DETECTING EXOPLANETS BY RADIAL VELOCITY METHOD

A project by:
Nikhil Paratkar (Id: 917158689)
Kunal Tambekar (Id: 917186964)
For CSC 877 : Topics in Big Data Analysis

INTRODUCTION : Quest for discovering planets and life outside of our solar system has intrigued a lot of researchers. For a long time, scientists and researchers struggled to find exoplanets which are too far to be observed with a telescope. After the confirmation of the first exoplanet in 1995, this area of research has attracted a lot of interest. Till date, there are around 3,500 confirmed exoplanets and billions if not trillions still waiting to be discovered.

The task of finding exoplanets seems next to impossible if we consider the fact that even the biggest of stars, which are several light years away appear to be just pinpoints when observed from Earth. The planets, which orbit these stars are nothing as compared to their star's mass and size, making the task of observing them with a telescope absolutely impossible. For several centuries, scientists and scholars have tried to find ways to detect such exoplanets that are dozens of light years away. An ingenious way came to light in the late 20th century where in scientists decided to observe the star instead of the planet orbiting it and record the minute effects that any orbiting body can have on it.

As ways of observing a star for detecting exoplanets, scientists came up with a few techniques viz.

1. Radial Velocity method
2. Transit photonomy
3. Microlensing

These techniques focus on different aspects of observing a star. For our project, as the name suggests, we focused on the radial velocity method for observing patterns in radial velocity changes of some known stars. In this report, we are going to discuss the radial velocity method, the data available for performing calculations and our efforts to make use of the available data. Our focus for this project was on finding out the orbital period of a planet that is in the orbit of a star under consideration. We will also discuss the results we came to and analysis on those results.

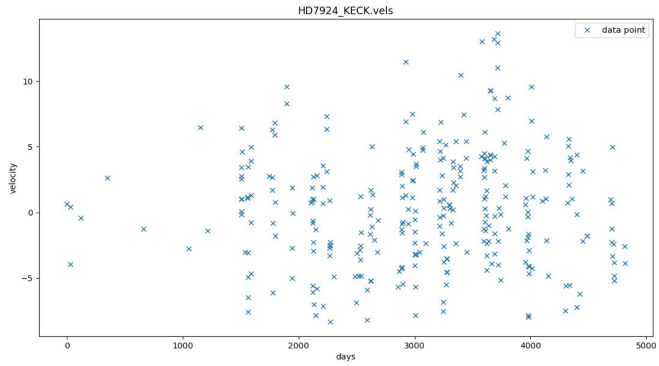
RADIAL VELOCITY : In purely scientific terms, the radial velocity of an object with respect to the point of observation, is the rate of change of the distance between the object and the observation point. In astronomy, radial velocity of a star is calculated using Doppler Spectroscopy. As we know according to Doppler effect, the frequency or wavelength of a wave changes when either the observer or the source is moving. Hence, the frequency of light coming from a star changes based on whether the star is moving towards or away from the observer. These changes in the frequency of light are termed as 'Redshift' or 'Blueshift'.

DATA : W. M. Keck observatory in Hawaii recently released a large dataset[1] of radial velocity observations of several stars. These observed recordings span upto 20 years in time and include measurements for more than 1,600 stars. This huge data released by the observatory was gathered as a part of a two decade planet hunting program, which made use of a spectrometer called HIRES. The data is categorized into different files for each star. For each star, the format of the data is as follows:

Julian Date	Velocity (m/s)	Error (m/s)	s_value	Halpha	Median photons/pixel	Exposure time (secs)
2452849.08461	-2.20	1.54	0.1431	-1.00	116037	5
2452850.04050	0.59	0.98	0.1471	-1.00	01063	12
2452851.13385	2.69	1.08	0.1714	-1.00	212819	10

Julian Date: It's the julian date timestamp of when the recording was made
Velocity: This is the radial velocity measurement, which is recorded in meters/second.
Error: This is the possible error in recording of the radial velocity using the spectroscope. This is also recorded in meters/second.
The rest of the data columns, which are- s_value, Halpha, Median photons per pixel and Exposure time were ignored for this project as they correspond to factors not under our consideration. The data that we have used for our project comes from the Julian Date, Velocity and the Error columns only. Using the values from these sections we have successfully tried to find the time periods of the orbiting exoplanet for any given star.

APPROACH : For a given set of data on a star, we started with plotting the radial velocity measurements for that star against time. For example, for star HD7924, when we plotted the graph, the results looked as follows:



We expected to see some pattern in the data that suggests periodicity. However, as we can see in the above graph, it was very difficult to observe any such pattern and similar graphs were observed for data on other stars. To reach to our goal of finding out an orbital period, we decided to apply different techniques to make sense of the provided data.

A major problem with the data we observed was that it was not evenly spaced. Since the data was gathered over a period of almost two decades, there was no consistency in recording the velocities. The velocity readings were taken non-periodically and therefore, there was no way to observe patterns with this unevenly spaced data. To tackle this problem, we decided to apply interpolation methods on the data, which would give us the missing data points between two successive readings.

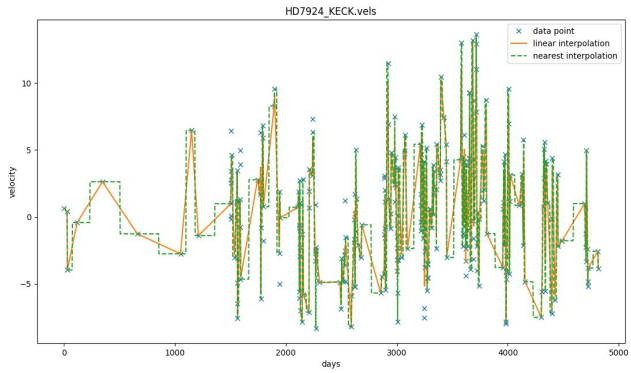
Interpolation: For this, we used scipy's interpolate library, in particular, the interp1d function. First, we started with linear interpolation which would give us points on a straight line between two consecutive data points.

f = interp1d(time, velocities, kind='linear')

Where ‘time’ is an array of the timestamps and ‘velocities’ is an array of the velocity measurements corresponding to the timestamps in the ‘time’ array. The above line of code returns a function ‘f’, which when passed an x-value, returns the interpolated y-value for the graph. For a new set of x-values or time-values, we decided to take 5 times the number of available values in the ‘time’ array.

t_new = np.linspace(time[0], time[length-1], length*5)

We then plotted our new graph with the new time-values and the interpolated velocity-values coming from the function ‘f’. Similarly, we tried some other interpolation techniques as well just to see which fits the best for our data. The other interpolation techniques included nearest interpolation, cubic interpolation, quadratic interpolation, etc. For cubic interpolation we observed that the interpolation function returns extreme interpolated values that are assigned according to spline interpolation of third order. Quadratic interpolation produced similar values, which did not fit our case. Therefore, to move ahead, we considered just linear and nearest interpolation. Following graph shows the interpolated values of linear and nearest interpolation.



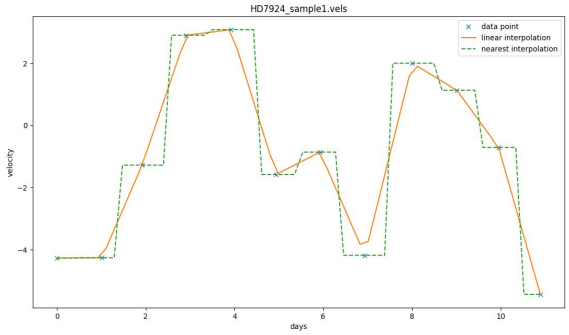
This experiment filled out the gaps between our data but did not solve our problem of finding the period. Out of the two interpolation methods, we moved ahead with the simplest, which is linear interpolation.

Fourier Transform: For finding out the orbital period of a possible planet, we tried making use of the fourier transform. The intuition behind using fourier transform was that if a planet is orbiting the star which we are observing, we should get a sinusoidal curve with the time Vs velocity plot. And fourier transform is one of the most popular methods to get frequency values from sinusoidal waveforms. Therefore, we tried using scipy library’s fftpack function to perform fourier transform on our data. Unfortunately, the fft function needed evenly spaced data for the sinusoidal curve and since we only had the linearly interpolated data, the frequency calculations we observed were not what we expected.

Selecting Smaller Dataset: At this point, we had the linearly interpolated data, which could not give us a clear view of the periodicity of the star-planet system. One of the primary problems we identified was that the recordings were measured over a very long period of time, typically thousands of

days. The planets that we were trying to find had expected orbital periods of much smaller number of days than this. For this reason and to get a better view into the data, we decided to consider only those readings, which were recorded on consecutive days. Such readings, according to our observation, did not span for more than 10 days in general.

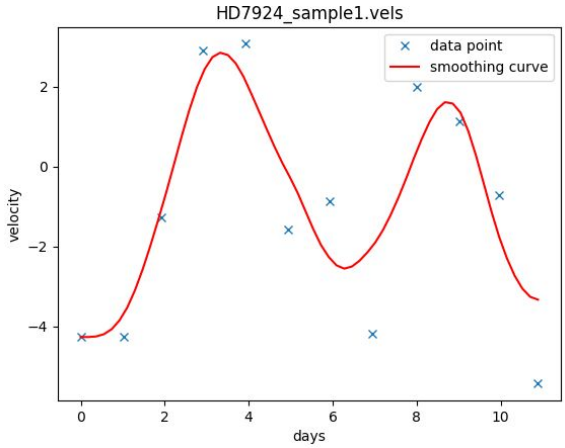
So the next task we overtook, was to find such smaller datasets from the dataset file of a star. We selected a few stars, which had enough data available for our consideration and we created new files containing only the recordings made on consecutive days for that star. After we had the new smaller dataset, we performed the same interpolation technique on this data and plotted the velocity values against time:



After plotting this new data, a pattern was visible for each of such newly created files. So the next task for us was to plot a sinusoidal curve with the help of this data so that we can calculate its time period.

Data Smoothing: To get a smooth curve out of the available data, we decided to try and apply different curve smoothing functions. The first function we tried was scipy.optimize library's curve_fit function. This function takes noisy sinusoidal wave data and plots a smooth curve. The problem we faced while trying to implement this method was that the method expects a function, which defines your waveform. For theoretical examples, we could easily pass a function for a sinusoidal wave with guess values for amplitude, phase and frequency. But for practical purpose, defining this function by just looking at the data was a very difficult task. There were too many unknown parameters that needed to be assumed before passing the function to the curve_fit method. So in the interest of time, we decided to go with another curve smoothing function we came across, called Savitzky Golay Filtering. Savitzky Golay filter is a digital filter that increases the signal-to-noise ratio without distorting the original signal and this is achieved using convolution. We found using this function for our new smaller dataset to be very simple. To this filter, we pass our linearly interpolated data, along with a window size and a polynomial order. We set the polynomial order to 3 which produced similar results as that of lower orders. In our implementation, we set the window size to 21 which suits our data when we have data spaced for approximately 10 days. We give the user the functionality to change the size of this window according to observations of the graph.

When with these settings, we passed our data to the Savitzky Golay function, we got a smooth curve as shown in the following graph:



Finding orbital period estimate: After we retrieved a smooth curve from our data, the next task was to extract the time period of the sinusoidal wave. For this, we wrote a function that calculates the distance between the consecutive peaks and returns it as an output. For calculating values of peaks, we used python's peakutils library. We called the output of this function, an orbital period estimate. This estimate, as seen in the previous steps, is calculated based on curve_fitting and observation of the available data. We now process the data on top of this estimated value to get a more accurate orbital period value.

Kalman Filter: Kalman filtering is an algorithm that gives an estimate of the unknown values by making use of the available data. The algorithm works in two phases: predict and update. For predicting initial estimate, the kalman filter makes use of the known physical laws and initial data in hand. Not only does it predict a new value, but also a covariance value is predicted. In the update phase, the new observed reading is taken into consideration and based on this new observed reading, we get a value of uncertainty in the prediction as well as its covariance relative to the prediction already made. The algorithm performs these steps recursively and takes into consideration the previously processed data for coming up with a new value of the unknown variable. Kalman Filter seems to be a very powerful tool for predicting the unknown values and it should fit our problem statement. However, Kalman filter expects to know the physical laws of bodies in order to predict the initial unknown value. Since we had very little knowledge of such laws relating to orbiting planets and the factors that affect radial velocity measurements, we felt underqualified to use this particular technique, given a small time to try it.

CALCULATIONS:

For further confirmation of the orbital period, we first process the data values provided in *.vels file format and convert it to csv files. We are using the binned folder as it results in clearer graphs and values are more accurate. This is because for some of the days the observatory made multiple observations and averaged these multiple values to get a single reading for that day, reducing the redundant data. Then we have used this data along with the coefficient of correlation (χ^2) to find the time period using that best fits the collected readings. We know that when we need to estimate or find the value that best fits a required data set we calculate the expected value using the input parameters and cross reference it with the observed value using the formula:

$$\chi^2 = \sum_k \frac{(v_{obs}(t_k) - v_{model}(t_k))^2}{\sigma_k^2}$$

Where,

v_{obs} is the observed value for input t_k and v_{model} is the expected value calculated using the radial velocity equation. The radial velocity is a periodic function with amplitude of K and periodicity equal to P. The equation for calculating the radial velocity can be given as follows:

$$v_{\mathrm{R}} = k \sin\left(\frac{2\pi(t - t_0)}{P}\right) + v_0$$

Here,

v_R is the radial velocity at time t. In our project we have considered v_0 as 0, to simplify the process and reduce the number of calculations involved. The $(t\text{-}t_0)$ in the equation corresponds to the timestamp values in our calculations and the values of P and K are varied throughout the iterations multiple times. To calculate the time period we have only one equation but 2 unknowns and hence we have used a brute force method but in a smart way. Just like actual astrophysicists, we start by making initial assumptions and then iterating over variations of guess values to get more refined results. We derived our initial guesses from the time series data which when plotted against time results in curve as shown below (when data is available in abundance and periodic manner).

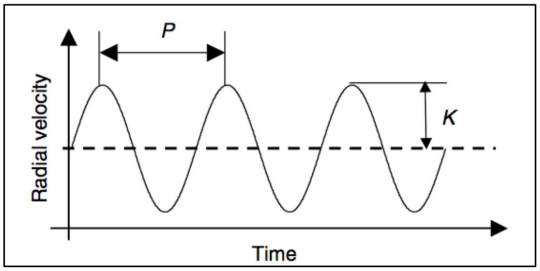


Figure 11 – Plot of radial velocity vs. time for the host star indicating how the period, P, and radial velocity semi-amplitude, K, can be determined from the data. (**Image Credit**: *Planetary Systems and the Origins of Life*, Cambridge University Press, 2007)

Where,

P = Time period, the peak to peak distance is considered to be the initial guess for the time period of the probable exoplanet that is revolving around the star system being observed

K = star's radial velocity semi amplitude.

Thus to get the minimum value of χ^2, we have to calculate the value of χ^2 for the complete data set by varying the values of K and P little by little at a time. We have done this using a brute force approach. But we don't do the calculation on complete range of estimation. We stop the iterations as soon as we think that we have found a good estimate of the unknown variable. Then we iterate over a smaller range of values and get a finer and even more refined value for the variable. To get the best estimation of the probable exoplanet we need to pass the data and process it through 3 functions we created.

To get a rough estimate of K we pass the data to `estimateBestGuessValueForK()` and this function returns an approx value of K which is passed to second function `calculateFinerValueOfK()` that returns a far more accurate value of K. We can get an even finer value of K if we increase the iterations and reduce the intervals even more. Then with the best value of K found we iterate over the data set to find the best value of time period (now we are left with one equation and one variable) using the third function `estimateBestTimePeriod()`. We iterate over smaller variations of each time period and calculate the value of χ^2 using the Time, Radial velocity and error in radial velocity. We then plot this data on a graph and select the value of P for which the value of Chi square is minimum and that points to the

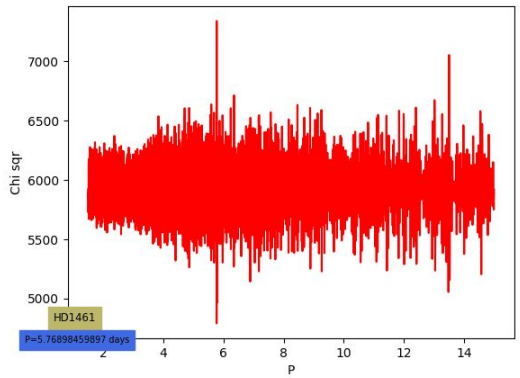
best value of the time period of a probable exoplanet that might be revolving around that star.

Below are few images of the χ^2 v/s P graphs that were generated from actual data in our program.

HD1461

$P_{calculated}$ = 5.76898 days

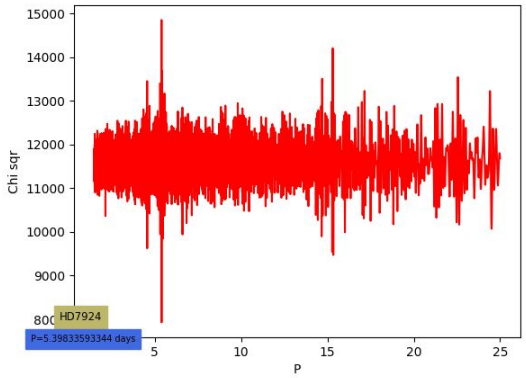
P_{actual} = 5.77 days



HD7924

$P_{calculated}$ = 5.39833 days

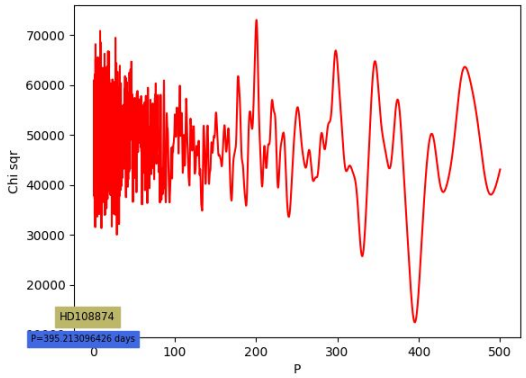
P_{actual} = 5.3979 days



HD108874

$P_{calculated}$ = 365.21 days

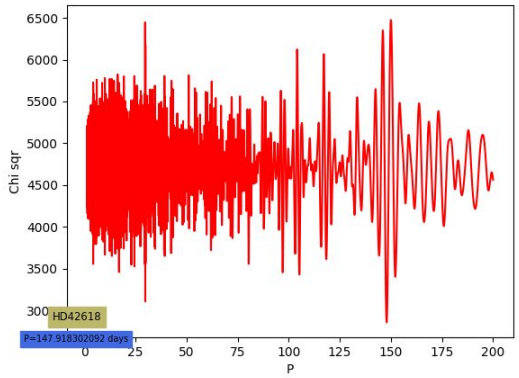
P_{actual} = 395 days



HD42618

$P_{calculated}$ = 147.91 days

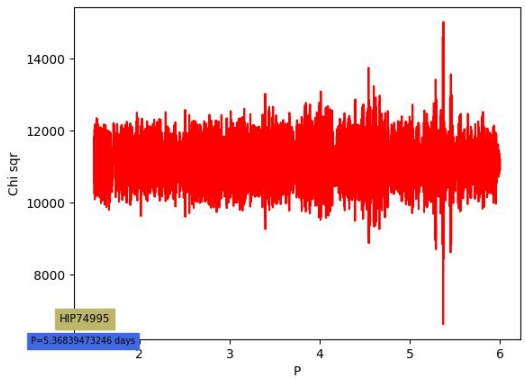
P_{actual} = 149.6days



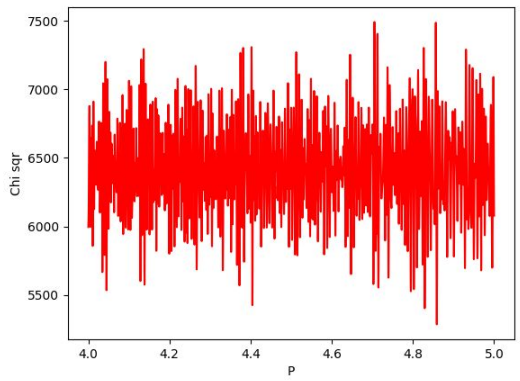
HIP74995

$P_{calculated}$ = 5.36839 days

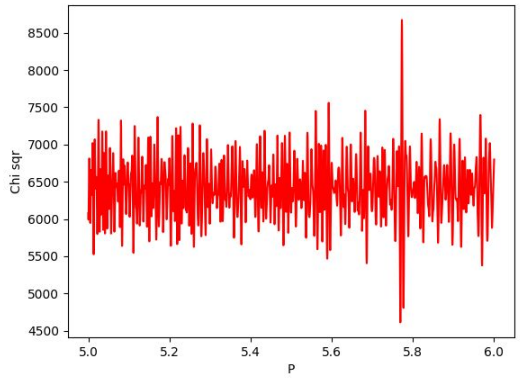
P_{actual} = 5.3686days



We noticed that there is some human intervention needed for the analysis of the results generated from the data because there will always be some minimum value for the curve that is generated and the correct way to identify if a star has an exoplanet is to check if there are spikes in the data. So to compare we have 2 graphs generated below.



Graph1: No planets in the star system



Graph2: Star system has a planet

The first one is for a star that doesn't have any exoplanet revolving around it and the second one is for the star that has some exo planet revolving around it. As we can see there is clearly a spike seen in the second graph whereas the data in the first one seems to be of almost same height.

Thus, we have not been able to clearly draw a boundary for identifying if a star system has a planet or not using only the code. But interestingly we have been able to find the time period of another exoplanet that may be revolving around the star under observation. We found out multiple spikes in the same graph and on further investigation we found that we had identified the probable time periods of other exoplanets in that star system. Multiple spikes that are generated may not be of significant magnitude and can be easily confused with false positives, but these definitely gives an estimate/probable value of the time period of another exoplanet that may be in the same star system.

THE BIG DATA COMPONENT:

Detecting exoplanets involves a great deal of computation on a huge data set that is gathered over a time span of several years by scientists in observatories around the world. The data that we used for predicting the exoplanets in our project has been collected for over past 20 years and has more than sixty thousand values corresponding to 1699 star systems. There are only limited number of equations that are available, that define the planetary motion but the number of variables are many when compared to the number of equations. Which leaves us with only one choice, that is, to start calculations by making the best assumption and then refining the values as we proceed by cross verifying the observed data with the experimental/ expected data readings. Not only do we have to deal with huge quantity of data, but the amount of computational

power needed to process data is also very high.

To get accurate values we need to run same computations over and over, many number of times, tweaking the input/ assumption value little by little and this process then eventually leads to more accurate results. During our experiment also we found that the data forms a 'V' curve.i.e. The values reach a certain point at which it coincides best with the actual data and then it starts deviating from the actual value again. The minima or the lowest point during this process corresponds to the lowest value of χ^2 which indicates the closest or the best value of the unknowns for that data set. Thus, the approach used for coming to the results by processing the raw data, the amount of data and the number of computations correspond to the Big Data.

VERIFICATION OF RESULTS:

Fortunately, till date more than 3000 exoplanets have been confirmed and more and more are being discovered every year. So we had a large number of star system's data to which coincided with the actual data and it was exciting to find out that our results were very accurate for the time periods we calculated using the data that was released. We used the data from the open exoplanet catalog to tally the results that we found after our calculation the error in the readings was also found out to be significantly low and the experiment can be considered as a success.

CONCLUSION:

It can be clearly said that we have been able to estimate a best possible value of the time period of the exoplanet revolving around a star system successfully. The time period calculation is the first step for calculating the other properties associated with the exoplanet such as, distance of planet from the parent star and the mass of the exoplanet. But for this we'll also need data from other methods such as, mass of star can be obtained from the luminosity of the star and this is used for the calculation of mass of the planet. Radial velocity data alone cannot be used for getting all the details about a planet for this project.

SPECIAL THANKS TO:

We would also like to thank Professor Stephen Kane from the Astrophysics department at SFSU for helping us clear some of our basic concepts and doubts about the topics related to discovery of exoplanets.

REFERENCES :

[1] http://home.dtm.ciw.edu/ebps/data/

[2] Lovis C. & D. Fischer 2010, "Radial_Velocity.pdf" *Exoplanets* Ed. S. Seager, University of Arizona Press, Tucson AZ

[3] Fulton et al. 2015 "Three Super-Earths Orbiting HD 7924" ApJ (accepted) arXiv:1504.06629

[4] Feroz, F., and M. P. Hobson. "Bayesian Analysis of Radial Velocity Data of GJ667C with Correlated Noise: Evidence for Only Two Planets." *Monthly Notices of the Royal Astronomical Society*. Oxford University Press, 29 Nov. 2013.

[5] http://home.strw.leidenuniv.nl/~keller/Teaching/ADA_2011/ADA_2011_L06_Exoplanets.pdf

[6]https://carnegiescience.edu/news/team-makes-planet-hunting-group-effort-finds-more-100-candidates

[7] http://scipy.github.io/old-wiki/pages/Cookbook/SavitzkyGolay

[8]http://joseph-long.com/writing/recovering-signals-from-unevenly-sampled-data/

[9] http://adamdempsey90.github.io/python/radial_velocities/radial_velocities.html

[10] http://www.astronomynotes.com/starprop/s10.htm

[11] Paul van der Werf, Henk Hoekstra, Jarle Brinchmann "The Exoplanet 51 Peg b" , 2015 http://home.strw.leidenuniv.nl/~jarle/Teaching/Practicum2015/assets/practicum2015-problem2.pdf

[12] Lectures from Professor Stephen Kane from Astrophysics department at SFSU http://www.physics.sfsu.edu/~skane/teaching/a405/lecture06.pdf