HomeworkE-exoplanetData

June 29, 2022

Due on Tuesday July 5, at 5:00pm.

In this homework, you're going to use some exoplanet data to practice reading in data files, storing data in `numpy` arrays or dictionaries, and making plots. Please read the instructions *carefully*, and complete all the requested steps in a jupyter notebook. Include all the code you write as "Code" cells and any written responses as "Markdown/Raw NBConvert" cells. Name your notebook (up at the top) `homework#_{identikey}` (replacing `{identikey}` with your identikey and `#` with the appropriate letter for this week's homework).

In your notebook, please indicate clearly at the start of a block of code which part of the assignment it is in response to.

Be sure to use comments throughout your code to explain how it works, variables, etc. You don't need to include full docstrings for any functions you write, but it *is* good practice! (1 pt)

As much as possible, make sure you're having Python do your calculations/limits/scaling for you. Anywhere you find yourself hardcoding a number, think about whether or not you could have Python calculate it dynamically for you. *(E.g. The difference between range(5) and range(len(masses)) in previous homework).*

1 Physics Background: Exoplanets, Known and Unknown

We now know of thousands(!) of extrasolar planets that transit (pass in front of) their host stars. For these planets, we can typically measure the planet's radius (from how much starlight it blocks) and orbital period (from how frequently the transits repeat). Many of these planets have been found by NASA's Kepler & TESS missions; some have been found by other projects.

The fraction of star light that a planet blocks, often known as the transit depth D, is the ratio of cross-sectional areas, given by

$$D = \frac{\pi R_{planet}^2}{\pi R_{star}^2} = \left(\frac{R_{planet}}{R_{star}}\right)^2$$

where R_{planet} is the planet's radius and R_{star} is the star's radius. A typical Jupiter-sized planet transiting a Sun-like star has a transit depth of 0.01, meaning it blocks 1% of the star's light.

The period of a planet tells us the distance at which the planet orbits its star (via Newton's and Kepler's laws). If that orbital distance is a and the luminosity of the star is L_{star}, then the energy flux F_{planet} the planet receives from the star is given by

$$F_{planet} = \frac{L_{star}}{4\pi a^2}$$

Expressed in units of the *insolation* (**in**coming **sol**ar radi**ation**) Earth receives from the Sun, this can be rewritten as

$$\frac{F_{planet}}{F_{Earth}} = \frac{L_{star}}{L_{Sun}} \times \left(\frac{1AU}{a}\right)^2$$

where F_{earth} is the insolation Earth receives from the Sun and L_{Sun} is the luminosity of the Sun. If a planet receives more than $1.5\times$ or less than $0.25\times$ Earth's flux, it probably can't have liquid water at its surface. We sometimes say that planets that are in this range are in their stars' "habitable zones."

2 Coding Assignment: Exoplanet Data

1. (2 pts) Download the plain text data files `exoplanets.csv` and `newplanet.lightcurve` from Canvas, and save them in the same directory as your `jupyter` notebook for this assignment. If you are on Jupyterhub, you'll need to upload them into your tutorial directory. Have a look at both files using a text editor, the Jupyter interface, or (if on JupyterHub, MacOS, or Linux) using the Linux command `!head <file name>` in a notebook code cell. It is always important to know what a data file looks like before you try to read it with code. In a Markdown/RawNBConvert cell, briefly describe, in your own words, how are each of these text files organized?

2. (3 pts) Read the `exoplanets.csv` file into Python. You can use any method we've discussed in class that you like, as long as the exoplanet names are stored in an array of strings and the other columns are stored in arrays of floating point numbers. This file contains many of the known transiting planets.

3. (3 pts) Make a plot of this exoplanet population, with the planet insolation (the energy flux received by the planet) on the x-axis and the planet radius (in Earth radii) on the y-axis. Make sure to label the axes, and set the scale of both the x and y axes to be logarithmic (instead of linear).

4. (3 pts) Make another plot, this time plotting the planet radius on the x-axis and the transit depth (the fraction of starlight the planet blocks) on the y-axis. Be careful with your units! Have a look at the units for the radii in the data file. Use log-log axes again, and set the limits of the y-axis of the plot to go from 10^{-6} to 1.

5. (2 pts) Make a Python dictionary called `exoplanets` that uses the names of exoplanets as keys, and their radii as values. You may want to start with an empty dictionary, and loop through elements in your arrays to populate the dictionary entries. Run the following commands to test that your dictionary works.

   ```
   print(exoplanets['Kepler-42b'])
   print(exoplanets['Kepler-42c'])
   print(exoplanets['Kepler-42d'])
   ```

6. (4 pts) Read the plain text data file `newplanet.lightcurve` into `numpy` arrays. This file contains the light curve (brightness as a function of time) measured for a new exoplanet. The time is expressed relative to the middle of the transit (in units of days), and the flux is expressed relative to the typical brightness of the star when there is no planet in front of it (e.g. out-of-transit flux = 1.0 = 100%). Make a plot of this light curve, and label the axes.

7. (3 pts) Look at your light curve plot. *By eye*, estimate the fraction of light the planet blocks when it's in front of the star (getting this to one significant figure is fine). Record this fraction in a variable in your notebook; this is the transit depth. *Note, you should be estimating the fraction of light LOST, not how much remains.* Using Python, calculate and store the physical radius of the planet (in Earth radii), assuming the star has a radius of 0.20 solar radii. Give this planet a name of your choosing(!), and add its radius to your dictionary of exoplanet radii.

8. (3 pts) Assume this planet has an orbital separation of $a = 0.11$ astronomical units, and that the star has a luminosity that is 0.004 times that of the Sun. Calculate (using Python) the insolation this planet receives from its star, relative to the amount Earth receives from the Sun. Remake the Radius vs Insolation plot you made in Q3, but then add in your new planet (giving it a different color and larger symbol size, so it stands out).

9. (1 pt) Briefly discuss what you find interesting about this new planet, the population of known exoplanets as a whole, or your philosophy about the search for exoplanets in general. This is an open-ended question; I mostly just want to make you think a bit about the amazing fact that we're discovering planets around other stars!

3 Turn in your Assignment

Your final version of your assignment should run from top to bottom without errors. (You should comment out or delete code bits that were just used for testing purposes.) To create a clean version, rerun your notebook using "Kernel|Restart & Run All." Be sure to save this final version (with output). To submit, click "File|Download As >|HTML (html)" inside jupyter notebook to convert your notebook into an HTML web page file. It should have a name like `homework#_{identikey}.html` (where # is the appropriate letter for this week's homework). Please upload this HTML file as your homework submission on Canvas.