In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import seaborn as sns
sns.set_style("darkgrid")
import warnings
warnings.filterwarnings("ignore")
import plotly.express as px
```

In [3]:
```
df=pd.read_csv("athlete_events.csv")
df.head(5)
```

Out[3]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [ ]:
```
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  object 
 2   Sex     271116 non-null  object 
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object 
 7   NOC     271116 non-null  object 
 8   Games   271116 non-null  object 
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  object 
 11  City    271116 non-null  object 
 12  Sport   271116 non-null  object 
 13  Event   271116 non-null  object 
 14  Medal   39783 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

In [5]:
```
df.describe()
```

Out[5]:

	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

In [6]:
```
df.describe(include="O")
```

Out[6]:

	Name	Sex	Team	NOC	Games	Season	City	Sport	Event	Medal
count	271116	271116	271116	271116	271116	271116	271116	271116	271116	39783
unique	134732	2	1184	230	51	2	42	66	765	3
top	Robert Tait McKenzie	M	United States	USA	2000 Summer	Summer	London	Athletics	Football Men's Football	Gold
freq	58	196594	17847	18853	13821	222552	22426	38624	5733	13372

In [ ]:

Identify the which columns have null value.

In [7]:
```
nan_values = df.isna()
nan_columns = nan_values.any()

columns_with_nan = df.columns[nan_columns].tolist()
print(columns_with_nan)
```

```
['Age', 'Height', 'Weight', 'Medal']
```

In [ ]:

Replace age,weight and height value with 0.

In [8]:
```
df[['Age','Height','Weight']] = df[['Age','Height','Weight']].fillna(0)
```

Replace Nan value with 0 in Medal column

In [9]:
```
df.Medal = df.Medal.fillna('0')
```

In [ ]:

Now convert the Age, Height, Weight value into integer.

In [10]:
```
cols = ['Age', 'Weight','Height']
df[cols] = df[cols].applymap(np.int64)
df.head(5)
```

Out[10]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24	180	80	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	0
1	2	A Lamusi	M	23	170	60	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	0
2	3	Gunnar Nielsen Aaby	M	24	0	0	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	0
3	4	Edgar Lindenau Aabye	M	34	0	0	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21	185	82	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	0

In [ ]:

Drop column name Game

In [11]:
```
df=df.drop(["Games"],axis=1)
df.head(5)
```

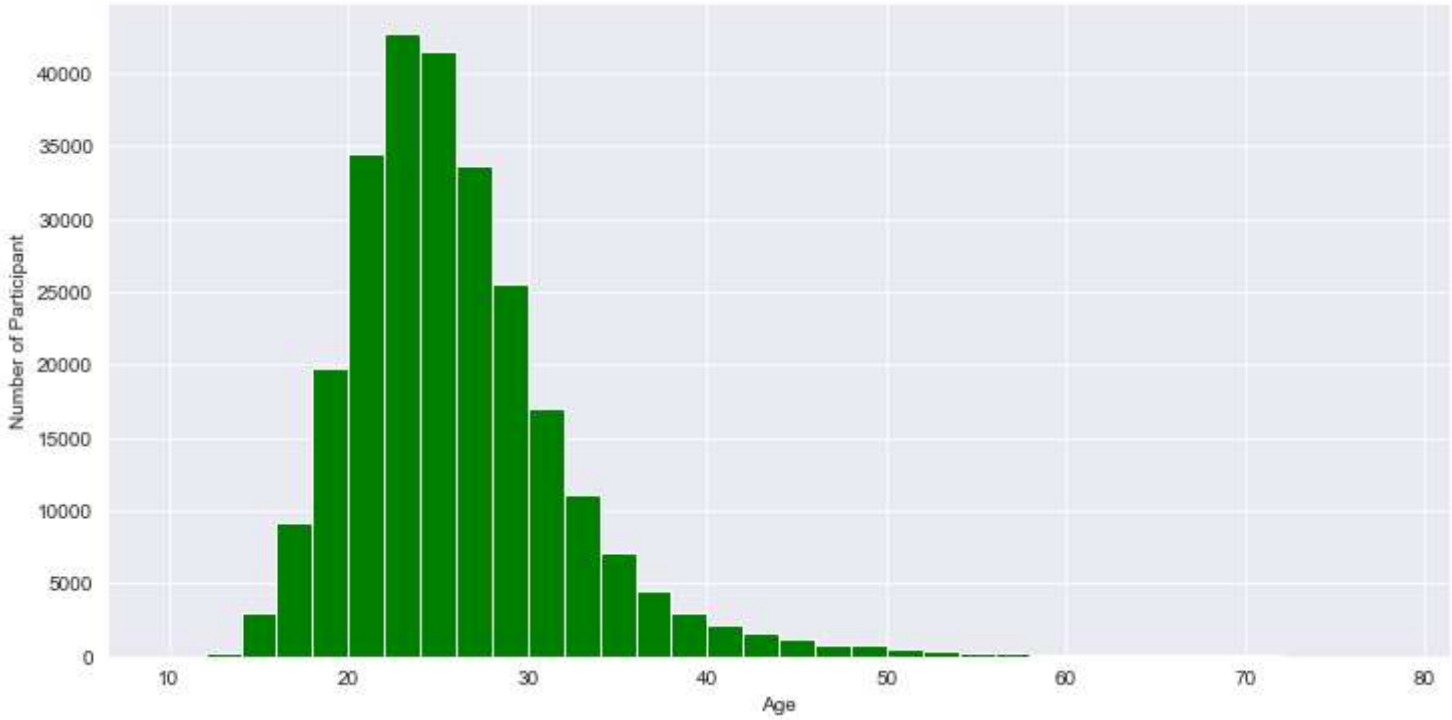
Out[11]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24	180	80	China	CHN	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	0
1	2	A Lamusi	M	23	170	60	China	CHN	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	0
2	3	Gunnar Nielsen Aaby	M	24	0	0	Denmark	DEN	1920	Summer	Antwerpen	Football	Football Men's Football	0
3	4	Edgar Lindenau Aabye	M	34	0	0	Denmark/Sweden	DEN	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21	185	82	Netherlands	NED	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	0

In [ ]:

Age Distribution of player

In [12]:
```
plt.figure(figsize=(12, 6))
plt.xlabel('Age')
plt.ylabel('Number of Participant')

plt.hist(df.Age, bins=np.arange(10,80,2), color='Green');
```



Observation: common age of players are 22-24.

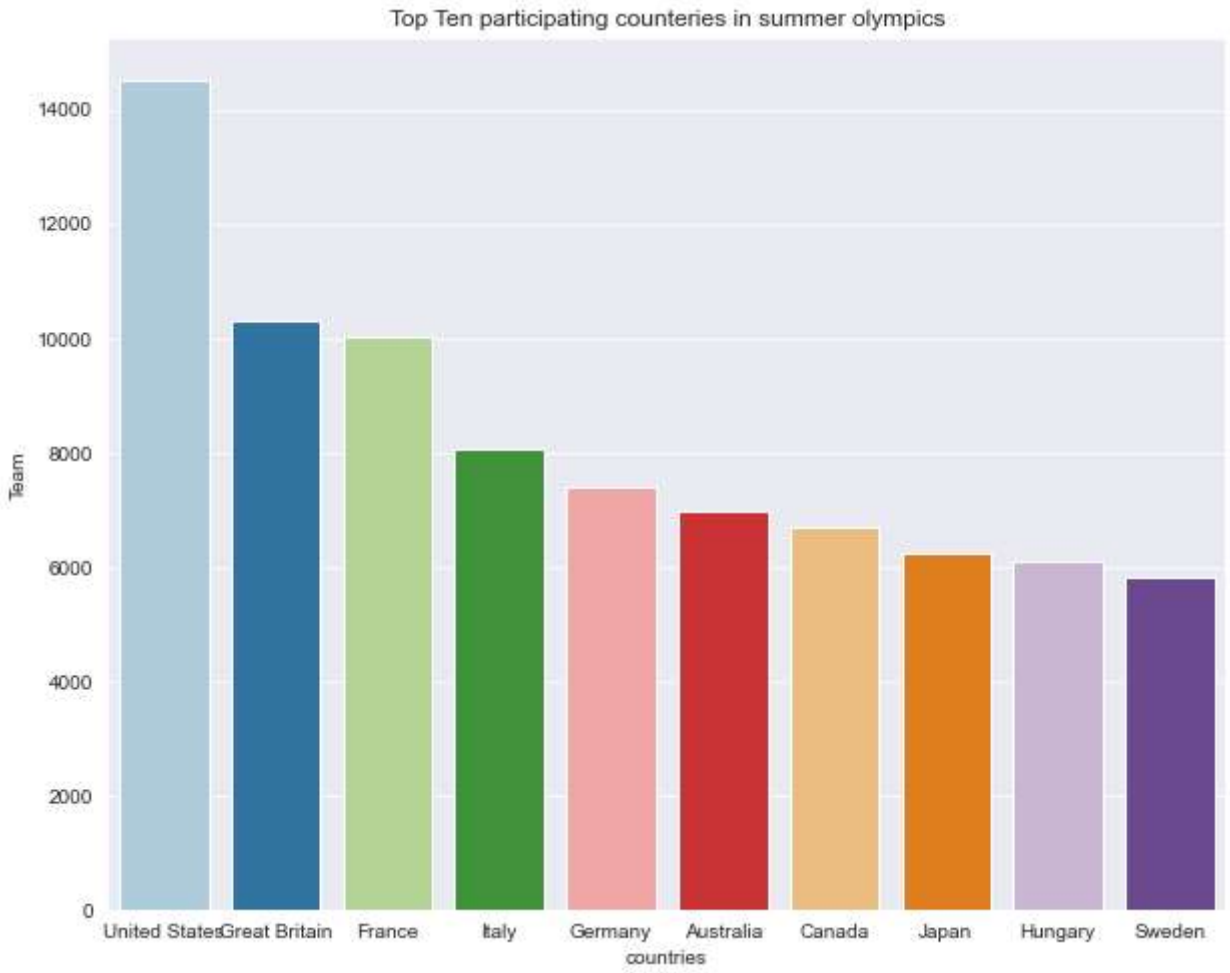
In [ ]:

Top Ten Countries participating in Summer olympics.

In [13]:
```
summer = df[df['Season']=='Summer'][['Team','Season']]
Top_10_summer = summer['Team'].value_counts().sort_values(ascending=False).head(10)
```

In [14]:
```
plt.figure(figsize=(10,8))
plt.title("Top Ten participating counteries in summer olympics")
plt.xlabel("countries")
plt.ylabel("number of participants")
sns.barplot(x=Top_10_summer.index , y=Top_10_summer , palette='Paired')
```

Out[14]:
```
<AxesSubplot:title={'center':'Top Ten participating counteries in summer olympics'}, xlabel='countries', ylabel='Team'>
```



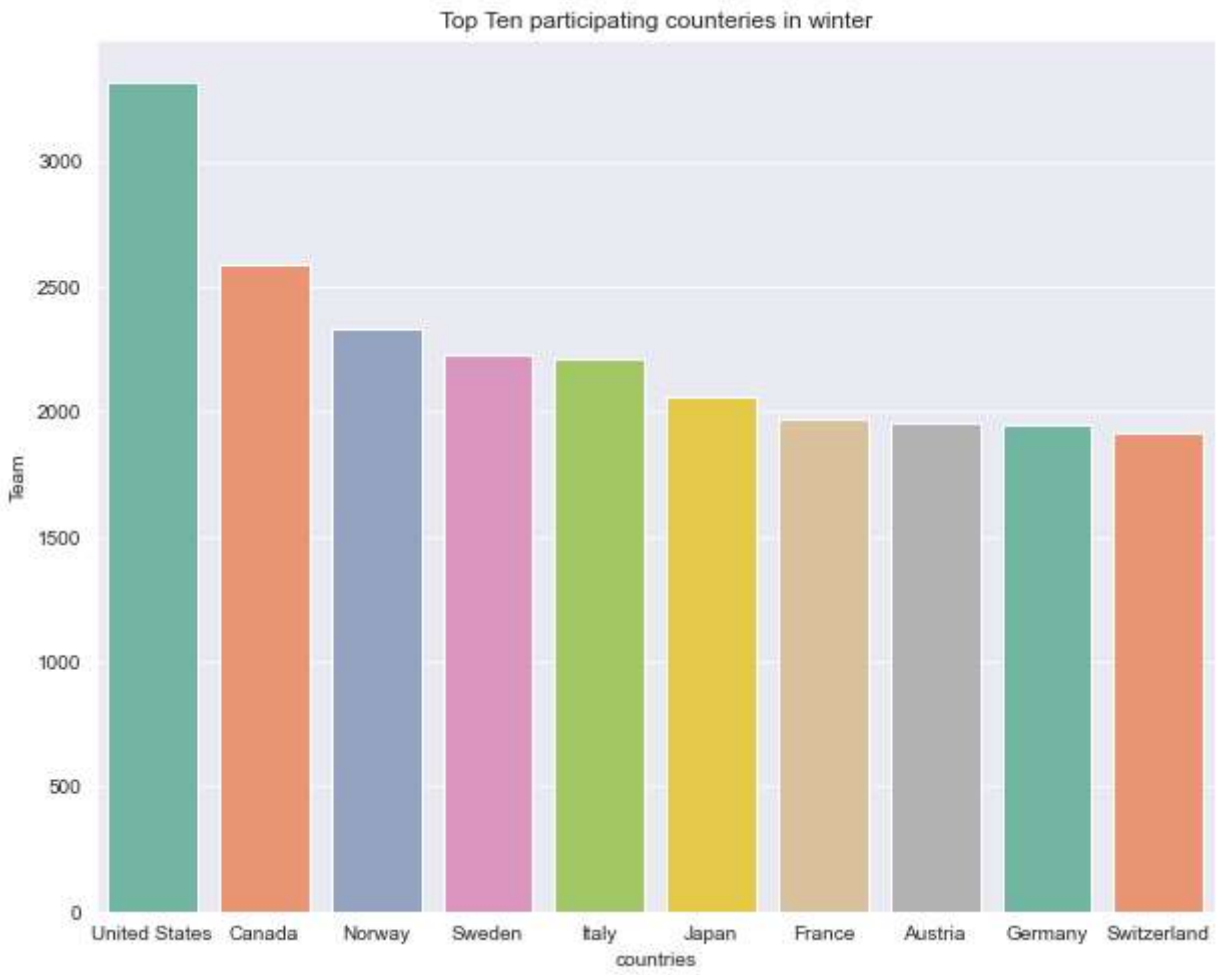
In [ ]:

Top Ten Countries Participating in Winet olympics

In [15]:
```
winter =  df[df['Season']=='Winter'][['Team','Season']]
Top_10_winter=winter['Team'].value_counts().sort_values(ascending=False).head(10)
```

In [16]:
```
plt.figure(figsize=(10,8))
plt.title("Top Ten participating counteries in winter")
plt.xlabel("countries")
plt.ylabel("number of participants")
sns.barplot(x=Top_10_winter.index , y=Top_10_winter , palette='Set2')
```

Out[16]: <AxesSubplot:title={'center':'Top Ten participating counteries in winter'}, xlabel='countries', ylabel='Team'>



In [ ]:

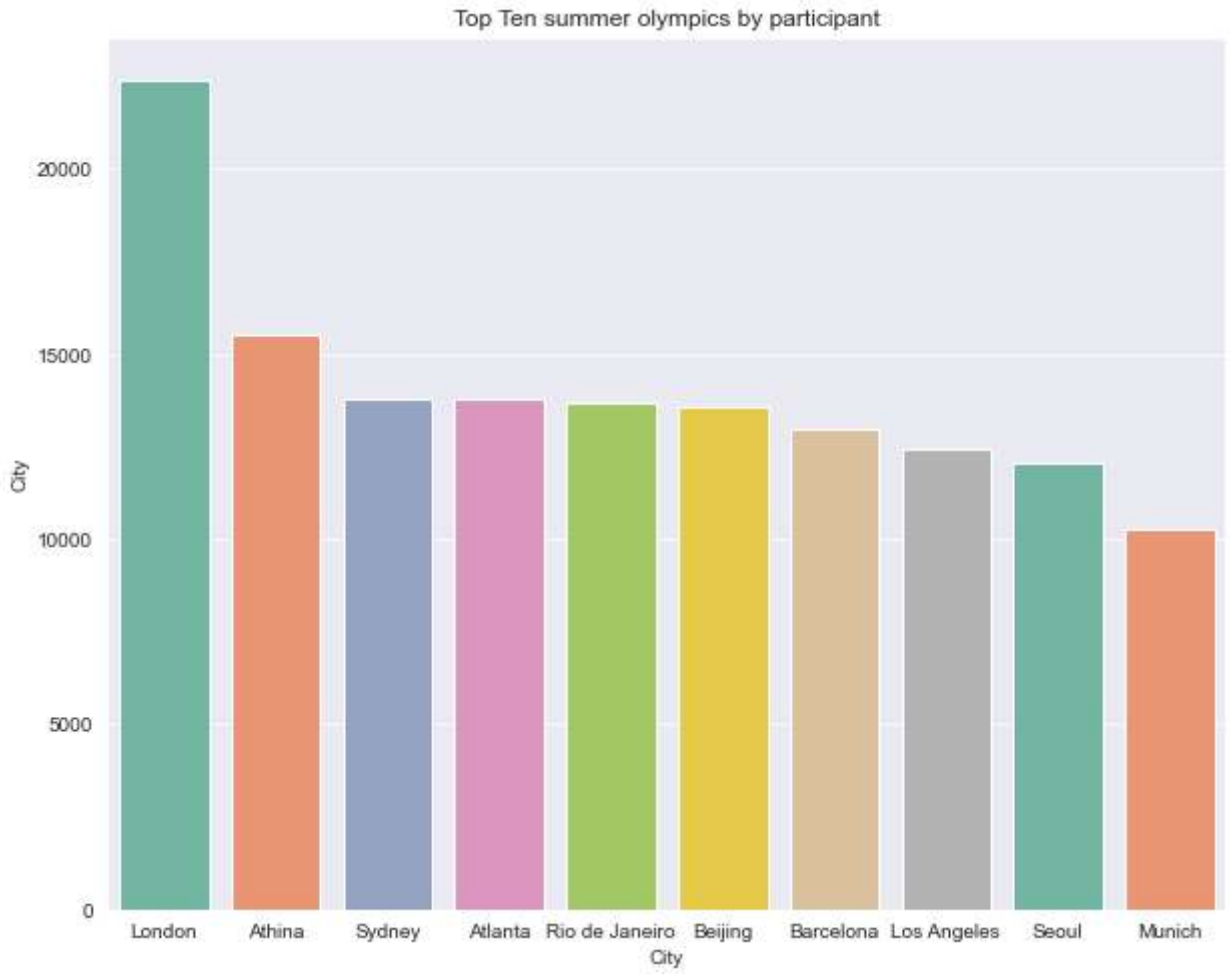
Which summer olympic had the largest number of participants?

In [17]:
```
summer = df[df['Season']=='Summer'][['City','Season']]
Top_10_summer = summer['City'].value_counts().sort_values(ascending=False).head(10)
```

In [18]:
```
plt.figure(figsize=(10,8))
plt.title("Top Ten summer olympics by participant")
```

```
plt.xlabel("City")
plt.ylabel("number of participants")
sns.barplot(x=Top_10_summer.index , y=Top_10_summer , palette='Set2')
```

Out[18]: <AxesSubplot:title={'center':'Top Ten summer olympics by participant'}, xlabel='City', ylabel='City'>



Observation: The large number of player participating in London Olympic

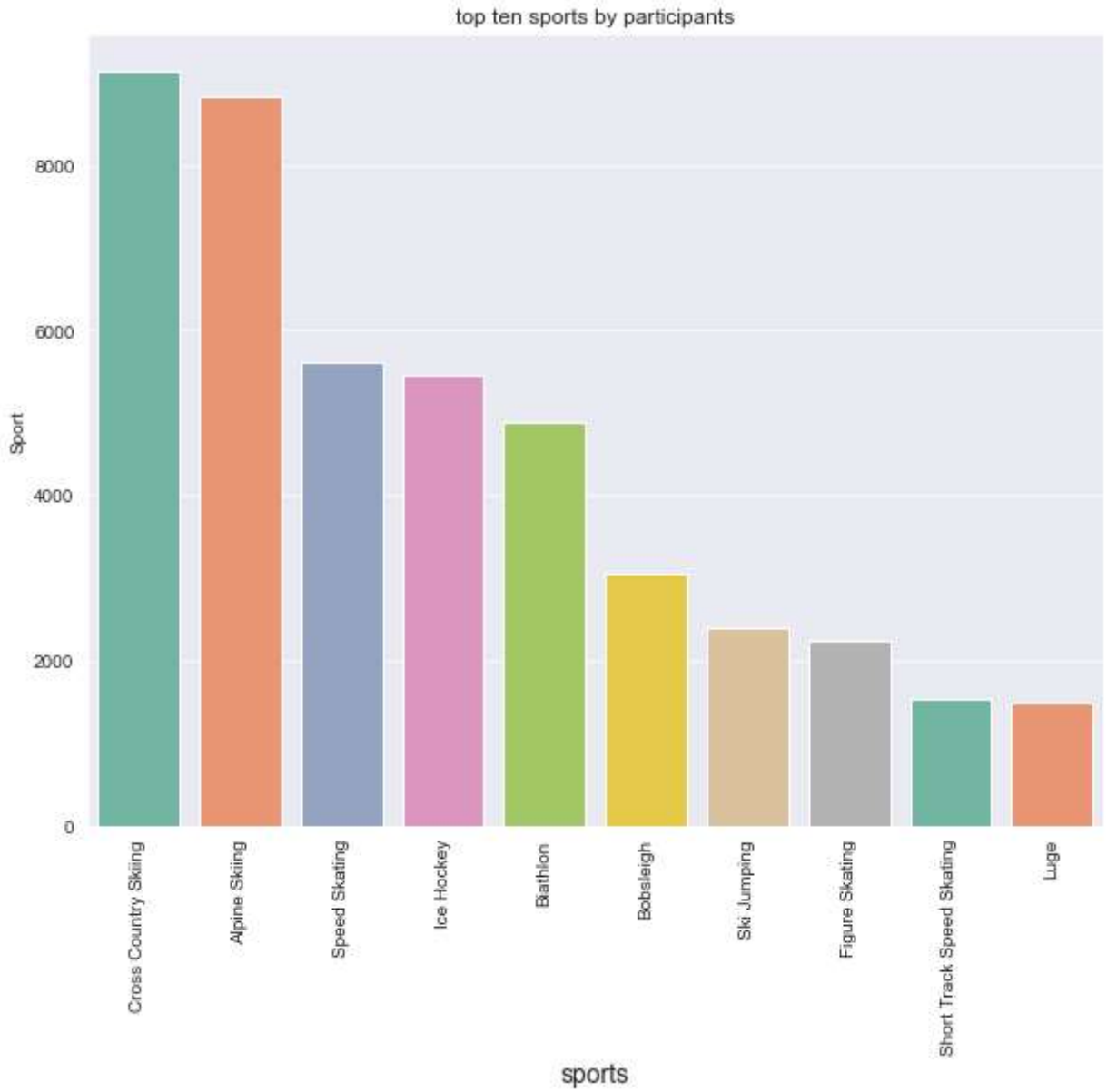
Which sport has the largest number of participanst in the winter Olympics?

In [19]:
```
winter =  df[df['Season']=='Winter'][['Sport','Season']]
```

```
Top_10_Sports=winter["Sport"].value_counts().sort_values(ascending=False).head(10)
```

In [20]:
```
plt.figure(figsize=(10,8))
plt.title("top ten sports by participants")
plt.xlabel("sports",size=14)
plt.ylabel("number of participants")
plt.xticks(rotation=90)
sns.barplot(x=Top_10_Sports.index , y=Top_10_Sports , palette='Set2')
```

Out[20]:
```
<AxesSubplot:title={'center':'top ten sports by participants'}, xlabel='sports', ylabel='Sport'>
```



Observation: Most player participate in Cross Country Skiing sport

In [ ]:

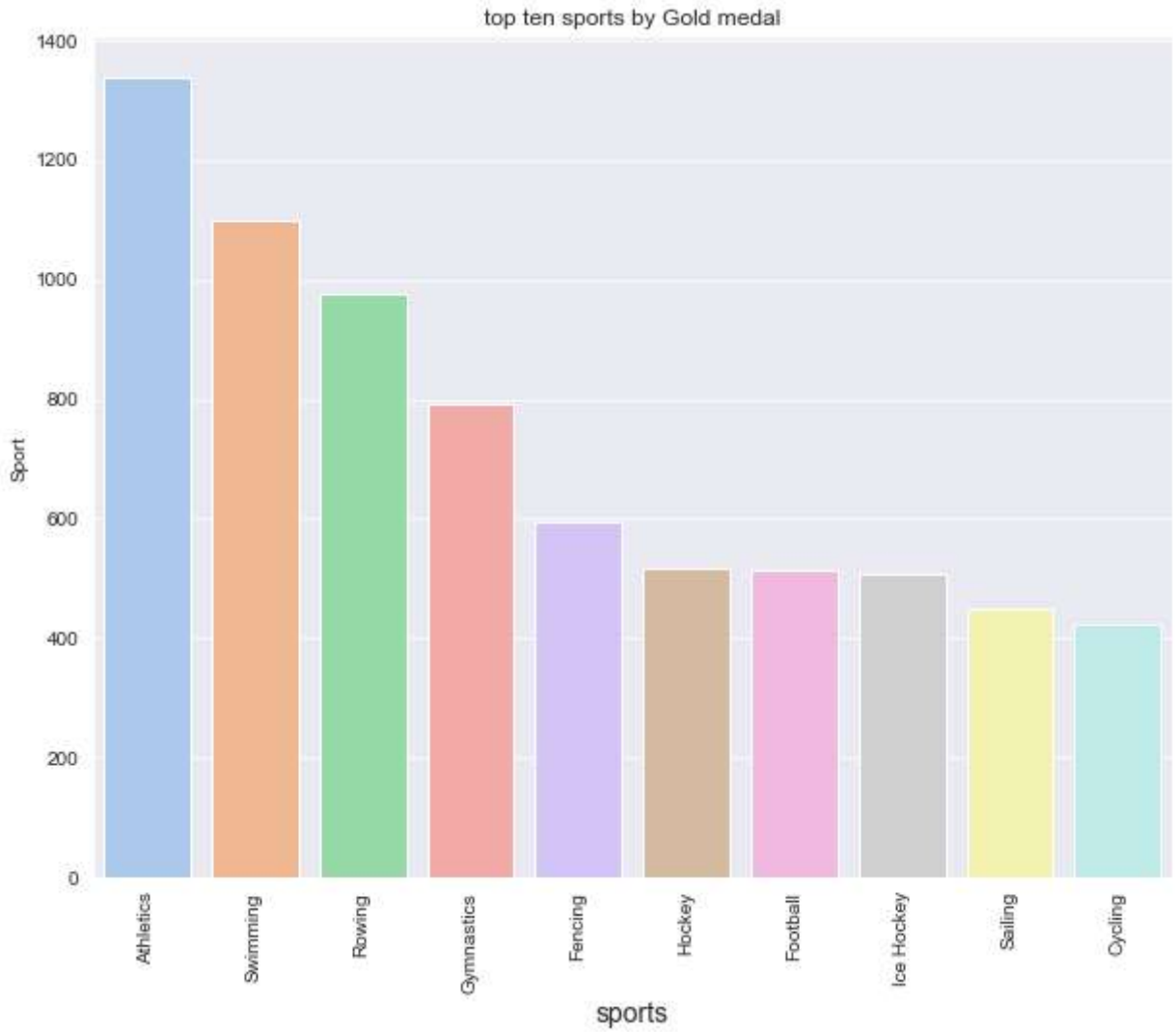
Which top 10 sports have most gold medal in Olympics?

In [21]:
```
category =  df[df['Medal']=='Gold'][['Sport','Medal']]
```

In [22]:
```
Top_10_Sports=category["Sport"].value_counts().sort_values(ascending=False).head(10)
```

In [23]:
```
plt.figure(figsize=(10,8))
plt.title("top ten sports by Gold medal")
plt.xlabel("sports",size=14)
plt.ylabel("number of participants")
plt.xticks(rotation=90)
sns.barplot(x=Top_10_Sports.index , y=Top_10_Sports , palette='pastel')
```

Out[23]:
```
<AxesSubplot:title={'center':'top ten sports by Gold medal'}, xlabel='sports', ylabel='Sport'>
```



Observation: Athletics have highest gold medal

In [ ]:

Which country has won highest gold medal in shooting?

In [24]:
```
category =  df[df['Sport']=='Shooting'] [['Team','Sport','Medal']]
```

In [25]:
```
gold =  category[category['Medal']=='Gold'] [['Team','Sport','Medal']]
```

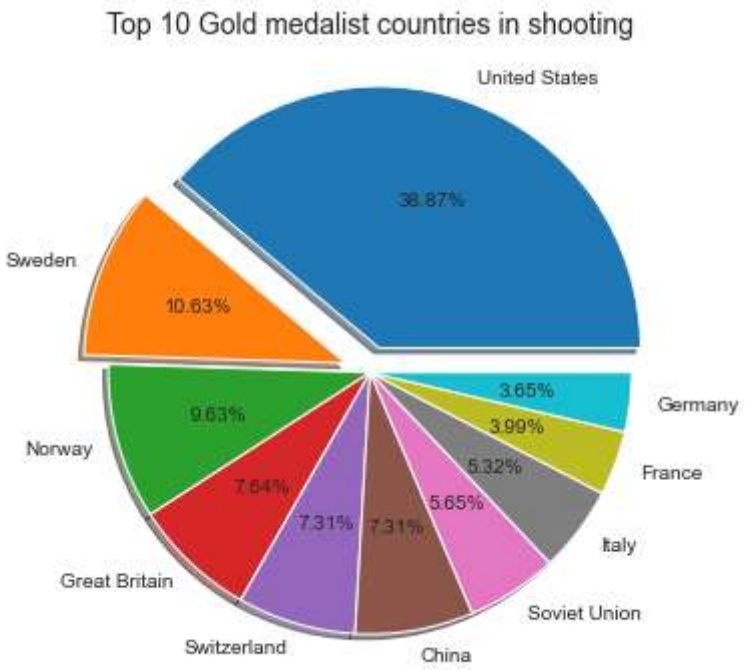
In [26]:
```
gold_shooting=gold["Team"].value_counts().sort_values(ascending=False).head(10)
gold_shooting
```

Out[26]:
```
United States    117
Sweden            32
Norway            29
Great Britain     23
Switzerland       22
China             22
Soviet Union      17
Italy             16
France            12
Germany           11
Name: Team, dtype: int64
```

In [ ]:
```

```

In [27]:
```
plt.figure(figsize=(10,6))
labels=gold_shooting.keys()
plt.pie(x = gold_shooting, labels=labels, autopct="%1.2f%%", shadow=True, radius=1,explode=[0.1,0.1,0,0,0,0,0,0,0,0])
plt.title("Top 10 Gold medalist countries in shooting", fontsize=14)
```

Out[27]:
```
Text(0.5, 1.0, 'Top 10 Gold medalist countries in shooting')
```



Observation: United States have won highest gold medal in shooting

In [ ]:

Which country have won highest Gold Medal in Hockey?

In [28]:
```
category =  df[df['Sport']=='Hockey'] [['Team','Sport','Medal']]
```

In [29]:
```
category_gold=category[category["Medal"]=='Gold'] [['Team','Sport','Medal']]
```

In [30]:
```
gold=category_gold["Team"].value_counts().sort_values(ascending=False).head(5)
```

In [31]:
```
# Now divide this column by 11 because hockey team have 11 players.
# here we sort out list by player team.
```

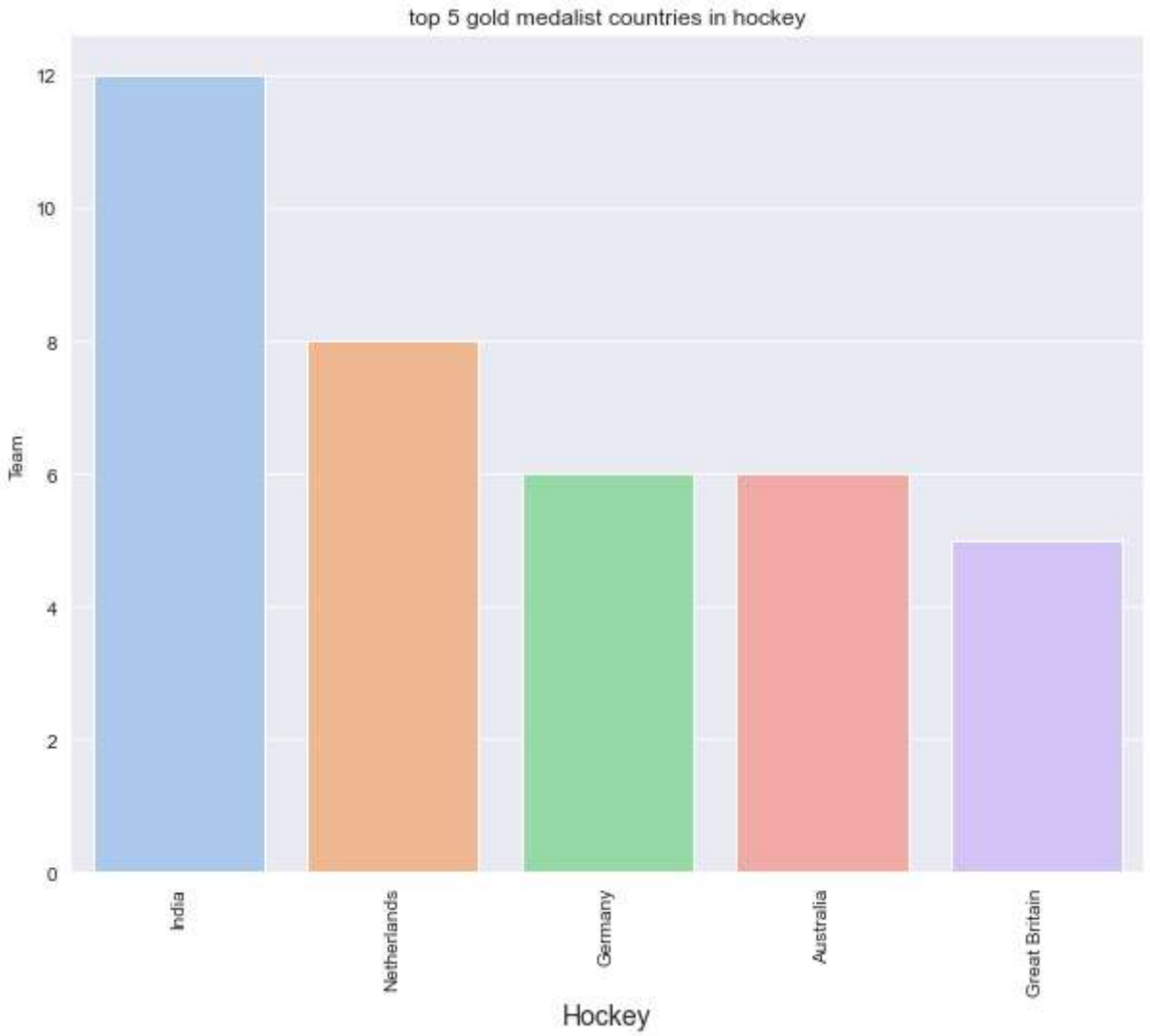
In [32]:
```
Gold=gold.div(11).round(2)
```

In [33]:
```
Gold=Gold.apply(np.ceil).head(5)
Gold
```

Out[33]:
```
India            12.0
Netherlands       8.0
Germany           6.0
Australia         6.0
Great Britain     5.0
Name: Team, dtype: float64
```

In [34]:
```
plt.figure(figsize=(10,8))
plt.title("top 5 gold medalist countries in hockey")
plt.xlabel("Hockey",size=14)
plt.ylabel("Count")
plt.xticks(rotation=90)
sns.barplot(x=Gold.index , y=Gold , palette='pastel')
```

Out[34]:
```
<AxesSubplot:title={'center':'top 5 gold medalist countries in hockey'}, xlabel='Hockey', ylabel='Team'>
```



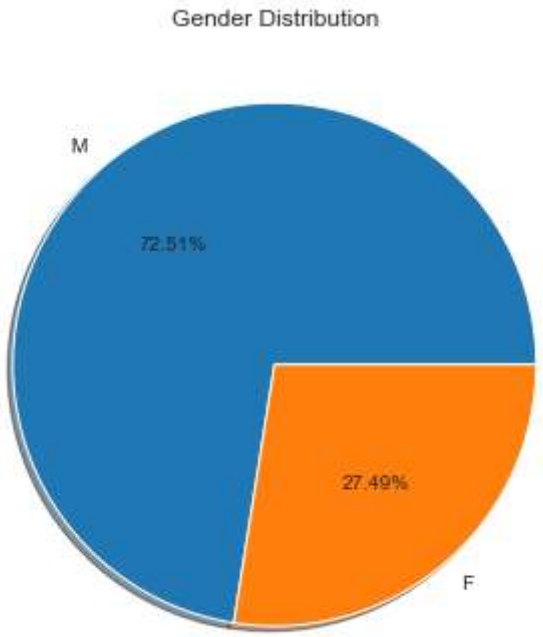
Observation: India have won highest Gold medal in Hockey

How many Male and Female Participate in Olympics?

In [35]:
```
gender_counts = df.Sex.value_counts()
gender_counts
```

Out[35]:
```
M    196594
F     74522
Name: Sex, dtype: int64
```

In [36]:
```
plt.figure(figsize=(12,6))
plt.title('Gender Distribution')
labels=gender_counts.keys()
plt.pie(x=gender_counts, labels=labels, shadow=True, autopct='%1.2f%%');
```



Observation: Male participations is higher compare to Female

In [ ]:
```
```

Top 10 Gold Medalist Country in winter Olympics

In [37]:
```
winter =  df[df['Season']=='Winter'][['Team','Season','Medal']]
```

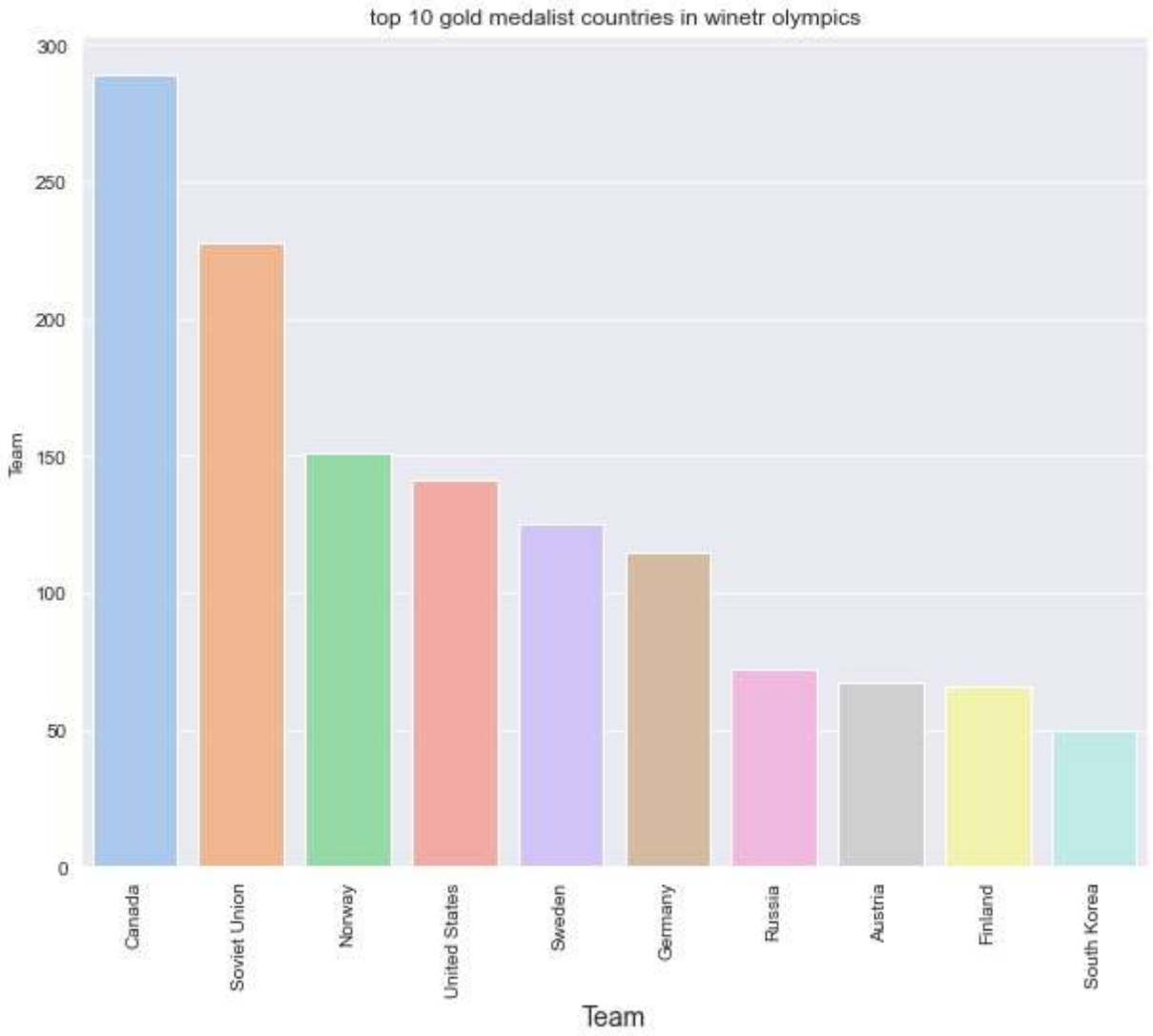
In [38]:
```
category_gold=winter[winter["Medal"]=='Gold'] [['Team','Season','Medal']]
```

In [39]:
```
top_10_gold=category_gold["Team"].value_counts().sort_values(ascending=False).head(10)
top_10_gold
```

Out[39]:
```
Canada           289
Soviet Union     228
Norway           151
United States    141
Sweden           125
Germany          115
Russia            72
Austria           67
Finland           66
South Korea       50
Name: Team, dtype: int64
```

In [40]:
```
plt.figure(figsize=(10,8))
plt.title("top 10 gold medalist countries in winetr olympics")
plt.xlabel("Team",size=14)
plt.ylabel("Count")
plt.xticks(rotation=90)
sns.barplot(x=top_10_gold.index , y=top_10_gold , palette='pastel')
```

Out[40]:
```
<AxesSubplot:title={'center':'top 10 gold medalist countries in winetr olympics'}, xlabel='Team', ylabel='Team'>
```



Observation: Canada have won Highest gold medal in Winter Olympics

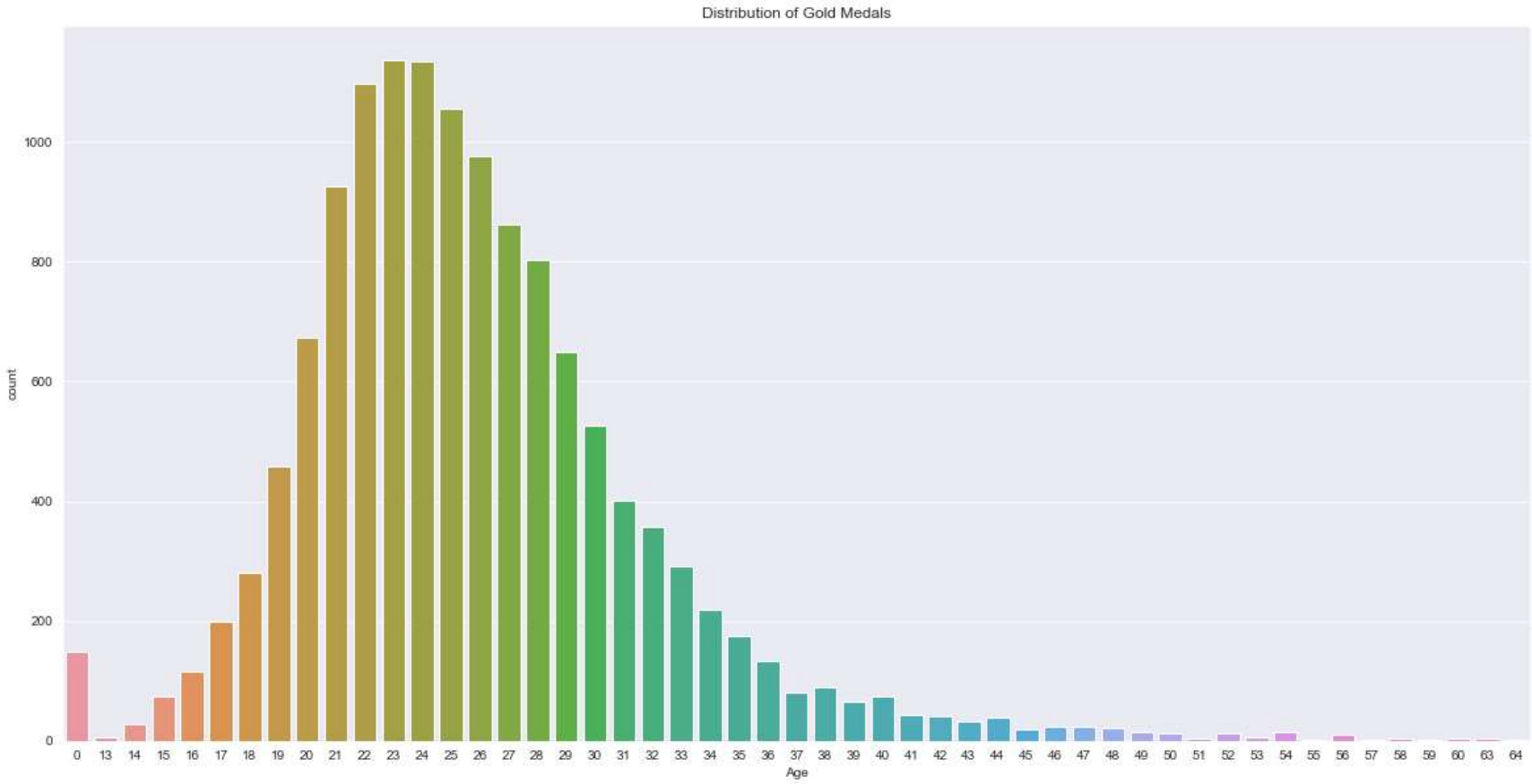
In [ ]:

Show Distribution between Age and Gold Medal.

In [41]:
```
goldmedal_age =  df[df['Medal']=='Gold'][['Age','Medal']]
```

In [42]:
```
plt.figure(figsize=(20, 10))
plt.tight_layout()
sns.countplot(goldmedal_age['Age'])
plt.title('Distribution of Gold Medals')
```

Out[42]:
```
Text(0.5, 1.0, 'Distribution of Gold Medals')
```



Observation: The average age of Gold medalist's players are 23-24

In [ ]:
```
```

In [43]:
```
# Now filter out players whose age>=50 and won gold medal.
```

In [44]:
```
age=df[df['Medal']=='Gold'][['Age','Medal','Team','Sport']]
age
```

Out[44]:

	Age	Medal	Team	Sport
3	34	Gold	Denmark/Sweden	Tug-Of-War
42	28	Gold	Finland	Gymnastics
44	28	Gold	Finland	Gymnastics
48	28	Gold	Finland	Gymnastics
60	20	Gold	Norway	Alpine Skiing
...	...	...	...	...
270981	23	Gold	Georgia	Judo
271009	28	Gold	Germany	Hockey
271016	29	Gold	Netherlands	Volleyball
271049	31	Gold	Netherlands	Rowing
271076	21	Gold	Soviet Union	Athletics

13372 rows × 4 columns

In [45]:
```
# filter out players whose age are 50 or higher.
```

In [46]:
```
senior=age[age["Age"]>=50][["Age","Medal","Team","Sport"]]
```

In [ ]:
```

```

How many players won Gold medal after age of 50?

In [47]:
```
senior.Medal.count()
```

Out[47]:
```
77
```

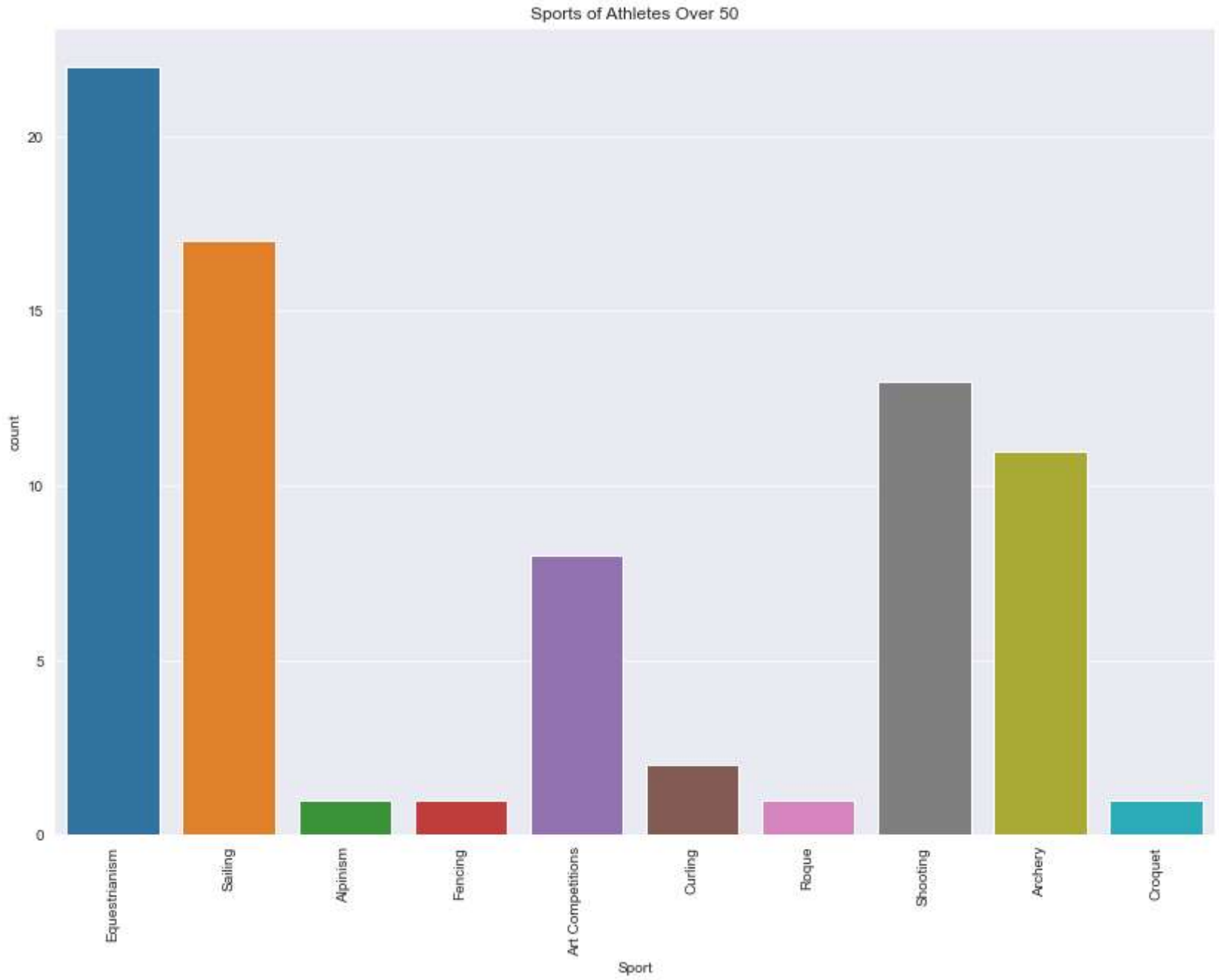
In [ ]:
```

```

IN which sports players above 50 won Highest Gold Medal?

In [48]:
```
sports=senior['Sport']
```

In [49]:
```
plt.figure(figsize=(14,10))
plt.tight_layout()
sns.countplot(sports)
plt.xticks(rotation=90)
plt.title('Sports of Athletes Over 50')
```

Out[49]:
```
Text(0.5, 1.0, 'Sports of Athletes Over 50')
```


Sports of Athletes Over 50
count
0
5
10
15
20
Equestrianism
Sailing
Alpinism
Fencing
Art Competitions
Curling
Roque
Shooting
Archery
Croquet
Sport

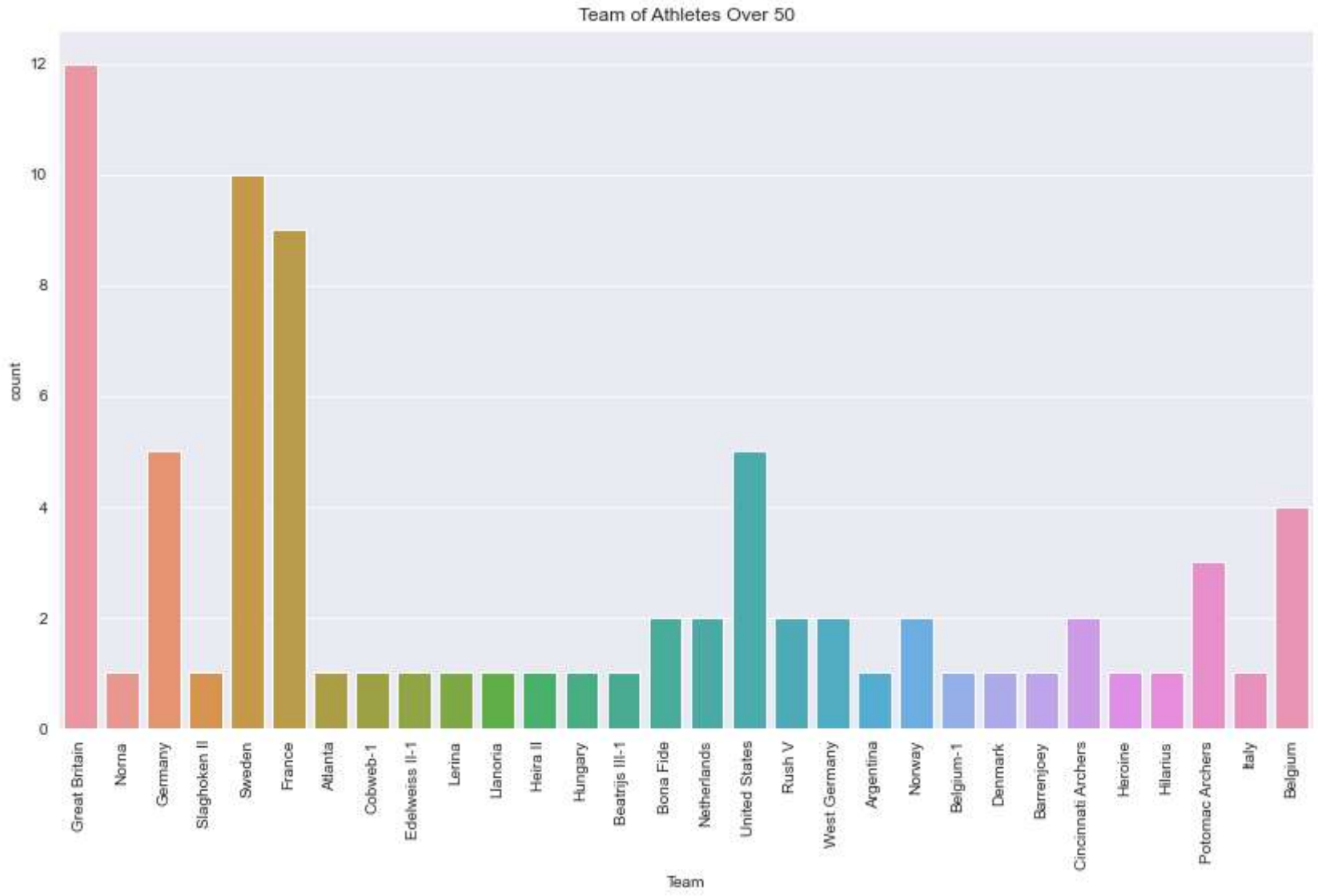
In [ ]:

Which Countries have more players whose age above 50 and win Gold Medal?

In [50]:
```
Team=senior["Team"]
```

In [51]:
```
plt.figure(figsize=(14,8))
plt.tight_layout()
sns.countplot(Team)
plt.xticks(rotation=90)
plt.title('Team of Athletes Over 50')
```

Out[51]:
```
Text(0.5, 1.0, 'Team of Athletes Over 50')
```



Observation:Great Britain has the most players who are over 50 and have won more Gold Medals.

In [ ]:

Women and Men participants in summer olympics by year.

In [52]:
```
women = df[(df.Sex=='F') & (df.Season=='Summer')][['Sex','Year']]
women = women.groupby('Year').count().reset_index()
```

In [ ]:
```

```

In [53]:
```
men = df[(df.Sex=='M') & (df.Season=='Summer')][['Sex','Year']]
men = men.groupby('Year').count().reset_index()
```

In [54]:
```
final = men.merge(women, on="Year", how="left")
final=final.fillna(0)
```

In [55]:
```
final.rename(columns={"Sex_x":"Male","Sex_y":"Female"}, inplace=True)
```

In [56]:
```
fig = px.line(final, x="Year",y=["Male","Female"])
fig.show()
```

Observation: Women's participation in summer Olympics increase.

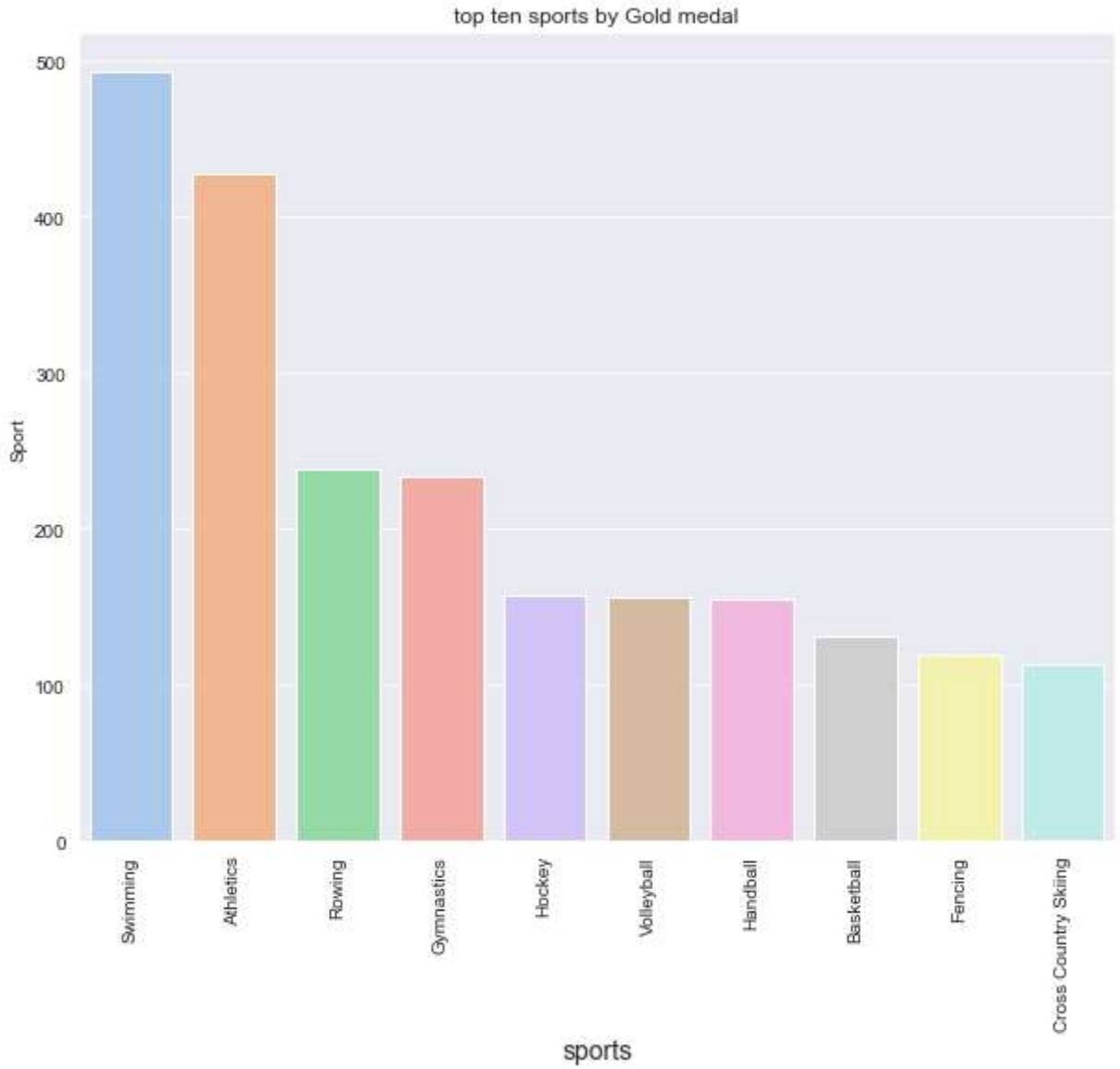
In which sport have women won the most Gold Medals?

In [57]:
```
participant = df[(df.Sex=='F') & (df.Medal=='Gold')][['Sport','Sex','Team','Medal']]
```

In [58]:
```
Top_10_Sport=participant["Sport"].value_counts().sort_values(ascending=False).head(10)
```

In [59]:
```
plt.figure(figsize=(10,8))
plt.title("top ten sports by Gold medal")
plt.xlabel("sports",size=14)
plt.ylabel("number of participants")
plt.xticks(rotation=90)
sns.barplot(x=Top_10_Sport.index , y=Top_10_Sport, palette='pastel')
```

Out[59]:
```
<AxesSubplot:title={'center':'top ten sports by Gold medal'}, xlabel='sports', ylabel='Sport'>
```



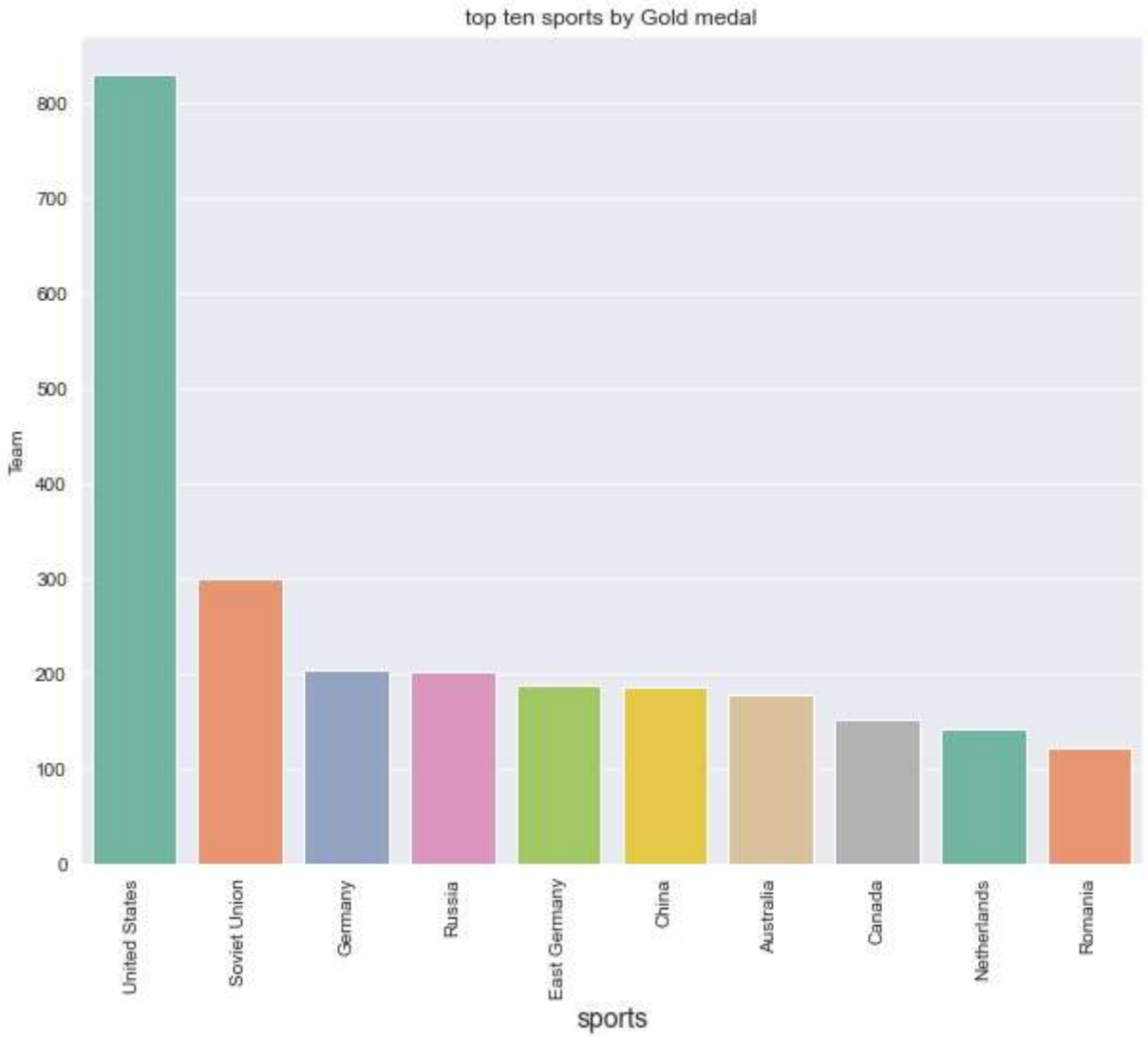
Observation: In Swimmin women have won most Gold Medal

Which country have the most women participating in the olympics?

In [60]:
```
Top_10_Country=participant["Team"].value_counts().sort_values(ascending=False).head(10)
```

In [61]:
```
plt.figure(figsize=(10,8))
plt.title("top ten sports by Gold medal")
plt.xlabel("sports",size=14)
plt.ylabel("number of participants")
plt.xticks(rotation=90)
sns.barplot(x=Top_10_Country.index , y=Top_10_Country, palette='Set2')
```

Out[61]:
```
<AxesSubplot:title={'center':'top ten sports by Gold medal'}, xlabel='sports', ylabel='Team'>
```



Observation: Women in the USA participate most in Olympics.

In [ ]:

In Which sports has britain get highest gold medals?

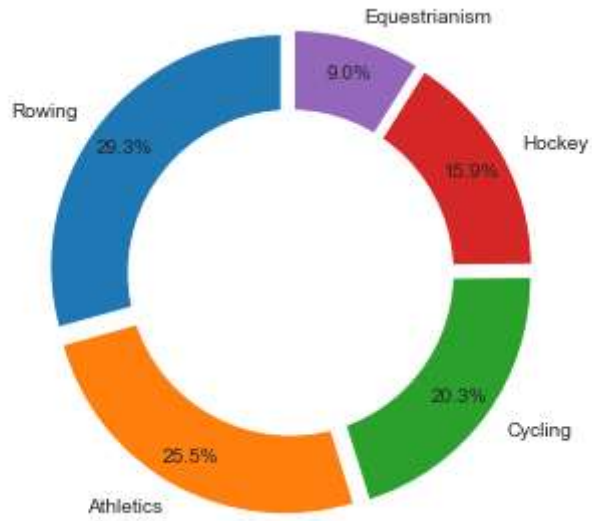
In [62]:
```
Britain = df[(df.Team=='Great Britain') & (df.Medal=='Gold')][['Sport','Team','Medal']]
```

In [63]:
```
GBR_Gold=Britain[['Sport','Medal']].groupby('Sport').count()
GBR_Gold.reset_index(inplace=True)
Top_sports = GBR_Gold.sort_values('Medal', ascending=False)
Top_sports.head()
```

Out[63]:

	Sport	Medal
20	Rowing	85
3	Athletics	74
7	Cycling	59
15	Hockey	46
9	Equestrianism	26

In [64]:
```
fig1, ax1 = plt.subplots()
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd']
explode = (0.05,0.05,0.05,0.05,0.05)
ax1.pie(Top_sports.head()['Medal'], colors = colors, labels= Top_sports.head()['Sport'], autopct='%1.1f%%', startangle=
       , pctdistance=0.85, explode = explode)

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

ax1.axis('equal')
plt.tight_layout()
plt.show()
```



Observation: Britain have won max gold medal in Rowing sports.

In [ ]: