Statistical Methods: Assignment 2

This is the second of three assignments, for which extensive help is available during the tutorials. It is worth 20% of the final course grade.

The deadline for this assignment is the end of Sunday 23 Jan (23:59h). Late work which is submitted up to a day after the deadline will receive a penalty of -20% of the awarded grade, while work submitted 1-2 days after the deadline will receive a penalty of -40% of the awarded grade. Work submitted later than this will not be assessed. The rubric for the assessment of all the assignments, listing the categories assessed and the requirements for each of them, will be provided separately on Canvas.

What you should submit
You should submit your work via Canvas. ***It must be in the form of two Jupyter notebooks, one for part A and one for part B.*** Make sure that you upload the correct files, and check that all the cells run successfully (***and in the correct order***, from start to finish) before you submit them!

When answering each question, use a cell with markdown to briefly explain your approach (a few sentences is fine) along with any assumptions you made. Explanations of your code should be included as comments (or docstrings for functions) in the code cells. Following your code and results obtained should summarise your findings (and interpret them if needed) in a markdown cell following the code cells which produce your results.

Remember that the usual plagiarism rules apply to your work: if you cut-and-paste code from somewhere/someone else you must cite the source (simply replacing variable names is not sufficient to make it your own!). We make an exception to this rule for code from the course's own material, which we allow you to use without citation. We also expect you to help each other, at least early on, and/or be inspired by methods you see online, so programming ***your own version*** (i.e. not cut and pasted) of someone else's method is fine and does not require citation.

Part A (10% of final course grade)

Do the programming challenge at the end of each episode, 6-9. Each challenge contributes 2.5% of the final course grade.

Part B (10% of final course grade)

For this part of the assignment you will be using a data file which contains astrometric and photometric data for 401448 stars observed by the Gaia mission, which have been identified as belonging to 1229 open star clusters in our galaxy. Gaia, launched in 2013, is ESA's successor to the Hipparcos astrometry satellite. Gaia continuously scans the sky using two telescopes set at a precisely known (via an on-board laser interferometer) angle to each other. The relative positions of many stars, focused on to two CCDs allows a precise astrometric solution that can measure the angles between the stars down to a precision of tens of microarcseconds – note that one microarcsec is $\approx 5 \times 10^{-12}$ radian! Besides measuring extremely accurate positions on the sky, Gaia provides measurement of the star's proper motion – its movement on the sky – and parallax – the annual angular motion against distant background objects as the Earth (and satellite) moves in its orbit. The parallax can be used to directly estimate the distance to the star.

The data you will be using is taken from an analysis of Gaia's Data Release 2 (DR2), a database of 1.3 billion sources (c.f. 2.5 million for Hipparcos!) which includes the astrometric data as well as photometry in 3 optical bands[1]. The open cluster study used a sophisticated clustering analysis of the astrometric information to identify which objects belong with high probability to a known open cluster that is along the line-of-sight. In this way, they were able to compile a list of cluster members for each of 1229 open clusters, which is contained, along with the Gaia astrometric and photometric data in `cluster_members.txt`.

[1] https://www.cosmos.esa.int/web/gaia/iow_20180316

The data columns are described in the additional file `data_description.txt`. The astrometric data are especially complex, with the method for making the astrometric solution leading to correlations between the different astrometric quantities. However, here you will only be focusing on the parallax measurements `plx` and `e_plx`, alongside the photometric apparent magnitude in the G band `Gmag` and the photometric colour `BP-RP`. Note that the magnitude is a logarithmic measure of the flux with a scale runs backwards, i.e. brighter sources have smaller magnitudes[2]. The cluster member data is limited to stars brighter than G magnitude 18. You will also use the `Cluster` name which the star is associated with and the membership probability `PMemb` which is less than 1 for a number of outlier cases that are not certain to be associated with that cluster.

Now do the following. All questions are equally weighted:

1. Read the entire data file on cluster members into a Pandas dataframe. When using `read_csv` you should include the argument `na_values='---'` so that the missing data (which are given this format in the file) are recorded as `NaN`. Then clean the dataframe of any rows with `NaN` values (this should remove nearly 5700 objects). From your cleaned dataframe, make a new dataframe for the data for cluster NGC2506, removing from it any stars with `PMemb`<1. Make a scatter plot of the apparent G magnitude vs BP-RP colour, putting the magnitude on the y-axis, which you should invert so that the lower-magnitude (brighter) stars are at the top of the plot). The resulting colour-magnitude diagram will be reminiscent of the Hipparcos colour luminosity plot from Episode 6, although with the stars from the top left missing, since the Hipparcos data was for field stars of all ages, while the open clusters have a specific age so that the bluest, most luminous stars (also shortest-lived) have already reached the end of their lives are are no longer seen.
2. We are going to use the parallax measurements to estimate the distance to NGC2506, but first we want to check that there are no flux-dependent biases in the parallax which might affect our results:
    a. First, plot a scatter plot of the parallax vs. the G magnitude for NGC2506. It should have a funnel-like appearance - briefly try to explain why it looks like that and why we would not expect to see this pattern for field stars (i.e. those along the line of sight but not associated with a single cluster).
    b. Next, use an appropriate test on your NGC2506 data (using one of the methods described this week) to determine whether the parallax depends on the G magnitude.
3. Now use the NGC2506 parallax data with Bayes' theorem, to calculate the posterior pdf for the distance d (in kpc) to NGC2506, using the formula $d = 1/p$ where p is the parallax in milliarcsec (mas). Gaia has a known 'zero-point' offset - a systematic error – in the parallax, so before you do your calculation you should first add a correction of 0.029 mas to the parallax measurements. You may assume that the corrected parallax measurements are normally distributed about the true parallax, with standard deviation given by the errors on the parallax measurements. Plot your posterior pdf and determine the 1-σ confidence interval on the distance and plot the interval on your pdf.
4. Finally, choose another open cluster in the data set (your choice!), remove stars with `PMemb`<1 and repeat question 3 to obtain the posterior distribution. Then plot this cluster and NGC2506 on the same colour-magnitude diagram, but using absolute G magnitudes (corrected to a common distance of 10 pc)[3], so that you can compare the diagrams for each cluster. For the purposes of estimating a distance, you may assume the best distance for each cluster corresponds to the maximum of the posterior pdf (known as the 'maximum likelihood estimate').

Hints:
To reshape a pandas data column into a reshaped numpy array use the `.values` method, e.g.:
`ngc2506['plx'].values.reshape(len(ngc2506),1)`

[2] The magnitude system was first developed in ancient Greece by Hipparcos (the original version). It was set up so that fainter stars could be added at the top of the scale, i.e. using larger numbers, because negative numbers had not been discovered at the time. The scale is logarithmic due to the response of the human eye.
[3] For absolute and apparent magnitudes M and m respectively: $M = m - 5\log_{10}(d) - 10$ if d is in kpc.