PyVO

Overview and community contribution refresher

Brigitta Sipőcz

Caltech/IPAC-IRSA

Active PyVO maintainers:

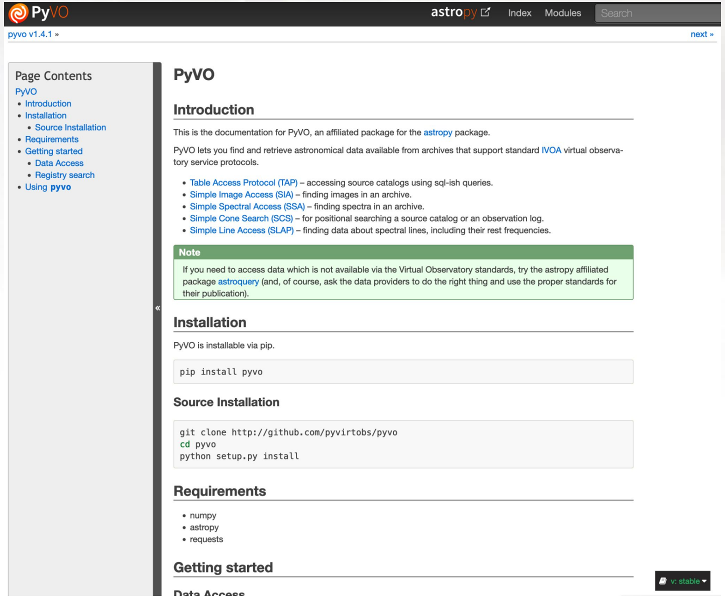
Adrian Damian (CADC), Markus Demleitner (Heidelberg University), Tom Donaldson (STScI)

PyVO overview

Standalone Python library to support standard IVOA virtual observatory service protocols.

Docs:
https://pyvo.readthedocs.io/en/stable/

Code:
https://github.com/astropy/pyvo/



Page Contents

PyVO
- Introduction
- Installation
  - Source Installation
- Requirements
- Getting started
  - Data Access
  - Registry search
- Using **pyvo**

PyVO

Introduction

This is the documentation for PyVO, an affiliated package for the astropy package.

PyVO lets you find and retrieve astronomical data available from archives that support standard IVOA virtual observatory service protocols.

- Table Access Protocol (TAP) – accessing source catalogs using sql-ish queries.
- Simple Image Access (SIA) – finding images in an archive.
- Simple Spectral Access (SSA) – finding spectra in an archive.
- Simple Cone Search (SCS) – for positional searching a source catalog or an observation log.
- Simple Line Access (SLAP) – finding data about spectral lines, including their rest frequencies.

Note

If you need to access data which is not available via the Virtual Observatory standards, try the astropy affiliated package astroquery (and, of course, ask the data providers to do the right thing and use the proper standards for their publication).

Installation

PyVO is installable via pip.

```
pip install pyvo
```

Source Installation

```
git clone http://github.com/pyvirtobs/pyvo
cd pyvo
python setup.py install
```

Requirements

- numpy
- astropy
- requests

Getting started

Data Access

PyVO recent history

- Community restart at IVOA Interop 2019 Paris
  - Maintainer group to expand over multiple institutions
  - Tap into the infrastructure ecosystem of community python
- More flexibility on standards
  - prototype feature (see Omar Laurino's 2022 Apr Interop talk)

PyVO present

v1.5 (to be released soon)

- Registry search now finds SIA v2 services
- .to_qtable() method to return astropy QTable
- Make regtap service aware

- Cleanups and deprecations

- Various bugfixes in v1.4.1 since last interop

PyVO near future

V1.6 (~by next interop?)

- Empowered by prototype feature:
  - cloud access utilities (see Tess Jaffe’s Science platform talk)

- Possibilities that needs champions
  - Consolidating VO relevant pieces into PyVO

    (astropy.samp, and astroquery.vo_conesearch)
  - Your feature

Collaborative software communities

- Community software
  - There is no central institute driving development
  - Open to and can handle community contributions
- A ***home platform***, accessible to anyone
  - GitHub, and services built around it
- Further reading on open-source communities
  - Nadia Eghbal/Asparouhova: Working in public

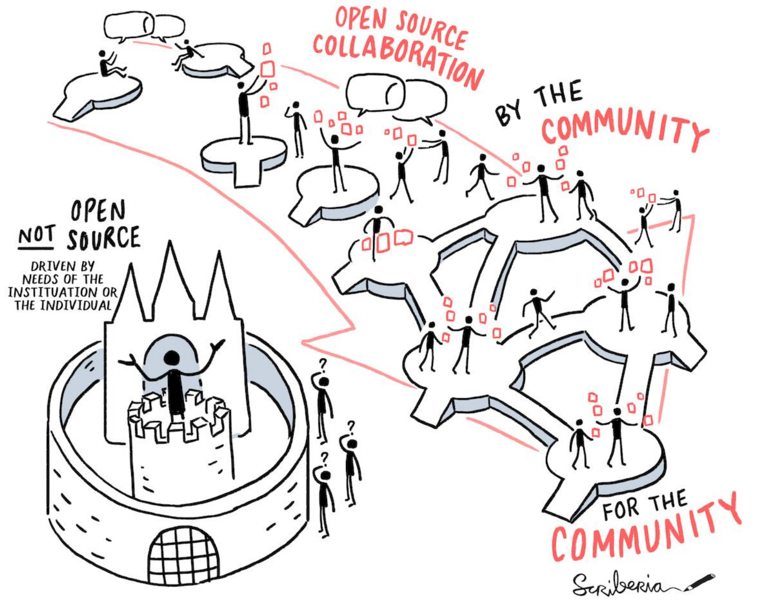

OPEN SOURCE COLLABORATION
BY THE COMMUNITY
FOR THE COMMUNITY
OPEN NOT SOURCE
DRIVEN BY NEEDS OF THE INSTITUTION OR THE INDIVIDUAL
Scriberia

Image by The Turing Way community and Scriberia used under a CC-BY licence.

Code on internet
≠
Open source
≠
Community development

Community

- **PyVO** is widely used in the astronomy community and beyond.
- Documentation, and rendered figures www.astroML.org
- Code and issue tracker on GitHub: https://github.com/astroML/astroML
- Feedback is welcome on missing features not available elsewhere in the ecosystem

Definitions for Reproducibility

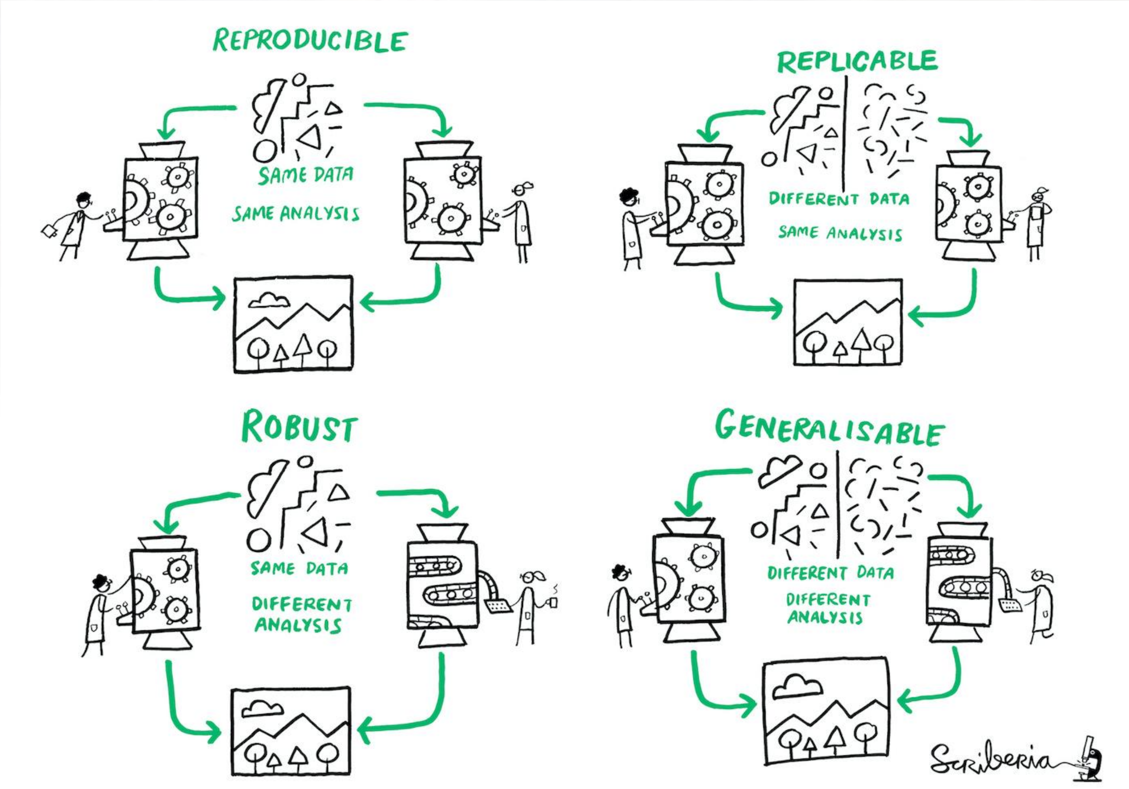


Image by The Turing Way community and Scriberia used under a CC-BY licence.

Jupyter and Jupyter Book ecosystem

- Jupyter is more than notebooks

- Modular, extensible, and powerful ecosystem for sharing and publishing

https://jupyterbook.org

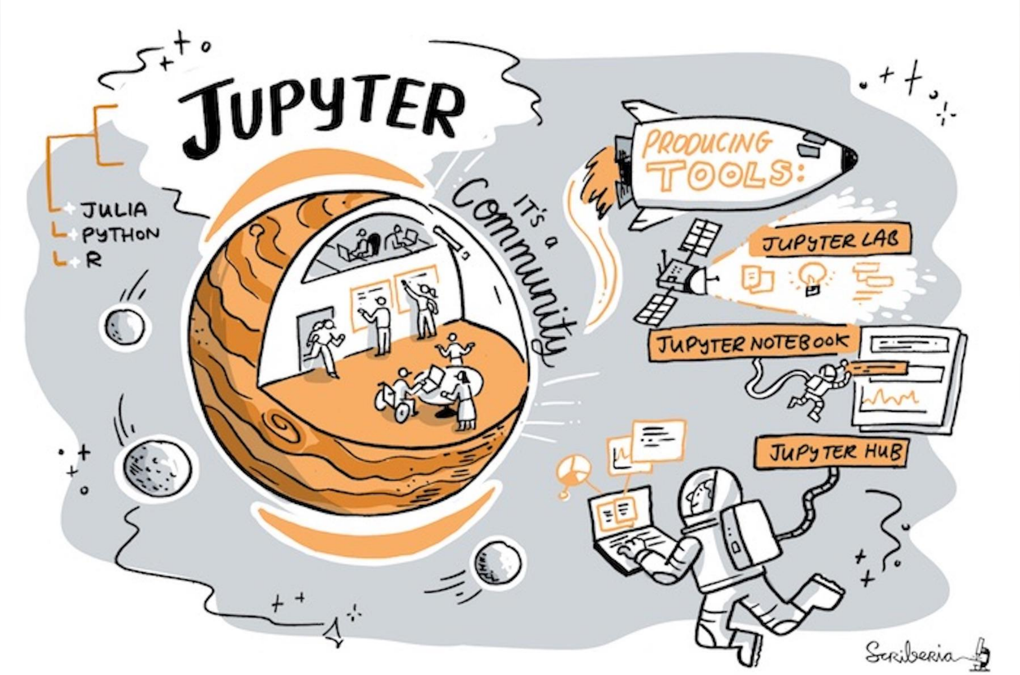
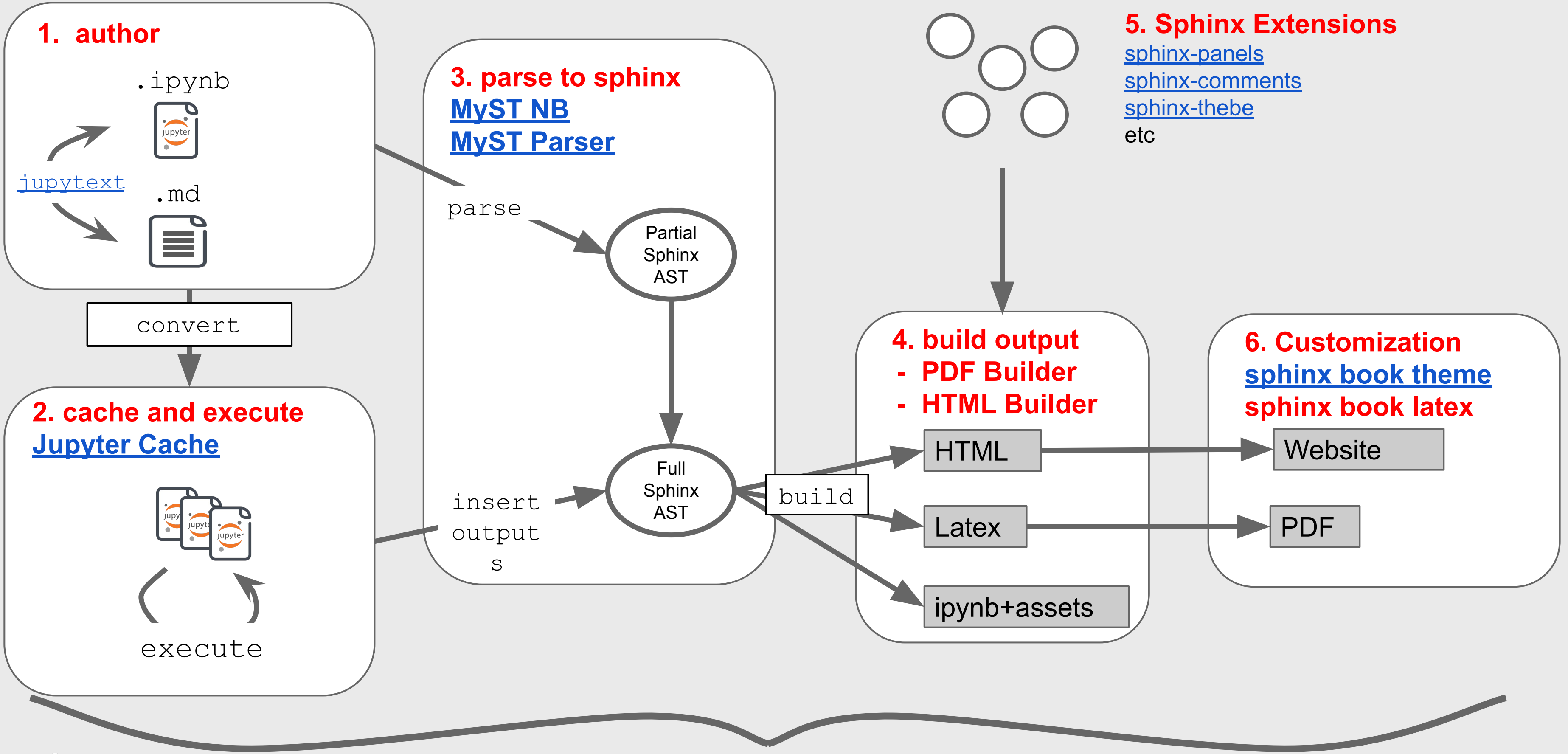


Image by *The Turing Way* community and Scriberia used under a CC-BY licence.

An overview of the new Jupyter Book build process



1. author

.ipynb

jupytext

.md

convert

2. cache and execute

Jupyter Cache

execute

3. parse to sphinx

MyST NB

MyST Parser

parse

Partial Sphinx AST

Full Sphinx AST

insert outputs

build

4. build output

- **PDF Builder**
- **HTML Builder**

HTML

Latex

ipynb+assets

5. Sphinx Extensions

sphinx-panels

sphinx-comments

sphinx-thebe

etc

6. Customization

sphinx book theme

sphinx book latex

Website

PDF

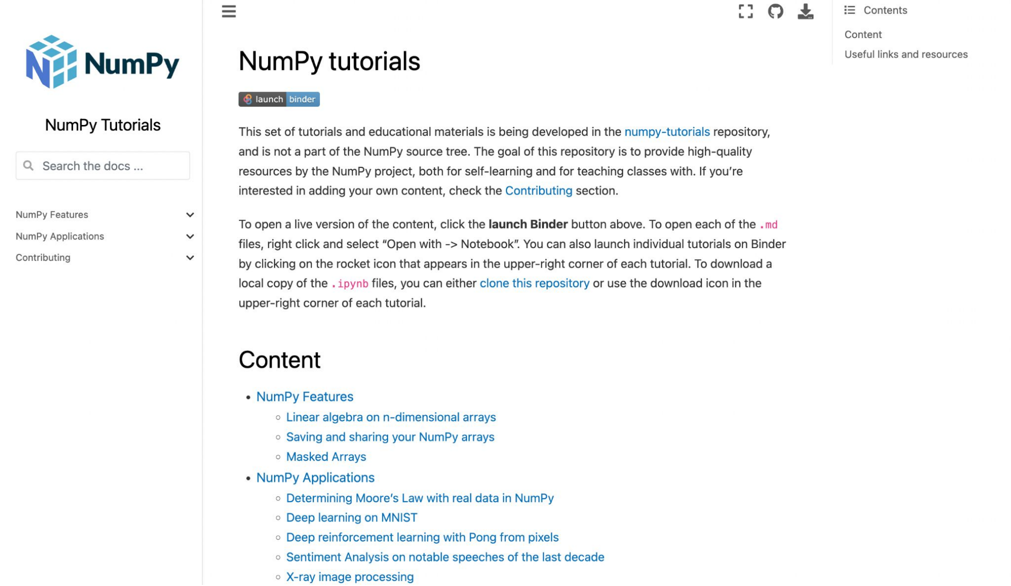
7. controller process

Jupyter Book

Slide by Chris Holdgraf

Jupyter Book users

- The Turing Way
  Handbook to reproducible, ethical and collaborative data science.
- Core scientific Python
  - NumPy
  - Scikit-learn
  - NetworkX
  - ...
- Astronomy resources
  - astroML
  - NAVO workshop
  - “Hello Universe” by MAST



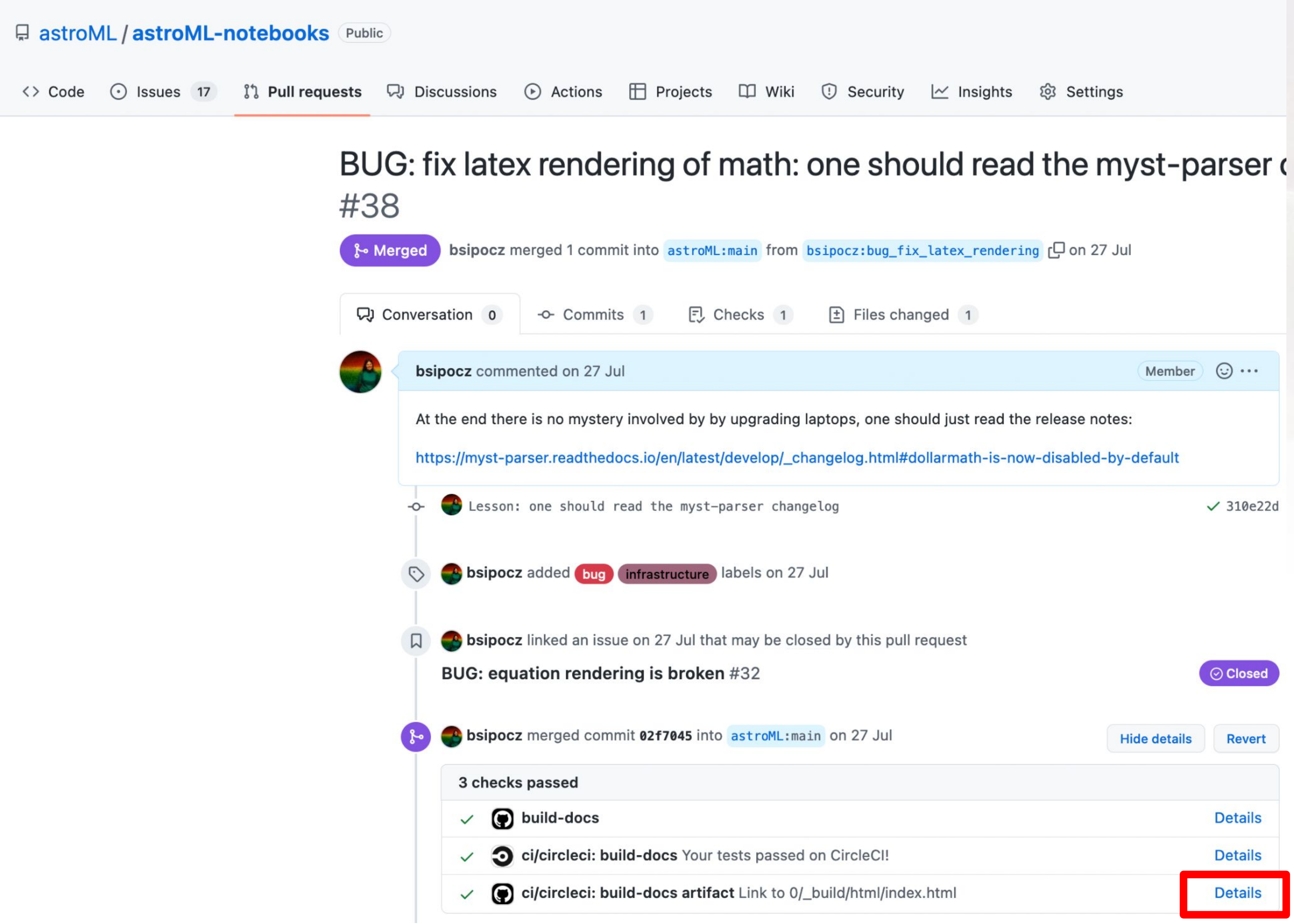
NumPy tutorials

This set of tutorials and educational materials is being developed in the numpy-tutorials repository, and is not a part of the NumPy source tree. The goal of this repository is to provide high-quality resources by the NumPy project, both for self-learning and for teaching classes with. If you're interested in adding your own content, check the Contributing section.

To open a live version of the content, click the **launch Binder** button above. To open each of the `.md` files, right click and select “Open with -> Notebook”. You can also launch individual tutorials on Binder by clicking on the rocket icon that appears in the upper-right corner of each tutorial. To download a local copy of the `.ipynb` files, you can either clone this repository or use the download icon in the upper-right corner of each tutorial.

Content

- NumPy Features
  - Linear algebra on n-dimensional arrays
  - Saving and sharing your NumPy arrays
  - Masked Arrays
- NumPy Applications
  - Determining Moore's Law with real data in NumPy
  - Deep learning on MNIST
  - Deep reinforcement learning with Pong from pixels
  - Sentiment Analysis on notable speeches of the last decade
  - X-ray image processing



BUG: fix latex rendering of math: one should read the myst-parser
#38

Merged bsipocz merged 1 commit into `astroML:main` from `bsipocz:bug_fix_latex_rendering` on 27 Jul

bsipocz commented on 27 Jul

At the end there is no mystery involved by by upgrading laptops, one should just read the release notes:

https://myst-parser.readthedocs.io/en/latest/develop/_changelog.html#dollarmath-is-now-disabled-by-default

`Lesson: one should read the myst-parser changelog` ✓ 310e22d

bsipocz added bug infrastructure labels on 27 Jul

bsipocz linked an issue on 27 Jul that may be closed by this pull request

BUG: equation rendering is broken #32

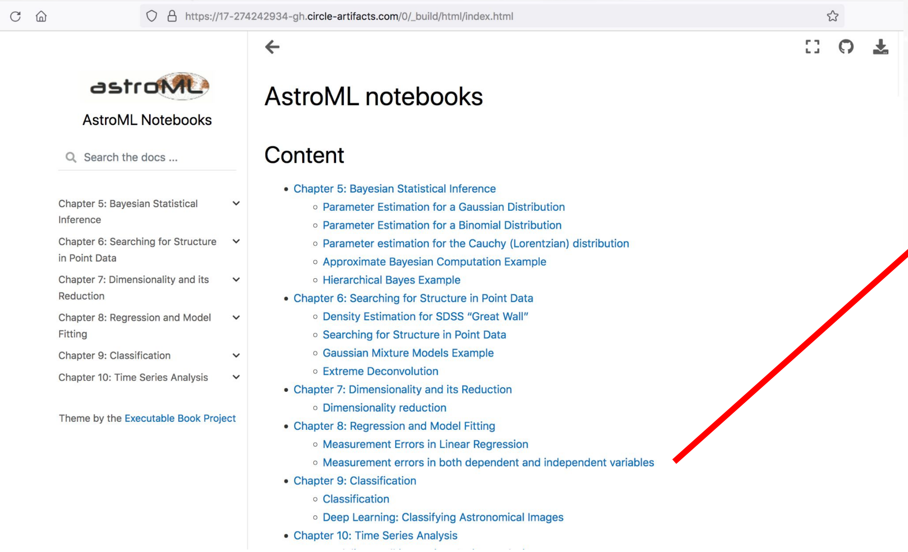
bsipocz merged commit **02f7045** into `astroML:main` on 27 Jul

3 checks passed

✓ **build-docs**

✓ **ci/circleci: build-docs** Your tests passed on CircleCI!

✓ **ci/circleci: build-docs artifact** Link to 0/_build/html/index.html



https://17-274242934-gh.circle-artifacts.com/0/_build/html/index.html

AstroML Notebooks

Search the docs ...

Chapter 5: Bayesian Statistical Inference

Chapter 6: Searching for Structure in Point Data

Chapter 7: Dimensionality and its Reduction

Chapter 8: Regression and Model Fitting

Chapter 9: Classification

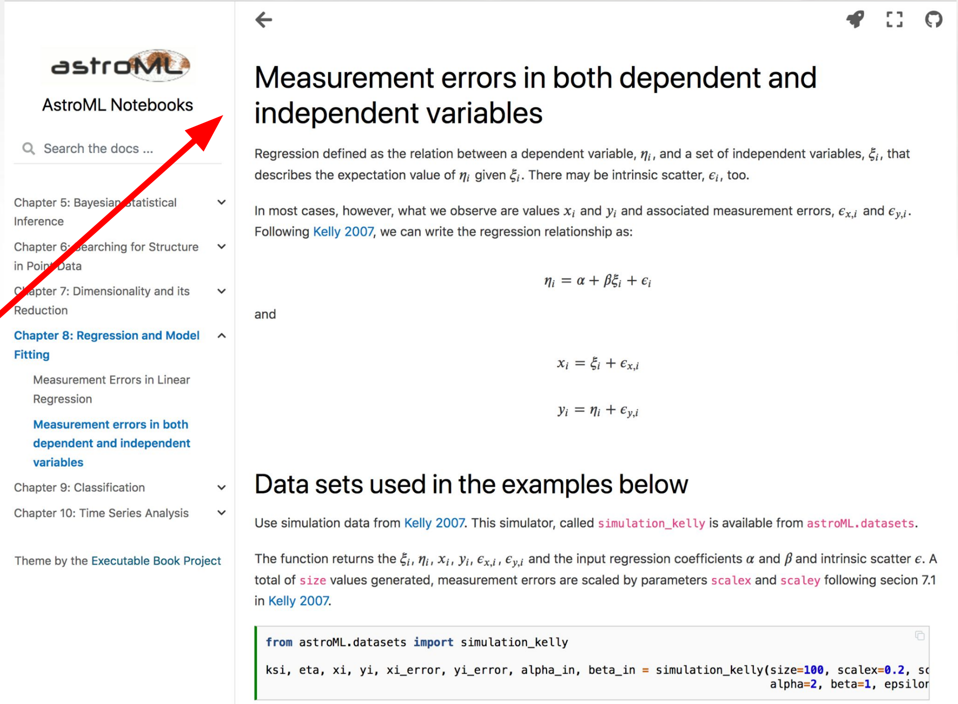
Chapter 10: Time Series Analysis

Theme by the Executable Book Project

AstroML notebooks

Content

- Chapter 5: Bayesian Statistical Inference
    - Parameter Estimation for a Gaussian Distribution
    - Parameter Estimation for a Binomial Distribution
    - Parameter estimation for the Cauchy (Lorentzian) distribution
    - Approximate Bayesian Computation Example
    - Hierarchical Bayes Example
- Chapter 6: Searching for Structure in Point Data
    - Density Estimation for SDSS “Great Wall”
    - Searching for Structure in Point Data
    - Gaussian Mixture Models Example
    - Extreme Deconvolution
- Chapter 7: Dimensionality and its Reduction
    - Dimensionality reduction
- Chapter 8: Regression and Model Fitting
    - Measurement Errors in Linear Regression
    - Measurement errors in both dependent and independent variables
- Chapter 9: Classification
    - Classification
    - Deep Learning: Classifying Astronomical Images
- Chapter 10: Time Series Analysis



AstroML Notebooks

Search the docs ...

Chapter 5: Bayesian Statistical Inference

Chapter 6: Searching for Structure in Point Data

Chapter 7: Dimensionality and its Reduction

Chapter 8: Regression and Model Fitting

Measurement Errors in Linear Regression

Measurement errors in both dependent and independent variables

Chapter 9: Classification

Chapter 10: Time Series Analysis

Theme by the Executable Book Project

Measurement errors in both dependent and independent variables

Regression defined as the relation between a dependent variable, η_i, and a set of independent variables, ξ_i, that describes the expectation value of η_i given ξ_i. There may be intrinsic scatter, ϵ_i, too.

In most cases, however, what we observe are values x_i and y_i and associated measurement errors, $\epsilon_{x,i}$ and $\epsilon_{y,i}$. Following Kelly 2007, we can write the regression relationship as:

$$\eta_i = \alpha + \beta\xi_i + \epsilon_i$$

and

$$x_i = \xi_i + \epsilon_{x,i}$$

$$y_i = \eta_i + \epsilon_{y,i}$$

Data sets used in the examples below

Use simulation data from Kelly 2007. This simulator, called `simulation_kelly` is available from `astroML.datasets`.

The function returns the ξ_i, η_i, x_i, y_i, $\epsilon_{x,i}$, $\epsilon_{y,i}$ and the input regression coefficients α and β and intrinsic scatter ϵ. A total of `size` values generated, measurement errors are scaled by parameters `scalex` and `scaley` following secion 7.1 in Kelly 2007.

```
from astroML.datasets import simulation_kelly

ksi, eta, xi, yi, xi_error, yi_error, alpha_in, beta_in = simulation_kelly(size=100, scalex=0.2, sc
                                                                           alpha=2, beta=1, epsilor
```

Next steps: community contributions

- Collect notebooks showcasing astro machine learning examples beyond the book material
- Promote to apply this stack with automations instead of one-off renderings for other notebook-based project

OPEN SOURCE COLLABORATION
BY THE COMMUNITY
FOR THE COMMUNITY
NOT OPEN SOURCE
DRIVEN BY NEEDS OF THE INSTITUATION OR THE INDIVIDUAL
Scriberia

Image by The Turing Way community and Scriberia used under a CC-BY licence.

Continuous integration

- Running tests **automatically** for all proposed code changes.

- Goal: Catch problems as soon as possible.

- Test as widely as possible for the supported use cases
  - Operating systems
  - Supported version numbers
  - Development version of dependencies
  - Ecosystem wide integration tests

CI services

GitHub integrated services, work out of box, most common ones:

- Github Actions
  - Large marketplace for many types of actions
  - Run in the cloud, free of charge for open source repos
  - Possible to self-host for private repos
- CircleCI
  - Excellent artifact support

Easy to understand and use configuration

Image by *The Turing Way* community and Scriberia used under a CC-BY licence.

Plenty of astronomy source code exists but

OPEN SOURCE COLLABORATION
BY THE COMMUNITY
FOR THE COMMUNITY
OPEN ~~NOT~~ SOURCE
DRIVEN BY NEEDS OF THE INSTITUATION OR THE INDIVIDUAL
Scriberia

Image by The Turing Way community and Scriberia used under a CC-BY licence.

Code on internet

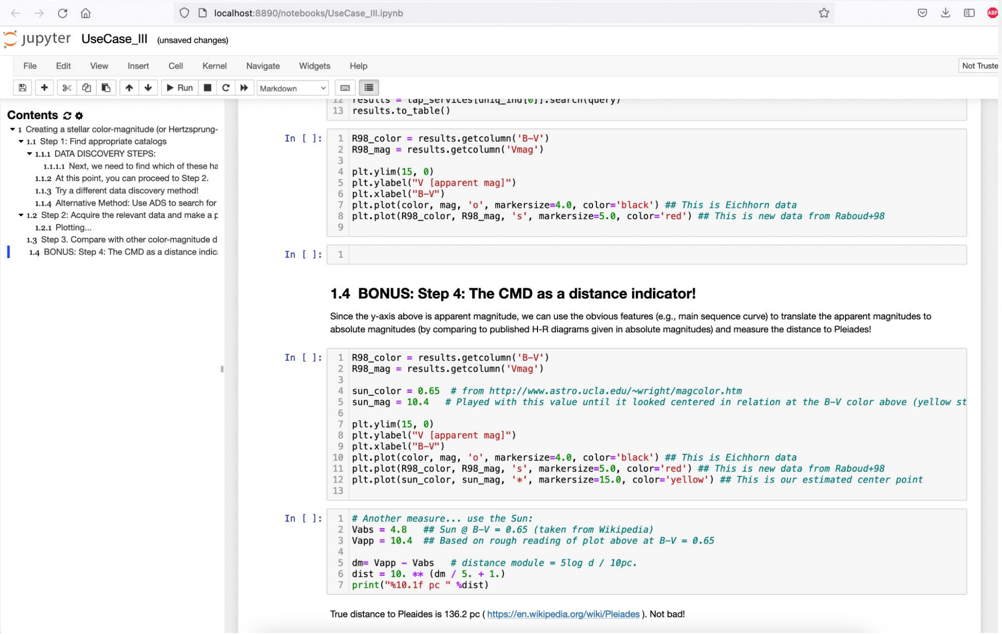
≠

Open source

≠

Open development

NAVO workshop tutorials



localhost:8890/notebooks/UseCase_III.ipynb
jupyter UseCase_III (unsaved changes)
File Edit View Insert Cell Kernel Navigate Widgets Help
Not Truste
Run
Markdown

Contents

- 1 Creating a stellar color-magnitude (or Hertzsprung-
  - 1.1 Step 1: Find appropriate catalogs
    - 1.1.1 DATA DISCOVERY STEPS:
      - 1.1.1.1 Next, we need to find which of these ha
    - 1.1.2 At this point, you can proceed to Step 2.
    - 1.1.3 Try a different data discovery method!
    - 1.1.4 Alternative Method: Use ADS to search for
  - 1.2 Step 2: Acquire the relevant data and make a p
    - 1.2.1 Plotting...
  - 1.3 Step 3. Compare with other color-magnitude d
  - 1.4 BONUS: Step 4: The CMD as a distance indic

```
results.to_table()
```

```
In [ ]:
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data from Raboud+98

```

```
In [ ]:
```

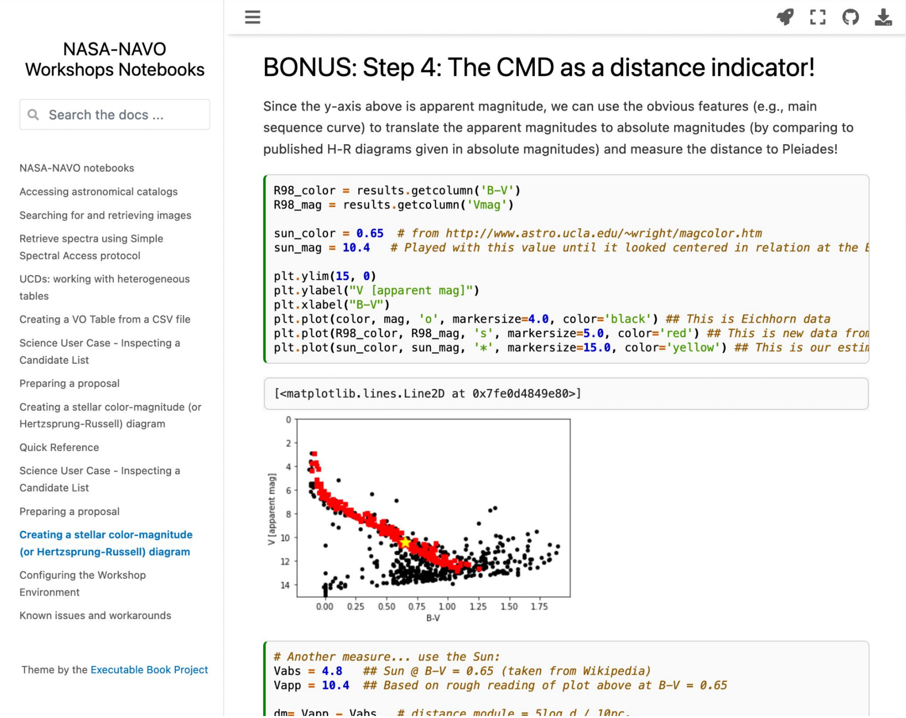
1.4 BONUS: Step 4: The CMD as a distance indicator!

Since the y-axis above is apparent magnitude, we can use the obvious features (e.g., main sequence curve) to translate the apparent magnitudes to absolute magnitudes (by comparing to published H-R diagrams given in absolute magnitudes) and measure the distance to Pleiades!

```
In [ ]:
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

sun_color = 0.65  # from http://www.astro.ucla.edu/~wright/magcolor.htm
sun_mag = 10.4   # Played with this value until it looked centered in relation at the B-V color above (yellow st

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data from Raboud+98
plt.plot(sun_color, sun_mag, '*', markersize=15.0, color='yellow') ## This is our estimated center point

```

```
In [ ]:
# Another measure... use the Sun:
Vabs = 4.8   ## Sun @ B-V = 0.65 (taken from Wikipedia)
Vapp = 10.4  ## Based on rough reading of plot above at B-V = 0.65

dm= Vapp - Vabs   # distance module = 5log d / 10pc.
dist = 10. ** (dm / 5. + 1.)
print("%10.1f pc " %dist)
```

True distance to Pleaides is 136.2 pc (https://en.wikipedia.org/wiki/Pleiades). Not bad!



NASA-NAVO Workshops Notebooks

Search the docs ...

- NASA-NAVO notebooks
- Accessing astronomical catalogs
- Searching for and retrieving images
- Retrieve spectra using Simple Spectral Access protocol
- UCDs: working with heterogeneous tables
- Creating a VO Table from a CSV file
- Science User Case - Inspecting a Candidate List
- Preparing a proposal
- Creating a stellar color-magnitude (or Hertzsprung-Russell) diagram
- Quick Reference
- Science User Case - Inspecting a Candidate List
- Preparing a proposal
- **Creating a stellar color-magnitude (or Hertzsprung-Russell) diagram**
- Configuring the Workshop Environment
- Known issues and workarounds

Theme by the Executable Book Project

BONUS: Step 4: The CMD as a distance indicator!

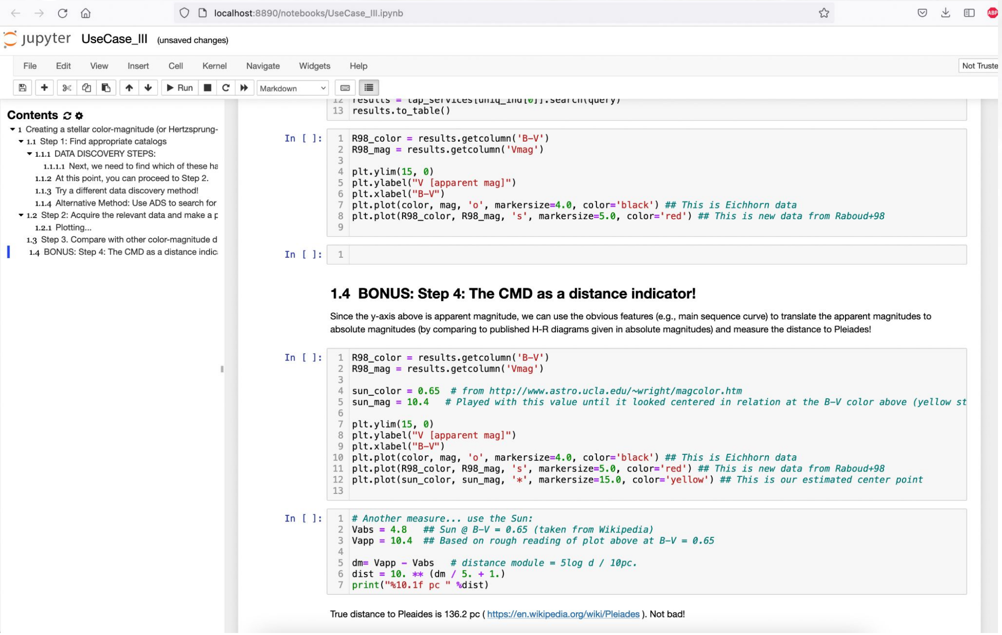
Since the y-axis above is apparent magnitude, we can use the obvious features (e.g., main sequence curve) to translate the apparent magnitudes to absolute magnitudes (by comparing to published H-R diagrams given in absolute magnitudes) and measure the distance to Pleiades!

```
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

sun_color = 0.65  # from http://www.astro.ucla.edu/~wright/magcolor.htm
sun_mag = 10.4   # Played with this value until it looked centered in relation at the E

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data fron
plt.plot(sun_color, sun_mag, '*', markersize=15.0, color='yellow') ## This is our estin
```

```
[<matplotlib.lines.Line2D at 0x7fe0d4849e80>]
```

```
# Another measure... use the Sun:
Vabs = 4.8    ## Sun @ B-V = 0.65 (taken from Wikipedia)
Vapp = 10.4   ## Based on rough reading of plot above at B-V = 0.65

dm= Vapp - Vabs   # distance module = 5log d / 10pc.
```

NAVO workshop tutorials



Contents

- 1 Creating a stellar color-magnitude (or Hertzsprung-
  - 1.1 Step 1: Find appropriate catalogs
    - 1.1.1 DATA DISCOVERY STEPS:
      - 1.1.1.1 Next, we need to find which of these ha
    - 1.1.2 At this point, you can proceed to Step 2.
    - 1.1.3 Try a different data discovery method!
    - 1.1.4 Alternative Method: Use ADS to search for
  - 1.2 Step 2: Acquire the relevant data and make a p
    - 1.2.1 Plotting...
  - 1.3 Step 3. Compare with other color-magnitude d
  - 1.4 BONUS: Step 4: The CMD as a distance indic:

```
results = tap_services[uniq_ind[0]].search(query)
results.to_table()
```

In [ ]:

```
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data from Raboud+98
```

In [ ]:

1.4 BONUS: Step 4: The CMD as a distance indicator!

Since the y-axis above is apparent magnitude, we can use the obvious features (e.g., main sequence curve) to translate the apparent magnitudes to absolute magnitudes (by comparing to published H-R diagrams given in absolute magnitudes) and measure the distance to Pleiades!

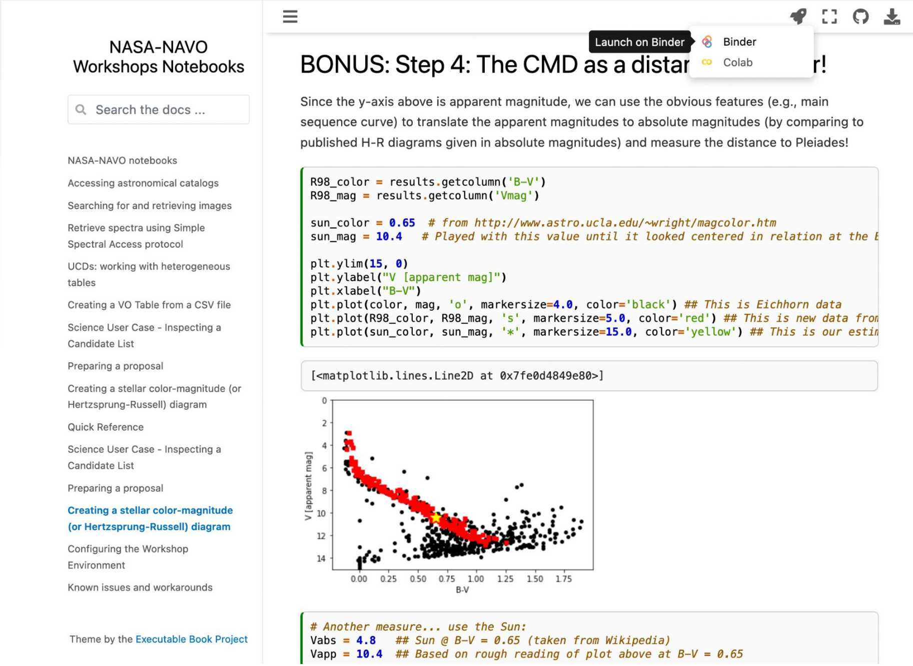
In [ ]:

```
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

sun_color = 0.65  # from http://www.astro.ucla.edu/~wright/magcolor.htm
sun_mag = 10.4   # Played with this value until it looked centered in relation at the B-V color above (yellow st

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data from Raboud+98
plt.plot(sun_color, sun_mag, '*', markersize=15.0, color='yellow') ## This is our estimated center point
```

In [ ]:

```
# Another measure... use the Sun:
Vabs = 4.8   ## Sun @ B-V = 0.65 (taken from Wikipedia)
Vapp = 10.4  ## Based on rough reading of plot above at B-V = 0.65

dm= Vapp - Vabs   # distance module = 5log d / 10pc.
dist = 10. ** (dm / 5. + 1.)
print("%10.1f pc " %dist)
```

True distance to Pleiades is 136.2 pc (https://en.wikipedia.org/wiki/Pleiades). Not bad!



NASA-NAVO Workshops Notebooks

- NASA-NAVO notebooks
- Accessing astronomical catalogs
- Searching for and retrieving images
- Retrieve spectra using Simple Spectral Access protocol
- UCDs: working with heterogeneous tables
- Creating a VO Table from a CSV file
- Science User Case - Inspecting a Candidate List
- Preparing a proposal
- Creating a stellar color-magnitude (or Hertzsprung-Russell) diagram
- Quick Reference
- Science User Case - Inspecting a Candidate List
- Preparing a proposal
- Creating a stellar color-magnitude (or Hertzsprung-Russell) diagram
- Configuring the Workshop Environment
- Known issues and workarounds

Theme by the Executable Book Project

BONUS: Step 4: The CMD as a distar r!

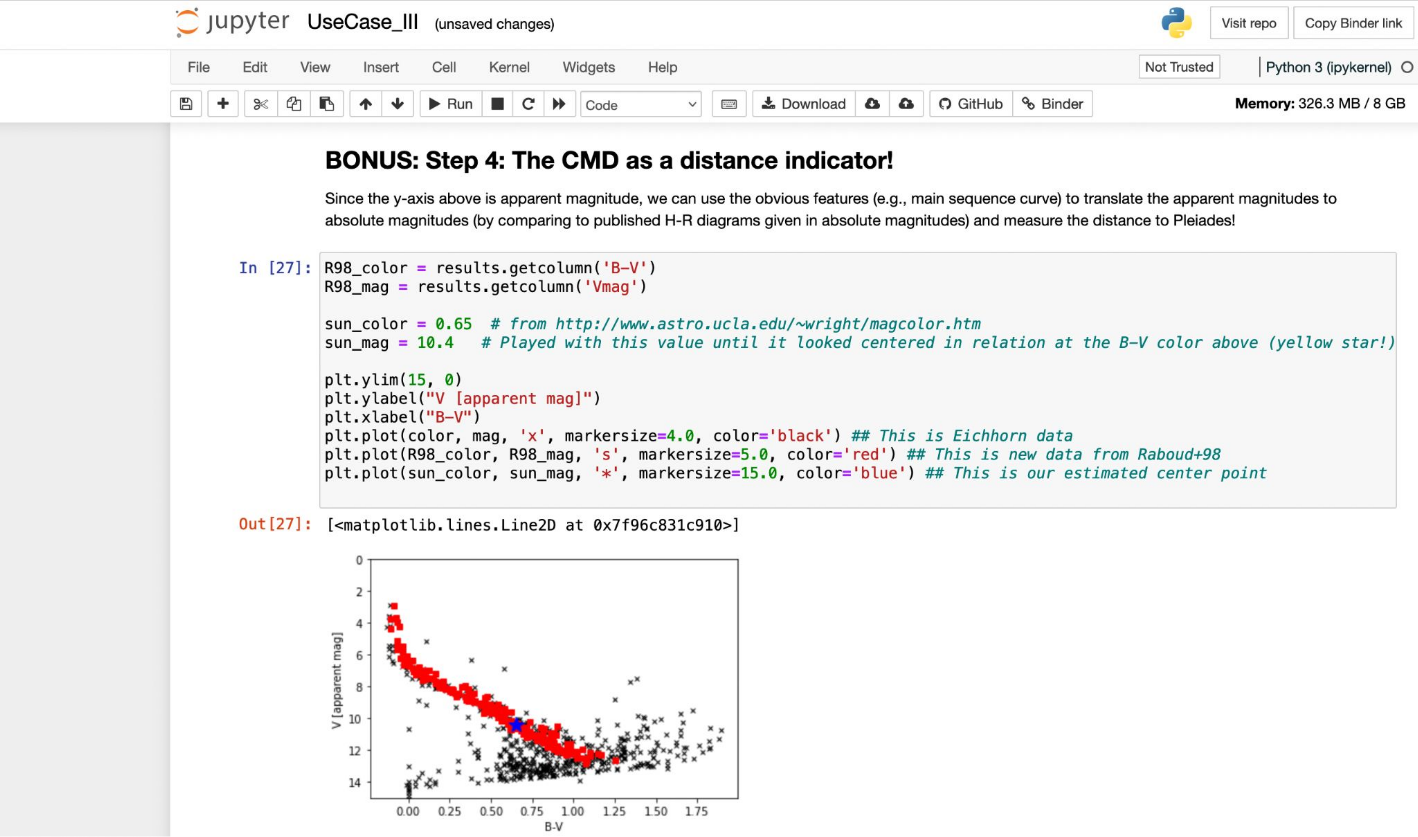
Since the y-axis above is apparent magnitude, we can use the obvious features (e.g., main sequence curve) to translate the apparent magnitudes to absolute magnitudes (by comparing to published H-R diagrams given in absolute magnitudes) and measure the distance to Pleiades!

```
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

sun_color = 0.65  # from http://www.astro.ucla.edu/~wright/magcolor.htm
sun_mag = 10.4   # Played with this value until it looked centered in relation at the E

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'o', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data fror
plt.plot(sun_color, sun_mag, '*', markersize=15.0, color='yellow') ## This is our estir
```

```
[<matplotlib.lines.Line2D at 0x7fe0d4849e80>]
```

```
# Another measure... use the Sun:
Vabs = 4.8   ## Sun @ B-V = 0.65 (taken from Wikipedia)
Vapp = 10.4  ## Based on rough reading of plot above at B-V = 0.65
```

NAVO workshop tutorials



BONUS: Step 4: The CMD as a distance indicator!

Since the y-axis above is apparent magnitude, we can use the obvious features (e.g., main sequence curve) to translate the apparent magnitudes to absolute magnitudes (by comparing to published H-R diagrams given in absolute magnitudes) and measure the distance to Pleiades!

In [27]:

```
R98_color = results.getcolumn('B-V')
R98_mag = results.getcolumn('Vmag')

sun_color = 0.65  # from http://www.astro.ucla.edu/~wright/magcolor.htm
sun_mag = 10.4   # Played with this value until it looked centered in relation at the B-V color above (yellow star!)

plt.ylim(15, 0)
plt.ylabel("V [apparent mag]")
plt.xlabel("B-V")
plt.plot(color, mag, 'x', markersize=4.0, color='black') ## This is Eichhorn data
plt.plot(R98_color, R98_mag, 's', markersize=5.0, color='red') ## This is new data from Raboud+98
plt.plot(sun_color, sun_mag, '*', markersize=15.0, color='blue') ## This is our estimated center point
```

Out[27]:

```
[<matplotlib.lines.Line2D at 0x7f96c831c910>]
```

Open development -
driving factor for community building

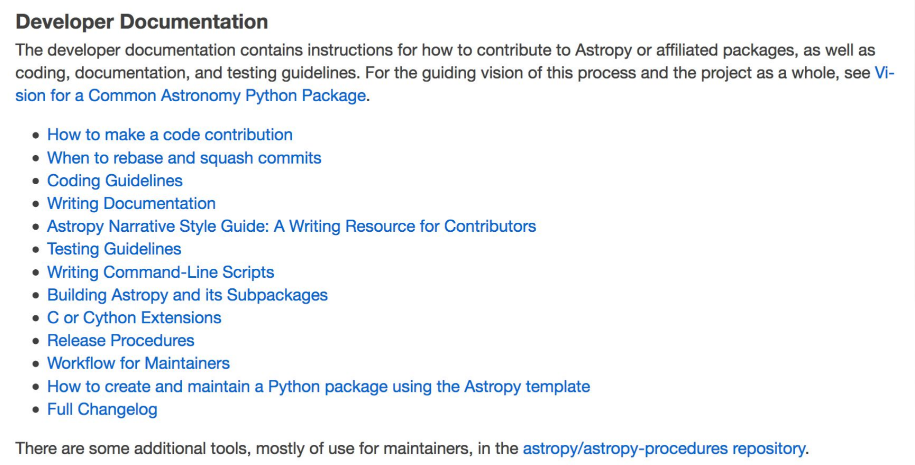
- Decisions are made ***in the open***
- Implementation details are e.g. ***on GitHub***
- Anyone can participate
  Feature implementation is driven by the timeline of individuals and institutions willing to do the work
- Most work done in ***remote collaboration***
  (but important to have a few ***face to face*** meetings)

Open source infrastructure

- **Detailed developer guides**
  - Include developer tutorial, e.g.

    - how to report bugs (MWE)

    - a fully worked out PR example
- **Large number of checks on PRs**
  - Extensive CI testing
    - including testing with development versions of upstream dependencies, e.g. Python, Numpy, Matplotlib
  - Documentation build, and rendered version shared
  - Bots checking for codestyle, milestone, changelog etc



Developer Documentation

The developer documentation contains instructions for how to contribute to Astropy or affiliated packages, as well as coding, documentation, and testing guidelines. For the guiding vision of this process and the project as a whole, see Vision for a Common Astronomy Python Package.

- How to make a code contribution
- When to rebase and squash commits
- Coding Guidelines
- Writing Documentation
- Astropy Narrative Style Guide: A Writing Resource for Contributors
- Testing Guidelines
- Writing Command-Line Scripts
- Building Astropy and its Subpackages
- C or Cython Extensions
- Release Procedures
- Workflow for Maintainers
- How to create and maintain a Python package using the Astropy template
- Full Changelog

There are some additional tools, mostly of use for maintainers, in the astropy/astropy-procedures repository.

What testing is

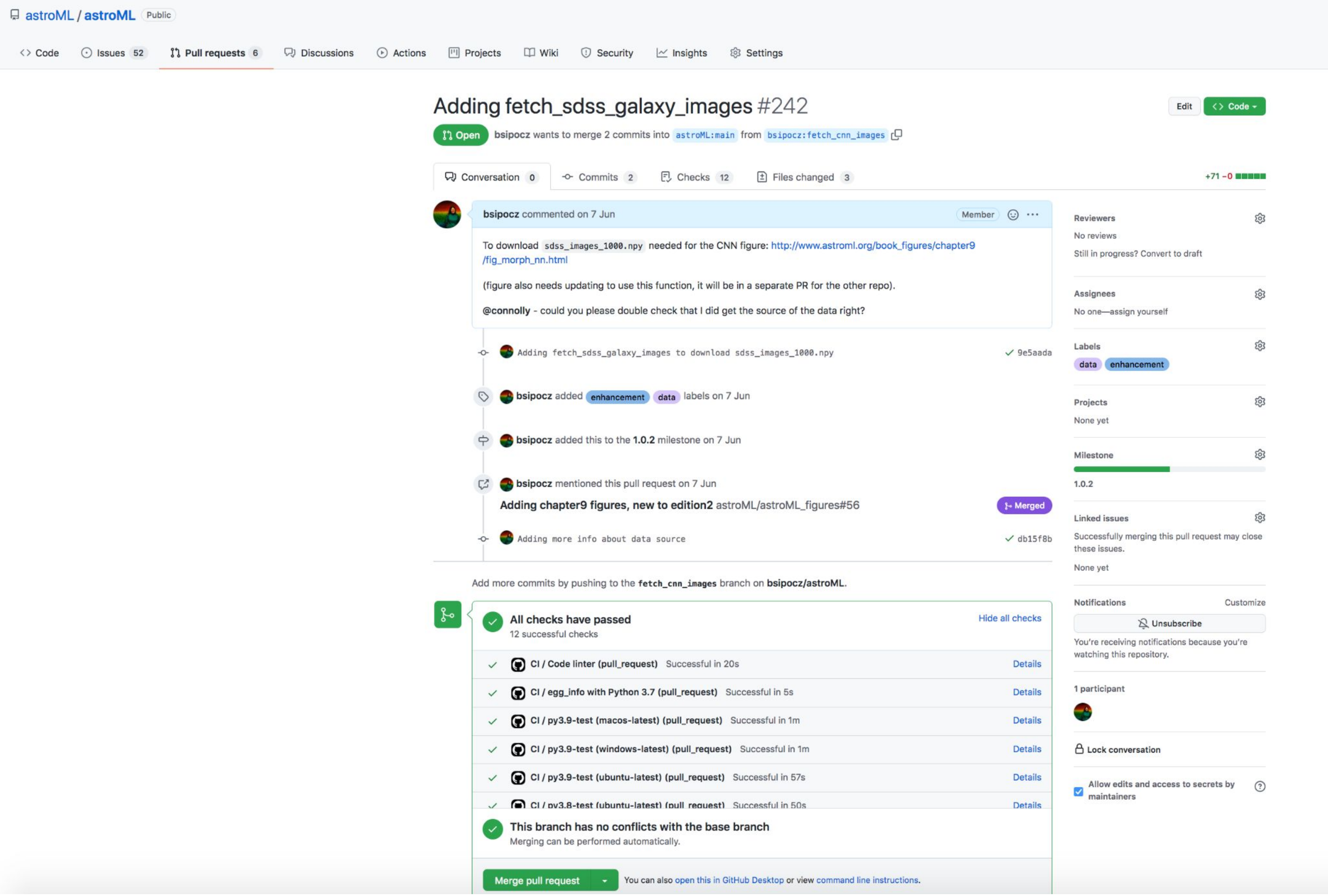
- Code that runs features with known expected result or behavior
- Known expected result can be e.g. a mathematical relation, trivial case for an algorithm or previously buggy but now fixed behavior.

What testing is

- Code that runs features with known expected result or behavior
- Known expected result can be e.g. a mathematical relation, trivial case for an algorithm or previously buggy but now fixed behavior.

Caveats

- Writing tests takes requires significant human resources, but
- Indispensable especially for collaborative projects where developers are joining and leaving, and no single person is able to oversee all the details



Adding fetch_sdss_galaxy_images #242

Open bsipocz wants to merge 2 commits into `astroML:main` from `bsipocz:fetch_cnn_images`

bsipocz commented on 7 Jun — Member

To download `sdss_images_1000.npy` needed for the CNN figure: http://www.astroml.org/book_figures/chapter9/fig_morph_nn.html

(figure also needs updating to use this function, it will be in a separate PR for the other repo).

@connolly - could you please double check that I did get the source of the data right?

`Adding fetch_sdss_galaxy_images to download sdss_images_1000.npy` ✓ 9e5aada

bsipocz added enhancement data labels on 7 Jun

bsipocz added this to the 1.0.2 milestone on 7 Jun

bsipocz mentioned this pull request on 7 Jun

Adding chapter9 figures, new to edition2 astroML/astroML_figures#56 — Merged

`Adding more info about data source` ✓ db15f8b

Add more commits by pushing to the `fetch_cnn_images` branch on bsipocz/astroML.

All checks have passed
12 successful checks

- ✓ CI / Code linter (pull_request) Successful in 20s — Details
- ✓ CI / egg_info with Python 3.7 (pull_request) Successful in 5s — Details
- ✓ CI / py3.9-test (macos-latest) (pull_request) Successful in 1m — Details
- ✓ CI / py3.9-test (windows-latest) (pull_request) Successful in 1m — Details
- ✓ CI / py3.9-test (ubuntu-latest) (pull_request) Successful in 57s — Details
- ✓ CI / py3.8-test (ubuntu-latest) (pull request) Successful in 50s — Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
data enhancement

Projects
None yet

Milestone
1.0.2

Linked issues
Successfully merging this pull request may close these issues.
None yet

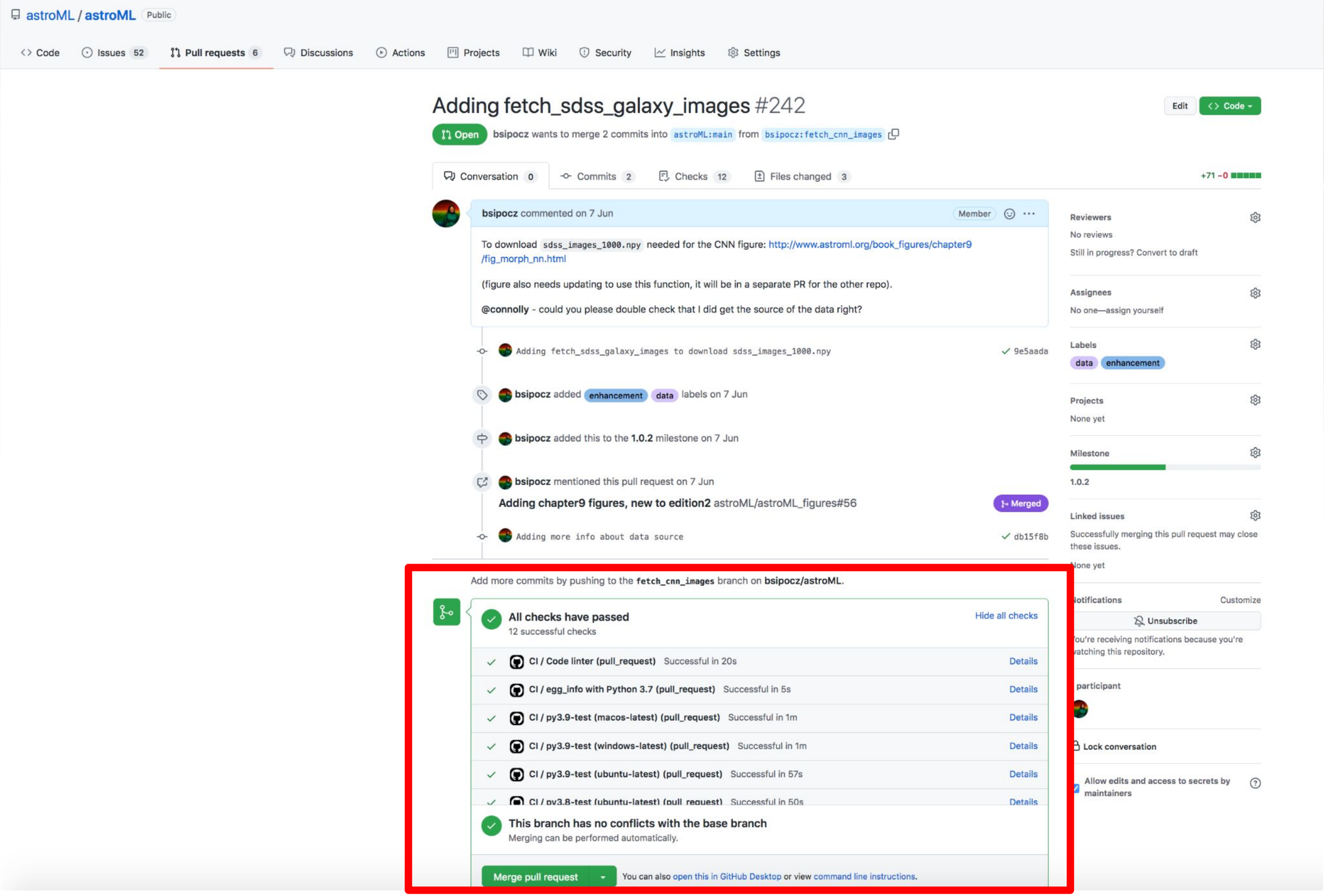
Notifications — Customize

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

☑ Allow edits and access to secrets by maintainers



Adding fetch_sdss_galaxy_images #242

Open bsipocz wants to merge 2 commits into `astroML:main` from `bsipocz:fetch_cnn_images`

bsipocz commented on 7 Jun — Member

To download `sdss_images_1000.npy` needed for the CNN figure: http://www.astroml.org/book_figures/chapter9/fig_morph_nn.html

(figure also needs updating to use this function, it will be in a separate PR for the other repo).

@connolly - could you please double check that I did get the source of the data right?

`Adding fetch_sdss_galaxy_images to download sdss_images_1000.npy` ✓ 9e5aada

bsipocz added enhancement data labels on 7 Jun

bsipocz added this to the 1.0.2 milestone on 7 Jun

bsipocz mentioned this pull request on 7 Jun

Adding chapter9 figures, new to edition2 astroML/astroML_figures#56 — Merged

`Adding more info about data source` ✓ db15f8b

Add more commits by pushing to the `fetch_cnn_images` branch on bsipocz/astroML.

All checks have passed
12 successful checks — Hide all checks

- ✓ CI / Code linter (pull_request) Successful in 20s — Details
- ✓ CI / egg_info with Python 3.7 (pull_request) Successful in 5s — Details
- ✓ CI / py3.9-test (macos-latest) (pull_request) Successful in 1m — Details
- ✓ CI / py3.9-test (windows-latest) (pull_request) Successful in 1m — Details
- ✓ CI / py3.9-test (ubuntu-latest) (pull_request) Successful in 57s — Details
- ✓ CI / py3.8-test (ubuntu-latest) (pull_request) Successful in 50s — Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.

Reviewers: No reviews. Still in progress? Convert to draft

Assignees: No one—assign yourself

Labels: data, enhancement

Projects: None yet

Milestone: 1.0.2

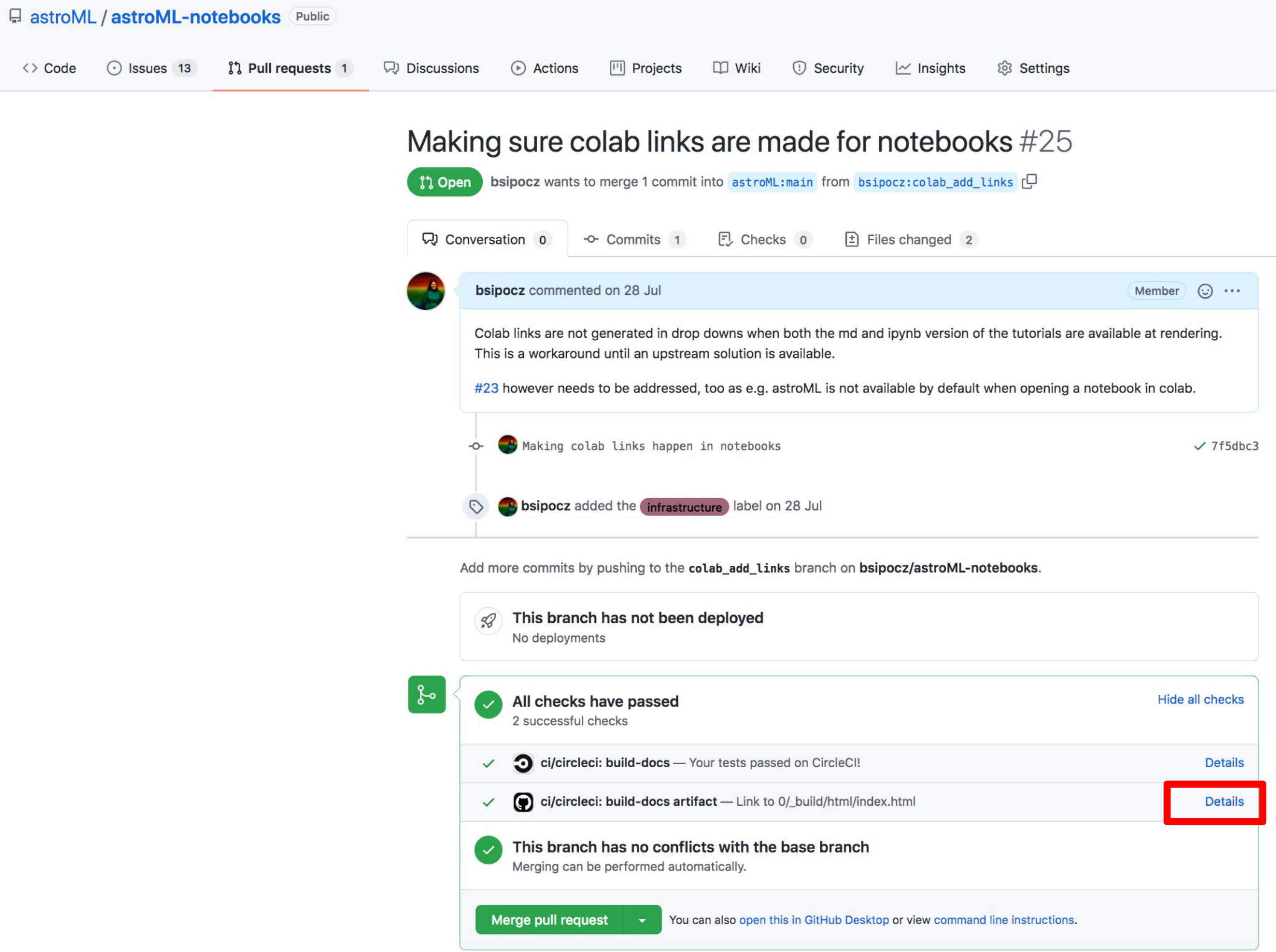
Linked issues: Successfully merging this pull request may close these issues. None yet

Notifications — Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

Allow edits and access to secrets by maintainers



Making sure colab links are made for notebooks #25

Open bsipocz wants to merge 1 commit into `astroML:main` from `bsipocz:colab_add_links`

bsipocz commented on 28 Jul Member

Colab links are not generated in drop downs when both the md and ipynb version of the tutorials are available at rendering. This is a workaround until an upstream solution is available.

#23 however needs to be addressed, too as e.g. astroML is not available by default when opening a notebook in colab.

`Making colab links happen in notebooks` ✓ 7f5dbc3

bsipocz added the infrastructure label on 28 Jul

Add more commits by pushing to the **`colab_add_links`** branch on **bsipocz/astroML-notebooks**.

This branch has not been deployed
No deployments

All checks have passed
2 successful checks

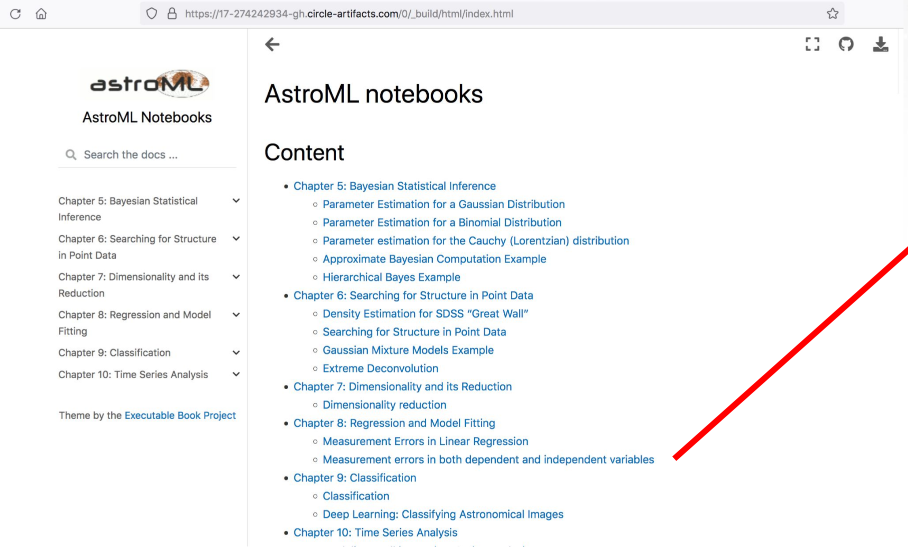
Hide all checks

✓ **ci/circleci: build-docs** — Your tests passed on CircleCI! Details

✓ **ci/circleci: build-docs artifact** — Link to 0/_build/html/index.html Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.



https://17-274242934-gh.circle-artifacts.com/0/_build/html/index.html

AstroML Notebooks

Search the docs ...

Chapter 5: Bayesian Statistical Inference

Chapter 6: Searching for Structure in Point Data

Chapter 7: Dimensionality and its Reduction

Chapter 8: Regression and Model Fitting

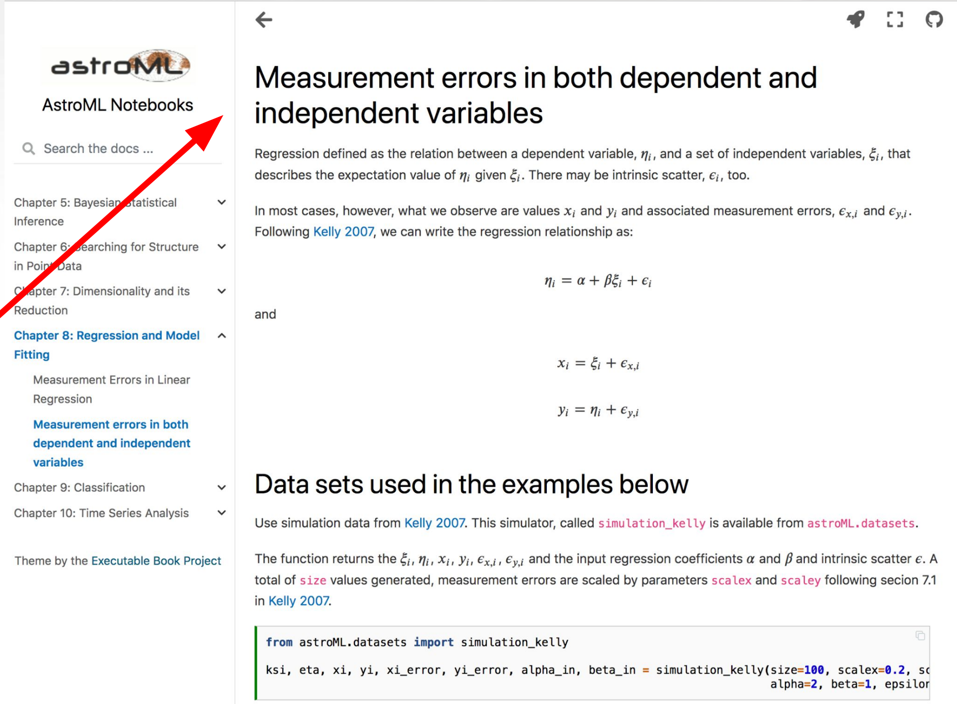
Chapter 9: Classification

Chapter 10: Time Series Analysis

Theme by the Executable Book Project

AstroML notebooks

Content



AstroML Notebooks

Search the docs ...

Chapter 5: Bayesian Statistical Inference

Chapter 6: Searching for Structure in Point Data

Chapter 7: Dimensionality and its Reduction

Chapter 8: Regression and Model Fitting

Measurement Errors in Linear Regression

Measurement errors in both dependent and independent variables

Chapter 9: Classification

Chapter 10: Time Series Analysis

Theme by the Executable Book Project

Measurement errors in both dependent and independent variables

Regression defined as the relation between a dependent variable, η_i, and a set of independent variables, ξ_i, that describes the expectation value of η_i given ξ_i. There may be intrinsic scatter, ϵ_i, too.

In most cases, however, what we observe are values x_i and y_i and associated measurement errors, $\epsilon_{x,i}$ and $\epsilon_{y,i}$. Following Kelly 2007, we can write the regression relationship as:

$$\eta_i = \alpha + \beta\xi_i + \epsilon_i$$

and

$$x_i = \xi_i + \epsilon_{x,i}$$

$$y_i = \eta_i + \epsilon_{y,i}$$

Data sets used in the examples below

Use simulation data from Kelly 2007. This simulator, called `simulation_kelly` is available from `astroML.datasets`.

The function returns the ξ_i, η_i, x_i, y_i, $\epsilon_{x,i}$, $\epsilon_{y,i}$ and the input regression coefficients α and β and intrinsic scatter ϵ. A total of `size` values generated, measurement errors are scaled by parameters `scalex` and `scaley` following secion 7.1 in Kelly 2007.

```
from astroML.datasets import simulation_kelly

ksi, eta, xi, yi, xi_error, yi_error, alpha_in, beta_in = simulation_kelly(size=100, scalex=0.2, sc
                                                                           alpha=2, beta=1, epsilor
```

CI services vs on premise

- Easier and/or practical especially user products:
  - standalone libraries
  - Jupyter notebooks
  - user and applications
- For larger, collaborative projects from code to book writing
- Endless possibilities and examples

- Serious caveat: it requires a containerized environment