UNIVERSITY of HOUSTON
COLLEGE OF EDUCATION
Curriculum & Instruction

Teaching Astronomy Remotely with Coding Activities

James Newland Bellaire High School, Bellaire, TX, University of Houston, Houston, TX

Bellaire High School
Physics and Astronomy

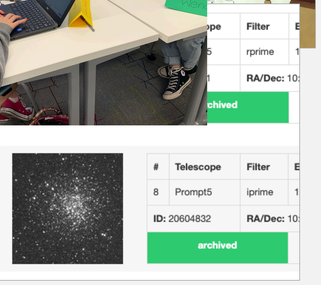
Computational Thinking

Data Practices	Collecting Creating Manipulating	Analyzing Visualizing
Modeling & Simulation Practices	Grasp Concepts Find/Test Solutions	Assessing Designing Constructing
Computational Problem Solving Practices	Prep Problems Programming Assess Solutions	Modular Solns Abstractions Debugging
Systems Thinking Practices	System as Whole Relationships in Systems	Thinking in Levels Defining System Complexity

Weintrop, et al., 2016[1]

Learning Science by Coding

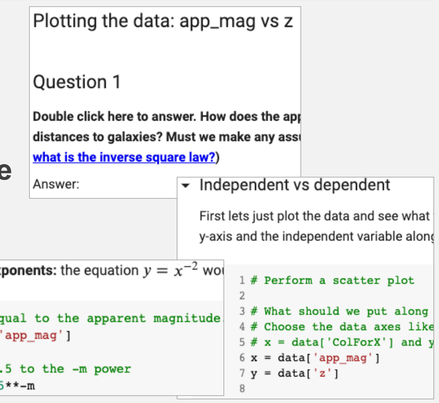
- Use of authentic datasets
- Basic data reduction techniques
- Well-known visualization tools
- Data science processes
- Guided inquiry model: interactive learning by coding



astropy seaborn scikit-learn

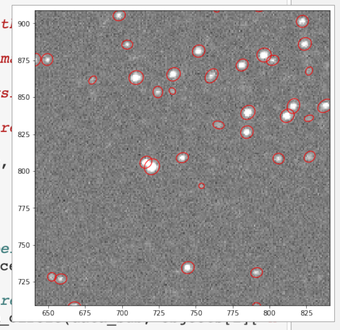
Applying CS Pedagogy

- Cognitive Load Theory[2]
- Worked examples[3]
- Human readable, well-commented code
- Mixed-media input with Markdown



Cluster Distance with Photometry

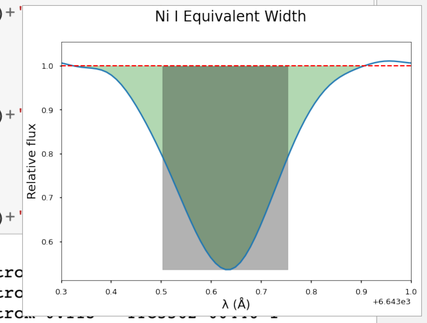
```
    This function consumes our list of extracted targets along with
    the data after subtraction, the background, and the center x and
    center y values.

    We then go through the list and find t
ar.
    Note: this will work ONLY for these im
star
    would require a tedious by-hand analys
    The flux for the calibration star is r
"""
def get_calib_flux(objects, data_sub, bkg,
    for i in range(len(objects)):
        curr_x = objects[i]['x']
        curr_y = objects[i]['y']

        # Get flux of calibration star (be
        if curr_x > cen_x-5 and curr_x < c
5 and curr_y < cen_y-81+15:
            # find the flux of object near
            flux, fluxerr, flag = sep.sum_
'], objects[i]['y'],
                                    3.0, err=bkg.globalrms, gain=1.0)
            calibration_flux = round(int(flux))
    return calibration_flux
```



Equivalent Width with Spectroscopy

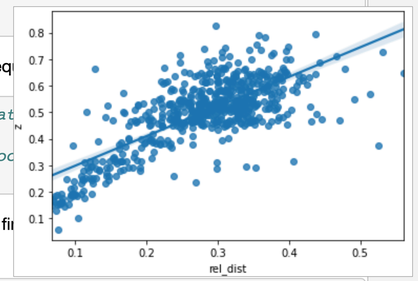
```
ni3 = np.round(ni3,4)
eu3 = np.round(eu3,4)
r3 = np.round(r3,3)
print(str(ni3)+'\t'+str(eu3)+'

ni4 = np.round(ni4,4)
eu4 = abs(np.round(eu4,4))
r4 = abs(np.round(r4,3))
print(str(ni4)+'\t'+str(eu4)+'

ni5 = np.round(ni5,4)
eu5 = np.round(eu5,4)
r5 = np.round(r5,3)
print(str(ni5)+'\t'+str(eu5)+'
```

```
Ni I EW          Eu II EW
0.1102 Angstrom 0.0066 Angstro
0.0716 Angstrom 0.0034 Angstro
0.1429 Angstrom 0.0161 Angstro
0.0585 Angstrom 0.0009 Angstrom 0.016   TYC5701-00197-1
0.142 Angstrom  0.0122 Angstrom 0.086   V*SX_Her
```



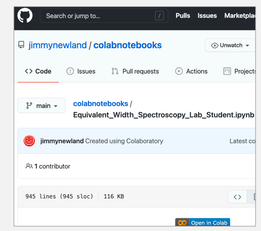
Hubble's Law with SDSS Galaxies

```
# Set m equal to the apparent magnitude.
m = data['app_mag']

# Raise 2.5 to the -m power
flux = 2.5**-m

Python HOWTO - square roots: the eq

# Use the inverse square relat

# np.sqrt finds the square roo
rel_dist = 1/np.sqrt(flux)

Python HOWTO - find the max: to fi
largest = my_list.max()

# Let's 'normalize' our relative distances.
# Based on our farthest galaxy, what's the relative dist. to th
e others?
farthest = rel_dist.max()
```



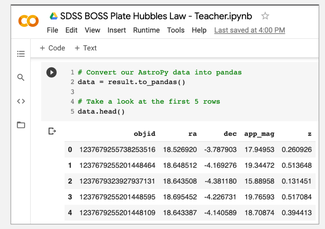
Code virtually: Google Colab and GitHub

- Projects scaffolded with new coders in mind
- Cloud computing allows for collaboration
- Only a browser and Internet connection required
- No software installation or configuration needed
- GitHub facilitates version control & open source



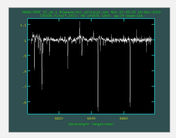
colab

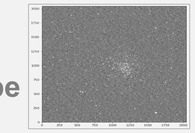
GitHub

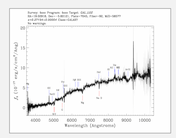


Authentic Datasets

- **Equivalent Width** – Data collected via Otto Struve Telescope and Sandiford Echelle Spectrograph at McDonald Observatory
- **Cluster Distance** – Photometric time series of RR Lyrae stars in NGC 3201 collected via Skynet Robotic Telescope Network
- **Hubble's Law** – SDSS Baryon Oscillation Spectroscopic Survey combined galaxy spectra and photometry legacy data







Find Out More

jimmynewland/colabnotebooks
DOI 10.5281/zenodo.4318058

https://git.io/JI6vE

Data, code, & more: https://wp.me/P3rYuP-6SAt6G

Acknowledgments

Thank you to Dr. Chris Sneden and Dr. Keely Finkelstein for their guidance and support. Thanks also to Eileen Grzybowski, Justin Hickey, and Olivia Kuper for help with data reduction and observing. Additionally, thanks to Dr. Sean Johnson and Dr. Britt Lundgren for help reducing the SDSS data.

References

[1]Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5
[2]Morrison, B. B., Dorn, B., & Guzdial, M. (2014). Measuring cognitive load in introductory CS. *Proceedings of the Tenth Annual Conference on International Computing Education Research - ICER '14*, 131–138. https://doi.org/10.1145/2632320.2632348
[3]Skudder, B., & Luxton-Reilly, A. (2014). Worked examples in computer science. *Conferences in Research and Practice in Information Technology Series*.