 = df[(df["Medal"]!="None") & (df["Season"]=="Summer")][["Team","Medal"]
```

In [244]:
```
data18
```

Out[244]:

	Team	Medal
3	Denmark/Sweden	Gold
37	Finland	Bronze
38	Finland	Bronze
41	Finland	Bronze
42	Finland	Gold
...	...	...
271078	Soviet Union	Silver
271080	Soviet Union	Bronze
271082	Poland	Bronze
271102	Russia	Bronze
271103	Russia	Silver

34088 rows × 2 columns

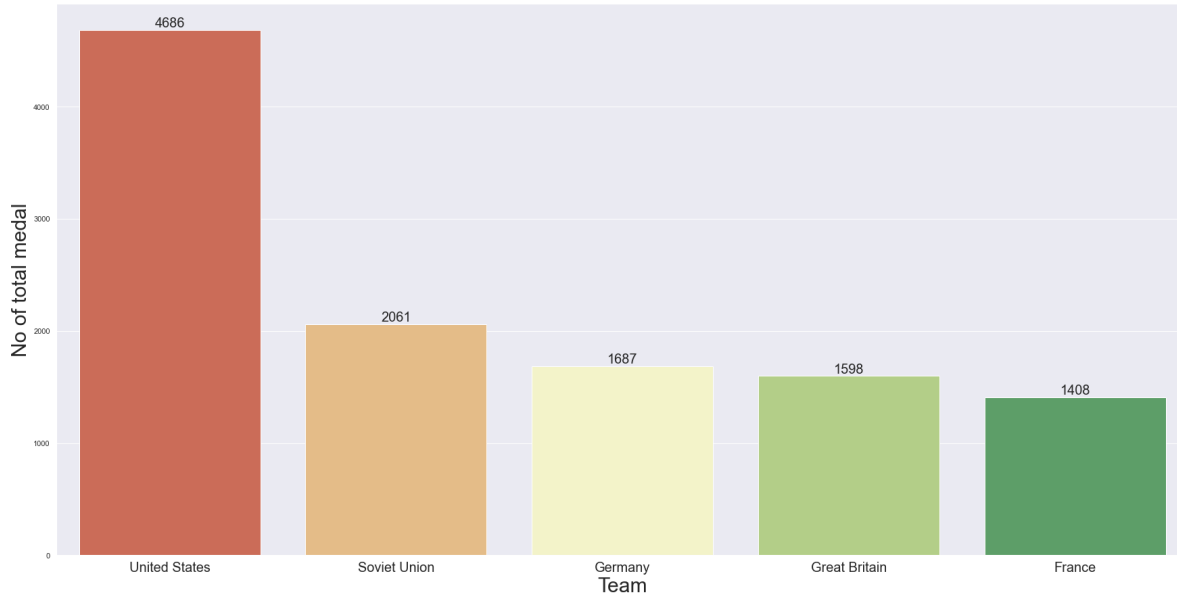
In [245]:
```
data19 = data18.value_counts("Team").reset_index(name = "No of total medal")[:
```

In [246]:
```
data19
```

Out[246]:

	Team	No of total medal
0	United States	4686
1	Soviet Union	2061
2	Germany	1687
3	Great Britain	1598
4	France	1408

In [247]:
```
fig = plt.figure(figsize=(30,15))
barplot = sns.barplot(x = "Team", y = "No of total medal", data=data19, palett
plt.xlabel("Team", fontsize = 30)
plt.ylabel("No of total medal", fontsize = 30)
plt.xticks(fontsize = 20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 20)
plt.show()
```



Athletes participation by gender over years for Olympics

In [248]:
```
df.columns
```

Out[248]:
```
Index(['ID', 'Name', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Game
s',
       'Year', 'Season', 'City', 'Sport', 'Event', 'Medal', 'region',
       'Age_Level'],
      dtype='object')
```

In [249]:
```
data20 = df.groupby(["Sex","Year"])["Name"].nunique().reset_index(name = "No o
```

In [250]:
```
data20
```

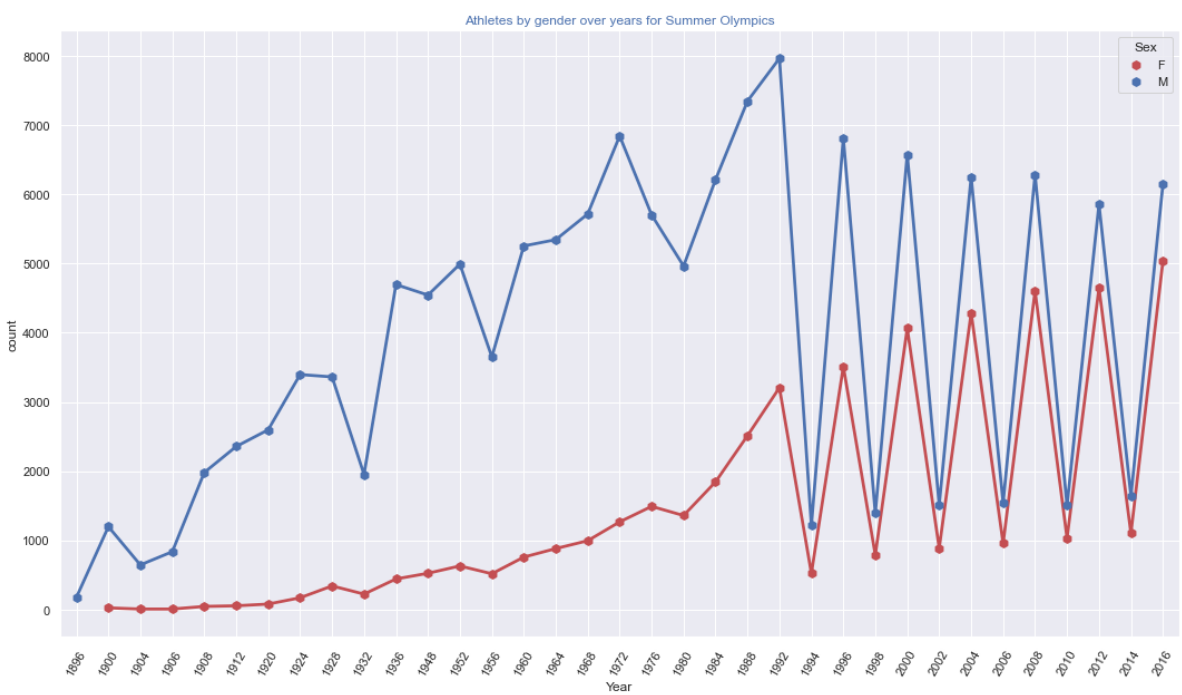
Out[250]:

	Sex	Year	No of athletes
0	F	1900	23
1	F	1904	6
2	F	1906	6
3	F	1908	44
4	F	1912	53
...	...	...	...
64	M	2008	6284
65	M	2010	1503
66	M	2012	5858
67	M	2014	1642
68	M	2016	6143

69 rows × 3 columns

In [251]:
```
fig = plt.figure(figsize=(18,10))
ax = sns.pointplot(x = data20["Year"] , y = data20["No of athletes"],
                   markers="h" , hue = data20["Sex"],palette = ["r","b"])
plt.grid(True)
plt.xticks(rotation = 60)
plt.ylabel("count")
plt.title("Athletes by gender over years for Summer Olympics",color="b")
```

Out[251]: Text(0.5, 1.0, 'Athletes by gender over years for Summer Olympics')



Great Britain Medals in Sports across Years

In [252]:
```
data21 = df[(df['Medal'] != 'None')][['Team',"Year","Sport"]].value_counts().r
```

In [253]:
```
data21
```

Out[253]:

	Team	Year	Sport	no of total Medal
0	United States	1968	Swimming	83
1	United States	1972	Swimming	74
2	United States	2016	Swimming	71
3	United States	2012	Swimming	68
4	United States	2008	Swimming	68
...	...	...	...	...
7291	Finland	2006	Alpine Skiing	1
7292	Romania	2000	Judo	1
7293	Finland	2004	Wrestling	1
7294	Romania	2000	Shooting	1
7295	South Africa	1924	Boxing	1

7296 rows × 4 columns

In [254]:
```
data21["Team"].nunique()
```

Out[254]: 498

In [255]:
```
list = []
for iteam in data21["Team"]:
    list.append(iteam.split(","))
```

In [256]:
```
list
```

Out[256]:
```
[['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['United States'],
 ['Soviet Union'],
 ['United States'],
 ['East Germany'],
 ['United States'],
 ['Australia'],
 ['East Germany'],
```

In [257]:
```
one_d = []
for item in list:
    for item1 in item:
        one_d.append(item1)
```

In [258]:
```
one_d
```

Out[258]:
```
['United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'United States',
 'Soviet Union',
 'United States',
 'East Germany',
 'United States',
 'Australia',
 'East Germany',
```

In [259]:
```
len(one_d)
```

Out[259]:
```
7305
```

In [260]:
```
uni_list = []
for item in one_d:
    if item not in uni_list:
        uni_list.append(item)
```

In [261]:
```
len(uni_list)
```

Out[261]: 507

In [262]:
```
data22 = data21[data21["Team"]=="Great Britain"][:20]
data22.sort_values(by = "Year")
```

Out[262]:

	Team	Year	Sport	no of total Medal
78	Great Britain	1908	Shooting	33
262	Great Britain	1908	Athletics	21
100	Great Britain	1912	Shooting	31
219	Great Britain	1912	Gymnastics	23
231	Great Britain	1912	Athletics	22
152	Great Britain	1920	Athletics	25
232	Great Britain	1924	Athletics	22
366	Great Britain	1964	Athletics	18
518	Great Britain	1980	Athletics	16
235	Great Britain	1984	Athletics	22
525	Great Britain	1988	Hockey	16

In [263]:
```
data24 = df[(df['Medal'] != 'None')][['Team',"Year"]].value_counts().reset_ind
```

In [264]:
```
data24
```

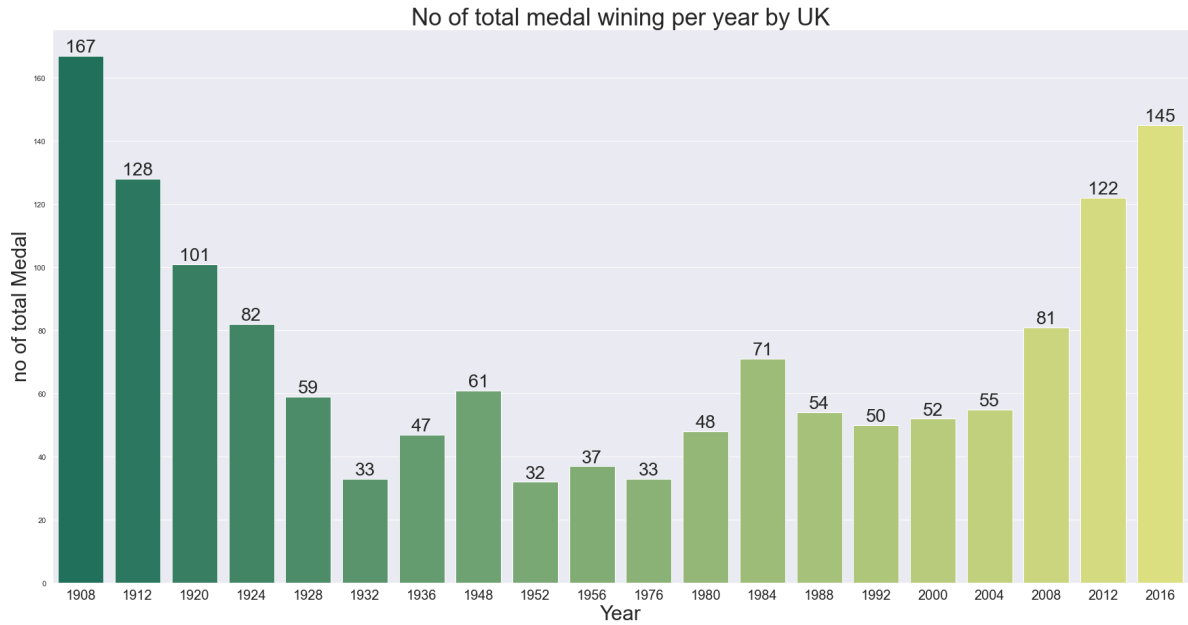
Out[264]:

	Team	Year	no of total Medal
0	Soviet Union	1980	488
1	United States	1984	359
2	Soviet Union	1988	352
3	Soviet Union	1976	336
4	United States	2008	309
...	...	...	...
2007	Portugal	2016	1
2008	Sweden	1900	1
2009	Mexico	1952	1
2010	Mongolia	2004	1
2011	Spain	1952	1

2012 rows × 3 columns

In [265]:
```
data25 = data24[data24["Team"]=="Great Britain"][:20]
data25
```

Out[265]:

	Team	Year	no of total Medal
35	Great Britain	1908	167
46	Great Britain	2016	145
54	Great Britain	1912	128
56	Great Britain	2012	122
74	Great Britain	1920	101
92	Great Britain	1924	82
95	Great Britain	2008	81
121	Great Britain	1984	71
163	Great Britain	1948	61
167	Great Britain	1928	59
185	Great Britain	2004	55

In [266]:
```
fig = plt.figure(figsize=(30,15))
barplot = sns.barplot(x = "Year", y = "no of total Medal", data=data25, palett
plt.xlabel("Year", fontsize = 30)
plt.ylabel("no of total Medal", fontsize = 30)
plt.xticks(fontsize = 20)
for bars in barplot.containers:
    barplot.bar_label(bars,fontsize = 27)
plt.title("No of total medal wining per year by UK", fontsize = 35)
plt.show()
```