

# Python for Variable Star Astronomy a status update

Matt Craig
Department of Physics and Astronomy

# Acknowledgements



- Current students
  - IsobelSnellenberger
  - Madelyn Madsen

- Former students
  - Adam Kline
  - Erin Aadland
  - Andy Block
  - •Jane Glanzer
  - Elias Holte
  - •Laura Maixner
  - •Stefan Nelson
  - Elizabeth Dougherty
  - Nathan Walker
  - •Laura Herzog
  - Michael Meraz
  - Connor Stotts
  - Nathan Heidt
- Colleagues (MSUM)
  - Juan Cabanela
  - Linda Winkler



#### Outline



- Motivation
- Educational materials
- Image viewer
- Source detection and photometry
- Data reduction
- •VSP

#### Motivation



- Prepare undergraduate students for career
  - Include some programming
  - Choose accessible language
  - Re-use as much community software as possible
- Do science: TESS/exoplanet follow-up

#### Goals



- Software interoperability
  - Read and write AstroImageJ photometry files
  - Generate/read AIJ source lists
- Data "interoperability"
  - TESS-style for reporting TESS data
  - AAVSO-style for reporting variable star data
- Let others do the hard work
  - Use the community-based astro software large institutions are developing
  - Make user interaction browser-based, built on tools widely used outside astro

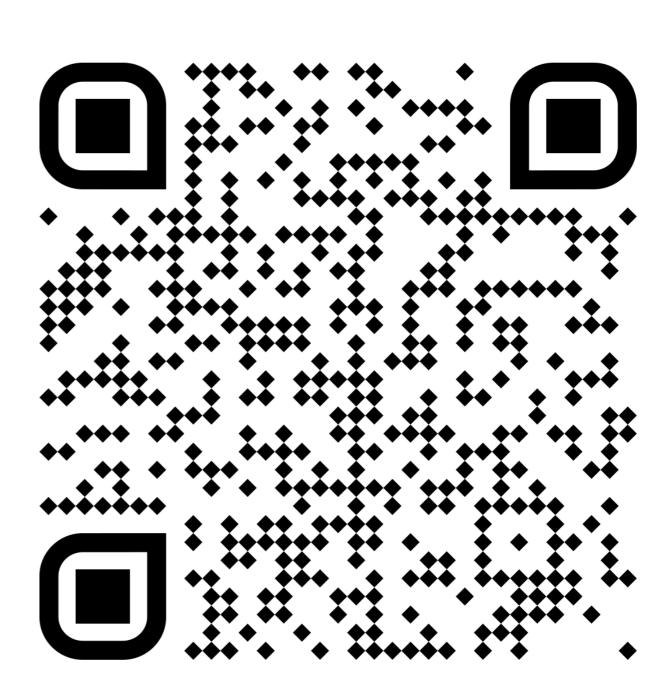
#### **Guide to Data Reduction**



•http://bit.ly/ccd-guide

#### **CCD Data Reduction Guide**

- 1. Understanding astronomical images
- 2. Overscan and bias images
- 3. Dark current and dark frames
- 4. Flat fielding
- 5. Calibrating science images
- 6. Finding and dealing with bad pixels



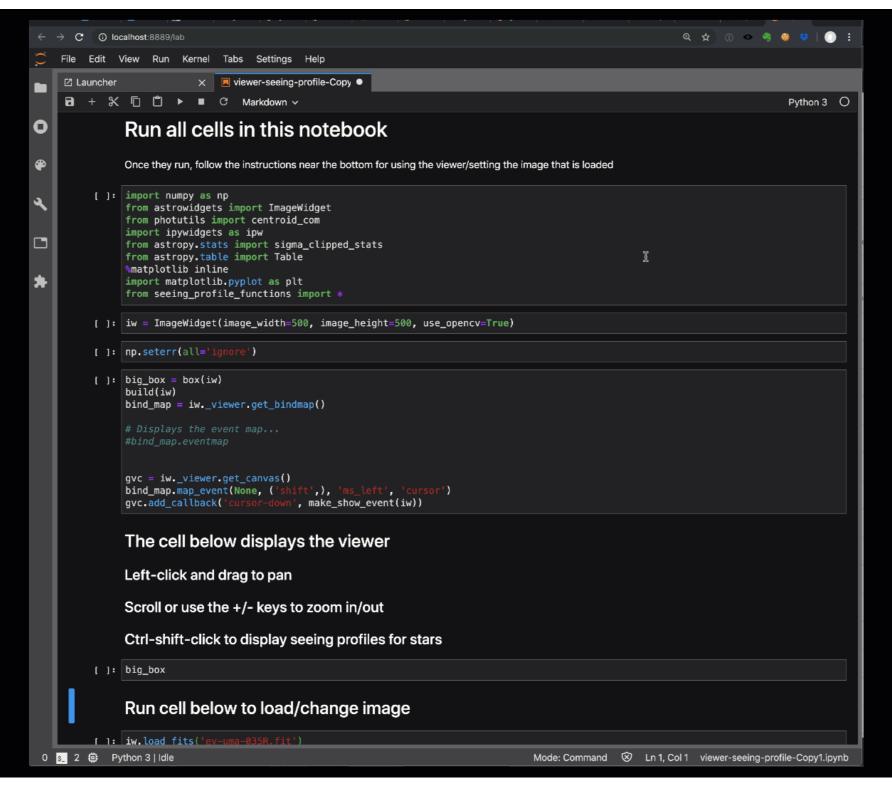
# Image viewer



- •Browser-based; can run
  - on cloud server
  - on local machine
- •Interaction with browser and Python uses
  - Jupiter framework
  - widely support by data science and finance

# Image viewer





# Image viewer



- Viewer design
  - Python designed to allow multiple implementations in future
  - •STScI considering its use internally
  - •LSST considering plugging their viewer into this framework

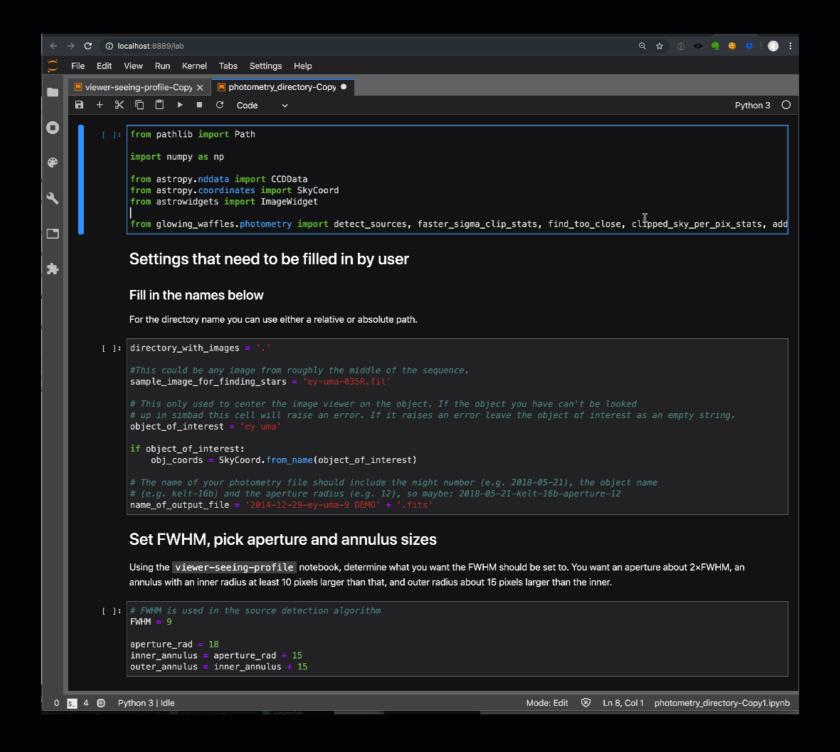




- Source detection using sample image
- •Perform aperture photometry on all images in folder for those sources











- Produces (per source per image):
  - Net counts
  - Instrumental magnitude
  - Filter
  - Sky background
  - •RA/Dec
  - Error (from CCD equation)

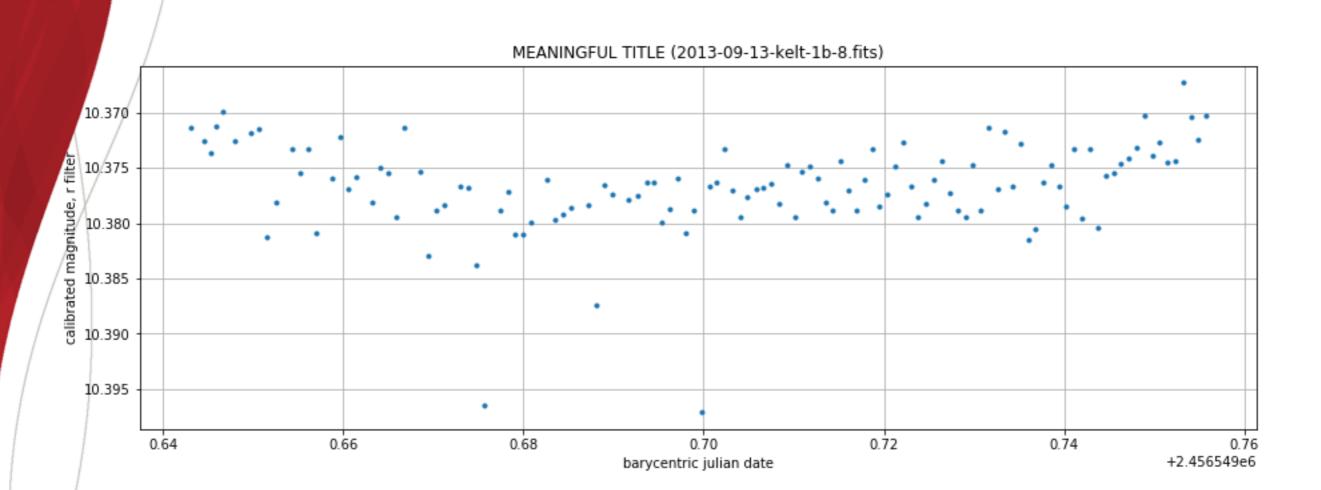




- Determine color correction and zero point for each frame
  - Use APASS stars in frame as standard stars
  - Prefer those with small error
- Apply corrections to all sources

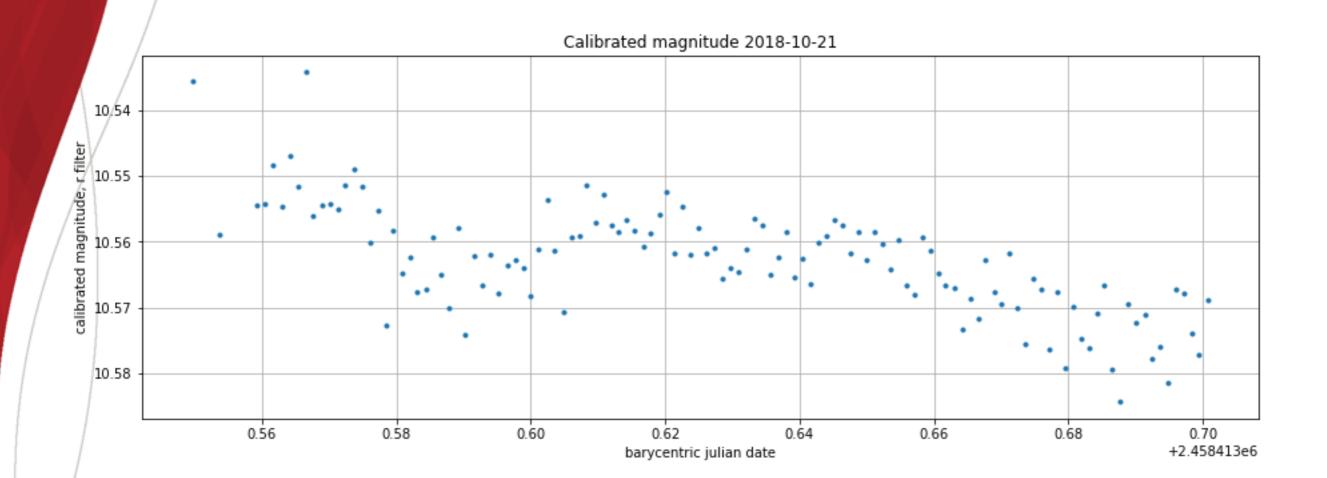






#### Sometimes





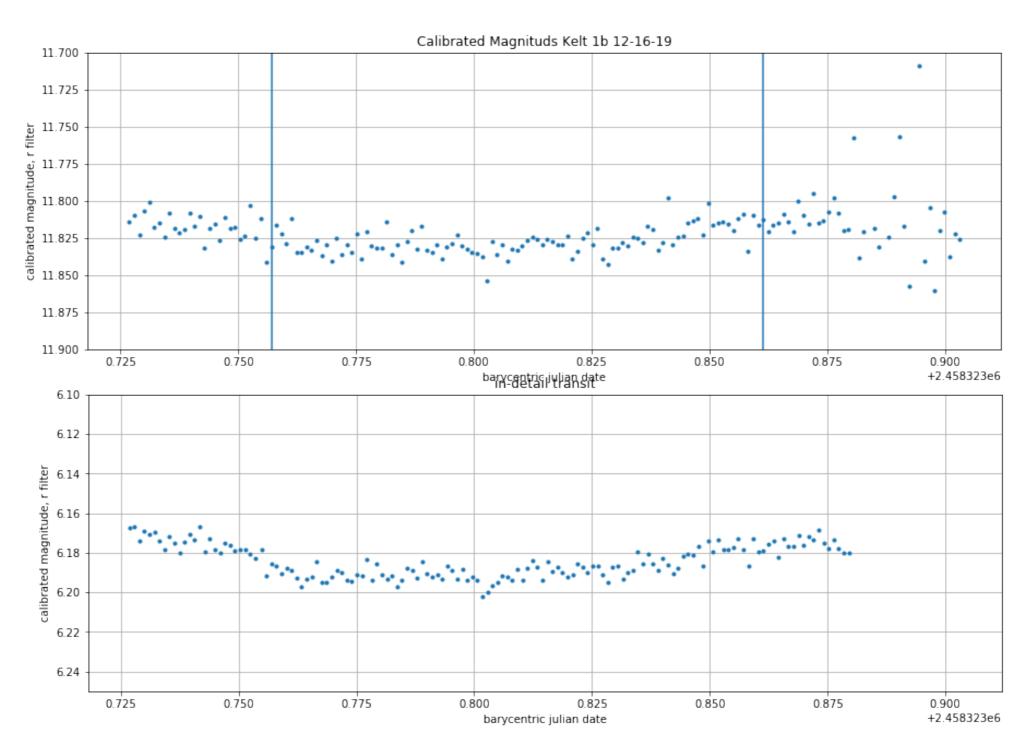
#### Differential photometry



- Need to do differential photometry in addition
  - •TESS/AIJ style
    - flux (count) ratio
    - target flux / (sum of comp fluxes)
  - AAVSO style:
    - target magnitude from
      - Difference between instrumental target and comp
      - Catalog magnitude of comp
    - average over all comp stars

### All-sky vs differential





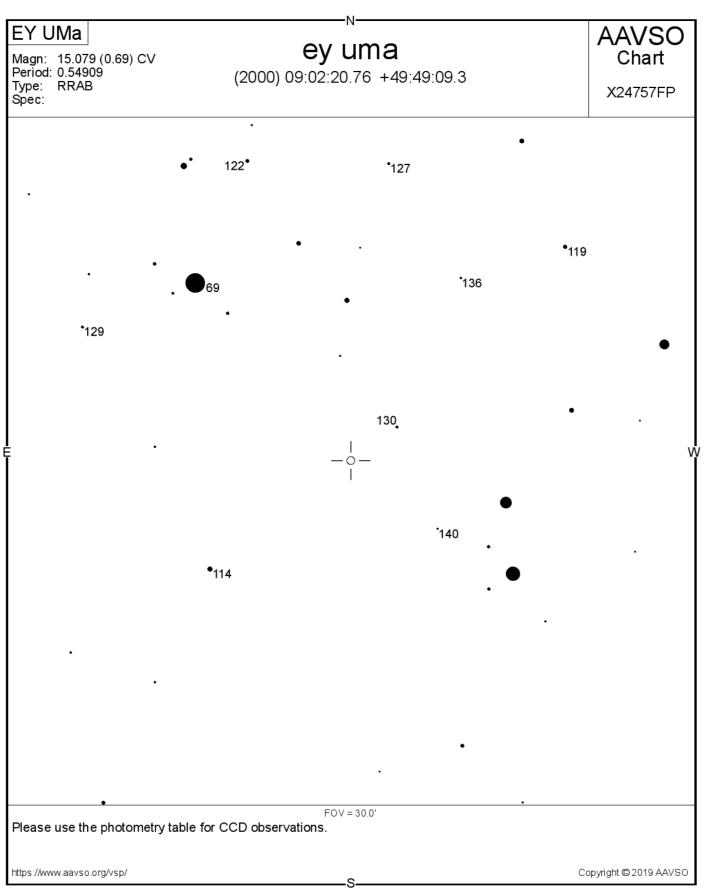




- Relatively isolated code
- Clearly defined result (single plot)

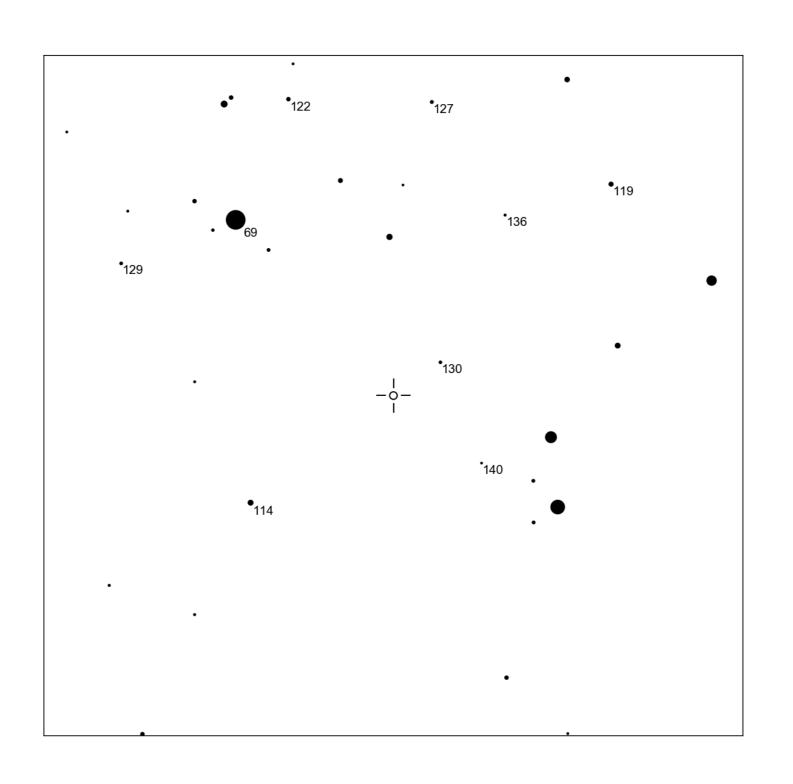
#### VSP: Chart for EY UMa











#### **VSP: Status**



- •Start: 3,500 lines of well-written Perl
- •Now: 190 lines of code
  - Really poorly written Python...
  - ...that uses current VSP API to get comp stars
- •Eventually: 500-600 lines of code

# Challenges



- Local computer
  - Installation is...painful
  - Relatively easy to break working install
  - Launch from a terminal
  - two or three platforms to support
- •Server:
  - Setup is...painful
  - Authentication is...more painful
  - Storage and CPU cost money

# Advantages



- Local computer
  - You control the compute
  - You have already paid for storage and CPU
- Server
  - Software for users easier to manage
  - Can provide large data files without downloads





Slides/links/how to try it out at:

http://bit.ly/aavso-2019

