



STScI | SPACE TELESCOPE
SCIENCE INSTITUTE

EXPANDING THE FRONTIERS OF SPACE ASTRONOMY

JWST Data Analysis Tools: Open Development of Community Software

Erik Tollerud



Astropy Coordination
Committee Member

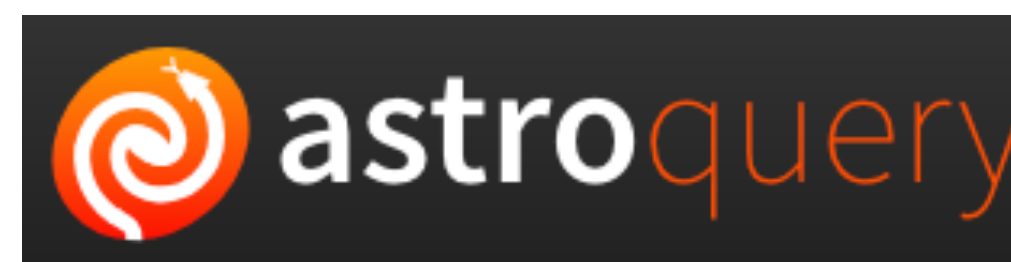
Data Analysis Tools Branch
Project Scientist





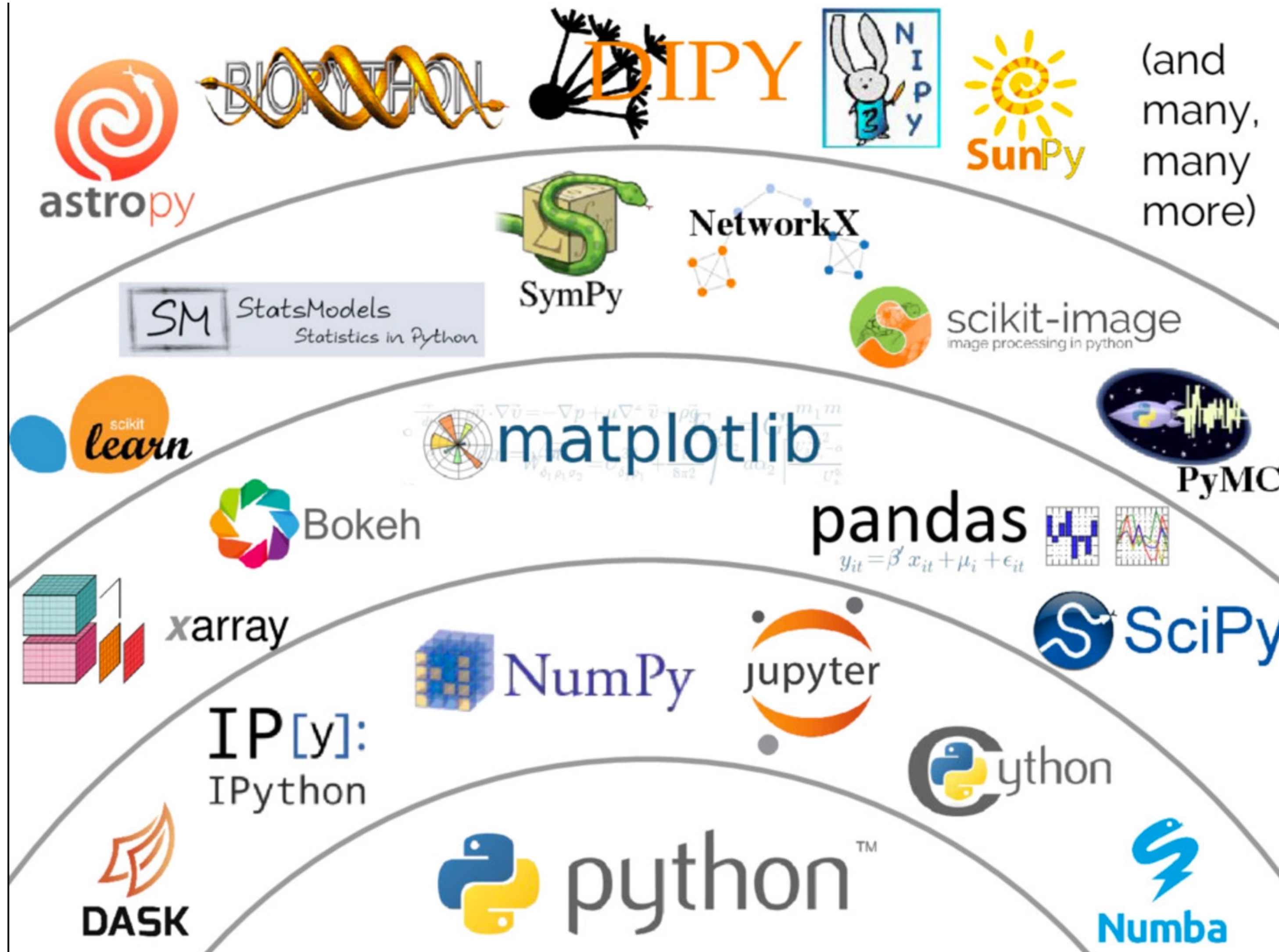
Software at STScI for the Community

Software that is meant to be used by the astronomy community to do their science.





JWST/STScI Community Software is Built on a Shared Foundation



**OPENNESS
LEADS TO
SHARING
THE LOAD**

**ALTHOUGH:
OPEN SOURCE
≠
OPEN
DEVELOPMENT**



What is Open Development?

- A way to build software that emphasizes processes where every step is done publicly.
- Anyone, internal or external, can participate as a 1st-class citizen.
- This makes the software also *by* the astronomy community. (Like **you!**)

GitHub



This repository Search Pull requests Issues Marketplace Explore

astropy / astropy Unwatch 163 Unstar 1,436 Fork 769

Code Issues 709 Pull requests 84 Projects 1 Wiki Insights Settings

Allow FITS tables with time columns (not written by Astropy) to be read by io.fits #6442

Open AustereCuriosity wants to merge 14 commits into astropy:master from AustereCuriosity:Time_read_fits

Conversation 87 Commits 14 Files changed 6 +595 -92

AustereCuriosity commented on Aug 14 • edited Contributor

This works for Chandra files and XMM files. I have added a test for reading Chandra files. The test needs to be more detailed and will be modified soon.

NOTE: The chandra file is huge, so I'll probably use another file or reduce it.

astropy-bot bot commented on Aug 14 • edited

Hi there @AustereCuriosity 🙌 - thanks for the pull request! I'm just a friendly bot that checks for issues related to the changelog and making sure that this pull request is milestone and labelled correctly. This is mainly intended for the maintainers, so if you are not a maintainer you can ignore this, and a maintainer will let you know if any action is required on your part 😊.

Everything looks good from my point of view! 👍

If there are any issues with this message, please report them [here](#)

bsipocz added this to the v3.0.0 milestone on Aug 14

bsipocz added io.fits Work in progress labels on Aug 14

bsipocz commented on Aug 14 Member

@AustereCuriosity - If you need this file, it can go into the astropy-data repository and then can be accessed as remote-data from here.

1

Reviewers: taldcroft (x), mhvk (✓)

Assignees: No one—assign yourself

Labels: Affects-dev, Affects-release, Enhancement, io.fits

Projects: None yet

Milestone: v3.0.0

Notifications: Unsubscribe

You're receiving notifications because you're subscribed to this repository.



What is Open Development?

- Starting from a Pull Request... Anyone can review
- Commenters may be scientists
- Or engineers
- The original author then chooses how to update it
- The maintainers just hit “merge”. The user has become a contributor.

GitHub



This repository Search Pull requests Issues Marketplace Explore

astropy / astropy Unwatch 163 Unstar 1,436 Fork 769

Code Issues 709 Pull requests 84 Projects 1 Wiki Insights Settings

Allow FITS tables with time columns (not written by Astropy) to be read by io.fits #6442

Open AustereCuriosity wants to merge 14 commits into astropy:master from AustereCuriosity:Time_read_fits

Conversation 87 Commits 14 Files changed 6 +595 -92

AustereCuriosity commented on Aug 14 • edited Contributor

This works for Chandra files and XMM files. I have added a test for reading Chandra files. The test needs to be more detailed and will be modified soon.

NOTE: The chandra file is huge, so I'll probably use another file or reduce it.

astropy-bot bot commented on Aug 14 • edited

Hi there @AustereCuriosity 🙌 - thanks for the pull request! I'm just a friendly bot that checks for issues related to the changelog and making sure that this pull request is milestone and labeled correctly. This is mainly intended for the maintainers, so if you are not a maintainer you can ignore this, and a maintainer will let you know if any action is required on your part 😊.

Everything looks good from my point of view! 👍

If there are any issues with this message, please report them [here](#)

bsipocz added this to the v3.0.0 milestone on Aug 14

bsipocz added io.fits Work in progress labels on Aug 14

bsipocz commented on Aug 14 Member

@AustereCuriosity - If you need this file, it can go into the astropy-data repository and then can be accessed as remote-data from here.

1

Reviewers: taldcroft, mhvk

Assignees: No one—assign yourself

Labels: Affects-dev, Affects-release, Enhancement, io.fits

Projects: None yet

Milestone: v3.0.0

Notifications: Unsubscribe

You're receiving notifications because you're subscribed to this repository.



Open Development is also Open *Planning*

- Design and planning of these tools also occurs in the open, in the same place the code lives.
- Contributions/opinions are accepted from anyone.

The screenshot shows the GitHub interface for the repository `spacetelescope/specviz`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository header shows 9 Watchers, 20 Stars, and 24 Forks. The main content area displays a list of 57 open issues, filtered by `is:issue is:open`. The issues list includes:

- Re-implement loader wizard** #416 opened 4 days ago by nmeal
- Encoding should be explicitly specified for linelist** #408 opened 14 days ago by astrofrog (Labels: bug, core)
- Roadmap for specutils v0.3 compatibility** #407 opened 19 days ago by weaverba137
- Incorporate XSonify into SpecViz** #405 opened 19 days ago by lfulmer
- Release specviz on PyPI** #404 opened 22 days ago by astrofrog
- specviz --version and --help launch specviz** #403 opened 22 days ago by astrofrog (Label: bug)
- Accessibility of specviz** #398 opened 28 days ago by stscicrawford (Labels: discussion, enhancement)
- Auto-crop the limits of the viewer to the data** #378 opened on Mar 21 by Cadair

You can do any of this right now!



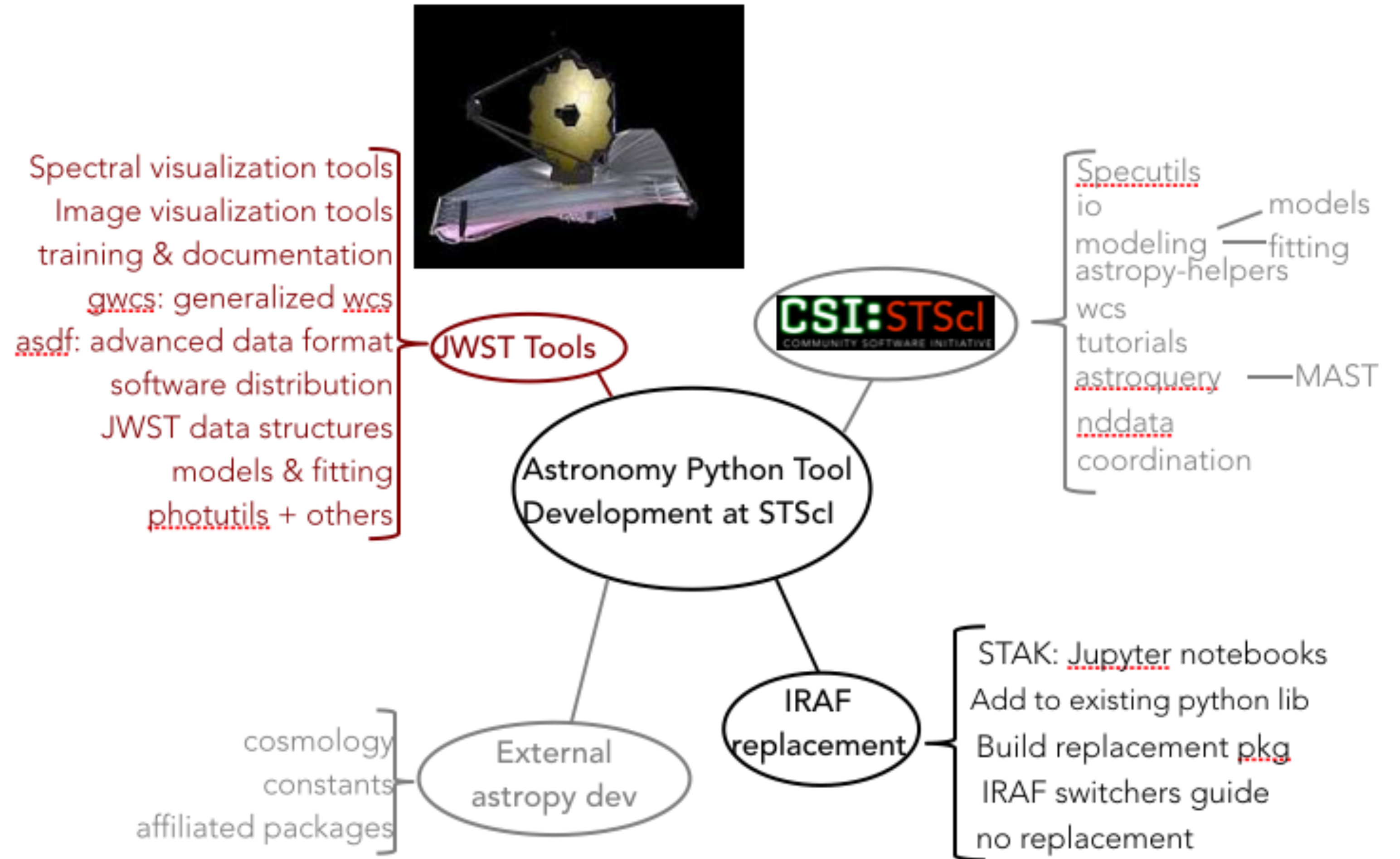
JWST and HST Data Analysis Tools are Open Developed

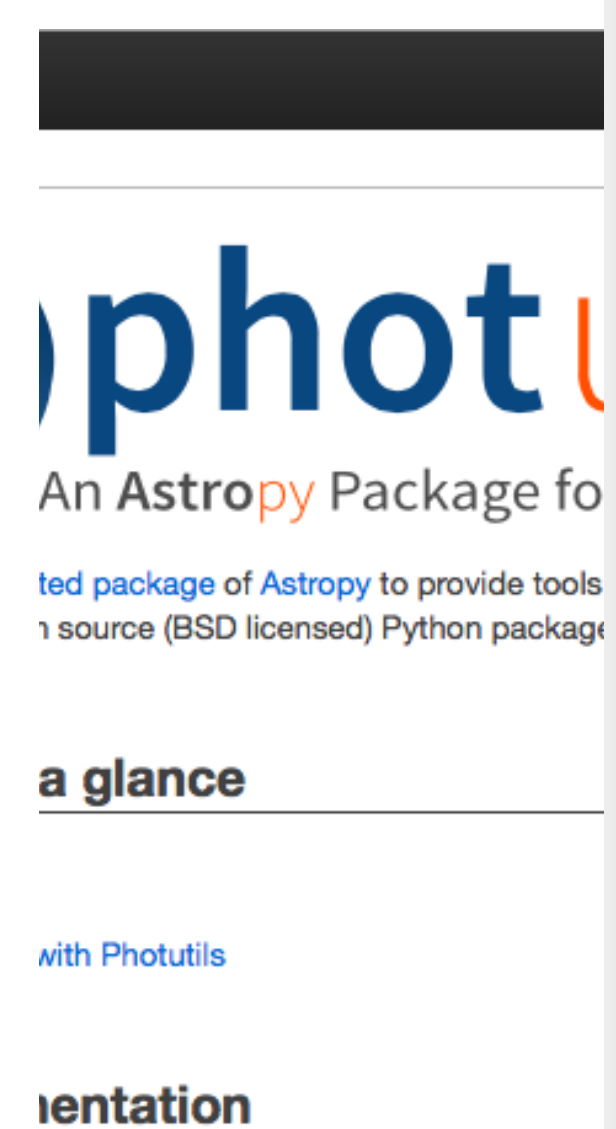
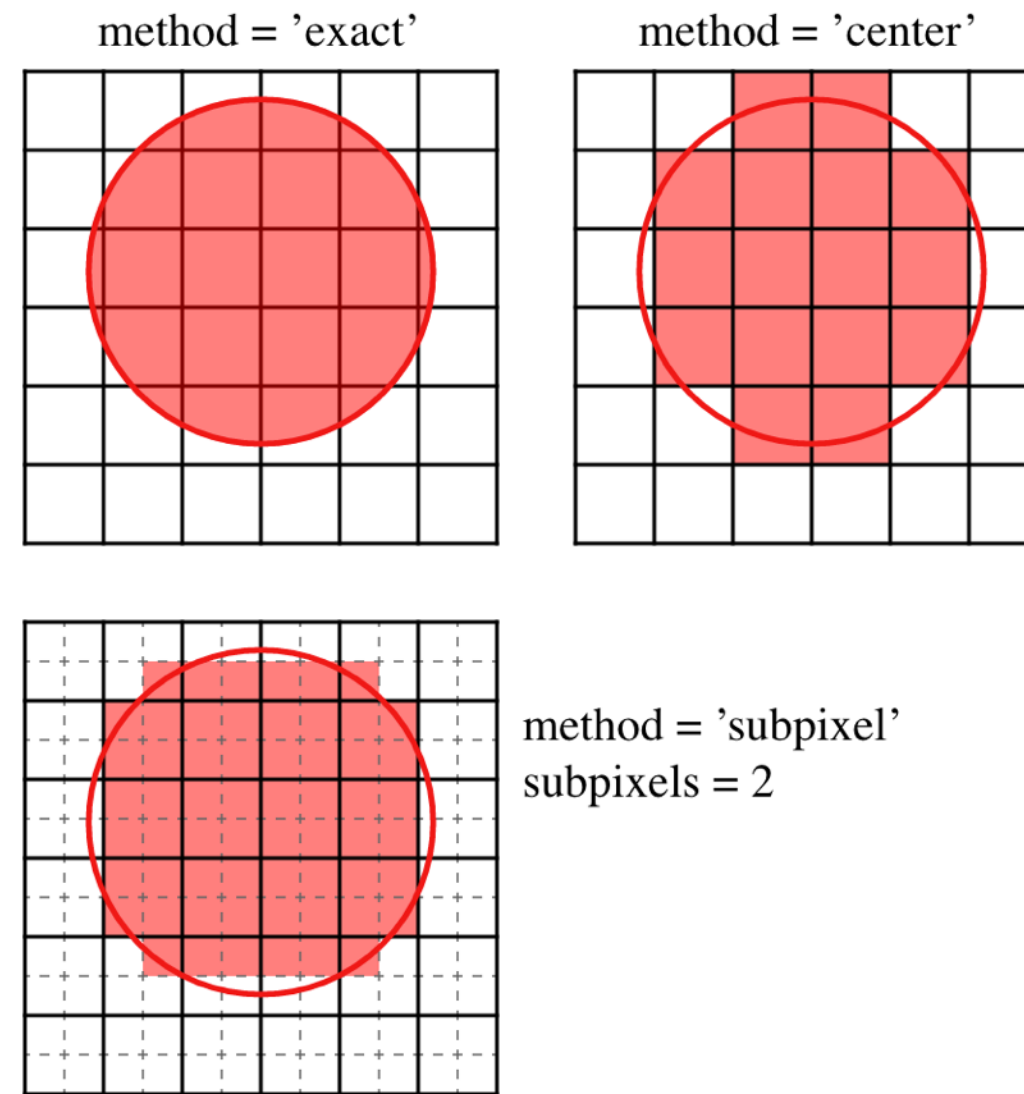


• DATs: Post-Pipeline Tools

- Analysis Tools
 - ▶ Photutils (w/Astropy)
 - ▶ Specutils (w/Astropy)
 - ▶ WebbPSF
- Visualization Tools
 - ▶ SpecViz
 - ▶ MOSViz
 - ▶ CubeViz
 - ▶ Imexam
 - ▶ Ginga
- STAK (“Python for IRAF-users”)

• JWST Pipeline





```
In [41]: from photutils import CircularAnnulus
positions = [(90.73, 59.43), (73.63, 139.41), (43.62, 61.63)]
aper = CircularAperture(positions, r=3)
bkg_aper = CircularAnnulus(positions, r_in=10., r_out=15.)
apers = [aper, bkg_aper]
```

Now, perform the photometry.

```
In [42]: phot = aperture_photometry(data, apers)
phot.rename_column('aperture_sum_0', 'aperture_sum')
phot.rename_column('aperture_sum_1', 'annulus_sum')
phot
```

```
Out[42]: <QTable length=3>
```

id	xcenter	ycenter	aperture_sum	annulus_sum
	pix	pix		
int64	float64	float64	float64	float64
1	90.73	59.43	0.0866436609693	0.0199107563833
2	73.63	139.41	0.393646538117	0.0358905305285
3	43.62	61.63	0.130109734904	0.0166684757391

- DATs: Post-Pipeline Tools

- Analysis Tools
 - ▶ **Photutils** (w/Astropy)
 - ▶ Specutils (w/Astropy)
 - ▶ WebbPSF

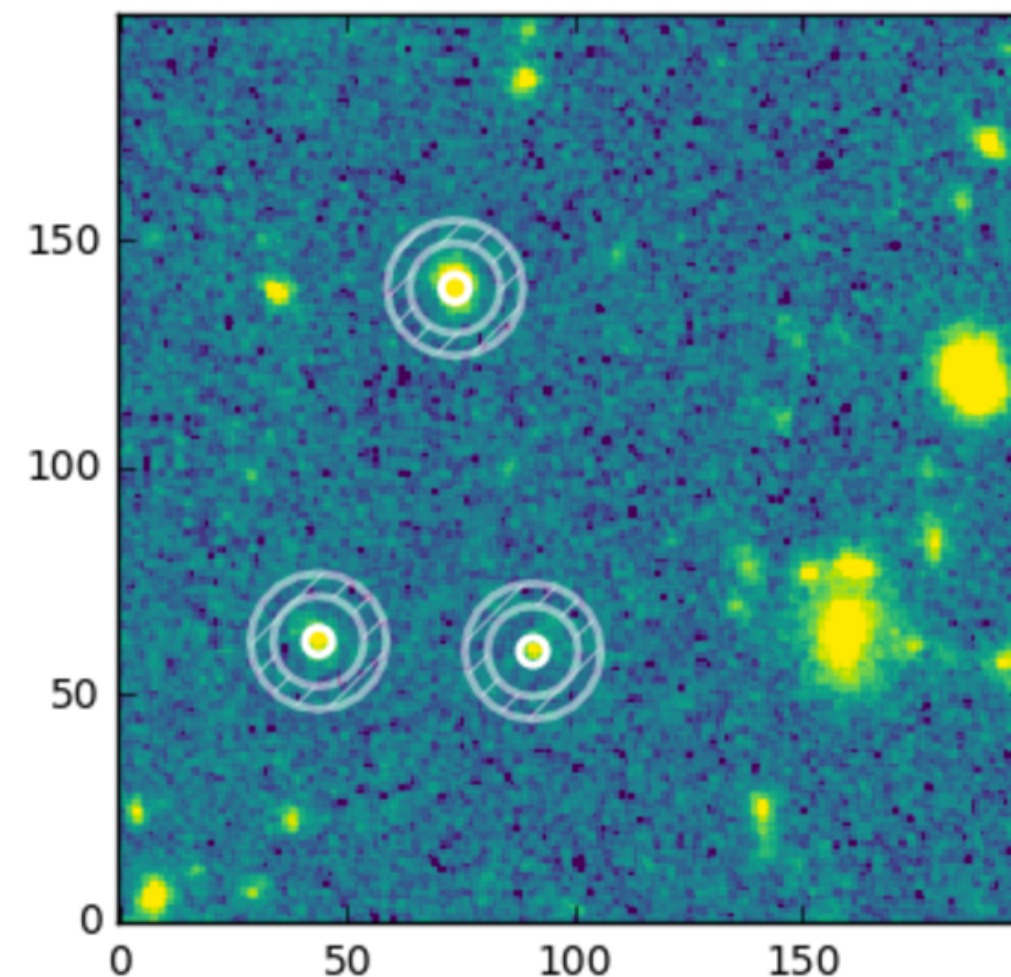
- Visualization Tools

- ▶ SpecViz
- ▶ MOSViz
- ▶ CubeViz
- ▶ Imexam
- ▶ Ginga

- STAK (“Python for IRAF-users”)

- JWST Pipeline

```
In [46]: plt.imshow(scale_image(data, scale='sqrt', percent=98.))
aper.plot(color='white', lw=2)
bkg_aper.plot(color='white', lw=2, hatch='///', alpha=0.5)
```

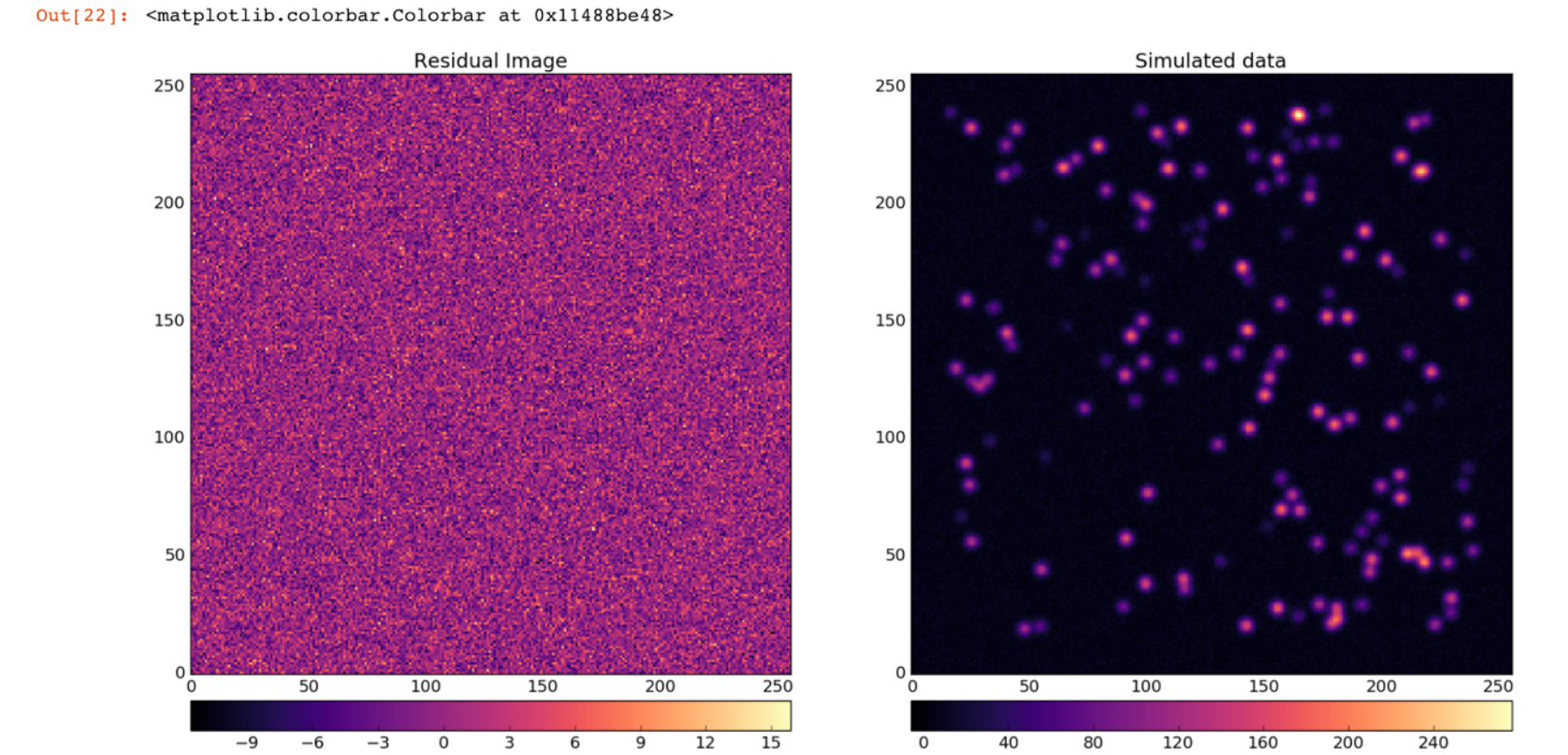


```
In [20]: from astropy.table import Table
positions = Table(names=['x_0', 'y_0'], data=[starlist['x_mean'], starlist['y_mean']])
```

```
In [21]: photometry_results = basic_photometry(image=image, positions=positions)
WARNING: Both positions and finder are different than None, which is ambiguous. finder is going to be ignored. [photutils.psf.photometry]
```

```
In [22]: plt.subplot(1,2,1)
plt.imshow(basic_photometry.get_residual_image())
plt.title('Residual Image')
plt.colorbar(orientation='horizontal', fraction=0.046, pad=0.04)

plt.subplot(1,2,2)
plt.imshow(image)
plt.title('Simulated data')
plt.colorbar(orientation='horizontal', fraction=0.046, pad=0.04)
```





Specutils

[HTTPS://SPECUTILS.READTHEDOCS.IO](https://specutils.readthedocs.io)



An Astropy-coordinated package with data structures and standard analysis functions for spectroscopy.

- DATs: Post-Pipeline Tools

- Analysis Tools

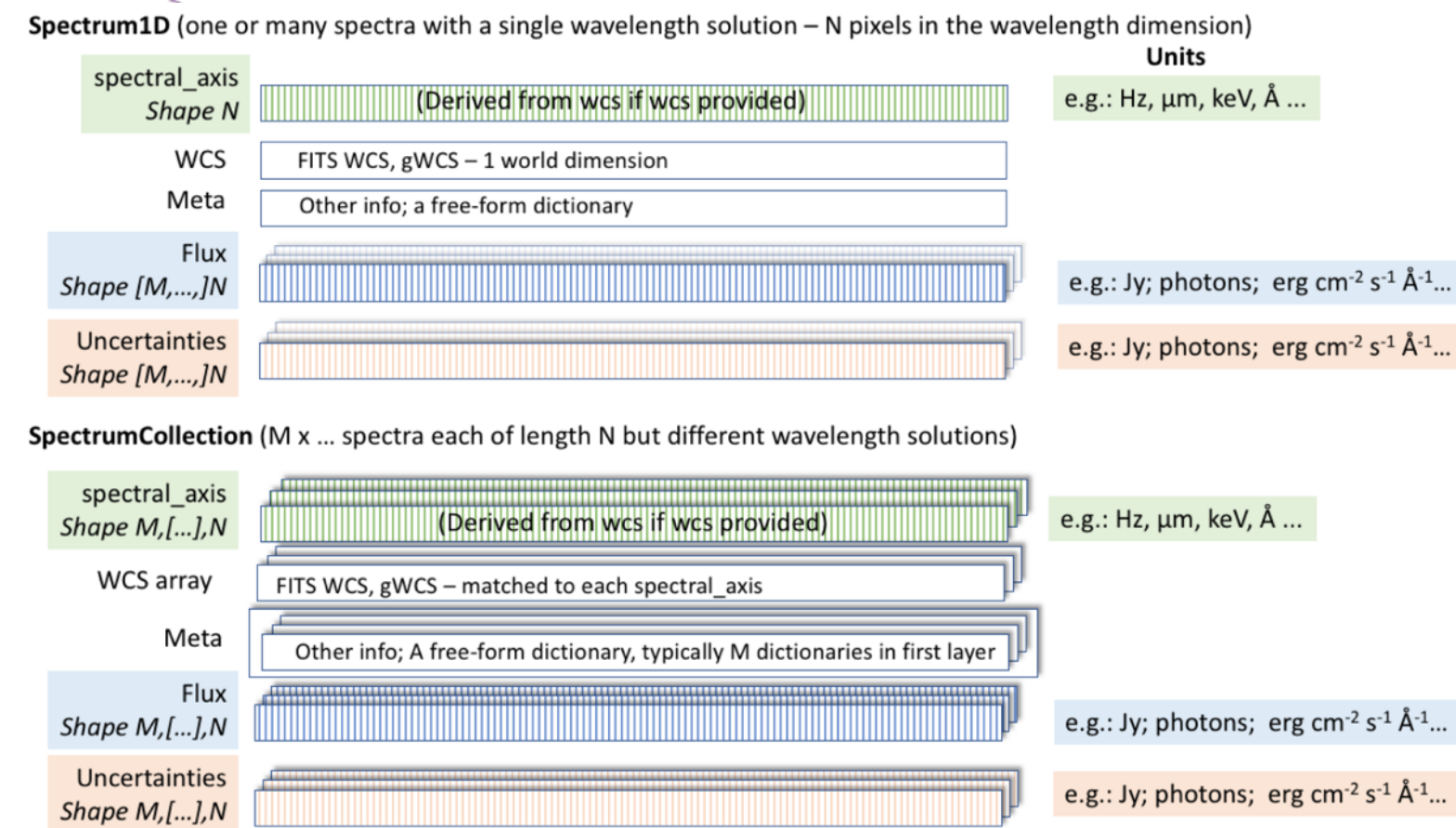
- ▶ Photutils (w/Astropy)
- ▶ **Specutils** (w/Astropy)
- ▶ WebbPSF

- Visualization Tools

- ▶ SpecViz
- ▶ MOSViz
- ▶ CubeViz
- ▶ Imexam
- ▶ Ginga

- STAK (“Python for IRAF-users”)

- JWST Pipeline



- Pythonic data structures of spectra^

- Analysis functions ->

- Flux, Centroids, FWHM
- Continuum fitting / subtraction
- Spectral arithmetic, respecting units
- Line modeling

```
In [ ]: import io
import warnings
warnings.simplefilter('ignore')

import numpy as np

from astropy.coordinates import SkyCoord
from astropy import units as u
from astropy import table

import specutils, specutils.fitting, specutils.analysis

from astroquery.sdss import SDSS

from IPython import display

%matplotlib inline
from matplotlib import pyplot as plt

We download a spectrum from the SDSS for this example... But the same specutils object will be available for e.g. NIRSpec as soon as there are actual data to load!

In [ ]: spec_file = SDSS.get_spectra(plate=6497, mjd=56329, fiberID=139)[0]
spec_file.writeto('spectrum.fits', overwrite=True)
spec = specutils.Spectrum1D.read('spectrum.fits', format='SDSS-III/IV spec')

In [ ]: url = 'http://skyserver.sdss.org/dr12/SkyserverWS/ImgCutout/getjpeg?ra={}&dec={}&width=256&height=256&scale=.1'.format(s
display.Image(url=url)

Create a continuum-subtracted spectrum, and plot it

In [ ]: basic_continuum = specutils.fitting.fit_generic_continuum(spec)
subspec = spec - basic_continuum(spec.spectral_axis)

plt.step(subspec.wavelength.to(u.angstrom), subspec.flux)
```



MOSViz
Powered by glue



A visualization tool for Multi-Object Spectrographs (heavily leverages Glue)

- DATs: Post-Pipeline Tools

- Analysis Tools

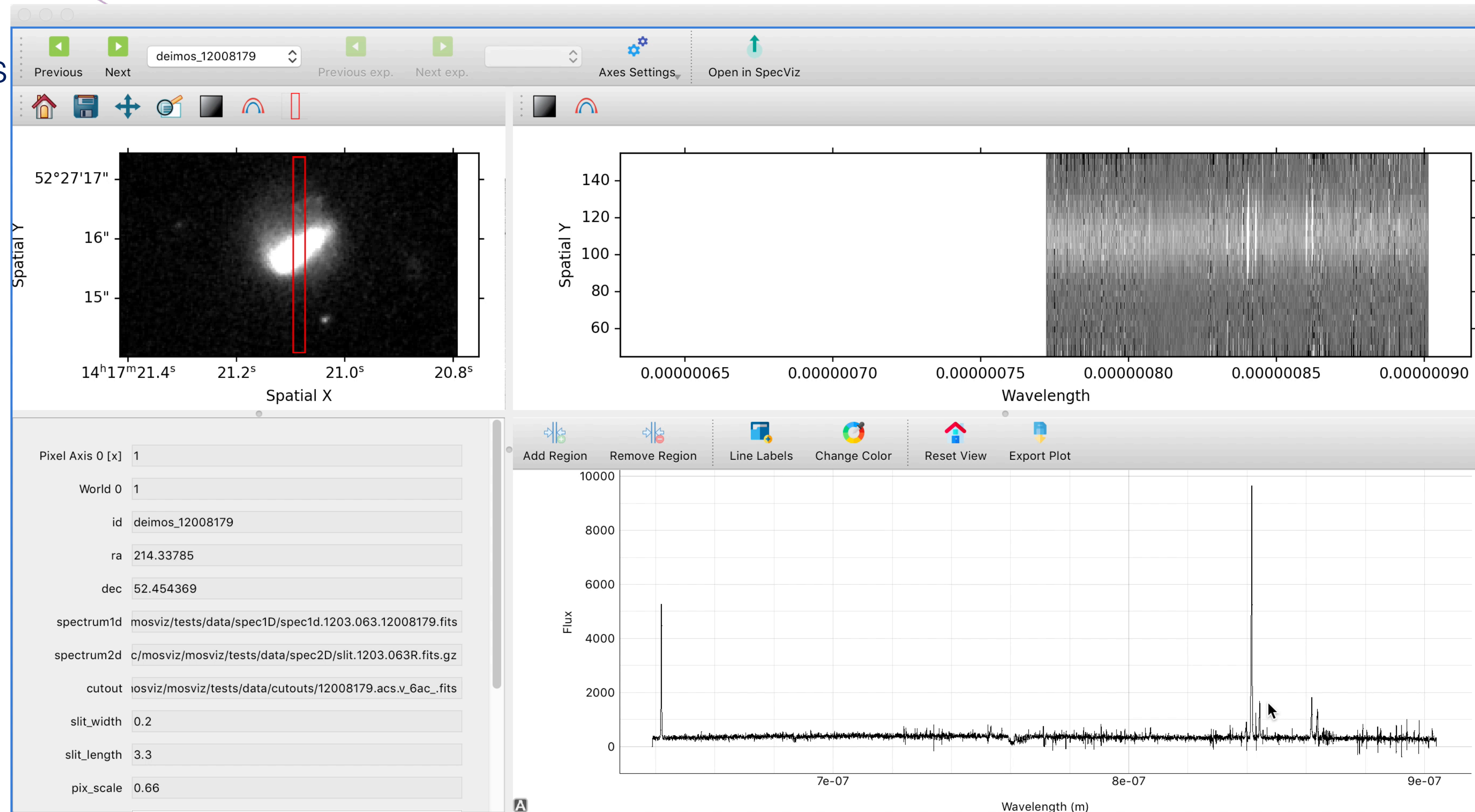
- ▶ Photutils (w/Astropy)
- ▶ Specutils (w/Astropy)
- ▶ WebbPSF

- Visualization Tools

- ▶ SpecViz
- ▶ **MOSViz**
- ▶ CubeViz
- ▶ Imexam
- ▶ Ginga

- STAK (“Python for IRAF-users”)

- JWST Pipeline





Ginga (+ stginga)



- DATs: Post-Pipeline Tools
 - Analysis Tools
 - ▶ Photutils (w/Astropy)
 - ▶ Specutils (w/Astropy)
 - ▶ WebbPSF
 - Visualization Tools
 - ▶ SpecViz
 - ▶ MOSViz
 - ▶ CubeViz
 - ▶ Imexam
 - ▶ **Ginga**
 - STAK (“Python for IRAF-users”)
- JWST Pipeline

```
voilable_astrowidgets.ipynl Python (astro36)

[ ]: import os
import sys

sys.path.insert(1, '/Users/erik/src/astrowidgets')
import astrowidgets

[ ]: default_file_name = 'https://astropy.stsci.edu/data/photometry/spitzer_exam
if os.path.exists('.to_load_astrowidgets'):
    with open('.to_load_astrowidgets') as f:
        s = f.read()
        if s.strip() != '':
            default_file_name = s.strip()
    os.unlink('.to_load_astrowidgets')

[ ]: import ipywidgets
from astrowidgets import ImageWidget

from astropy.nddata import CCDData
from astropy import units as u

w = ImageWidget()
zoom_slider = ipywidgets.FloatSlider(min=.1, max=5, step=.1, value=.4)

def f(data):
    w.zoom_level = zoom_slider.value
    zoom_slider.observe(f, 'value')

[ ]: to_load_widget = ipywidgets.Text(default_file_name)
to_load_widget.layout.width='26em'
hdunum_widget = ipywidgets.IntText(0)
hdunum_widget.layout.width = '50px'
load_button = ipywidgets.Button(description='Load')
load_button.layout.width = '5em'

def load_func(button):
```

An image viewer in Python (Works both out of and inside Notebooks)



In Development: More Interactive Notebooks



- DATs: Post-Pipeline Tools
 - Analysis Tools
 - ▶ Photutils (w/Astropy)
 - ▶ Specutils (w/Astropy)
 - ▶ WebbPSF
 - Visualization Tools
 - ▶ SpecViz
 - ▶ MOSViz
 - ▶ CubeViz
 - ▶ Imexam
 - ▶ Ginga
 - STAK (“Python for IRAF-users”)
- JWST Pipeline

The screenshot shows a Jupyter Notebook titled "Example_spec_redshifting" running on a local host. The notebook contains a plot of a spectrum with two vertical dashed lines indicating regions of interest. Below the plot, there is a text block explaining the manual process of plotting and fiddling with view region sizes. This is followed by an "Interactive version" section with instructions on how to use the tool. The notebook also shows several code cells: one for importing a helper library, one for creating a line finder object, one for displaying the line finder, one for getting regions, and one for creating a table to display the results. The table columns are 'name' (dtype='S10') and 'centroid' (dtype=float, unit=u.angstrom).

Will be at <https://spacetelescope.github.io/notebooks/>



STScI Supports Community Open-Developed Projects

Contributions to astronomy and scientific libraries are a part of tool development for the missions. This lets us both leverage these libraries and “give back” our knowledge.





These Tools Are Also Used by Other Organizations/Institutes



ASDF - Advanced Scientific Data Format



**This is how STScI is helping build a new generation
of software tools that are not just for astronomy,
but also *by* and *for*, the astronomy *community*.**

