Olympic Mini Project

(revised September 21, 2023)
Project 1: Programming & Data Visualization
Data, Tables (6.1-6.4), Visualization (7.1, 7.2), Cross-classifying (8.3)

In celebration of the Olympic spirit we will analyze trends in a data set which spans the from the 1896 Athens games to Rio in 2016. With this data we will explore trends in medals awarded, sports, and countries, as well as any host country advantage. The dataset is from Kaggle (https://www.kaggle.com), a Data science dataset, coding, and competition site. The mini-project represents your first chance to try out your coding and data skills to address specific questions without template code. Look to your previous labs and our work in class for ideas.

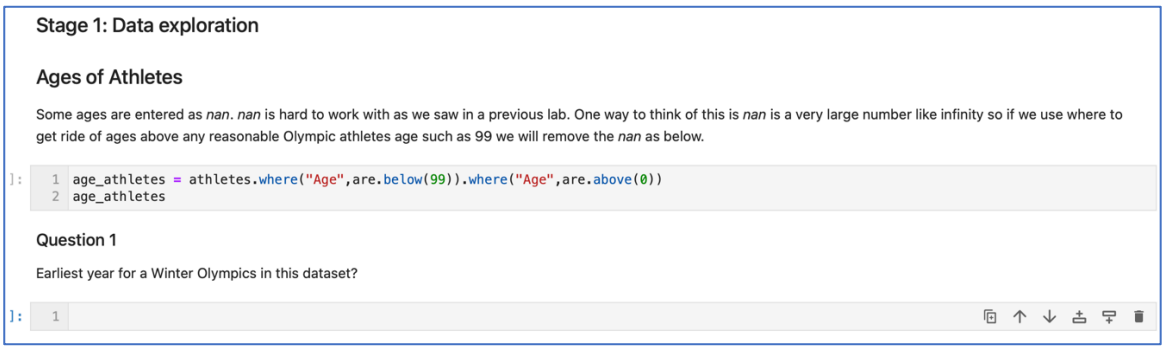
We will limit our project to data from the Winter Olympics by using the where method [**.where("Season","Winter")**] which leaves us with 18,923 individual athletes and 48,564 athlete/event datapoints (Many athletes compete in multiple events and/or over multiple Olympics). The dataset contains the following columns (Data source: Source: Kaggle https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results)

1. **ID** - Unique number for each athlete
2. **Name** - Athlete's name
3. **Sex** - M or F
4. **Age** - Integer
5. **Height** - In centimeters
6. **Weight** - In kilograms
7. **Team** - Team name
8. **NOC** - National Olympic Committee 3-letter code
9. **Games** - Year and season
10. **Year** - Integer
11. **Season** - Summer or Winter
12. **City** - Host city
13. **Sport** - Sport
14. **Event** - Event
15. **Medal** - Gold, Silver, Bronze, or nan

The dataset encodes missing values for athlete Age as “nan” but in the case of medals athletes who do not win Gold, Silver, or Bronze are also encoded “nan” which is a string. When examining ages for certain questions, we can remove which are labelled “nan” by restricted the “Age” column to between 0-99 using the **.where** method and **are.between(0,99)**. For this mini-project this case you will start with a new Python Jupyter notebook on the Temple.2i2c.cloud server based on the Olympic template. Be sure to use comments (lines starting with #) and markdown cells (create a new cell and change it to markdown from the top menu, see: https://www.earthdatascience.org/courses/intro-to-earth-data-science/file-formats/use-text-files/format-text-with-markdown-jupyter-notebook/)

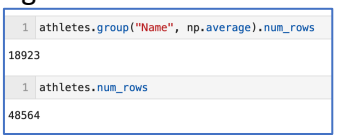
Stage 1: Data exploration

Address each question below with code cells and markdown cells. Make sure the markdown is clear so that each question and the results and analysis are well organized. For example see below:



1. What is the earliest year for a Winter Olympics in this dataset? Check this value (https://olympics.com/en/olympic-games/olympic-results), does it fit the data?

2. Examine the distribution of the age of all Olympians with a histogram. What do you find? A better way to get a view of the distribution of ages is a five number summary which includes the min, max, median, mean, and standard deviation. To get the five number summary (min, max, median, mean, and standard deviation) using np.min, np.max, np.median, np.mean, and np.std respectively on the corresponding column array or better yet create a function to compute and display this given arguments of table name and column label. Since a given athlete can appear in multiple events, a better way to examine the age distribution of athletes is to group the data by name using a function such as np.average as in below.



3. Now examine the age distribution of only gold medal winners with both a five number summary, and with a histogram. Compare to the distributions from question 2 and 3. Are gold medal winners older, younger, or about the same on average?

4. Now look at number of athletes and medals for top 10 countries. Use the *.group* method. What are the top ten countries in number of athletes? The "nan" in the "Medal" column simply means an athletes did not win a medal but still participated. Get the five number summary (min, max, median, mean, and standard deviation) again.

5. What are the top ten countries in number of Gold, silver, bronze medals, and total medals? You should have four sets of top ten countries for each of the scenarios. Again, get the five number summary (min, max, median, mean, and standard deviation). Hint: **`.where("Medal",are.not_equal_to("nan"))`** to get only medal winners. Consider how to create a column for the sum of the three medal categories.

6. What are the top 5 sports in terms of number of athletes?

7. Which sports (top 5) have awarded the most medals?

8. Which sports (top 5) awarded the most medals in Lake Placid, New York (1980, https://www.lakeplacid.com/do/activities/olympic-sites).

9. Remember medals are awarded to each participant on a team, how does this effect the results you found above?

Stage 2: Time trends and comparative results

1. Plot the trend in number of athletes per year.
   Hint: `athletes.group("Year").plot("Year","count")`

2. Plot the number of medals per year.

3. Plot the number of gold medals per year excluding “Ice Hockey”, why hockey?

4. Plot an overlay of gold, silver, and bronze medals as a function of year on the same plot excluding hockey.

5. Compare the US and Norway medal counts as a function of year by overlaying their counts. Hint: You could create separate tables for the US and Norway using an appropriate .where method. Now these tables can be combined using the Table .append method which merges two tables for instance, **NORUSA = US.append(Norway)**. You may also find *.pivot* useful.

6. Now use a scatter plot (**.scatter()**) to look at the number of athletes per year for the US versus that for Norway. What trends do you see?

7. Use a scatter to plot the number of athletes for each country versus the number of medals. See tip below.

Question 7. Use a scatter to plot the number of athletes for each country versus the number of medals.

Use the below datascience Table approach to create a column for whether an athletes has a medal, True or False.

To get a table with just US athletes, create the same for Norway and use the Table .append method to create a Table with only these two countries.

```
US = athletes.where('Team',are.equal_to("United States"))
```

```
athletes_NORUSA=athletes_NORUSA.with_columns("Medal num",athletes_NORUSA.column("Medal")!="nan","Athlete Number",1)
athletes_NORUSA
```

```
athletes_NORUSA.pivot("Medal num","Year")
```

Upload the .html and .ipynb files to Canvas to complete your work.