Chapter 15

Redetecting Kepler exoplanets

For this assignment you will use data from the Kepler telescope and the NASA exoplanet archive[1]. With the Kepler data you can re-detect previously discovered planets.
Goals for this week:

- demonstrate an ability to design and implement simple programs,
- demonstrate an ability to collect and access tabular data (in text formats),
- access data in FITS files, and run some data analysis on them (fitting and creating periodograms in this case).

This exercise will be graded as follows, you can earn at most 9 points by completing the exercise and handing in on time. Your grade g, awarded on a scale from 1 to 10, is calculated as follows:$g = 1 + s$, where s are the points you earned. For this exercise the points are awarded for the following things:

(3 points) For proper programming style and a programs that are clearly structure.

(2 points) For the assignment in Section 15.1.

(4 points) For the assignment in Section 15.2.

15.1 Part 1: creating an all sky plot

In this part of the assignment you will create a plot of the positions on the sky of the known exoplanets. For this part of the assignment you have to hand in a program, and add to your report the all sky plot you create (with for reference the data file that you used to create it, so that it may be checked).

- The NASA exoplanet archive also provides data on all known exoplanets. This data can be downloaded as simple text files like the ones you processed before. See `https://exoplanetarchive.ipac.caltech.edu/`.
- **Assignment:** Download the sky positions of the known exoplanets, and write a parser for these data. You are allowed to use a library functionality to do the parsing. There are many libraries you may choose to use, like: the Python standard library `csv` module, the Numpy `genfromtxt` function, the Pandas `read_csv` funtion, etc.
  Be sure to include all data files you use in your submission. As a tip: for this exercise you do not need to download all the more than twenty columns from the Exoplanet data archive table.
- **Assignment:** Create an all sky plot of the confirmed exoplanets in the archive using the Hammer projection in Matplotlib. You may have to convert the right ascension and declination to degrees first.

[1] See `http://exoplanetarchive.ipac.caltech.edu/`

15.2 Re-detecting planets in Kepler data

For this assignment you will re-detect previously known exoplanets in the Kepler data. For this part of the assignment you have to hand in a program and a report.

- The Kepler mission observed a specific path of sky (the "Kepler field") for several years, repeatedly measuring the magnitudes of the stars in that field[2] The lightcurves created by Kepler show exoplanet transits, and by tracking these over a long time the orbits of these exoplanets can be reconstructed. Each of the stars in the Kepler field has a Kepler ID (sometimes called KOI for Kepler Object of Interest) that can be used to retrieve the data files, these IDs should not be confused with the numbering of the Kepler discoveries (given to planets like Kepler 5b).

- Kepler data files are in the FITS (Flexible Image Transport System) format that is used often in Astronomy. If you want to open FITS files in a Python program you should use the Astropy library (you may come across mentions of PyFits, but that is an older library). Astropy is a large library with a lot of functionality beyond reading FITS files. FITS files contain both header and data sections. The header generally contains information about the observed data and the data section the actual data.
  Assignment: Check that you have Astropy installed. If it is not, you you must install it.

- Find the earliest Kepler exoplanet discoveries, we will use one of these because they easiest to detect in the data (which is exactly why the were detected first by Kepler). The data can be downloaded from the Mikulski Archive for Space Telescopes (MAST)[3].
  Assignment: Download some data files for the Kepler-7b planet. Be sure to choose data files with lightcurves whose baseline does not jump around too much and that shows a number of dips (transits) so that it is easy to reconstruct the orbital period of the planet.
  Note: the website can generate plots for these data files so that you can check them before downloading.

- **Assignment:** Write a function to open the FITS file and extract the raw time-series data and not the baseline corrected one — because part of this assignment is to correct the baseline of the data. Create a plot of the photometry data, make sure your time axis is correct. To access the data in the FITS file use the `astropy.io.fits` module.

- **Assignment:** Inspect your time-series and choose a range of data that has no breaks, and filter out any not-a-number values as these will cause problems in the following analysis.

- Scipy provides a way of creating Lomb-Scargle periodograms, we will use that functionality (do not use the Lomb Scargle functions from other libraries like AstroML). Make sure you find the Lomb-Scargle periodogram function in Astropy.

- Scipy's implementation of these periodograms expects data that is centered around zero. The raw Kepler data is not like that.
  Assignment: Fit a low order polynomial function to the data (so that the planet transits are not also fit) and subtract this from the data. You will now have data that is close to being centered around zero. Furthermore subtract the median of the data to get even closer to zero.
  If you did this part of the assignment correctly you will still clearly see the planet transits while the baseline is now centered around zero.

- **Assignment:** Use Scipy's periodogram functionality to find the period of the planet's orbit. Create a plot of the periodogram and indicate where the planet's signal can be found. Include this plot in your report, and make sure to also include the period you extracted and the

[2] Hardware failure meant that it has since moved to observing other patches of sky.
[3] See `https://archive.stsci.edu/kepler/`.

actual period of the planet (from literature, or in this case it is admissible to quote the value you may find on Wikipedia). Provide relevant plots.

- **Assignment:** Extend your program such that it can process several light curve data files in one run of the program. Show that it works for several planets and several light curves per planet. Provide relevant plots.

15.3 Handing in

Check that the following things are included in your submission:

- a report describing your data sources and methods,
- plots you were asked to create,
- all code you wrote to perform your data analysis (able to be executed),
- any ancillary files needed to run your programs.

Furthermore, check that your code conforms to proper style and that it has a clear structure. Check that your code does not depend on hard coded paths, as those will not be present on the computer used for grading — make everything work with relative paths (and structure the tarball that you hand in such that the program will run without having to move files around).