Bootstrapping in Error Analyses: Determine the Distance to 47 Tuc

Background:

The 47 Tuc Globular Cluster is a relatively nearby globular cluster in the Milky Way estimated to be about 13 billion years old. Only visible from southern latitudes, 47 Tuc is near the Small Magellanic Cloud on the sky (though at very different distances). In this lab, we'll be using astrometric data to isolate the cluster from surrounding field stars and calculate a distance to the cluster. We will bootstrap in order to estimate an uncertainty in our distance determination.

Please see http://simbad.u-strasbg.fr/simbad/sim-id?Ident=47+Tuc for more.

Skills:

Pandas makes bootstrapping very easy thanks to built-in sampling functions. You may need some or all of the following, depending on your approach.

- To read in a csv file and inspect the results:

```
#Read in data from .csv into a Data Frame
data = pd.read_csv("/kaggle/input/myDataIsHere.csv")
#Display first 5 rows of data:
data.head()
```

- To create a new column in a data frame:

```
#Create a new column from existing columns
data['newColumn'] = data.column1 - data.someColumn
```

- To make a cut on one Data Frame and store the results in a new Data Frame:

```
#                ----cuts go here------                          , the ':' means all columns
Tuc47 = data.loc[((cut1)&(cut2)&(some cut here)&(data.distance < 15)), :]
```

- To obtain a sample of a Data Frame or Pandas Series. Note n is the number of data points you'd like to draw (with replacement) and random_state just lets you pick a random number to make your random sampling reproducible.

```
Tuc47_bootstrap = Tuc47.distance.sample(n = 12442,
                                        replace = True,
                                        ignore_index = True,
                                        random_state = i)
```

- To "convert" a Python list into a pandas Data Frame object:

```
#create a new, empty Data Frame object
newDF = pd.DataFrame()
#create a new column in this DF object and assign a list to it
newDF['createThisColumn'] = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

Problems:

1. The Simbad database has a parallax measurement for 47 Tuc. What distance should we expect to find when we eventually measure the distance to the cluster?

2. Design a query to obtain the following data from the Gaia DR3 database:
    a. Columns:
        i. Parallax
        ii. Proper Motion in the Right Ascension direction
        iii. Proper Motion in the Declination direction
        iv. Heliocentric longitude (l)
        v. Heliocentric latitude (b)
    b. Conditions:
        i. About two degrees in galactic longitude (l) centered on the central value in Simbad
        ii. About two degrees in galactic longitude (b) centered on the central value in Simbad
        iii. Parallax greater than zero (removes bad parallax measurements)
        iv. Random_index < 300000000
    c. Download your query as a .csv file.

3. Create a Kaggle notebook and upload the csv file in the data panel. Read in the csv file into a pandas Data Frame object named 'data'. Create a new column named 'distance' and calculate this from the parallax measurement of each star.

4. Neglecting internal motions, the stars in the cluster are co-moving through space. Including internal motions causes a spread in proper motions compared to the single pair of proper motion values given in Simbad. Plot the proper motions and reference Simbad in order to come up with a series of cuts to apply to your data frame such that you cut away most field stars while keeping as many cluster members as possible. Name the new Data Frame 'Tuc47'. Let's also cut out any stars that are 15 kpc away or further, as we know the 47 Tuc globular cluster is inside the Milky Way:

```
Tuc47 = data.loc[((cut1)&(cut2)&(some cut here)&(data.distance < 15)), :]
```

5. With your Tuc47 Data Frame, find the mean distance to the member stars, in units of kpc. What do you find?

6. Since there is only one 47 Tuc cluster, we can't "repeat the experiment" several times like experimentalists can. Instead, we can bootstrap in order to estimate the uncertainty in our distance determination. That is:
    a. Determine the number of stars in your Tuc 47 object.
    b. Make use of the built-in pandas method (function) sample() in order to create a new, bootstrapped sample.
    c. Find the mean of the new bootstrapped sample and append it to a list.
    d. Repeat steps b and c until you have a list of 50 different means.
    e. Determine the mean (.mean()) of this array and the standard deviation (.std()), and report your final distance assessment for 47 Tuc.

7. How does your distance determination compare to the result in Simbad?