What Are Data Analysis Tools? The Things After the Pipeline

- DATs: Post-Pipeline Tools
  - Analysis Tools
    - Astropy
    - Photutils
    - Specutils
  - Visualization Tools
    - Python Imexam (+ds9)
    - Ginga
    - SpecViz
    - (MOSViz)
    - (CubeViz)
Data Analysis Software at STScI is for the Community

Software that is meant to be used by the astronomy community to do their science. This talk is about some newer developments of interest to the HST user community.

1. IRAF
2. Astropy Project
3. photutils, specutils, imexam, astroquery
• IRAF was amazing for its time, and the DAT for generations of astronomers

IRAF - Image Reduction and Analysis Facility
NOAO is transitioning IRAF to an end-of-support state, and has taken NOAO's IRAF distribution offline pending a final copyright and licensing review of the source code. Users interested in new IRAF installations during this review period may wish to consider the following two distributions:

1. The AstroConda Legacy Software Stack
2. The IRAF Community Distribution

STScI will no longer support PyRAF as of Oct 1st, 2019. Users will still be able to access an installation of PyRAF through Astroconda. However, STScI will no longer answer help calls related to PyRAF installation, bugs, or other usage questions. All support currently provided for other packages in Astroconda will continue.

What is the long-term replacement strategy?
Python's Scientific Ecosystem Has what Astro Needs

Python is now the single most-used programming language in astronomy.

And 3rd-most in industry... Largely because of the vibrant scientific and numerical ecosystem (=Data Science):

(Which HST's pipelines helped start!)
IRAF<->Python is not 1-to-1. Hence STAK

Supporting both STScI's internal scientists and the astronomy community at large (viewers like you) requires that the community be able to transition from IRAF to the Python-world. This is the purpose of the STAK notebooks


Fully supported via the STScI Help Desk

(And it is all open for the user community to contribute to and improve:)

https://github.com/spacetelescope/stak-notebooks
Example: IRAF’s `images.imfilter` tasks map to several different packages and functions or objects in Python.

(And all the code is runnable if you download the notebook)
Inspired by this, a larger effort continues to write and serve Jupyter notebooks that show specific use cases to show the community how to use the Python-based DATs on HST data.
New Development: Deep Synergy w/ JWST DATs

- Data Analysis Tools
  - Astropy
  - Photutils (w/Astropy)
  - Specutils (w/Astropy)
  - WebbPSF
- Visualization Tools
  - Python Imexam (+ds9)
  - Ginga
  - SpecViz
  - (MOSViz)
  - (CubeViz)

JWST DATs are almost entirely in Python, and meet many of HST's needs with only minor additions.

w/Harry Ferguson, Susan Kassin
STScI Community Software is Built on a Shared Foundation

Core Shared Philosophy:
Open Development (≠ Open Source)

Diagram credit: Jake VanderPlas
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!)
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.
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.

You can do any of this right now!
Astropy

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

- **Visualization Tools**
  - Python Imexam (+ds9)
  - Ginga
  - SpecViz
  - (MOSViz)
  - (CubeViz)

- Units and “Quantities” (arrays with units that act the way you’d expect). Integrated with comprehensive astro-appropriate physical constants
- Date/time good to nanoseconds over a Hubble time
- Celestial coordinates and their transformations
- Table manipulation, including many arcane astro formats

- **nddata**: Image analysis and interoperability data structures
- Astro-appropriate convolution
- WCS (pixel ↔ sky mapping)
- Extensible I/O: **FITS**, VOTable, hdf5, custom
- Astro-relevant data models and compatible fitting/optimization
- Common Astrostatistics tools
- Cosmology tools

http://docs.astropy.org
• **Data Analysis Tools**
  - Astropy
  - **Photutils** (w/Astropy)
  - Specutils (w/Astropy)
  - WebbPSF

• **Visualization Tools**
  - Python Imexam (+ds9)
  - Ginga
  - SpecViz
  - (MOSViz)
  - (CubeViz)
A Demo of photutils on ACS data

Aperture Photometry with photutils

Not shown here: Crowded-Field PSF Photometry (a la DAOPHOT/DOLPHOT) in the same framework.

What is aperture photometry?

The most common method to measure the flux from a celestial source is aperture photometry. This kind of photometry measures the amount of flux within a region of defined shape and size (an aperture) surrounding a source. The ideal aperture would capture all of the flux emitted from the desired source, and none of the flux emitted from the surrounding sky or nearby sources. Especially when performing photometry on image data that includes a number of sources with varying size and shape, it is important to perform aperture corrections to account for imperfect apertures and better constrain photometric errors.

The photutils package provides tools for performing photometry with apertures of various shapes.

What does this tutorial include?

This tutorial covers how to perform aperture photometry with photutils, including the following methods:

- Creating Apertures
  - Circular Apertures
  - Elliptical Apertures
  - Sky Apertures with WCS
Specutils

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

- Pythonic data structures of spectra

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

Data Analysis Tools
- Astropy
- Photutils (w/Astropy)
- Specutils (w/Astropy)
- WebbPSF

Visualization Tools
- Python Imaging (+ds9)
- Ginga
- SpecViz
- (MOSViz)
- (CubeViz)

https://specutils.readthedocs.io
A Demo of *specutils* on STIS data

```python
import numpy as np
from matplotlib import pyplot as plt
from IPython import display
from astropy import units as u
from astropy import modeling
from astropy.table import Table
from astropy.visualization import quantity_support
import specutils
import analysis, fitting
```

Start by Loading a specific PID's STIS dataset from MAST

```python
from astroquery.mast import Observations
obses = Observations.query_criteria(proposal_id=9117)
sub_obs = obses[obses['target_name']=='M82-A-P051']
sub_obs
```

```python
prods = Observations.get_product_list(sub_obs[0])
prods
```
Imexam (in Python)

- Data Analysis Tools
  - Astropy
  - Photutils (w/Astropy)
  - **Specutils** (w/Astropy)
  - WebbPSF
- Visualization Tools
  - **Python Imexam** (+ds9)
  - Ginga
  - SpecViz
  - (MOSViz)
  - (CubeViz)

Lead: Megan Sosey
Specviz

- Data Analysis Tools
  - Astropy
  - Photutils (w/Astropy)
  - Specutils (w/Astropy)
  - WebbPSF

- Visualization Tools
  - Python Imexam (+ds9)
  - Ginga
  - SpecViz
  - (MOSViz)
  - (CubeViz)
But not everything comes for free

- Some of these tools require more directed effort to apply newly-developed JWST tools to HST.
- Grism tools have been developed but tend to be problem-specific. JWST is more general but a lot of details are missing for HST.
- Sophisticated PSF modeling tools exist for JWST (and WFIRST), and could replace tinytim, but need work to be backported and tested against HST’s cameras.
- Tools for empirical PSFs for astrometry and photometry are being developed for JWST from knowledge of HST, but need customization for HST.
- (Your favorite problem could go here!)
- We may be able to do this if you tell us it’s a priority, but we want to know what you want.
This is how STScI is helping build a new generation of data analysis tools that are not just for science with HST, but also by and for, its community.

But we need your input! Comments/questions?
Possibilities include:
Should we focus on grism tools/aXe replacement?
Or better PSFs/tinytim replacement?