0. Title: your title should start with the package name, feel free to use a Combination Title, e.g. NEMO: where did the fish go? Author: don't forget to author your work, a date is always nice too.

Title: PySpecKit: Delving into the Composition of the Universe.

Author: Debika Biswas

Date: 11 May 2025

1. Name of the package, describe what is the basic aim of what the package does or solve?

The name of this package is PySpecKit Spectroscopic Toolkit. This package lets us analyze spectroscopic data from various astronomical objects. While there are instruments that look at specific wavelengths, there aren't many options if I want to look at multiple wavelengths. This package fits the spectra using Gaussian, Voigt profile fitting, and baseline/continuum fitting.

2. Why/how did you select this package?

I selected this package first because I am an Astronomy major and this seemed like an interesting project to work on. I am also already familiar with astropy, so I wanted to learn about something different. Lastly, this semester my research has to do with fitting the spectra of a comet using Gaussians and using that data to find out how many molecules of CO_2 and H_2O are in the nucleus and coma of the comet.

3. How old is the package? Does it have a genealogy, i.e. what related codes came before or after. Are there other codes you can find that solve the same problem?

This package seems to have originated in September of 2011, but development started in 2009. Traditionally the Image Reduction and Analysis Facility or IRAF was used for line fitting of spectra. As of right before the development of PySpecKit, IRAF had stopped development. Because there were no equivalent tools, PySpecKit started being created. This was also before astropy had been created, so it could not be used to solve the problem. After initial development, PySpecKit grew to handle more sophisticated models. In 2012, PySpecKit moved to GitHub and has been there ever since.

There seem to be similar codes that have been developed that fit spectra of objects. One example, which is what I used for my research, is astropy.models. This part of the astropy package fits lines to spectra. For example: Gaussians, lines, and polynomials are a few that astropy.models can utilize. Specutils is another example of an astropy package that can fit lines to spectra. This link was used for the information to answer this question.

4. Is it still maintained, and by the original author(s)? Are there instructions on how to contribute to this project?

PySpecKit is still being maintained by Adam Ginsburg who was also the original author. On how to contribute to the project, they can be emailed at pyspeckit@gmail.com. Issues with PySpecKit can also be posted on their issues page.

5. **Evaluate how easy it was to install and use. What commands did you use to install?**

This package was easy to install. I used a !pip install to do so. I imported PySpecKit after installing it. I also thought that it was pretty easy to use. There were a few quirks I had to work out like subtracting out the baseline/continuum, as this was the only way the code seems to make an accurate fit. Other than this it wasn't too hard to maneuver.

6. **Does it install via the "standard" pip/conda, or is it more complex?**

   All it needs is a !pip install. It is not more complex than this.

7. **Is the source code available? For example, "pip install galpy" may get it to you, but where can you inspect the code?**

   Yes, the source code is available here. Using this page, I can inspect the code.

8. **Is the code used by other packages (if so, give one or two examples). ASCL codes have citations via their ADS link. See also 22.**

Yes astropy is an affiliated package that is used in combination with PySpecKit. Astropy is a python package designed to help with Astronomy calculations and plotting.

9. **How is the code used? Is it command line, python script, or a jupyter notebook, or even a web interface?**

   The code was a python script that I used in a jupyter notebook.

10. **Provide examples using the code. If you prefer to use a jupyter notebook instead of a python script, that's ok. See also 12.**

```
import matplotlib.pyplot as plt
data = np.genfromtxt("ngc253_hb.tab")

wavelength = data[:,0]
flux = data[:,1]

fig, ax = plt.subplots()
ax.scatter(wavelength, flux, marker = '.')
ax.set_xlabel("Wavelength (Angstroms)")
ax.set_ylabel("Flux (unitless)")
ax.set_title("Wavelength vs Flux (visualizing data)")
```

Above is code I wrote to visualize the data.
Below is the code I used to find the fit taken from the source code page on github under the documentation and examples page.

```python
import numpy as np
import pyspeckit as pspeck
from astropy import units as u

# sigma = 10.
# center = 50.

# Add noise
y_below = flux[:207]
y_above = flux[217:]
y_noise = np.concatenate((y_below, y_above))
stddev = np.std(y_noise)
print(stddev)

noise = np.random.randn(wavelength.size)*stddev
error = stddev*np.ones_like(flux)
# data = noise+flux

# this will give a "blank header" warning, which is fine
sp = pyspeckit.Spectrum(data=flux, error=error, xarr=wavelength, xarrkwargs = {'unit': "AA"},
                        unit='Flux (unitless)')

sp.plotter()
sp.baseline()

# Fit with automatic guesses
sp.specfit(fittype='gaussian')
```

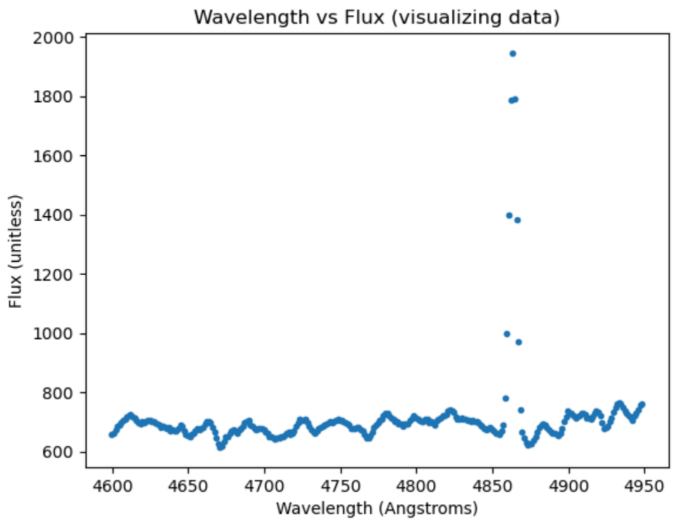
```python
# Fit with input guesses
# The guesses initialize the fitter
# This approach uses the 0th, 1st, and 2nd moments
amplitude_guess = flux.max()
center_guess = 4863.45849609375
width_guess = 5
guesses = [amplitude_guess, center_guess, width_guess]
sp.specfit(fittype='gaussian', guesses=guesses)

sp.plotter(errstyle='fill')
sp.specfit.plot_fit()
```

11. **Does the package produce figures, or are you on your own? Is matplotlib used?**

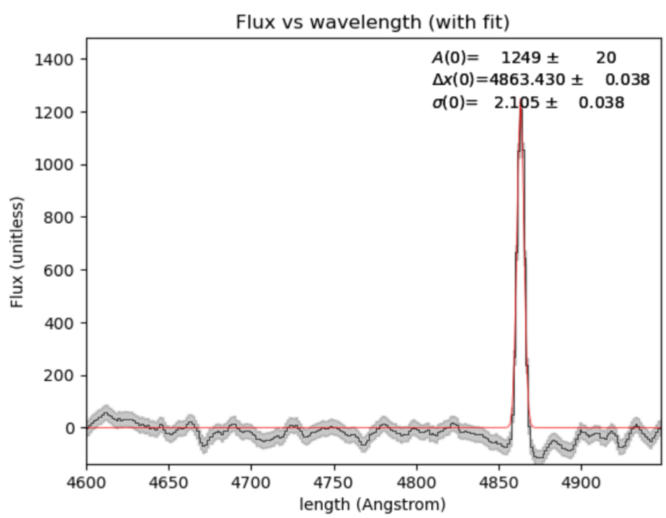
Yes it produces a figure of the spectra fit to the gaussian. The GUI was built using matplotlib but you don't need to import it.

12. **Your code and report should show at least one figure, and create a nice figure caption explaining what it shows. Your notebook should show how the figure was made (i.e. be reproducible). Second figure is optional, but only use it when you need to illustrate something extra.**

I originally just plotted the data file to just visualize the data, as we've been taught to do in class. That plot is below.



I then used the code found to fit the spectra of the galaxy. That figure is below.



Caption: This figure shows the Flux versus Wavelength of NGC253 through H-β. It shows the best fit of the data to a Gaussian curve. In the top right I see the best fit values for amplitude, mean, and standard deviation as well as their errors on the values.

13. Is the package pure python? or does it need accompanying C/C++/Fortran code?

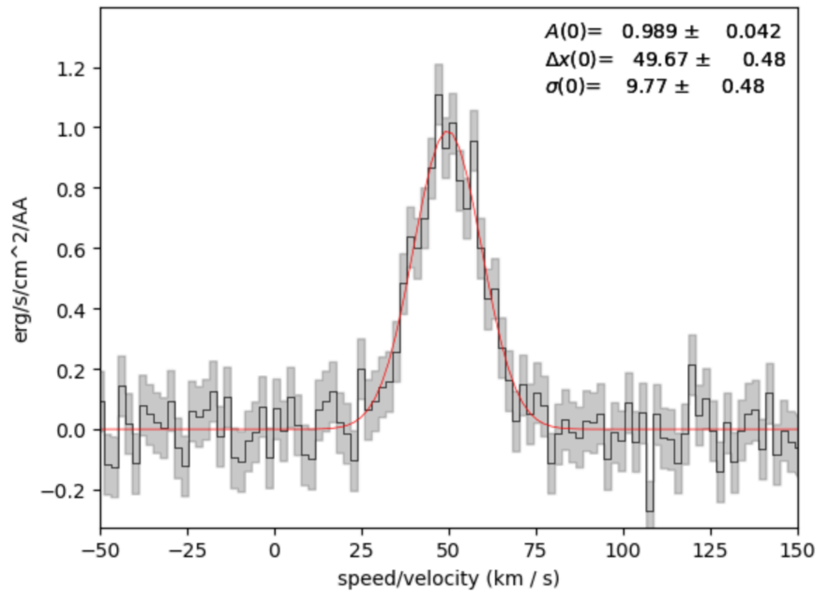
It is able to run only using python.

14. What is the input to the package? Just parameters, or dataset(s), or can they be generated from scratch?

The example code I found was just a spectra produced from scratch, but in my case I loaded in the data and plotted it.

15. What is the output of the package? Just parameters, or dataset(s)?, or just a screen output you would need to capture.

The output was the plot of the spectra with the fit. The figure below shows the figure the sample code made with the data created from scratch.



16. Does the code provide any unit tests, regression or benchmarking?

Yes the code does get tested for different versions of python and different computers. This information was found under the actions page on github. I can see that it has been tested 78 times, so this is a good indicator to show that PySpecKit is well maintained.

17. How can you feel confident the code produces a reliable result? (see also previous question)

I can take a look on, visually speaking, how well the fit lines up with the data. I can also see that the errors on the amplitude, mean, and standard deviation are less than 10% of the values. This can tell us that the fit is found with relative certainty and the errors are not large enough to negate the values I found.

18. What (main) python package(s) does it use or depend on (e.g. numpy, curve_fit, solve_ivp) - how did you find this out?

It uses numpy, astropy, and matplotlib. I found this out from a paper on the PySpecKit.

19. What kind of documentation does the package provide? Was it sufficient for you?

The github readme file had a documentation link where I found the example code I used. It was also enough information for me to work with.

20. If you use this code in a paper, do they give a preferred citation method?

They have preferred a preferred citation method. In the documentation link they have two papers to cite. PySpecKit: Python Spectroscopic Toolkit and Pyspeckit: A Spectroscopic Analysis and Plotting Package .

21. **Provide any other references you used in your report.**
    All of the sources I used have been linked in the previous questions.

22. **Can you find two other papers that used this package? E.g. use ADS citations for ASCL based code. See also 8.**
    Yes there are other papers that use this package. High-energy spectra of LTT 1445A and GJ 486 reveal flares and activity and Powerful Outflows of Compact Radio Galaxies.

23. **Did you have to learn new python methods to use this package? Or was the class good enough to get you through this project.**
    No, I did not have to learn new python methods to use this package. The class was good enough to get me through this package.

24. **Final Disclaimer: you need to state if you have prior experience in using the package or the data, or this is all new to you. In addition, if you collaborated in a group, as long as this is your work.**
    I have worked on a similar research project in the past. For this semester, I used data from JWST's NIRSpec to learn about the spectra of comet 22P/Kopff. This data was in a datacube format, and I used astropy.models. While these projects were similar, I hadn't worked with the same package or the same type of data before.

    I worked in a group on this project with Jasmin Mohommadi, Margaret Haswell, and Zya Woodfork.