Ethan Crowley and Robert Rogers

N/A

Python DeCal

26 April 2024

Python DeCal Final Project Report

For our project, originally going to work with vectors and arrays. We wanted to do something involving dot product and/or cross product using NumPy arrays, but after a while, we decided against this. There were too many variables and factors to consider, and we could not find any good data sets to plot this data. So, we decided to turn our attention to the stars. We knew that there were plenty of good datasets and equations for astral bodies, so we ended up looking at data sets for stars.

Our first order of business was to find a dataset. We went straight to the NASA Exoplanet Archive and started snooping around. After a bit of searching, we found the Stellar Hosts spreadsheet. We decided that this would be the best option due to the large amount of data for a multitude of facets for stars. There was so much data that we could not even scroll to the bottom without using the sidebar. After looking through the data we decided to look at a star's brightness compared to its mass. While the mass was given in solar masses in the Stellar Mass column, there was no luminosity column. Luckily the V (Johnson) Magnitude was provided so we could use that in place of luminosity.

After choosing these two variables, we immediately ran into another problem. Other factors affect a star's brightness, not just mass. To rectify this we choose a specific spectral type

of star, class G stars. These stars are the same type of stars as the Sun and share a similar temperature range of 5,300-6,000 degrees Kelvin. This helped eliminate other factors that would interfere with the graph. Instead of using code to select just G-class stars, we went back to NASA Exoplanet Archive and filtered the data there, redownloading the CSV file.

One problem we could not deal with in the NASA Exoplanet Archive was the amount of blank data. This was screwing with our graph so we wrote a few quick lines of code to get rid of it. We then resaved the CSV file with the blank data removed. With that, we were ready to really start coding.

The libraries we used were NumPy, Pandas, and Matplotlib. Writing the graph code did not take that long, but we kept running into errors. On the CVS file, it dictated what the column names were, but after a while, we figured out that they were just lying. So we directly manipulated the CVS file in Numbers to change the column names for Stellar Mass and V (Johnson) Magnitude so the graph would actually plot. After resaving the CSV file, we finally got a plot.

After quickly changing the plot type to scatter from plot, we got to work making a line of best fit. This turned out to be pretty tricky as the graph was not built super well for a line of best fit. In the end, used NumPy polyfit and poly1d to make the line. Using this line, we got a polynomial function to use for our equation. After some minor tweaks to make the plot a bit more readable, we were nearly done.

All we had to do to finish up was add some error bars. This was pretty simple, as we just used our polynomial function to create some residuals and the standard deviation function from NumPy (np.std) to finish the error bars. Overall this was a fun project to complete. While the

errors we had were frustrating, it was satisfying to overcome them and finish with a completed project.