Practicing Data Visualization and Cutting Data II:

1. In Kaggle, go to the following dataset (https://www.kaggle.com/datasets/austinhinkel/47tuc-globular-cluster-line-of-sight-rand-subset) and click "New Notebook." This dataset contains Gaia data for the positions and motions of stars in the vicinity of an ancient globular cluster in the Milky Way: 47 Tucanae, or 47 Tuc for short. However, there are plenty of other stars that do not belong to the cluster which happen (by chance!) to be along the same line of sight.

2. Using your previous notebooks as guides, write code that imports numpy, matplotlib's pyplot, and pandas. Read in the data into a pandas DataFrame object named rawData. Use the len() function on this DataFrame to determine the number of stars in it. How many stars are there?

3. Now use the .head() method on your rawData DataFrame. What columns are available in the data?

4. Create a new column in your DataFrame called 'distance' and compute it from the parallax.

5. Make a plot of longitude (l) vs latitude (b) for all of the stars in your DataFrame.

6. Because 47 Tucanae is a globular cluster, its stars are gravitationally bound to it and therefore move through space together. As such, all of the clusters stars should share very similar velocities ("proper motions"). Let's create a new DataFrame object called data47Tuc and assign it to the result of a rawData.loc[(cut1)&(cut2)&…, :] command that keeps only stars with a very specific range of proper motions. Stars belonging to 47 Tuc have the following proper motions in the ra ("Right Ascension") direction AND the dec (Declination) direction:
    a. pmra must be less than 7.25 mas/yr
    b. pmra must be greater than 3.25 mas/yr
    c. pmdec must be less than -0.50 mas/yr
    d. pmdec must be greater than -4.50 mas/yr

7. Now that we have a new DataFrame object, let's plot longitude (l) vs latitude (b) again for the cut data. Do you notice a difference between the current plot and the plot from problem 5? Explain.

8. Finally, let's find the mean of the distance column in your data47Tuc DataFrame. To do this, use the following command: print(data47Tuc.loc[:, 'distance'].mean()). Keep in mind this distance will be in kiloparsecs (or kpc). Does your result agree with a quick search of the internet for the distance to this cluster? Explain.