Pre-ThaiPASS exercise

August 2024

Instructions: All attendees of ThaiPASS'24 should attempt the following questions before arrival. Working on these questions will ensure that you have the necessarily basic astronomy knowledge and Python skills for ThaiPASS (e.g. plotting, working with data files, installing new libraries). These questions don't require an expert knowledge in astronomy or Python.

Selected answers are given at the back.

We are happy to answer questions! Please write to the instructor for each question directly. The instructors are:

1. Dr Dimitris Stamatellos dstamatellos@uclan.ac.uk

2. Dr Krittapas Chanchaiworawit krittapas@narit.or.th

3. Dr Claire Cashmore c.cashmore@hull.ac.uk

4. Dr Teeraparb Chantavat teeraparbc@nu.ac.th

For other non-technical and logistical queries about ThaiPASS, write to Dr Siri Chongchitnan siri.chongchitnan@warwick.ac.uk.

1. **Orbits of exoplanets** *by Dr. Dimitris Stamatellos*

   Kepler's laws describe the motions of the planets in the Solar System. They were formulated by Kepler in the 1600's (see this Wikipedia article).

   A simplified version of Kepler's third law that connects the planet's orbital period, P (the time it takes for the planet to make a full orbit around its host star), to the planet's distance from the star, a, is as follows:

$$P^2 = \frac{a^3}{M_\star}\,, \tag{1}$$

   where $M_\star$ is the mass of the host star. In this equation, the period of the planet needs to be in years, the distance from the star in AU (astronomical unit; the distance between the Earth and the Sun), and the mass of the star in solar masses.

   (a) In Python, define a function that returns the distance of a planet from its host star when you input the planet's period P and the mass M of the star (input units should be years and AU; output units AU). Don't forget to import the relevant packages (numpy and matplotlib.pyplot).

   Use this function to find and output the distance between the Earth and the Sun, and the distance between the Jupiter and the Sun (note that the period of Jupiter is 12 yr).

   (b) Use this function to make a numpy array containing the semi-major axis lengths of planets with orbital periods between 0.1 and 5 years (every 0.1 yr). Do this for stars between 0.1 and 2 times the mass of the Sun (for 5 values between 0.1 and 2)

   Hint: Use the numpy functions `arange` and `linspace`.

   (c) Plot the semi-major axis versus the period of the planet for each of the 5 stars (5 lines), using logarithmic axes. Label your axes, give a title to the plot, use different colour lines for each stellar mass. Provide a legend for your graph.

2. **Exploring properties of AGN** *by Dr. Krittapas Chanchaiworawit*

   Active Galactic Nuclei or AGN are a class of objects constituting (at least) an accreting central engine (supermassive black hole) residing within a host galaxy. The energy released from this central engine is in both kinematics (winds, outflows, and jets) and electromagnetic energy (from X-ray to infrared). However, not all galaxies with supermassive black holes have AGN. They could, for example, possess these AGN briefly in their lifetime. So, they are quite sparse in their distribution in space compared to 'normal' galaxies.

   Try the following question to gain a basic understanding of the distribution and physical properties of AGN. It's not essential to complete all questions, but do as many as you can.

   (a) Download this table containing data of low-luminosity AGN. Then, use Numpy's `loadtxt` function or Pandas' `read_csv` function to read the content of the table into your Python notebook.

   (b) Use Matplotlib's `pyplot.scatter` to plot the distribution of the AGN in sky projection (i.e. a plot of Dec against RA).

   Hence, find the average number of such AGN per square-degree. **Hint:** remember that the RA values are closer in physically as the Declination increases northward and southward.

   (c) Identify the population of AGN with signal-to-noise in the i'-band (snMedian_i) larger than 40 and which are apparently brighter than 16 magnitude, with redshift < 0.4. How many such AGN are there in the table?

   Pick a set of fiber ID, plate number, and plate MJD (mjd_plate) of these AGN. (**Hint:** use `numpy.where`.)

   (d) Install `AstroPy` and `Astroquery`.

   Then, use SDSS class `get_spectra` to query the spectrum of the selected AGN using its fiber ID, plate number, and plate MJD. Then, plot the flux density (y-axis) and wavelength (x-axis) in linear scale. (**Hint:** The result of the query will be in HDU List. There are multiple layers in this file. Try reading "header" of layer[0][1] and [0][2]. This will give you the idea of where to find wavelength, flux density, redshift, and so on.)

   (e) Locate the Balmer emission lines on the spectrum (such as Hα and Hβ). Then, fit a simple Gaussian profile to one of these emission lines using `SciPy`'s `optimize.curve_fit`.

   What is the Full-Width at Half-Maximum of this line in km/s. (**Hint:** Remember that the object is redshifted.)

3. **Orbits of stars** *by Dr. Claire Cashmore*

   Stars move in orbits around the centre of mass of a galaxy due to the force of gravity. The magnitude of the orbital velocity depends on the enclosed mass (the mass inside the orbit) and the distance of the star from the centre of the galaxy (the radius). In this exercise we will explore the orbital velocity of stars in a galaxy and construct a rotation curve. There are two data files that you will need for these tasks, you can find them on the ThaiPASS'24 GitHub page.

   (a) The file rc100224.dat contains the orbital velocities of stars in a galaxy and their radial distance from the centre of the galaxy. The first column of the file contains the radius (kpc: kiloparsec) and the second column contains observed orbital velocity (km/s: in kilometres per second). Read in the data from the file and plot the radial distance on the x-axis and the orbital velocity on the y-axis.

   (b) Now you will use real mass distribution data from a model of the Andromeda galaxy (M31) to calculate the orbital velocities of stars. The mass distribution is of only baryonic matter (stars and gas). You will use the mass distribution to calculate and plot the rotation curve of the galaxy.

   You should use the data file M31_mass2.dat

   The first column of the file contains the radius (pc: parsec) and the second column contains the mass at that radius.(Note that the units of radius are different here, there are 1000pc in 1kpc, ensure to use units of kpc in equation 2).

   You can then use the equation below, derived from Kepler's Laws of Planetary Motion to calculate the orbital velocities of stars as a function of radial distance and plot this as a line graph.

$$v = \sqrt{\frac{GM(r)}{r}} \tag{2}$$

   You can use the following value for the gravitational constant, $G = 4.30091 \times 10^{-6} \frac{kpc km^2}{M_\odot s^2}$.

   From the data file, remember that you need to use the cumulative/enclosed mass, $M(r)$, so add up the mass contained inside the sphere as the radius increases (hint: use a loop to do this). The mass for a star at a given radius should include the total mass within that radius.

   You should then compare your graph to the one you made in the first task and comment on any differences you find.

   We will discuss the reasons for this at ThaiPASS'24!

4. **Useful Python packages for cosmology** *by Dr. Teeraparb Chantavat*

   Your task is to install Python packages that we shall use in the cosmology session at ThaiPASS. They are `GetDist` and `CLASS`.

   (a) **GetDist**: Follow the instructions at https://github.com/cmbant/getdist for installing GetDist on your machine.

   After installing, try the tutorial at https://getdist.readthedocs.io.

   I have prepared the data for you here. The data comprises 6 columns $x_1, y_1, x_2, y_2, x_3, y_3$ (i.e. three pairs of data). Plot the 2D contours of all the data in one plot using GetDist. We will learn about GetDist in more details in our session at ThaiPASS.

   (b) **CLASS**: The Cosmic Linear Anisotropy Solving System (CLASS) is a code to calculate many important quantities in cosmology especially something we call "power spectra". In cosmology, there are many types of power spectra, including "linear power spectrum" (when discussing large-scale structure formation) and "angular power spectrum" (when discussing the cosmic microwave background), and many more. We will learn about these two power spectra in our session and discuss why they are so important in understanding the history of the Universe.

   CLASS can be downloaded at https://lesgourg.github.io/class_public/class.html. The installation guide can also be found on that website. Warning: installing CLASS could be tricky especially for Mac users. (Don't worry if you don't manage to install CLASS on your machine.)

Answers

1. (**Orbits of exoplanets**) Your graph should look something like fig. 1.

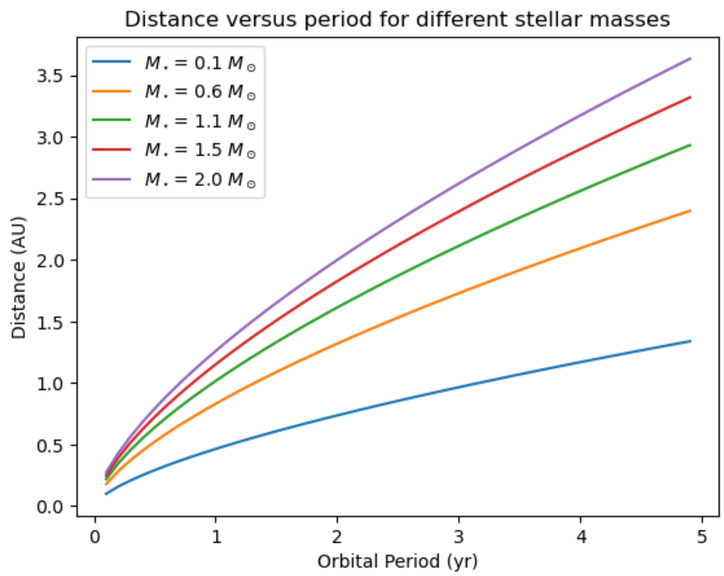

Figure 1: Distance vs. period for planets orbiting stars of various masses.

2. (**Properties of AGN**)

   b) On average, the number of these AGN is 3-4 objects per square degree. (How does this compare to the number of galaxies down to i'=21 magnitude?) The distribution is shown in fig. 2.

   c) The total number of 'bright' AGN from this data set is 329.

   d) Fig. 3 shows an example of a spectrum of an AGN.

   e) Fig. 4 shows a simple Gaussian profile fit to an emission feature of the AGN spectrum shown in fig. 3.

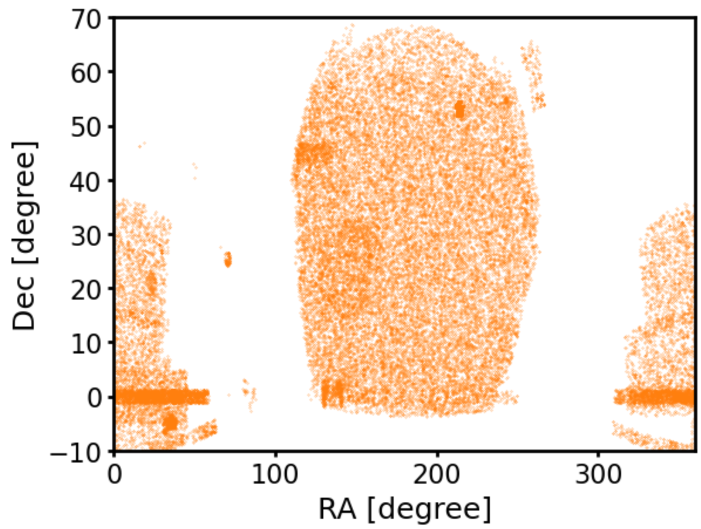

Figure 2: Distribution of low-luminosity AGN in sky projection.

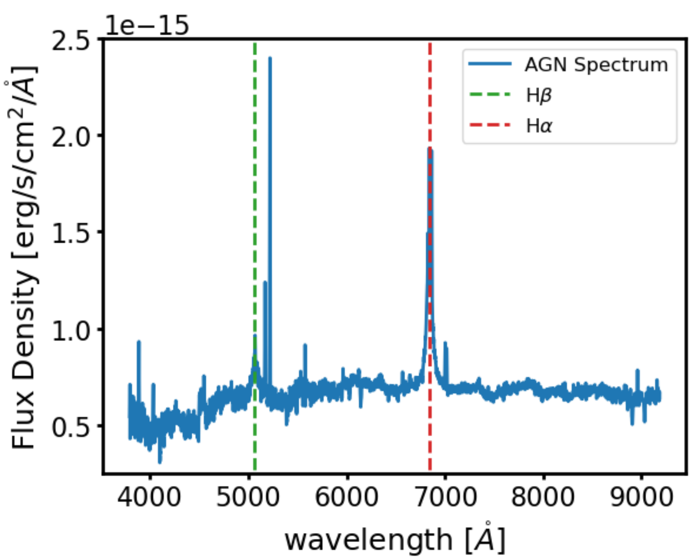

Figure 3: Spectrum of an AGN with ObjID 1805993529994930176 (SpecID). The green and red dashed lines indicate the Hβ and Hα emission lines, respectively.

.

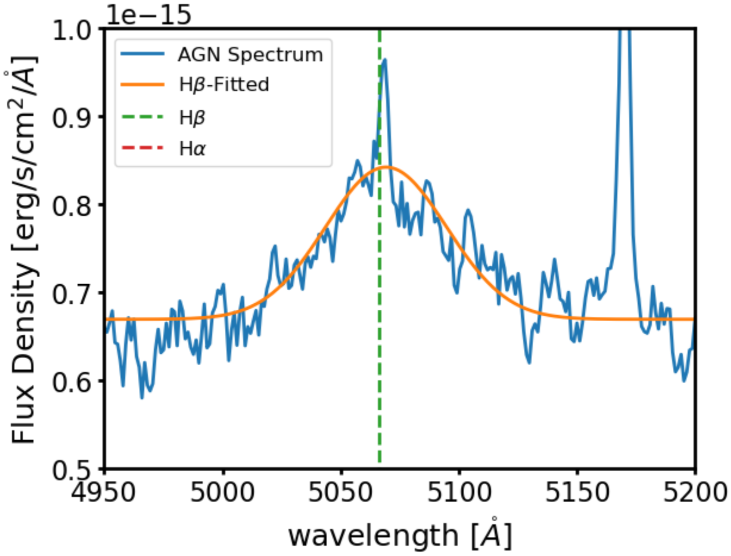

Figure 4: The best-fit Gaussian profile of the Hβ emission of the same AGN as in fig. 3.

3. (**Orbits of stars**)

   Your graph should look like that shown in fig. 5

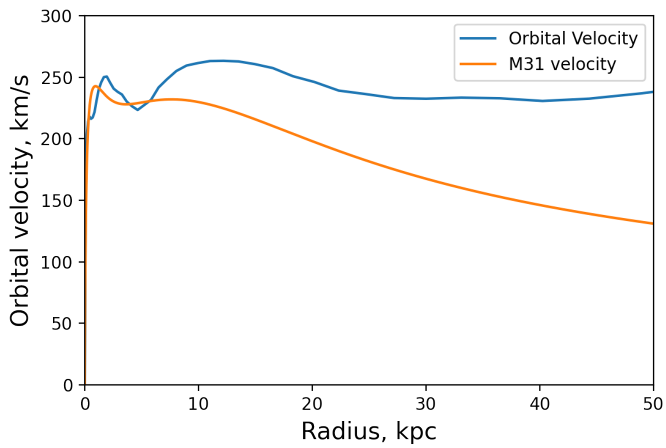

Figure 5: Graph for question 3 showing the orbital velocity of stars form observational data (blue line) and a model of the orbital velocity of the stars calculated considering the baryonic components only (orange line)

4. (**Useful Python packages for cosmology**)

a) Your contour plot should look like a little flower!

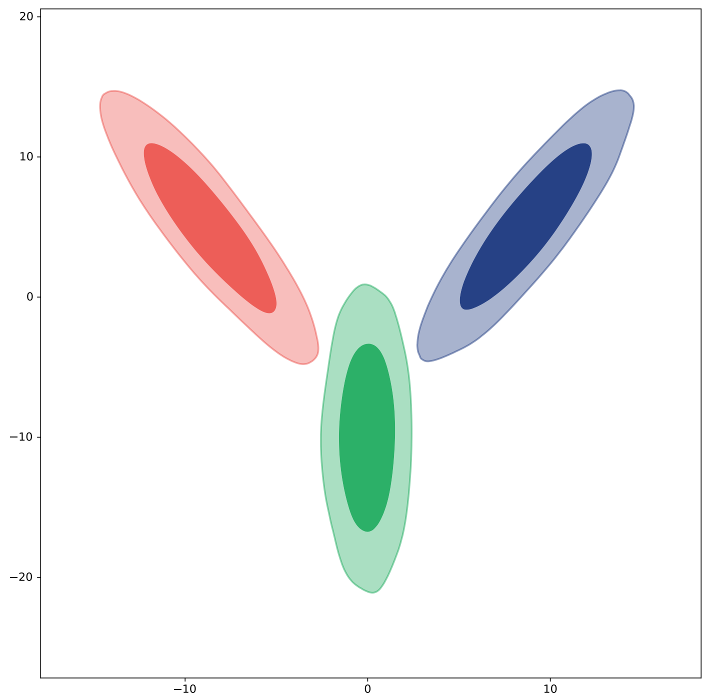

Figure 6: Solution to the GetDist exercise.