Project Proposal

1. Which client/dataset did you select and why?

Client: SportsStats

SportsStats is a sports analysis firm partnering with local news and elite personal trainers to provide “interesting” insights to help their partners. Insights could be patterns/trends highlighting certain groups/events/countries, etc. for the purpose of developing a news story or discovering key health insights.

The data set being used: Olympics Dataset - 120 years of data

I selected the SportsStats client set as I have always had an interest in sports and being able to use the skills I learned throughout this course to dive deeper into sports related data is very appealing.

In [54]:
```python
import pandas as pd
from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
```

In [55]:
```python
athlete_events = pd.read_csv(r'C:\Users\Shay\Documents\SQL _Data_Science\Captstone_Project\athlete_events.csv')
%store athlete_events
```

```
Stored 'athlete_events' (DataFrame)
```

In [56]:
```python
noc_regions = pd.read_csv(r'C:\Users\Shay\Documents\SQL _Data_Science\Captstone_Project\noc_regions.csv')
```

In [57]:
```python
athlete_events.head()
```

Out[57]:

	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 [58]:
```python
noc_regions.head()
```

Out[58]:

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

In [59]:
```python
athlete_events.isnull().sum()
```

Out[59]:
```
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
dtype: int64
```

Looking at the null values in each category, we have a significant number of nulls within 'Age', 'Height' and 'Weight' The nulls under the 'Medal' column is expected as not every athlete is going to receive a medal.

In [60]:
```python
pysqldf('''SELECT COUNT(*) Total
           FROM athlete_events''')
```

Out[60]:

	Total
0	271116

First I am going to split the events into two categories since we are working with data from the Olympics, 'Summer' and 'Winter'.

In [61]:
```python
summer_games = pysqldf('''SELECT ID, Name, Sex, Age, Height, Weight, Team, NOC, Games, Year, Season, City, Sport, Event, Medal
                        FROM athlete_events
                        WHERE Season = "Summer"''')
%store summer_games
```

```
Stored 'summer_games' (DataFrame)
```

In [62]:
```python
summer_games.head()
```

Out[62]:

	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	None
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	None
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	None
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	8	Cornelia "Cor" Aalten (-Strannood)	F	18.0	168.0	NaN	Netherlands	NED	1932 Summer	1932	Summer	Los Angeles	Athletics	Athletics Women's 100 metres	None

In [63]:
```python
winter_games = pysqldf('''SELECT ID, Name, Sex, Age, Height, Weight, Team, NOC, Games, Year, Season, City, Sport, Event, Medal
                        FROM athlete_events
                        WHERE Season = "Winter"''')
%store winter_games
```

```
Stored 'winter_games' (DataFrame)
```

In [64]:
```python
winter_games.head()
```

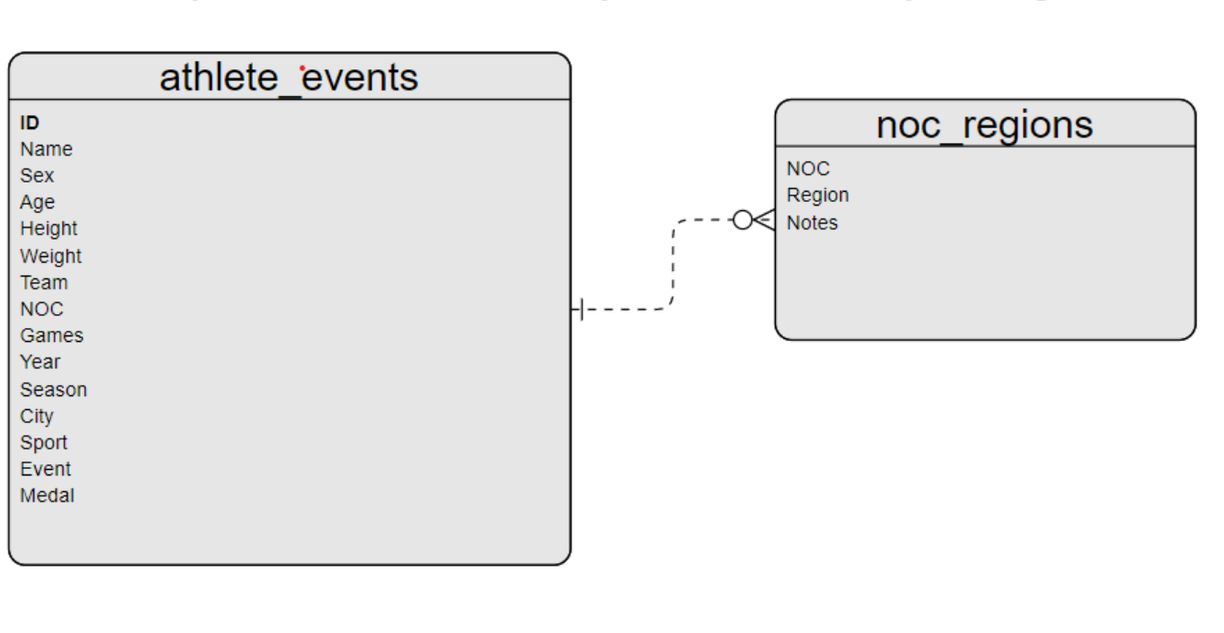
Out[64]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	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	None
1	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 1,000 metres	None
2	5	Christine Jacoba Aaftink	F	25.0	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville	Speed Skating	Speed Skating Women's 500 metres	None
3	5	Christine Jacoba Aaftink	F	25.0	185.0	82.0	Netherlands	NED	1992 Winter	1992	Winter	Albertville	Speed Skating	Speed Skating Women's 1,000 metres	None
4	5	Christine Jacoba Aaftink	F	27.0	185.0	82.0	Netherlands	NED	1994 Winter	1994	Winter	Lillehammer	Speed Skating	Speed Skating Women's 500 metres	None

ERD

In [65]:
```python
from IPython.display import Image
Image(filename = r"C:\Users\Shay\Documents\SQL _Data_Science\Captstone_Project\ERD.png", width = 800, height = 400)
```

Out[65]:



Questions

Question 1: Is there more men or women competing in the Olympics, how has this changed over time

Question 2: Countries with the most medals and how they compare between Summer and Winter

Question 3: Age distribution, oldest and youngest Olympians

Hypothesis

1: I would think there are more men competing in the Olypmics currently but I am very interested in seeing how this has changed overtime and would think in current years it will be a lot more evenly divided.

2: Countries with the most medals are presuming to be the largest countries by population. e.g USA, China etc.

3: I would Imagine the age gap would be quite significant and will be interesting to see which sports they are most divided by.

Approach

1: First step is to review the table to get familiar with the data. I then ran a simple SQL query to get the total counts of males and females that have ever participated. I then declared a variable to store all of the info on males and another variable to store all the info on females using CAST, CASE and SUM functions to tally up the total number of males and femalesso I can visualize it.

I created a simple line chart to take a look at the data and see how the male/female participation rates have changed over time, once I plotted the graph it was difficult to read and understand as the numbers significantly dropped every 2 years for the winter Olympics and then spiked back up for the Summer Olpympics so it was difficult to grasp the actual difference between males and females.

I then split this out into two categories, participation for Summer Olympics and for Winter Olympics. Once I plotted the graphs broken out it was much easier to read and its easy to tell that there was a huge gap in gender participation in the early years but since the 1980 there has been a steady climb in female participants so eventually this should become very close.

2: I wanted to see the countries with the most medals won over the entire Olympics period (this dataset covers 120 years of data). As expected in my hypothesis USA had a signifcant lead over other countries.

I then broke it down to see the Summer and Winter figures. Once again as expected for Summer games it was USA leading the pack. The winter games were led by Canada with USA coming in second and then followed by Scandavian countries such as Norway, Sweeden and Finland which is expected for the Winter games.

3: This question was a lot easier but it was something that I was interested in knowing. I simply assigned a variable of ages and ran a SQL query to return all the ages and the count of each of these ages and then grouped by ages. I then plotted this onto a line chart as its always easier to get an understanding of your data when you can visualize it.

Question 1:

Question 1: Is there more men or women competing in the Olympics, how has this changed over time

In [66]:
```python
pysqldf('''SELECT Sex, COUNT(Sex)
           FROM athlete_events
           WHERE Year >= 2000
           GROUP BY Sex''')
```

Out[66]:

	Sex	COUNT(Sex)
0	F	36040
1	M	49218

As expected there are recently more men competing in the Olumpics compared to women but lets take a look at how this has changed over time

In [67]:
```python
male = pysqldf('''SELECT Year,
                        CAST(SUM(CASE WHEN Sex = 'M' THEN 1 ELSE 0 END) AS INT) Male
                        FROM athlete_events
                        GROUP BY year''')

male.tail()
%store male
```

```
Stored 'male' (DataFrame)
```

In [68]:
```python
female = pysqldf('''SELECT year,
                          CAST(SUM(CASE WHEN Sex = 'F' THEN 1 ELSE 0 END) AS INT) Female
                          FROM athlete_events
                          GROUP BY year''')

female.tail()
%store female
```

```
Stored 'female' (DataFrame)
```

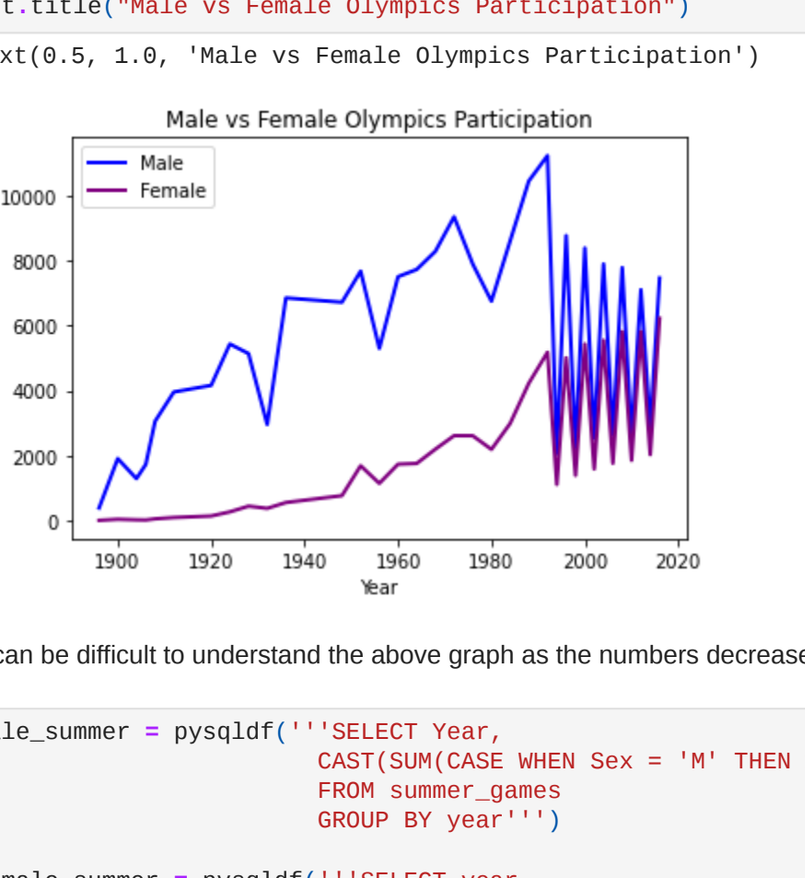
In [69]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (3.5.1)
Requirement already satisfied: packaging>=20.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (4.31.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (3.0.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.4.0)
Requirement already satisfied: cycler>=0.10 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.17 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.22.3)
Requirement already satisfied: six>=1.5 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [70]:
```python
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
```

In [71]:
```python
plt.plot(male.Year, male.Male, marker='', color='blue', linewidth=2, label='Male')
plt.plot(female.Year, female.Female, marker='', color='purple', linewidth=2, label='Female')
plt.legend(loc=2)
plt.xlabel("Year")
plt.ylabel("# Athletes")
plt.title("Male vs Female Olympics Participation")
```

Out[71]:
```
Text(0.5, 1.0, 'Male vs Female Olympics Participation')
```



It can be difficult to understand the above graph as the numbers decreased significantly every 2 years due to the switch between Summer and Winter games. I will split this out int two different graps, one for Summer and one for Winter and it will be much easier to read and understand.

In [72]:
```python
male_summer = pysqldf('''SELECT Year,
                        CAST(SUM(CASE WHEN Sex = 'M' THEN 1 ELSE 0 END) AS INT) Male
                        FROM summer_games
                        GROUP BY year''')

female_summer = pysqldf('''SELECT year,
                          CAST(SUM(CASE WHEN Sex = 'F' THEN 1 ELSE 0 END) AS INT) Female
                          FROM summer_games
                          GROUP BY year''')
```

In [73]:
```python
male_summer.tail()
```

Out[73]:

	Year	Male
24	2000	8390
25	2004	7897
26	2008	7786
27	2012	7105
28	2016	7465

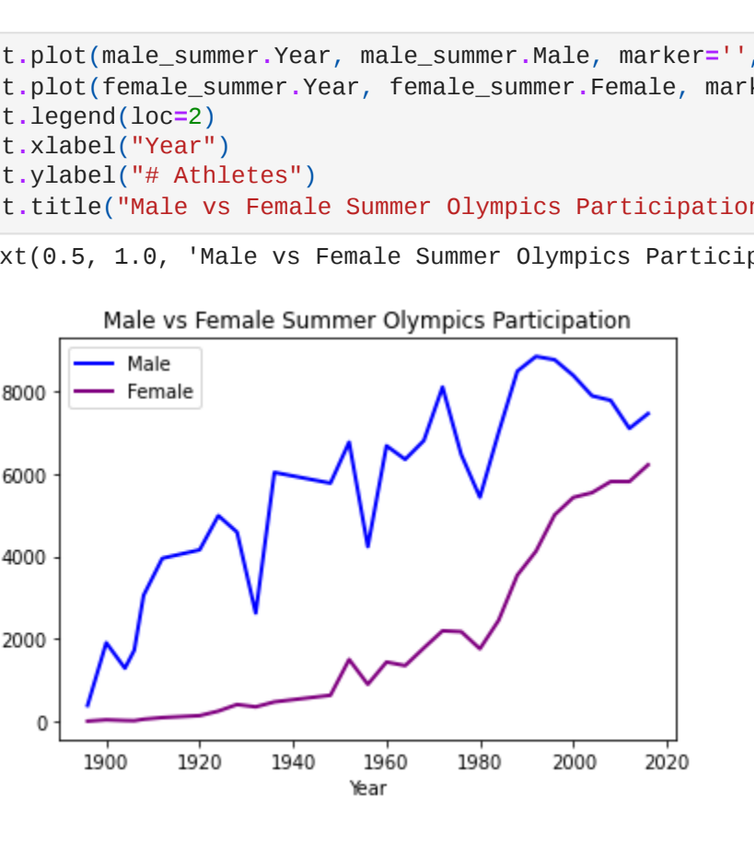
In [74]:
```python
female_summer.tail()
```

Out[74]:

	Year	Female
24	2000	5431
25	2004	5546
26	2008	5816
27	2012	5815
28	2016	6223

In [75]:
```python
plt.plot(male_summer.Year, male_summer.Male, marker='', color='blue', linewidth=2, label='Male')
plt.plot(female_summer.Year, female_summer.Female, marker='', color='purple', linewidth=2, label='Female')
plt.legend(loc=2)
plt.xlabel("Year")
plt.ylabel("# Athletes")
plt.title("Male vs Female Summer Olympics Participation")
```

Out[75]:
```
Text(0.5, 1.0, 'Male vs Female Summer Olympics Participation')
```



This graph is much easier to read now and its easy to tell that there was a huge gap in gender participation in the early years but since the 1980 there has been a steady climb in female participants so eventually this should become very close.

Now I will do the same for the Winter games.

In [76]:
```python
male_winter = pysqldf('''SELECT Year,
                        CAST(SUM(CASE WHEN Sex = 'M' THEN 1 ELSE 0 END) AS INT) Male
                        FROM winter_games
                        GROUP BY year''')

female_winter = pysqldf('''SELECT year,
                          CAST(SUM(CASE WHEN Sex = 'F' THEN 1 ELSE 0 END) AS INT) Female
                          FROM winter_games
                          GROUP BY year''')
```

In [77]:
```python
male_winter.tail()
```

Out[77]:

	Year	Male
17	1998	2221
18	2002	2527
19	2006	2625
20	2010	2555
21	2014	2868

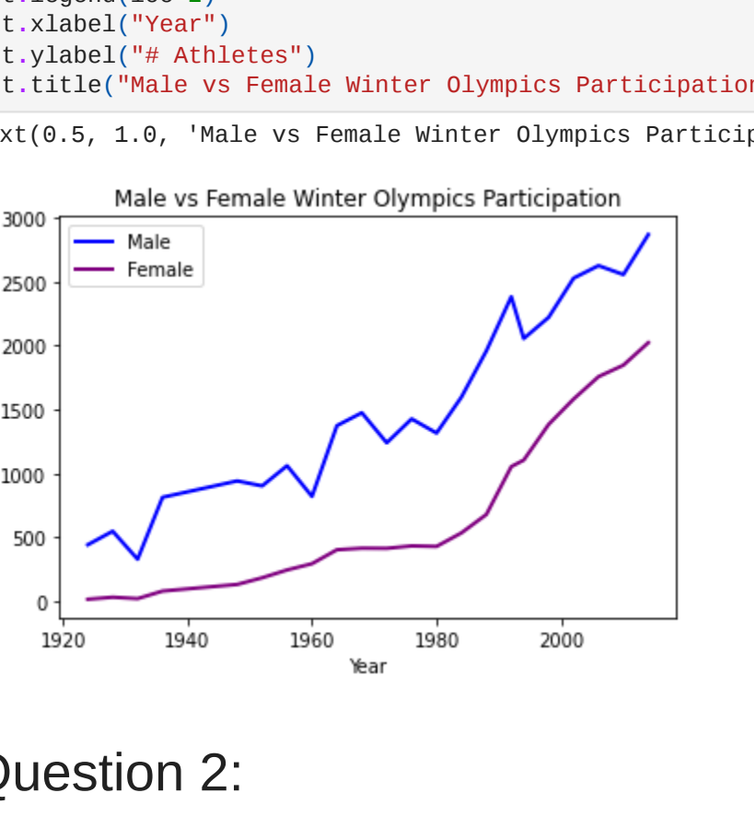
In [78]:
```python
female_winter.tail()
```

Out[78]:

	Year	Female
17	1998	1384
18	2002	1582
19	2006	1757
20	2010	1847
21	2014	2023

In [79]:
```python
plt.plot(male_winter.Year, male_winter.Male, marker='', color='blue', linewidth=2, label='Male')
plt.plot(female_winter.Year, female_winter.Female, marker='', color='purple', linewidth=2, label='Female')
plt.legend(loc=2)
plt.xlabel("Year")
plt.ylabel("# Athletes")
plt.title("Male vs Female Winter Olympics Participation")
```

Out[79]:
```
Text(0.5, 1.0, 'Male vs Female Winter Olympics Participation')
```



Question 2:

Question 2: Countries with the most medals and how they compare between Summer and Winter

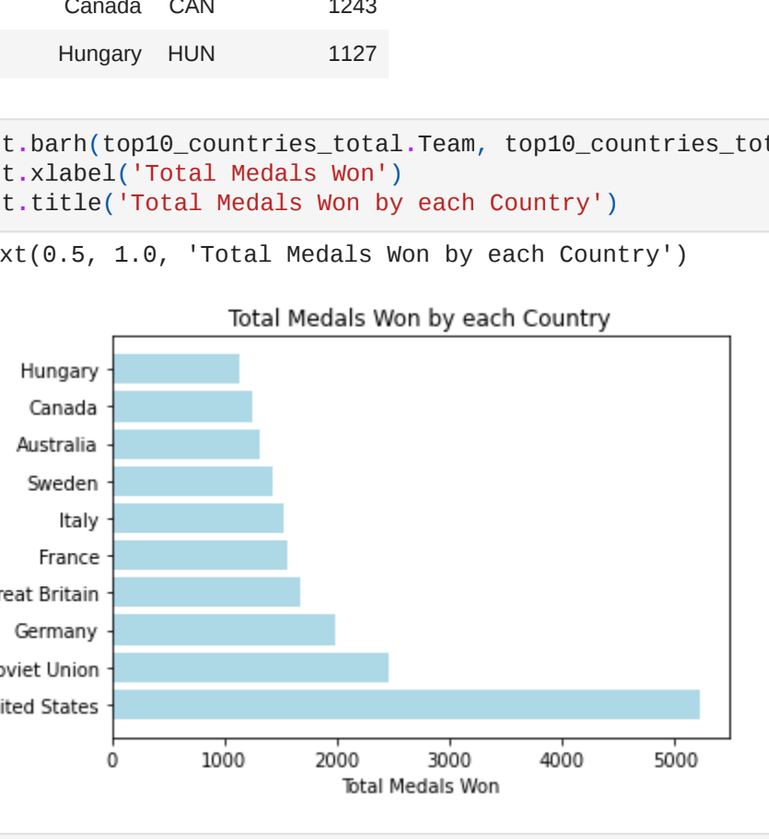
In [80]:
```python
top10_countries_total = pysqldf('''SELECT Team, NOC, COUNT(Medal) Total_Medals
                                  FROM athlete_events
                                  GROUP BY Team
                                  ORDER BY COUNT(Medal) DESC
                                  LIMIT 10''')

top10_countries_total
```

Out[80]:

	Team	NOC	Total_Medals
0	United States	USA	5219
1	Soviet Union	URS	2451
2	Germany	GER	1984
3	Great Britain	GBR	1673
4	France	FRA	1550
5	Italy	ITA	1527
6	Sweden	SWE	1434
7	Australia	AUS	1306
8	Canada	CAN	1243
9	Hungary	HUN	1127

In [81]:
```python
plt.barh(top10_countries_total.Team, top10_countries_total.Total_Medals, color='lightblue')
plt.xlabel('Total Medals Won')
plt.title('Total Medals Won by each Country')
```

Out[81]:
```
Text(0.5, 1.0, 'Total Medals Won by each Country')
```



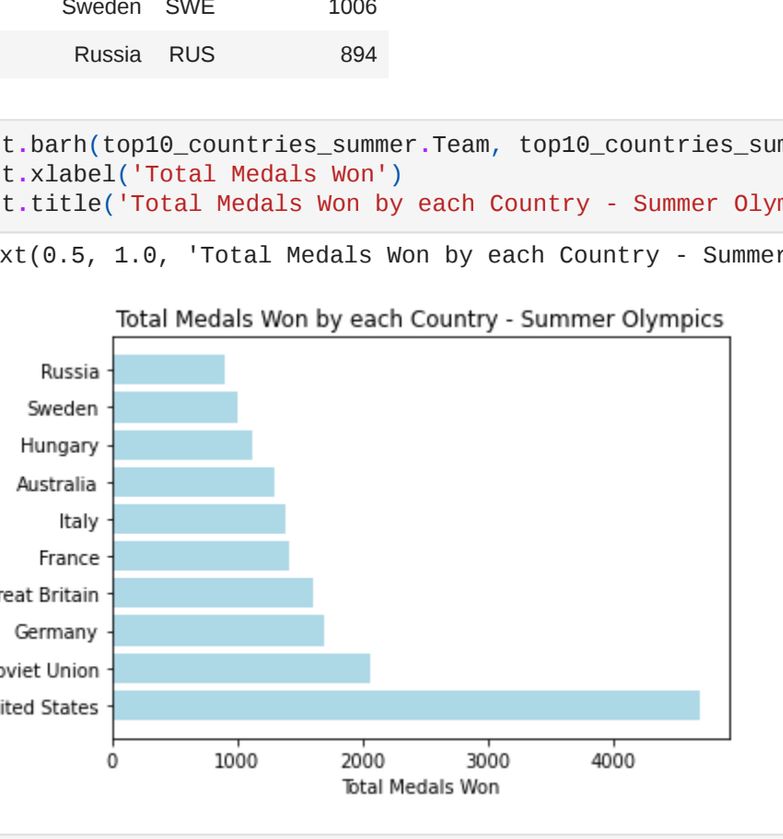
In [82]:
```python
top10_countries_summer = pysqldf('''SELECT Team, NOC, COUNT(Medal) Total_Medals
                                  FROM summer_games
                                  GROUP BY Team
                                  ORDER BY COUNT(Medal) DESC
                                  LIMIT 10''')
top10_countries_summer
```

Out[82]:

	Team	NOC	Total_Medals
0	United States	USA	4686
1	Soviet Union	URS	2061
2	Germany	GER	1687
3	Great Britain	GBR	1598
4	France	FRA	1408
5	Italy	ITA	1384
6	Australia	AUS	1290
7	Hungary	HUN	1123
8	Sweden	SWE	1006
9	Russia	RUS	894

In [83]:
```python
plt.barh(top10_countries_summer.Team, top10_countries_summer.Total_Medals, color='lightblue')
plt.xlabel('Total Medals Won')
plt.title('Total Medals Won by each Country - Summer Olympics')
```

Out[83]:
```
Text(0.5, 1.0, 'Total Medals Won by each Country - Summer Olympics')
```



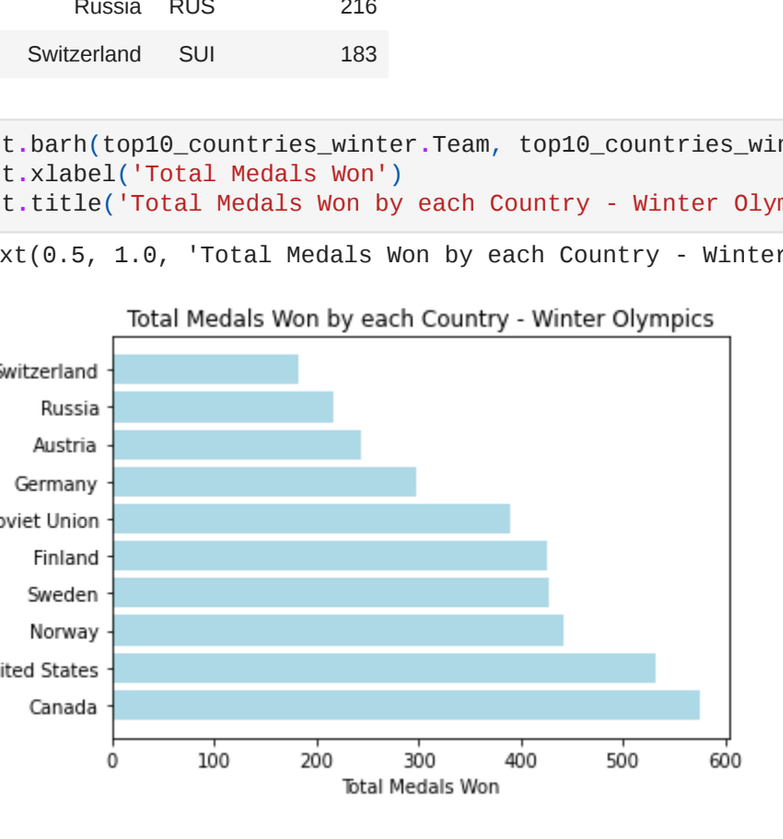
In [84]:
```python
top10_countries_winter = pysqldf('''SELECT Team, NOC, COUNT(Medal) Total_Medals
                                  FROM winter_games
                                  GROUP BY Team
                                  ORDER BY COUNT(Medal) DESC
                                  LIMIT 10''')
top10_countries_winter
```

Out[84]:

	Team	NOC	Total_Medals
0	Canada	CAN	575
1	United States	USA	533
2	Norway	NOR	443
3	Sweden	SWE	428
4	Finland	FIN	426
5	Soviet Union	URS	390
6	Germany	GER	297
7	Austria	AUT	244
8	Russia	RUS	216
9	Switzerland	SUI	183

In [85]:
```python
plt.barh(top10_countries_winter.Team, top10_countries_winter.Total_Medals, color='lightblue')
plt.xlabel('Total Medals Won')
plt.title('Total Medals Won by each Country - Winter Olympics')
```

Out[85]:
```
Text(0.5, 1.0, 'Total Medals Won by each Country - Winter Olympics')
```



Question 3:

Question 3: Age distribution, oldest and youngest Olympians

In [86]:
```python
ages = pysqldf('''SELECT Age, COUNT(Age) Count_Of_Ages
               FROM athlete_events
               GROUP BY Age''')

ages
```

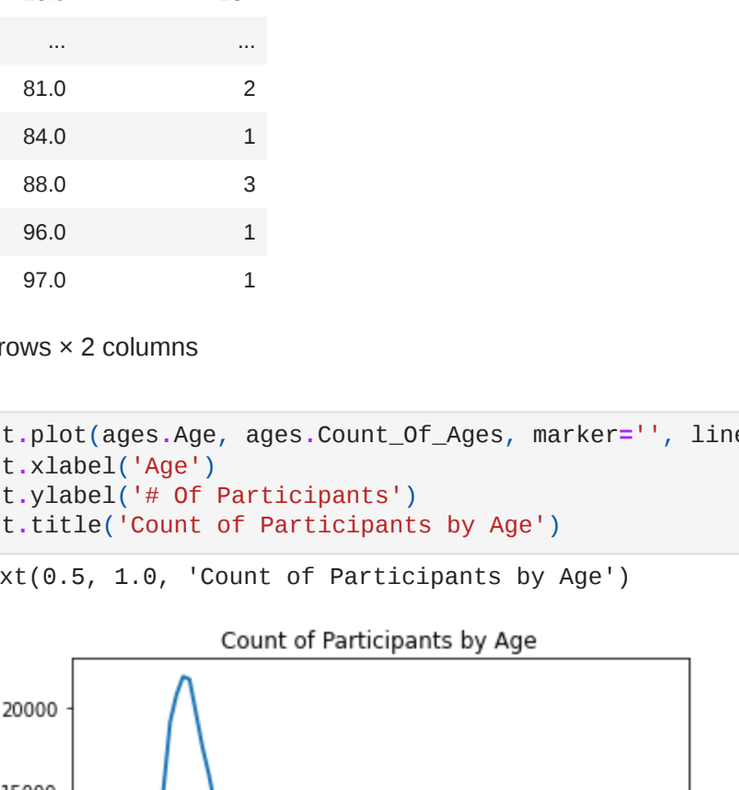
Out[86]:

	Age	Count_Of_Ages
0	NaN	0
1	10.0	1
2	11.0	13
3	12.0	39
4	13.0	187
...	...	...
70	81.0	2
71	84.0	1
72	88.0	3
73	96.0	1
74	97.0	1

75 rows × 2 columns

In [87]:
```python
plt.plot(ages.Age, ages.Count_Of_Ages, marker='', linewidth = 2)
plt.xlabel('Age')
plt.ylabel('# Of Participants')
plt.title('Count of Participants by Age')
```

Out[87]:
```
Text(0.5, 1.0, 'Count of Participants by Age')
```



In [88]:
```python
avg_age = pysqldf('''SELECT AVG(Age) Average_Age
               FROM athlete_events''')

avg_age
```

Out[88]:

	Average_Age
0	25.556898

In [89]:
```
pip install nbconvert
```

```
Requirement already satisfied: nbconvert in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (6.4.4)
Requirement already satisfied: pygments>=2.4.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (2.11.2)
Requirement already satisfied: beautifulsoup4 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (4.10.0)
Requirement already satisfied: testpath in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.6.0)
Requirement already satisfied: jupyterlab-pygments in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: nbformat>=4.4 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (5.2.0)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (1.5.0)
Requirement already satisfied: bleach in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (4.1.0)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.5.13)
Requirement already satisfied: traitlets>=5.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (5.1.1)
Requirement already satisfied: jinja2>=2.4 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (3.0.3)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.4)
Requirement already satisfied: jupyter-core in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (4.9.2)
Requirement already satisfied: defusedxml in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbconvert) (0.7.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jinja2>=2.4->nbconvert) (2.1.1)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (7.1.2)
Requirement already satisfied: nest-asyncio in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.4)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from nbformat>=4.4->nbconvert) (4.4.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from beautifulsoup4->nbconvert) (2.3.1)
Requirement already satisfied: six>=1.9.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from bleach->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: packaging in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from bleach->nbconvert) (21.3)
Requirement already satisfied: pywin32>=1.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jupyter-core->nbconvert) (303)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (0.18.1)
Requirement already satisfied: attrs>=17.4.0 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (21.4.0)
Requirement already satisfied: pyzmq>=13 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (22.3.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: tornado>=4.1 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\shay\appdata\local\programs\python\python310\lib\site-packages (from packaging->bleach->nbconvert) (3.0.7)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

In [90]:
```python
import nbconvert
```

In [ ]:

In [ ]: