Using SDSS Data and Python in the Compilation and Composition of a Hertzsprung-Russell Diagram

Caden Gobat [1]

(PHYS 2023, Fall 2019)

Kalvir S. Dhuga [1]

(Advising Professor)

[1] *Department of Physics*
The George Washington University
Washington, DC 20052, USA

ABSTRACT

The Hertzsprung-Russell (H-R) diagram is a foundational visualization tool of modern astronomy that plots stars' luminosity against their temperature, spectral class, or color (all essentially synonymous). Here I make use of stellar spectral and parallax data collected by the Sloan Digital Sky Survey to fit and plot a representative H-R diagram using Python and Matplotlib. The result is a high-precision, visually appealing graphic, which is presented here, along with the methods used to generate it.

Keywords: astronomical databases: miscellaneous — methods: data analysis — Hertzsprung–Russell and C–M diagrams — stars: luminosity function, mass function

1. INTRODUCTION

Although the study of the stars on the whole is a practice that dates back millennia, the characterization of patterns and relationships among their spectra and other properties began to be truly solidified at the beginning of the twentieth century with works such as Hertzsprung (1911) and Russell (1912), who plotted stars' luminosities against their colors and temperatures. Doing so brings to light a number of revelations about stellar classification (main sequence, dwarfs, giants, etc.), life cycles, and a multitude of other insights.

In order to create such a diagram, the provided observational data from the Sloan Digital Sky Survey (Blanton et al. 2017) was imported and processed using a series of Python algorithms implemented by the author. Python was chosen over a more simplistic software such as Excel or Google Sheets because it provides more precision and automation in the handling of the data, especially for large datasets. Using packages such as Pandas (McKinney 2010) and astropy (Astropy Collaboration et al. 2013) allows for greater control in the manipulation of the data as well as a more readily scalable method to analyze much larger datasets. Overall, the use of Python is a more advanced, robust technique that is much more efficient than manual spreadsheet computations on large datasets. Additionally, Matplotlib (Hunter 2007) enables a much higher degree of control over visualizations and plots of the data than most spreadsheet programs.

2. DATA COLLECTION & PARSING

Data on stellar properties was provided in two separate catalog excerpt files: one main list (`HR-project-stars(1).xlsx`), which included stars from a variety of spectral and luminosity classes, and one that contained exclusively white dwarfs (`wd-stars.xlsx`). For the main list of stars, data was provided for the following variables: apparent magnitude in the V band, right ascension, declination, parallax angle, B-V color, and spectral class. The white dwarfs list contains all of the same information categories, with the addition of a column containing absolute magnitude in the V band as well.

cgobat@gwu.edu

dhuga@gwu.edu

Because the primary list did not contain information about the absolute magnitudes of the stars, it was necessary to calculate using apparent magnitude and distance. Because an object's apparent brightness is a function of both how bright it actually is and how far away it is, knowing two of these values allows for the calculation of the third. In this case, the formula for performing this conversion to absolute visual magnitude (`MV`) is given in Eq. (1), where `par` is the measured parallax angle in milliarcseconds and `Vmag` is the apparent visual magnitude.

$$\mathrm{M_V} = \mathrm{Vmag} + 5 \times \log(\frac{\mathrm{par}}{100}) \tag{1}$$

In order to execute this calculation, and the rest of the procedures described here, the two spreadsheets were converted into .csv format in order to be loaded into a Python environment through the Pandas module for analysis. After generating an `MV` column for the main list, the two datasets contained all of the same variables, and could be compiled into one master catalog. Each column of the white dwarfs list was appended to the corresponding column of the main list. Then, absolute luminosities could be calculated for all of the stars using Eq. (2) (provided).

$$\mathrm{L/L_\odot} = 2.512^{4.83-\mathrm{M_V}} \tag{2}$$

With this, the dataset now contains all of the information necessary for the creation of the diagram.

3. COLOR & TEMPERATURE

One of the major breakthroughs of the early twentieth century was the characterization of the black-body radiation spectrum. This revealed that a black-body radiator (such as a star) has a color that is directly related to its temperature. Given that the color index of each star is provided in the dataset (B–V index, the magnitude in the V band subtracted from that in the B band—referred to as `BV` for convenience), their temperatures can be inferred. First, a model must be developed that relates the two variables numerically. To this end, a calibration dataset of empirical observations of color and temperature for 14 objects was provided (`stars-color-color-diagrams.xls`).

The following equation was initially suggested to fit the color-temperature calibration data:

$$\begin{aligned}\log T = a_0 + a_1(\mathrm{BV}) + a_2(\mathrm{BV})^2 \\ +a_3(\mathrm{BV})^3 + a_4(\mathrm{BV})^4\end{aligned} \tag{3}$$

With provided coefficient weights of:

$$a_0 = +3.986;$$
$$a_1 = -0.558;$$
$$a_2 = +0.498;$$
$$a_3 = -0.324;$$
$$a_4 = +0.078$$

As a check, a non-linear least-squares fit was performed to optimize the a coefficients This yielded a model that overfit the data, making the coefficients deviate significantly from the originals.

However, the accepted[1] formula is given in Eq. (4), and was first presented by Ballesteros (2012), who derived it by considering stars as black-bodies, which is a fairly appropriate approximation for most purposes.

$$T = 4600\ \mathrm{K}(\frac{1}{0.92(\mathrm{BV}) + 1.7} + \frac{1}{0.94(\mathrm{BV}) + 0.62}) \tag{4}$$

All three of these fits are plotted in Fig. (1), which shows that the accepted model is actually reproduced more closely by the original polynomial coefficients, rather than the optimized ones. This is likely because the fitting engine attempted to overly compensate for erroneous fluctuations in the data, making it actually less reflective of the real relationship.

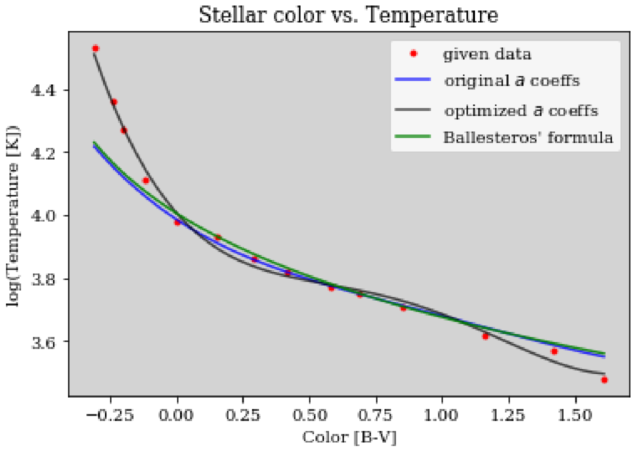


Figure 1. The various fits between color and temperature of several exemplar stars (data plotted in red). The blue line plots the fourth-order polynomial (Eq. 3) using the given a-coefficients. The black uses the optimized coefficient weights, and shows an obvious discrepancy between the original and fitted polynomials. The green line shows that the accepted formula derived from black-body radiation theory is much more in line with the original model than the fitted one.

[1] implemented in packages such as PyAstronomy

Given the close similarity between the original-coefficient fourth-order polynomial and Ballesteros' formula (on this domain, anyway), it does not particularly matter which is used for the rest of the analysis.

Now, with a model to convert between B-V color index and temperature, the surface temperature of each star in the catalog can be determined.

4. THE H-R DIAGRAM: RESULTS AND CONCLUSIONS

There are essentially three different variables that can be used as the horizontal axis of an H-R diagram: color index, temperature, and spectral class. All of these values are essentially synonymous, and converting between them is a simple task. For the purposes of this diagram, all three were used in some way. The data were originally plotted using the color index as the abscissa and luminosity as the ordinate. Then, temperatures were matched with these colors (using the formulas described previously) and charted on a secondary horizontal axis at the top of the diagram. Finally, the labels on the main x-axis were replaced by the spectral classes (O, B, A, F, G, K, M) at their respective locations and color index was denoted using a scaled color scheme, keyed using a colorbar on the right of the diagram.

The Python code block used to generate the plot itself out of the formatted data is provided in Fig. (2), with a link to the complete source code used for analysis in Appendix B (readers are encouraged to follow the link to gain a full appreciation for how the code works). The finished product itself is presented in Fig. (3), as generated by the Matplotlib code shown. Both figures appear on the following page.

In general, the creation of the H-R diagram was a success, as the groups that one expects to appear indeed reveal themselves: there is a clear main sequence spanning the plot from upper left to lower right, with a distinctly separate group of dwarfs to the lower left. The giants branch also extends upward and to the right from the main sequence, as it should.

This project confirmed the empirical nature of the relationships displayed on H-R diagrams, and provided a valuable exercise in computational analysis of real observational data.

ACKNOWLEDGMENTS

This research made use of Astropy, a community-developed core Python package for Astronomy.

Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the Participating Institutions. SDSS-IV acknowledges support and resources from the Center for High-Performance Computing at the University of Utah. The SDSS web site is www.sdss.org. SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration.

This report was typeset using AASTEX v6.3, a resource provided by the American Astronomical Society.

Facilities: SDSS

Software: Python, pandas, Matplotlib, astropy

REFERENCES

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068

Ballesteros, F. J. 2012, EPL (Europhysics Letters), 97, 34008, doi: 10.1209/0295-5075/97/34008

Blanton, M. R., Bershady, M. A., Abolfathi, B., et al. 2017, AJ, 154, 28, doi: 10.3847/1538-3881/aa7567

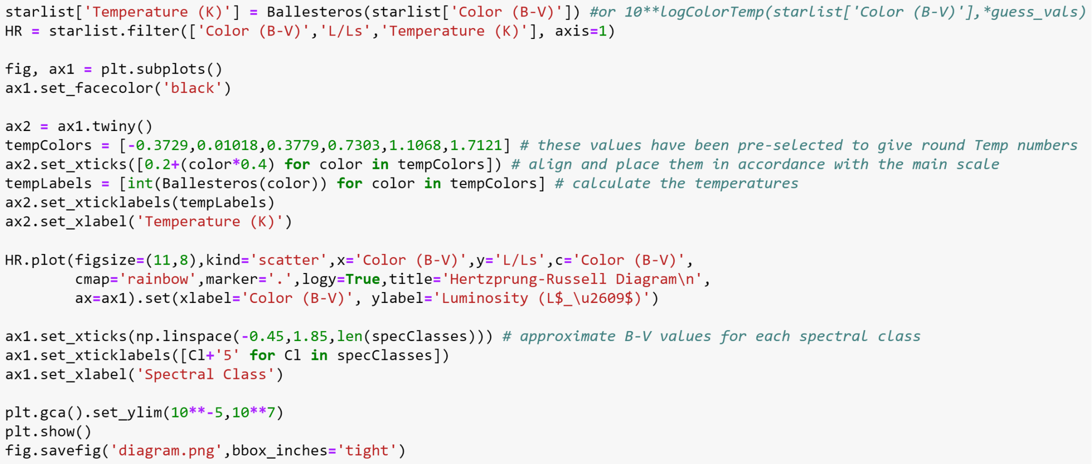
Hertzsprung, E. 1911, Publikationen des Astrophysikalischen Observatoriums zu Potsdam, 63

Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90, doi: 10.1109/MCSE.2007.55

McKinney, W. 2010, in Proceedings of the 9th Python in Science Conference, ed. S. van der Walt & J. Millman, 51 – 56

Pérez, F., & Granger, B. E. 2007, Computing in Science and Engineering, 9, 21, doi: 10.1109/MCSE.2007.53

Russell, H. N. 1912, Proceedings of the American Philosophical Society, 51, 569,579



```
starlist['Temperature (K)'] = Ballesteros(starlist['Color (B-V)']) #or 10**logColorTemp(starlist['Color (B-V)'],*guess_vals)
HR = starlist.filter(['Color (B-V)','L/Ls','Temperature (K)'], axis=1)

fig, ax1 = plt.subplots()
ax1.set_facecolor('black')

ax2 = ax1.twiny()
tempColors = [-0.3729,0.01018,0.3779,0.7303,1.1068,1.7121] # these values have been pre-selected to give round Temp numbers
ax2.set_xticks([0.2+(color*0.4) for color in tempColors]) # align and place them in accordance with the main scale
tempLabels = [int(Ballesteros(color)) for color in tempColors] # calculate the temperatures
ax2.set_xticklabels(tempLabels)
ax2.set_xlabel('Temperature (K)')

HR.plot(figsize=(11,8),kind='scatter',x='Color (B-V)',y='L/Ls',c='Color (B-V)',
        cmap='rainbow',marker='.',logy=True,title='Hertzprung-Russell Diagram\n',
        ax=ax1).set(xlabel='Color (B-V)', ylabel='Luminosity (L$_\u2609$)')

ax1.set_xticks(np.linspace(-0.45,1.85,len(specClasses))) # approximate B-V values for each spectral class
ax1.set_xticklabels([Cl+'5' for Cl in specClasses])
ax1.set_xlabel('Spectral Class')

plt.gca().set_ylim(10**-5,10**7)
plt.show()
fig.savefig('diagram.png',bbox_inches='tight')
```

Figure 2. Once the data were analyzed, this code block generates and saves the H-R diagram as it appears below.

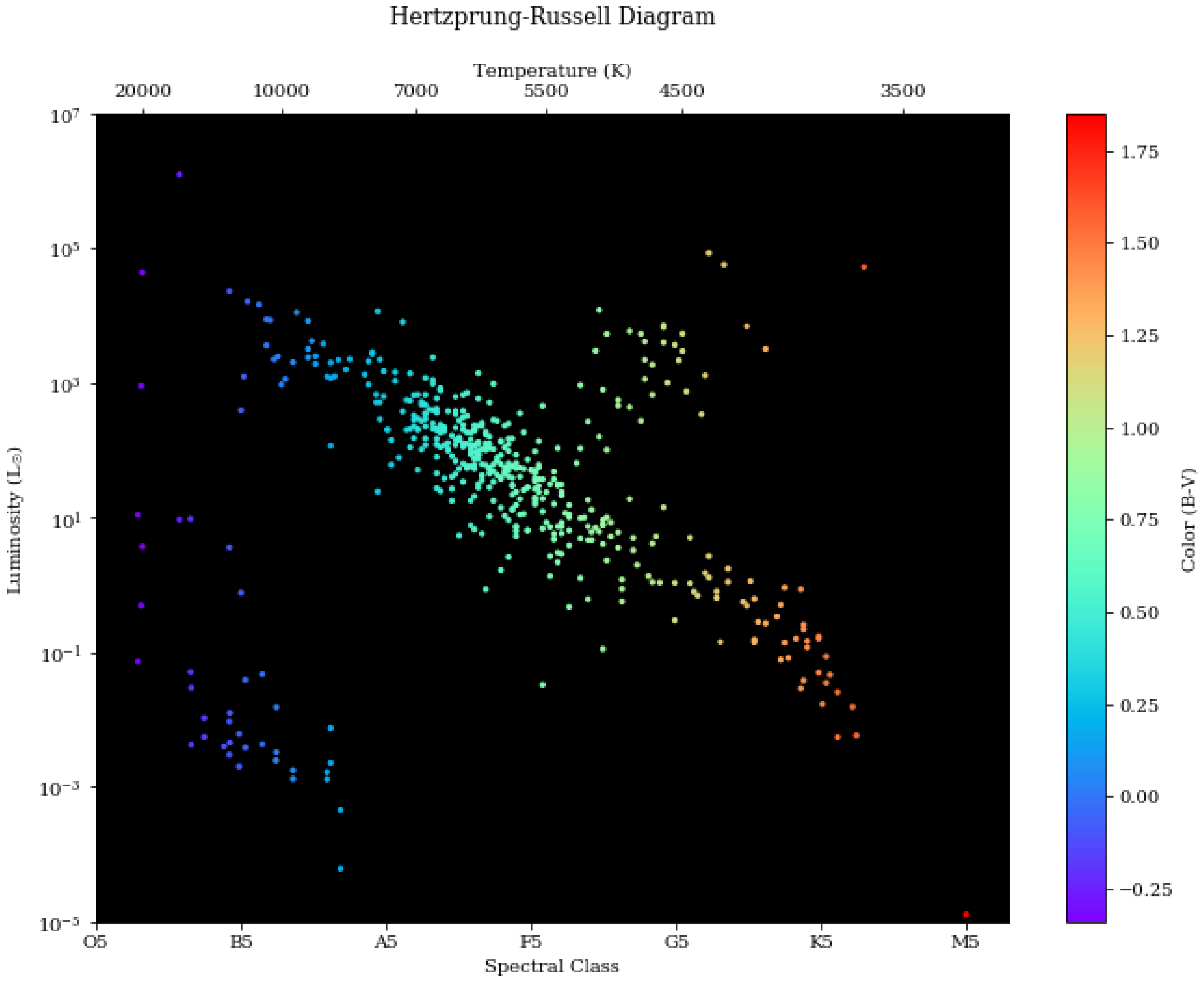


Figure 3. The H-R diagram created using the data provided for this project. On the bottom horizontal axis is the approximate spectral class, while the top denotes corresponding effective temperatures. On the horizontal axis is the stars' luminosities in units of solar luminosities, on a logarithmic scale. The colorbar to the right provides information as to the B-V color index of each star, which is essentially another measure of surface temperature.

APPENDIX

A. SKY CHART

For reference purposes, astropy and Matplotlib were used to create a visualization of the locations of the given stars using the provided right ascension and declination information.

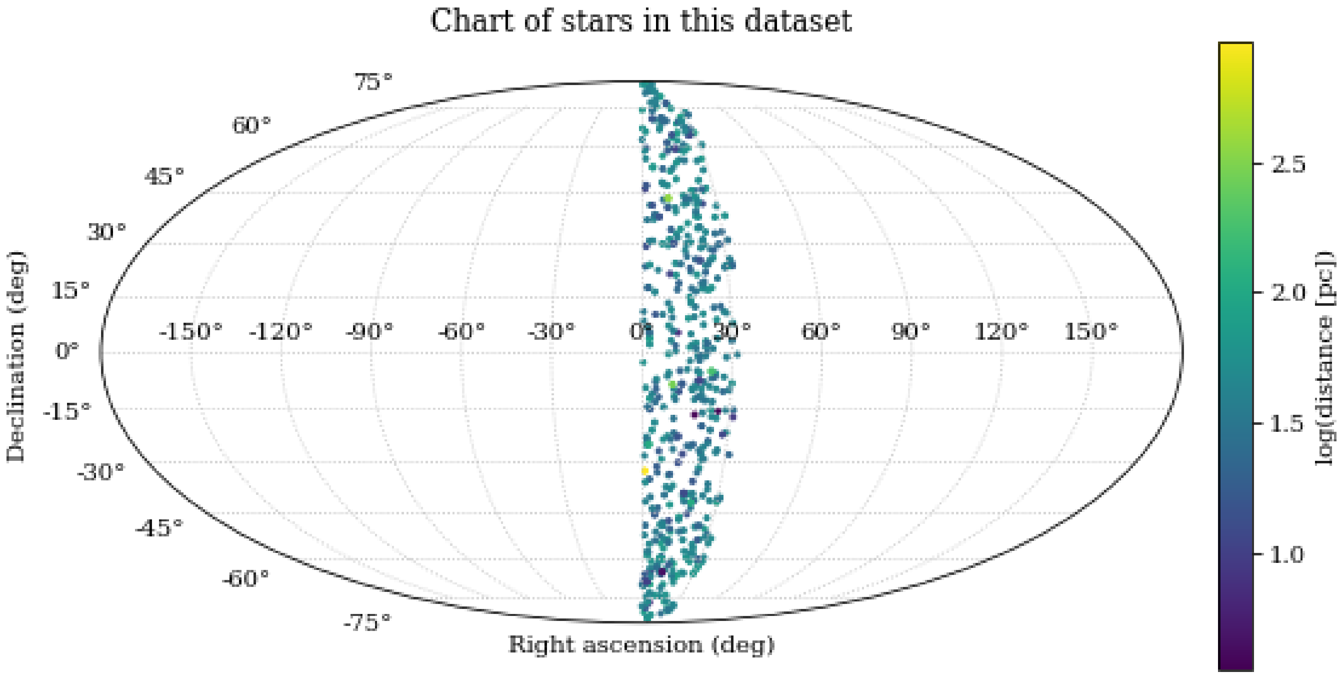


Figure 4. A Mollweide projection of the sky with the stars in this dataset charted. The points' colors correspond to their distances from Earth.

It is noteworthy that stars in this excerpt are shown to all be located between 0° and 30° of right ascension, spanning the full ±90° of declination. Additionally, the color mapping shows that most stars are located at a distance on the order of ~10^1 to 10^2 parsecs away.

B. REFERENCE LINKS

- Complete source code - Jupyter or Colab notebook (requires IPython: Pérez & Granger 2007)
- Main list of stellar data (accessible to PHYS 2023 course members only)
- List of white dwarf data (accessible to PHYS 2023 course members only)
- Star color-temperature calibration data (accessible to PHYS 2023 course members only)
- Information on the H-R diagram provided by SDSS