



**astropy**

A Community Python Library for Astronomy

# **THE ASTROPY PROJECT: UPDATES AND RAMBLINGS**

**Erik Tollerud**

@eteq

Yale University

Astropy Coordinating Committee Member

Hubble Fellow

# ASTROPY'S ORIGIN STORY

Q. How do I use python to convert from Equatorial J2000 RA/Dec to Galactic coordinates (as of 2011)?

A. Use any of:

- pyast
- Astrolib
- Astropysics
- Kapteyn
- EphemPy
- PyAst
- PyAstro
- Probably more...

Lots of wasted effort!

Mutually incompatible!

# ASTROPY'S ORIGIN STORY

Everyone agreed this was bad.

Do we as a community really need yet another separate python library for astronomy and yet another attempt at building a core set of routines ported from the IDL library?

Marshall Perrin on "astropy" mailing list, June 2011

# ASTROPY'S ORIGIN STORY

Everyone agreed this was bad.

(Agreement ends up *crucial* to shared development.)

A grassroots discussion started in June 2011, followed by a series of votes (~100 astronomers).

The Result:  **astropy**

Check out <http://bit.ly/astropyvision> for the original “vision”

# THE ASTROPY PROJECT AND PACKAGE

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

Core package “astropy”  $\neq$  “Astropy project”

Core package is what’s in github repo *astropy/astropy*

Project includes all the affiliated packages and associated community

Three-member coordination committee: currently me, Tom R, Perry

# WHAT IS THE PHILOSOPHY BEHIND THE CODE?

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

# WHAT IS THE PHILOSOPHY BEHIND THE CODE?

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

This means both *by*  
and *for* the  
community

# WHAT IS THE PHILOSOPHY BEHIND THE CODE?

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

This means both *by*  (Professional) Astronomers help write it  
and *for* the  
community

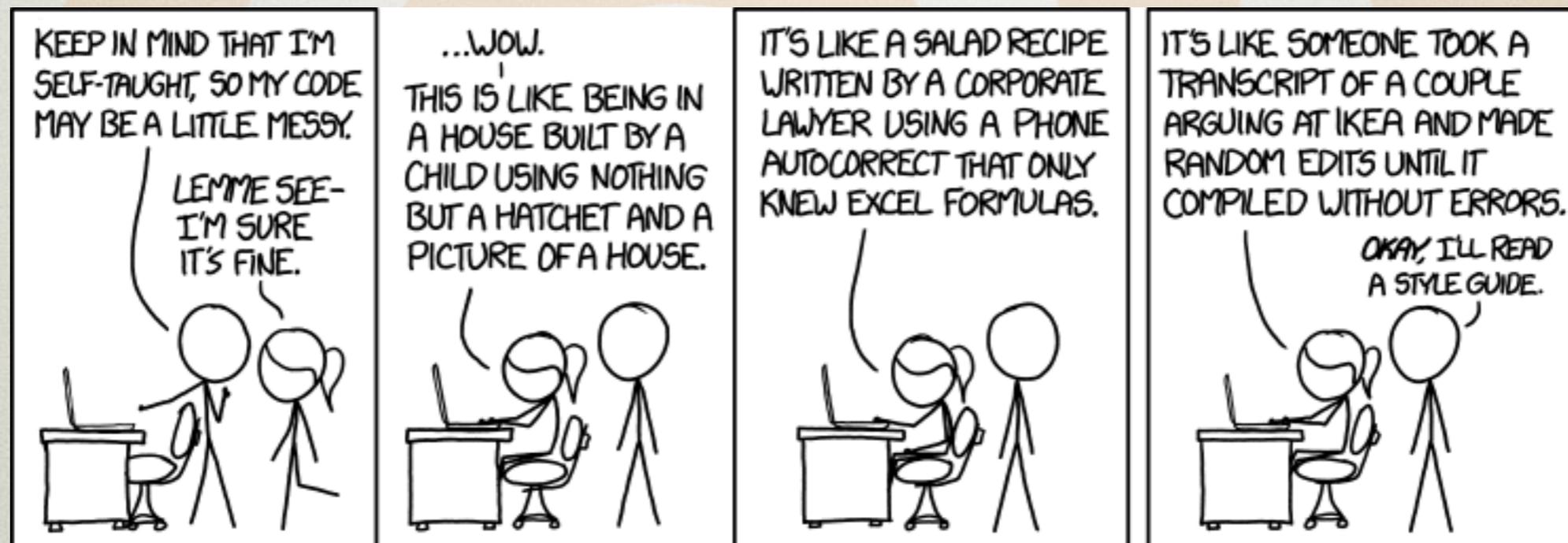
# WHAT IS THE PHILOSOPHY BEHIND THE CODE?

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

This means both *by* (Professional) Astronomers help write it  
and *for* the community  
It should be familiar for them as part of their day-to-day work

# WHAT IS THE PHILOSOPHY BEHIND THE CODE?

But at the same time, it  
should address this problem:



By showing how things *can* be done!  
(Which means more collaboration with software  
folks, and pushing researcher's comfort zones)

# WHAT'S IN THE ASTROPY CORE PACKAGE

Best place to look is always

<http://docs.astropy.org>

# WHAT'S IN THE ASTROPY CORE PACKAGE

Best place to look is always

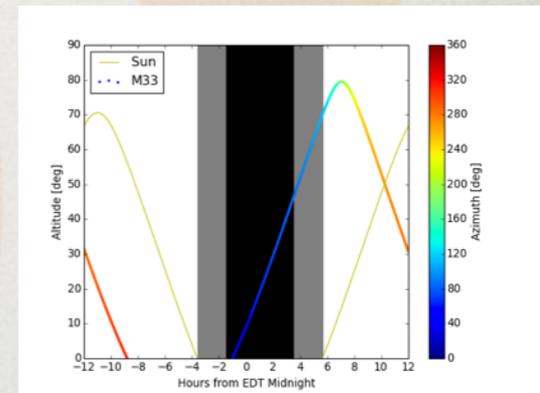
<http://docs.astropy.org>

- 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 (and other astro) coordinates
- Image analysis and interoperability data structures.
- WCS
- FITS, VOTable, hdf5, extensible I/O
- Table manipulation, including many arcane astro formats
- Cosmology tools
- Astrostatistics convenience functions
- Data modeling and fitting
- Configuration and plumbing to make it all work

# WHAT WAS BIG IN v1.0?

Released February

- Promise of Long-Term Support (“v1.0”)
- Support for Celestial  $\Leftrightarrow$  AltAz coordinate transformations
- Optimized table I/O, basically C-speed
- modeling is ready for prime-time
- tables columns can be arbitrary fancy things (“mixin columns”)
- 676 other enhancements!



# WHAT WILL BE BIG IN v1.1?

July / August?

- Better Quantities (log / functional units? distributions? masked quantities?)
- Velocity support in coordinates?
- Support quantities and uncertainties in modeling?
- Ephemerides?
- You can help decide this week!

# HOW DO YOU LEARN MORE ABOUT USING ASTROPY?

<Cruz's talk!>

Talk to one of us this week

Astropy mailing list

astropy-dev mailing list

(The new facebook group?)

**astropy** Tutorials

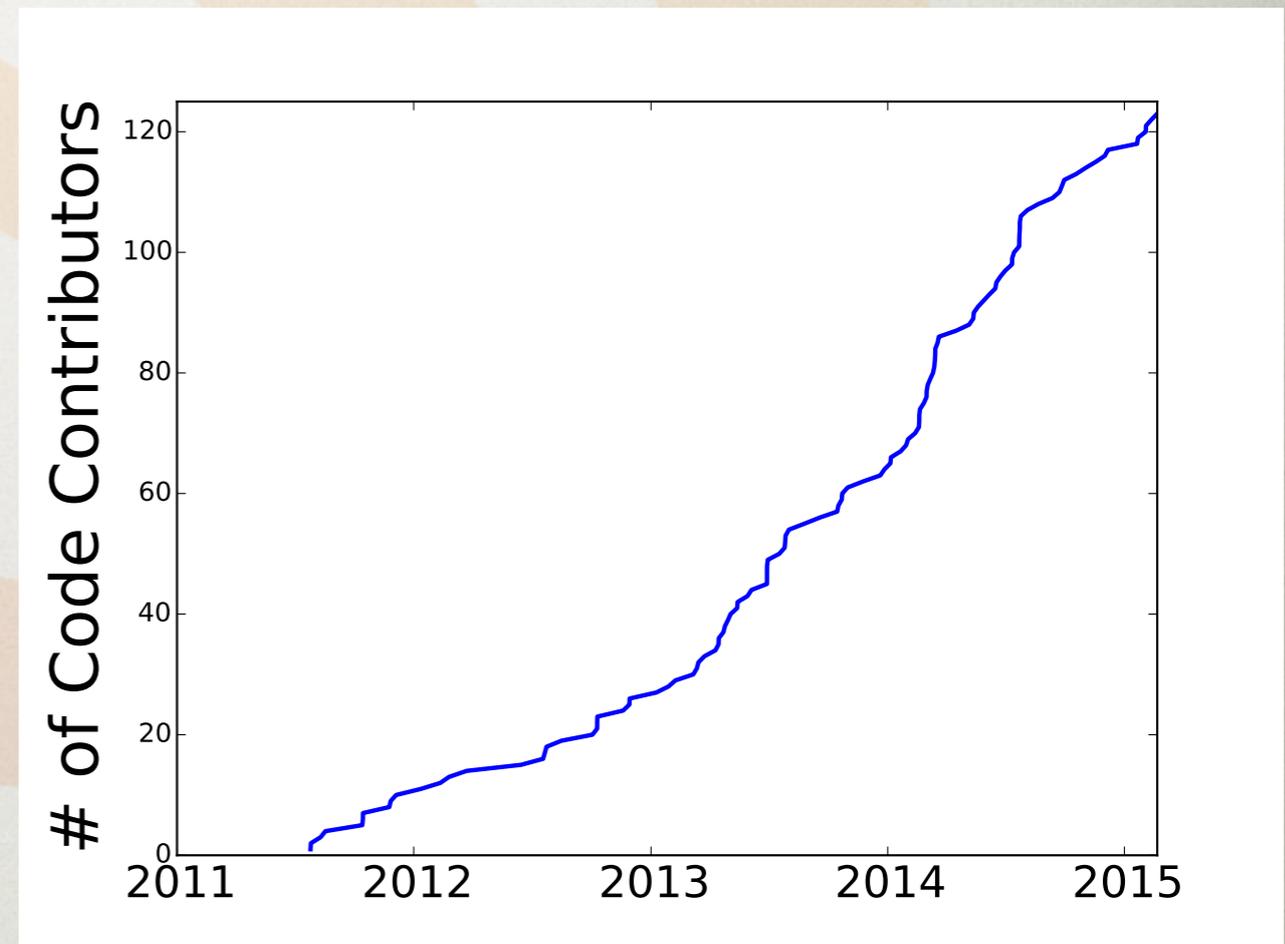
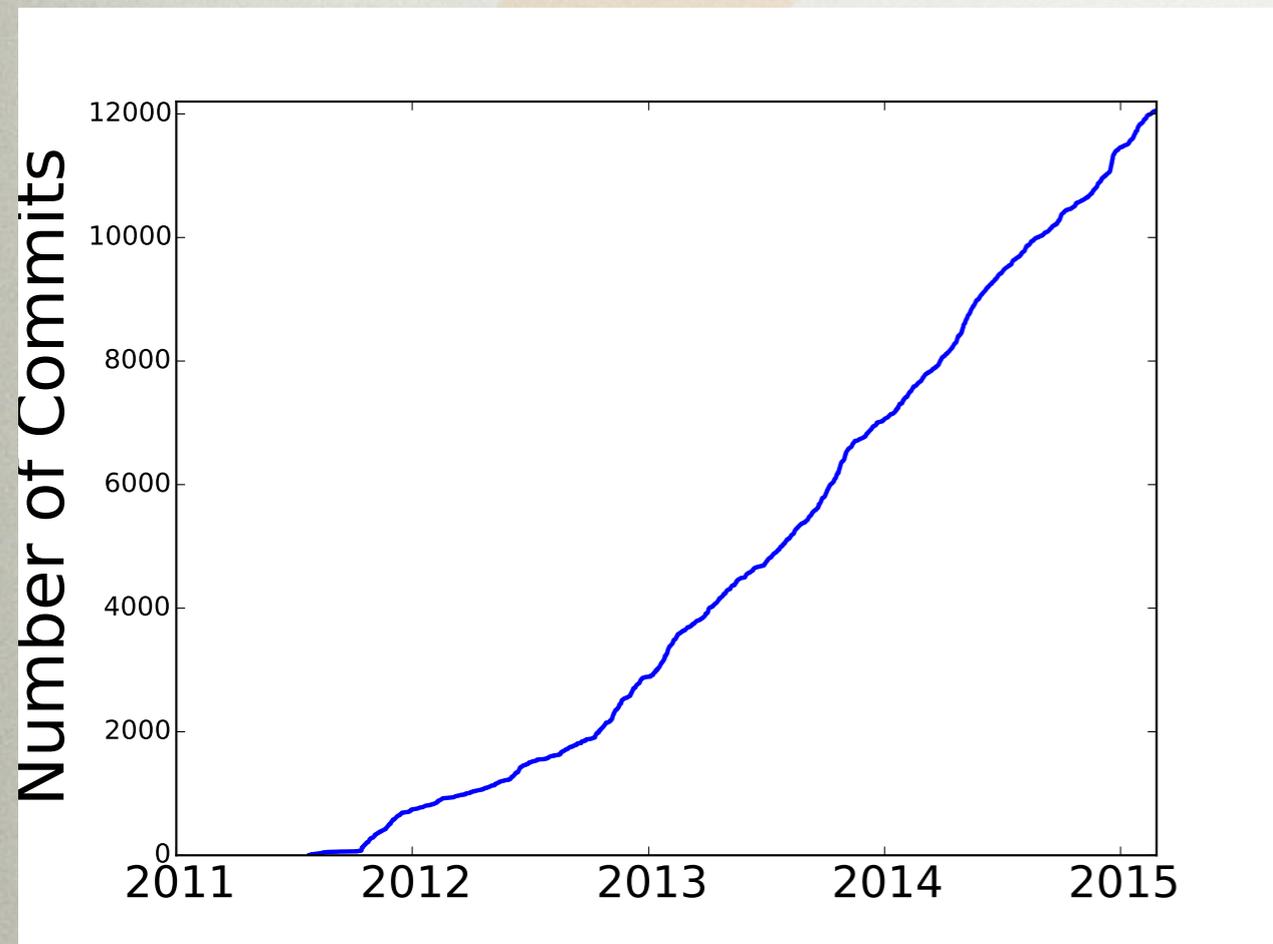
<http://tutorials.astropy.org>

<http://docs.astropy.org>

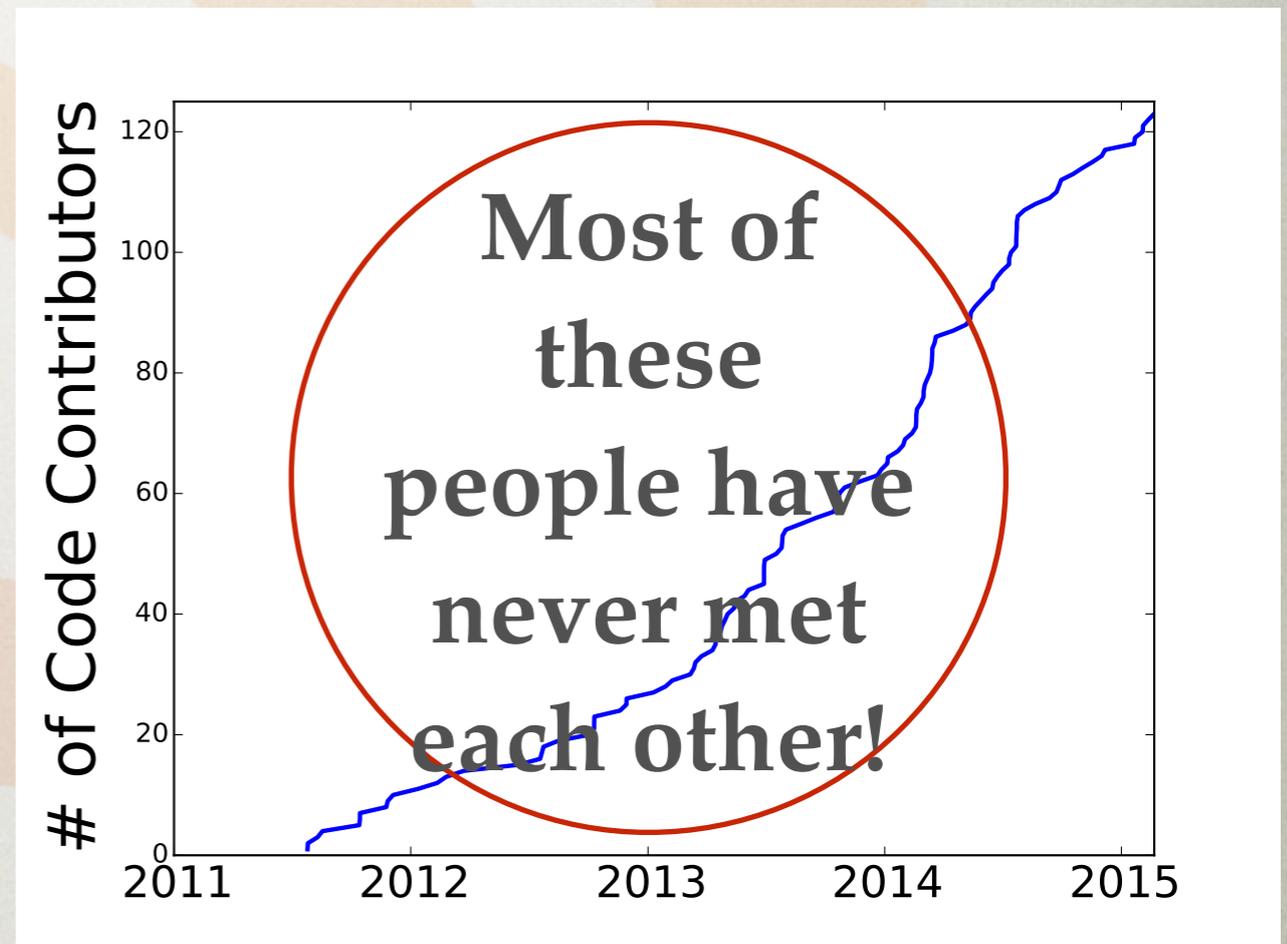
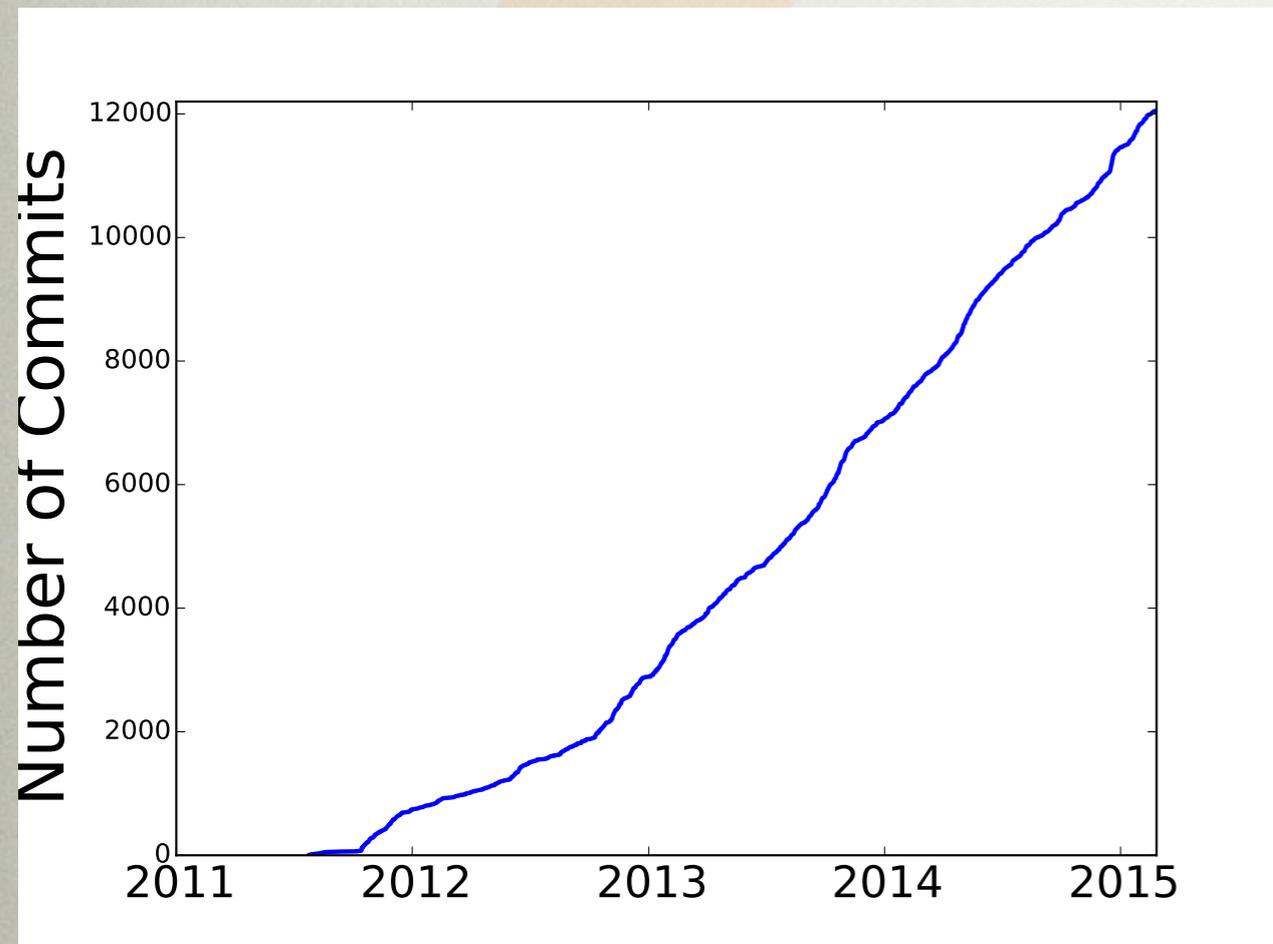
# THE “BY” PART

The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.

# CONTRIBUTIONS ARE GROWING



# CONTRIBUTIONS ARE GROWING



# WHAT ARE THE KEY ELEMENTS?



# WHAT ARE THE KEY ELEMENTS?

github  
SOCIAL CODING



astropy / astropy Unwatch 86 Unstar 530 Fork 375

## Issue 3601 -- astropy.stats.funcs.bootstrap now accepts bootfuncs with multiple outputs #3628

Open ezbc wants to merge 7 commits into `astropy:master` from `ezbc:issue3601`

Conversation 8 Commits 7 Files changed 3 +115 -6

**ezbc** commented 26 days ago

I have addressed [Issue 3601](#). I implemented the option for users to supply a function with multiple outputs to `bootfunc`. They can control which `bootfunc` outputs to retain with `output_index`.

This function could be sped up if the `if` statements were moved outside of the loop.

**ezbc** added some commits 26 days ago

- `initial commit` 84afdce
- `fully functional, needs indices to be output_index` b69447d
- `bootstrap: reworked indices variable to be more pythonic` e7ac659
- `updated changes log` b7c1515
- `updated changes log for issue3601` 9e01308

**embray** added `stats` `Affects-release` labels 26 days ago

**embray** commented 26 days ago Collaborator

@ezbc Your PR so far has the changelog entry and some tests which look good, but is missing the actual change to the function. Is it intentionally not implemented yet?

`definite commit of funcs.py` 594ddd0

**Labels**  
`Affects-release`  
`stats`

**Milestone**  
v1.1.0

**Assignee**  
No one—assign yourself

**Notifications**  
Unsubscribe  
You're receiving notifications because you were mentioned.

**5 participants**

Lock pull request

# WHAT ARE THE KEY ELEMENTS?

github  
SOCIAL CODING



 **ttshimiz** commented 4 days ago ✎ ✕

@ezbc, @bsipocz, @eteq Yeah that would definitely be an acceptable solution to me. Examples and an explanation in the documentation would really help to show people how to use bootstrap with functions that have multiple inputs and outputs. It didn't even occur to me to use lambda to just define a new function that only returned the first output of spearmanr. In my view, simpler is always better. Thanks for addressing this issue!

 **✓ All is well** — 3 successful checks Show all checks

**This pull request can be automatically merged.** 🗨 Merge pull request  
You can also merge branches on the [command line](#).

 **Write** **Preview** 📄 Markdown supported 🖥 Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Close pull request Comment

 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.



# WHAT ARE THE KEY ELEMENTS?

github  
SOCIAL CODING



**embroy** commented 26 days ago Collaborator

@ezbc Your PR so far has the changelog entry and some tests which look good, but is missing the actual change to the function. Is it intentionally not implemented yet?

5 participants Lock pull request

definite commit of funcs.py 594ddd0

**ezbc** commented 26 days ago

@embroy This was not intentional, sorry about that. I must have committed funcs to a different branch. I believe the changes are now pushed to the correct branch, issue3601, and are now ready to be merged.

**bsipocz** commented on an outdated diff 26 days ago Show outdated diff

**bsipocz** commented on the diff 26 days ago

```
astropy/stats/tests/test_funcs.py View full changes
```

	@@ -236,6	+238,60 @@
236	238	def test_bootstrap():
237	239	bootresult = np.mean(funcs.bootstrap(bootarr, 10000, bootfunc=np.m
238	240	assert_allclose(np.mean(bootarr), bootresult, atol=0.01)
	241	+ # test a bootfunc with several output values
	242	+ # return just bootstrapping with one output from bootfunc
	243	+ with NumpyRNGContext(42):
	244	+     bootarr = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 0],
	245	+                       [4, 8, 8, 3, 6, 5, 2, 8, 6, 2]]).T
	246	+
	247	+     answer = np.array((0.19425, 0.02094))
	248	+
	249	+     bootresult = funcs.bootstrap(bootarr, 2,
	250	+                               bootfunc=spearmanr)

**bsipocz** added a note 26 days ago Collaborator

I think you may want to put this new part in a separate function as it depends on scipy. The old test can run without scipy, and this new function gets the decorator and runs only when scipy is available.

Add a line note

# WHAT ARE THE KEY ELEMENTS?

github  
SOCIAL CODING



embray commented 26 days ago Collaborator ✎ ✕

@ezbc Your PR so far has the changelog entry and some tests which look good, but is missing the actual change to the function. Is it intentionally not implemented yet?

definite commit of funcs.py 594ddd0

ezbc commented 26 days ago ✎ ✕

@embray This was not intentional, sorry about that. I must have committed funcs to a different branch. I believe the changes are now pushed to the correct branch, issue3601, and are now ready to be merged.

bsipocz commented on an outdated diff 26 days ago ✎ ✕ Show outdated diff

bsipocz commented on the diff 26 days ago <>

astropy/stats/tests/test\_funcs.py View full changes

@@	-236,6	+238,60	@@	def test_bootstrap():
236	238			bootresult = np.mean(funcs.bootstrap(bootarr, 10000, bootfunc=np.m
237	239			assert_allclose(np.mean(bootarr), bootresult, atol=0.01)
238	240			
	241	+		# test a bootfunc with several output values
	242	+		# return just bootstrapping with one output from bootfunc
	243	+		with NumpyRNGContext(42):
	244	+		bootarr = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 0],
	245	+		[4, 8, 8, 3, 6, 5, 2, 8, 6, 2]]).T
	246	+		
	247	+		answer = np.array((0.19425, 0.02094))
	248	+		
	249	+		bootresult = funcs.bootstrap(bootarr, 2,
	250	+		bootfunc=spearmanr)

bsipocz added a note 26 days ago Collaborator ✎ ✕

I think you may want to put this new part in a separate function as it depends on scipy. The old test can run without scipy, and this new function gets the decorator and runs only when scipy is available.

Add a line note

# WHAT ARE THE KEY ELEMENTS?

github  
SOCIAL CODING



 **ttshimiz** commented 4 days ago ✎ ✕

@ezbc, @bsipocz, @eteq Yeah that would definitely be an acceptable solution to me. Examples and an explanation in the documentation would really help to show people how to use bootstrap with functions that have multiple inputs and outputs. It didn't even occur to me to use lambda to just define a new function that only returned the first output of spearmanr. In my view, simpler is always better. Thanks for addressing this issue!

 **✓ All is well** — 3 successful checks Show all checks

**This pull request can be automatically merged.**  
You can also merge branches on the [command line](#).

 **Merge pull request**

 **Write** **Preview** Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

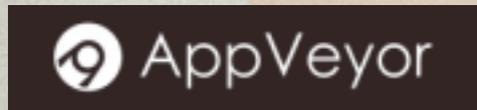
**Close pull request** **Comment**

 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.



# WHAT ARE THE KEY ELEMENTS?

py.test



A screenshot of a GitHub pull request page. At the top, a comment from user 'ttshimiz' is visible, dated '4 days ago'. The comment text reads: '@ezbc, @bsipocz, @eteq Yeah that would definitely be an acceptable solution to me. Examples and an explanation in the documentation would really help to show people how to use bootstrap with functions that have multiple inputs and outputs. It didn't even occur to me to use lambda to just define a new function that only returned the first output of spearmanr. In my view, simpler is always better. Thanks for addressing this issue!'. Below the comment is a green status bar indicating 'All is well — 3 successful checks' with a 'Show all checks' link. A message states 'This pull request can be automatically merged.' with a 'Merge pull request' button. At the bottom of the pull request view, there is a 'Write' tab, a 'Preview' tab, and a text area for 'Leave a comment'. A 'ProTip!' is displayed at the bottom: 'Add .patch or .diff to the end of URLs for Git's plaintext views.' The footer of the page includes copyright information for GitHub, Inc. and various navigation links.



# WHAT ARE THE KEY ELEMENTS?



## SPHINX

PYTHON DOCUMENTATION GENERATOR



### Read the Docs

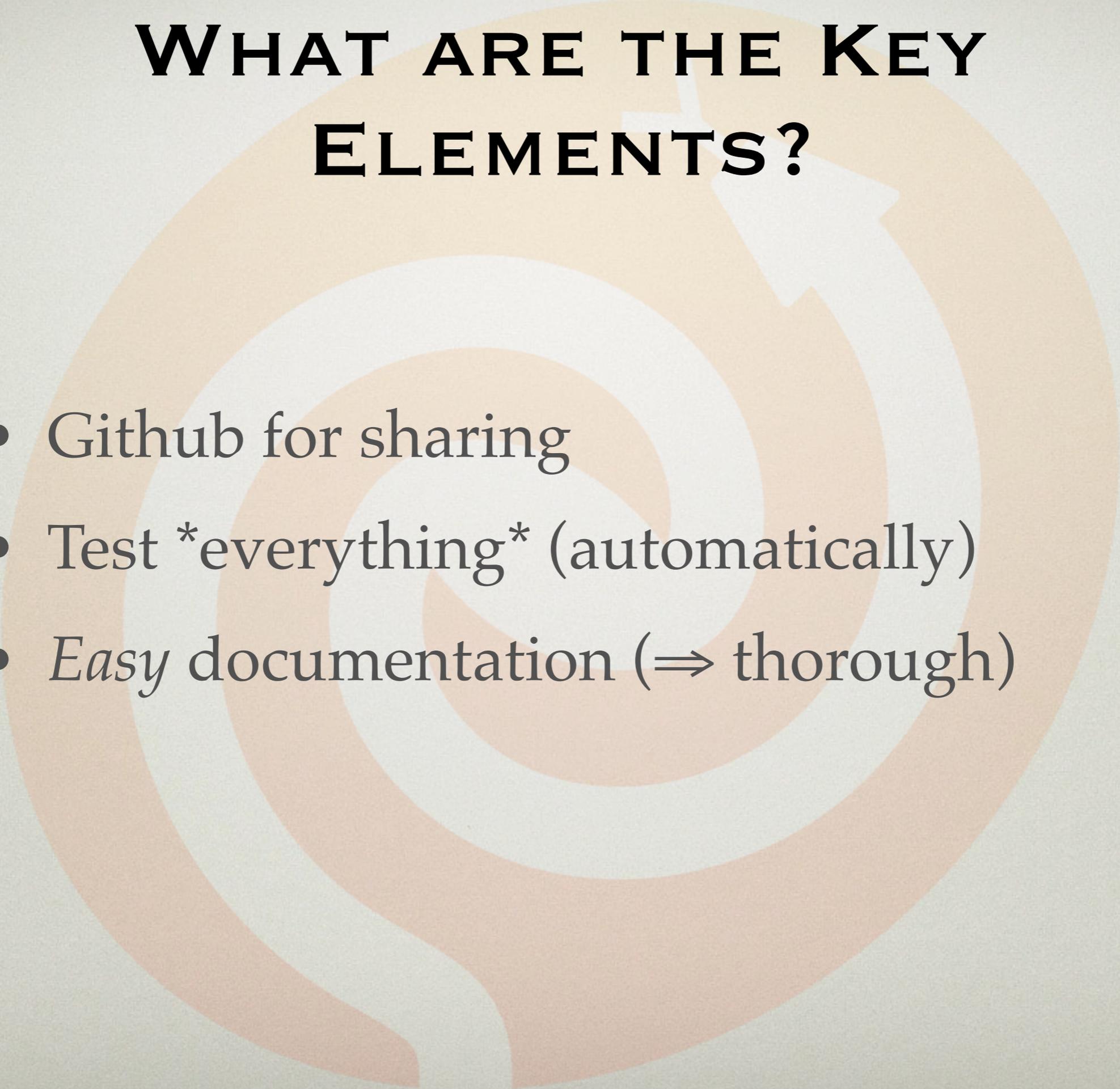
Create, host, and browse documentation.

The screenshot shows the Read the Docs website for the `astropy.table` package. The page title is "Data Tables (`astropy.table`)". The left sidebar contains a "Page Contents" section with a tree view of the documentation structure, including "Introduction", "Getting Started", "Using `table`", "Reference/API", and "astropy.table Module". The main content area is titled "Data Tables (`astropy.table`)" and has an "Introduction" section. The introduction text states that `astropy.table` provides functionality for storing and manipulating heterogeneous tables of data in a way familiar to `numpy` users. A list of features is provided, including: initializing a table from various input data structures; modifying a table by adding or removing columns, changing column names, or adding new rows; handling tables with missing values; including table and column metadata; specifying descriptions, units, and output formatting; interactive scrolling for long tables; creating new tables by selecting rows or columns; performing `Table operations` like database joins and concatenation; manipulating multidimensional columns; methods for reading and writing `Table objects` to files; and hooks for subclassing `Table` and its component classes. A "Note" box highlights a change in the internal implementation of the `Table` class starting with version 1.0, noting that it no longer uses `numpy` structured arrays as the core data container. The "Getting Started" section begins by describing the basic workflow for creating a table, accessing elements, and modifying it, with examples showing a simple case and a more complex one using the `astropy.table` documentation. A code block at the bottom shows the following Python code:

```
>>> from astropy.table import Table
>>> a = [1, 4, 5]
>>> b = [2.0, 5.0, 8.2]
>>> c = ['x', 'y', 'z']
>>> t = Table([a, b, c], names=('a', 'b', 'c'), meta={'name': 'first table'})
```



# WHAT ARE THE KEY ELEMENTS?

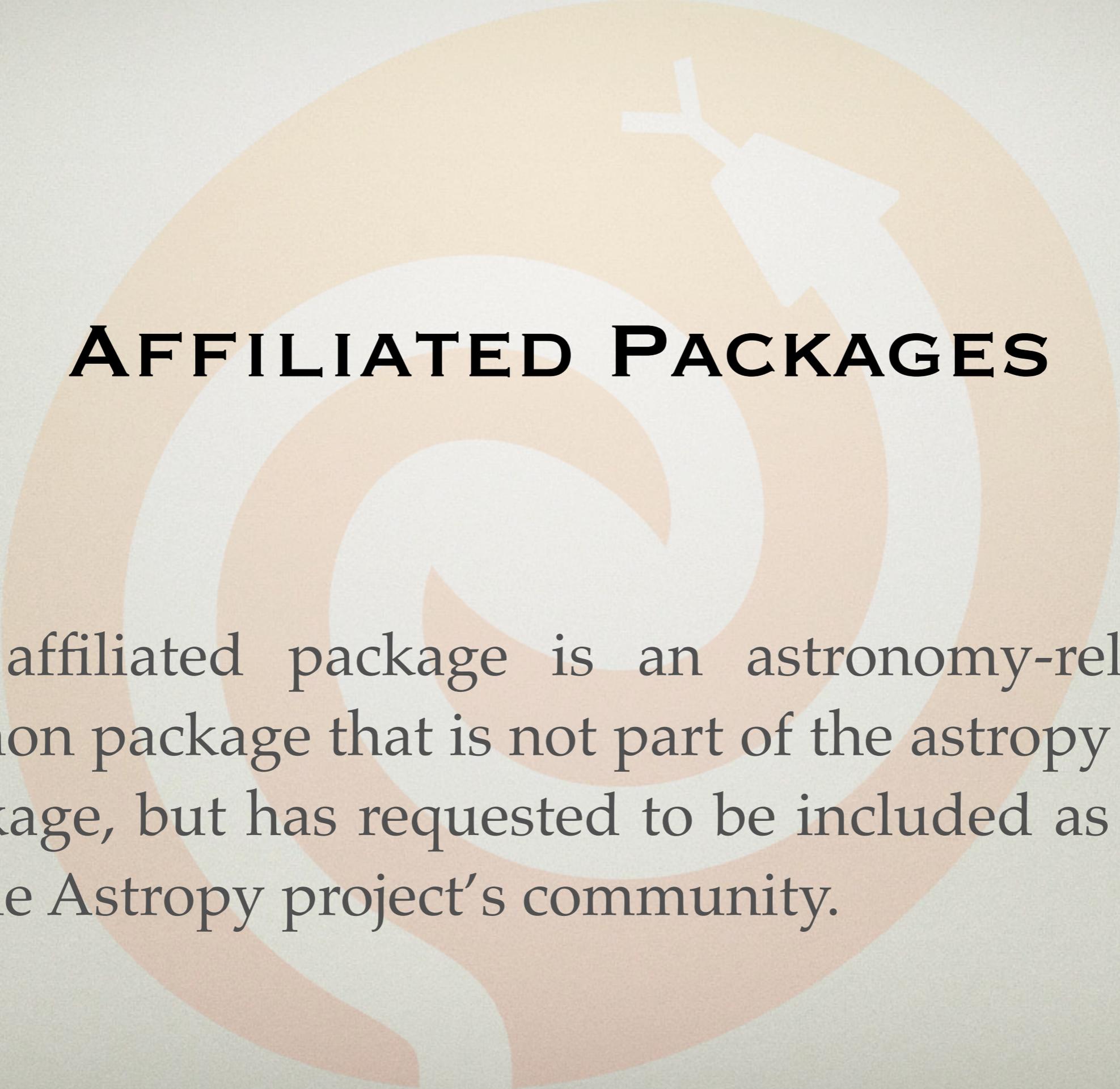


- Github for sharing
- Test *\*everything\** (automatically)
- *Easy* documentation ( $\Rightarrow$  thorough)

# WHAT ARE THE KEY ELEMENTS?

- Github for sharing
- Test *\*everything\** (automatically)
- *Easy* documentation ( $\Rightarrow$  thorough)

Question for discussion: can we do this with “ordinary” research code / collaborations?

The background features a large, faint Astropy logo. It consists of a stylized orange and white spiral that forms a large 'A' shape, with a white telescope icon positioned at the top right of the spiral's outer edge.

# **AFFILIATED PACKAGES**

An affiliated package is an astronomy-related Python package that is not part of the astropy core package, but has requested to be included as part of the Astropy project's community.

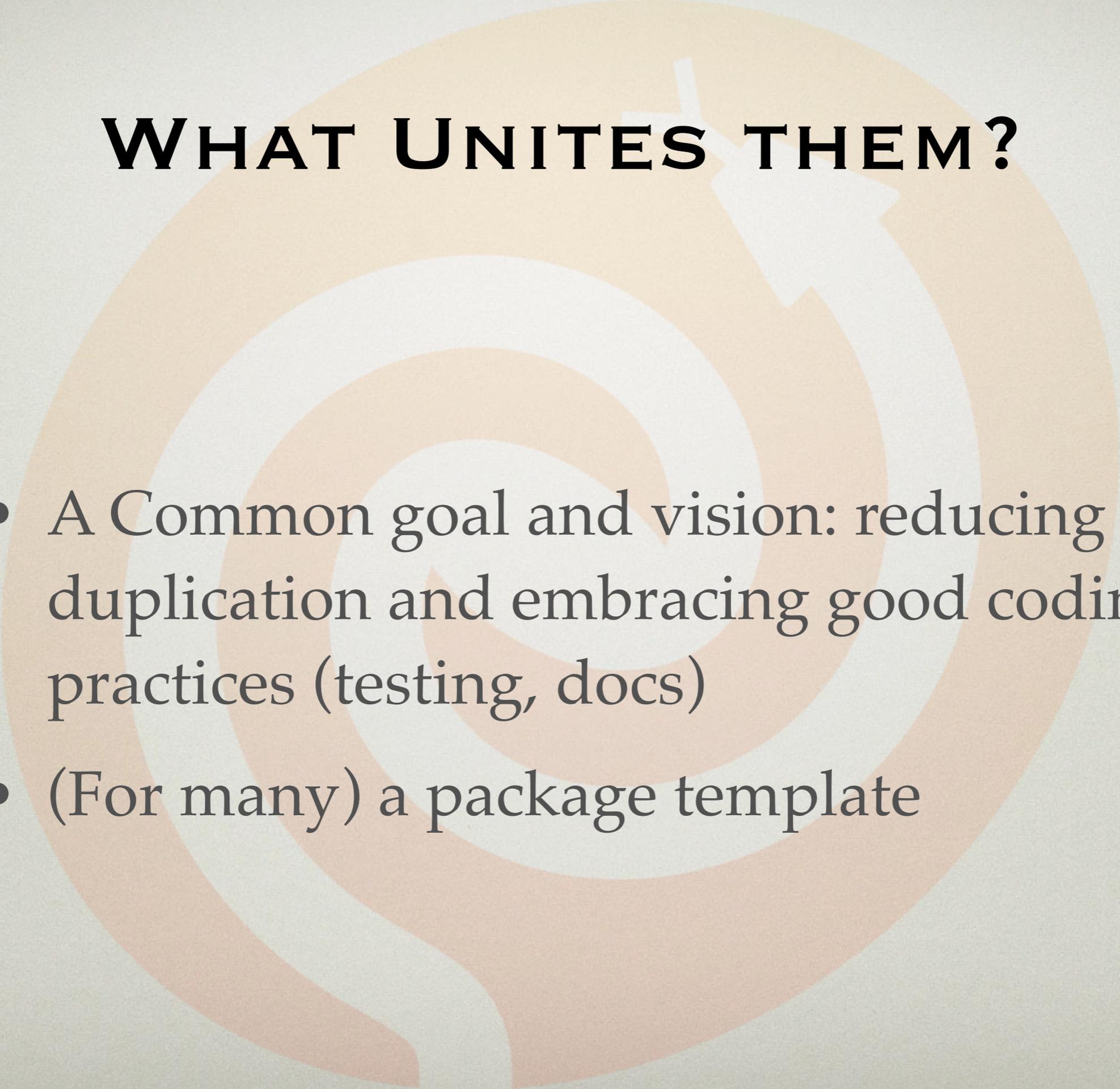
# AFFILIATED PACKAGES

Best place to look is always

<http://affiliated.astropy.org>

- APLpy: astronomical plotting
- astroML: astro machine learning (companion to a textbook)
- astroquery: access to internet resources <see Ginsburg talk>
- ccdproc: ccd reductions <see Crawford talk?>
- gammapy: gamma-ray astronomy
- ginga: interactive image viz
- imexam: quick image analysis
- montage-wrapper: image mosaicing
- photutils: photometry
- pydl: simple IDL ports
- pyregion: ds9 region files
- PyVO: VO access
- snocosmo: supernova light curves fitting / typing / etc
- specutils: spectroscopy
- spherical\_geometry: spherical polygons / regions
- WCSAxes: WCS-aware matplotlib plots

# WHAT UNITES THEM?



- A Common goal and vision: reducing duplication and embracing good coding practices (testing, docs)
- (For many) a package template

# AFFILIATED PACKAGE TEMPLATE

- Contains a ready-to-go “copy” of the astropy package layout. (Leans heavily on *astropy-helpers*)
- Provides documentation tools, testing framework, cython, configuration, etc.
- Docs on how to actually make it all work!

The screenshot shows the GitHub repository page for 'astropy / package-template'. The repository is maintained by '@astrofrog' and has 173 commits, 1 branch, 2 releases, and 13 contributors. A merge pull request #112 from 'eteq/add-badge' is currently open. The repository contains several files and folders, including 'astropy\_helpers', 'cextern', 'docs', 'licenses', 'packagename', '.gitignore', '.gitmodules', '.travis.yml', 'MANIFEST.in', 'README.rst', 'TEMPLATE\_CHANGES.md', 'ah\_bootstrap.py', 'ez\_setup.py', 'setup.cfg', and 'setup.py'. The README.rst file is displayed, featuring the title 'Astropy affiliated package template', a 'powered by Astropy' badge, and a description: 'This is the template for affiliated packages of the Astropy project. Astropy affiliated packages are astronomy-related Python packages that have requested to be part of the Astropy project's community. Such packages can make use of the setup, installation, and documentation infrastructure developed for the astropy core package simply by using this template to lay out the package. For more information, see: Detailed instructions for using this template, The Affiliated Packages section of the Astropy web site, and This template's Github code repository. Status reports for developers show a 'build passing' status.

# AFFILIATED PACKAGE PHILOSOPHY: DOES IT WORK?

“Y’all’s work is great! [the affiliated package template] has made our life so much easier.”

Andrew Hearin, working on  
“halotools” affiliated package to-be,  
Feb 2015

“Huh? I don’t get it. This is confusing.”

Probably wants to be anonymous,  
while attempting to adopt the template  
Dec 2014

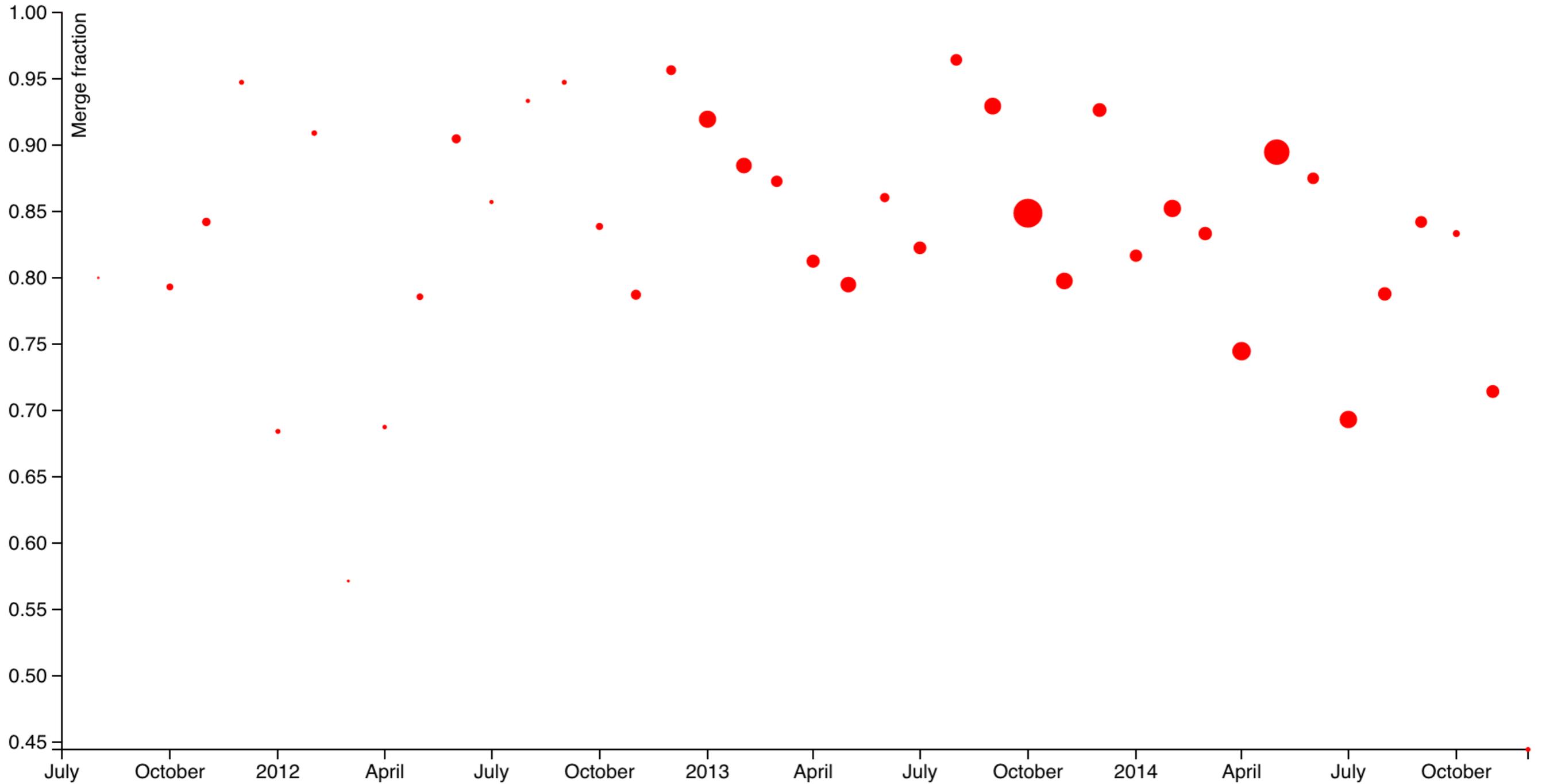
“I don’t see why I should care. My  
code doesn’t have units.”

Definitely wants to be anonymous,  
on using the template for a package  
(which does astrophysics)  
Jan 2015



**WHAT ABOUT  
CHALLENGES?**

# KEEPING UP WITH THE PRS'S



# GIVING CREDIT WHERE CREDIT IS DUE

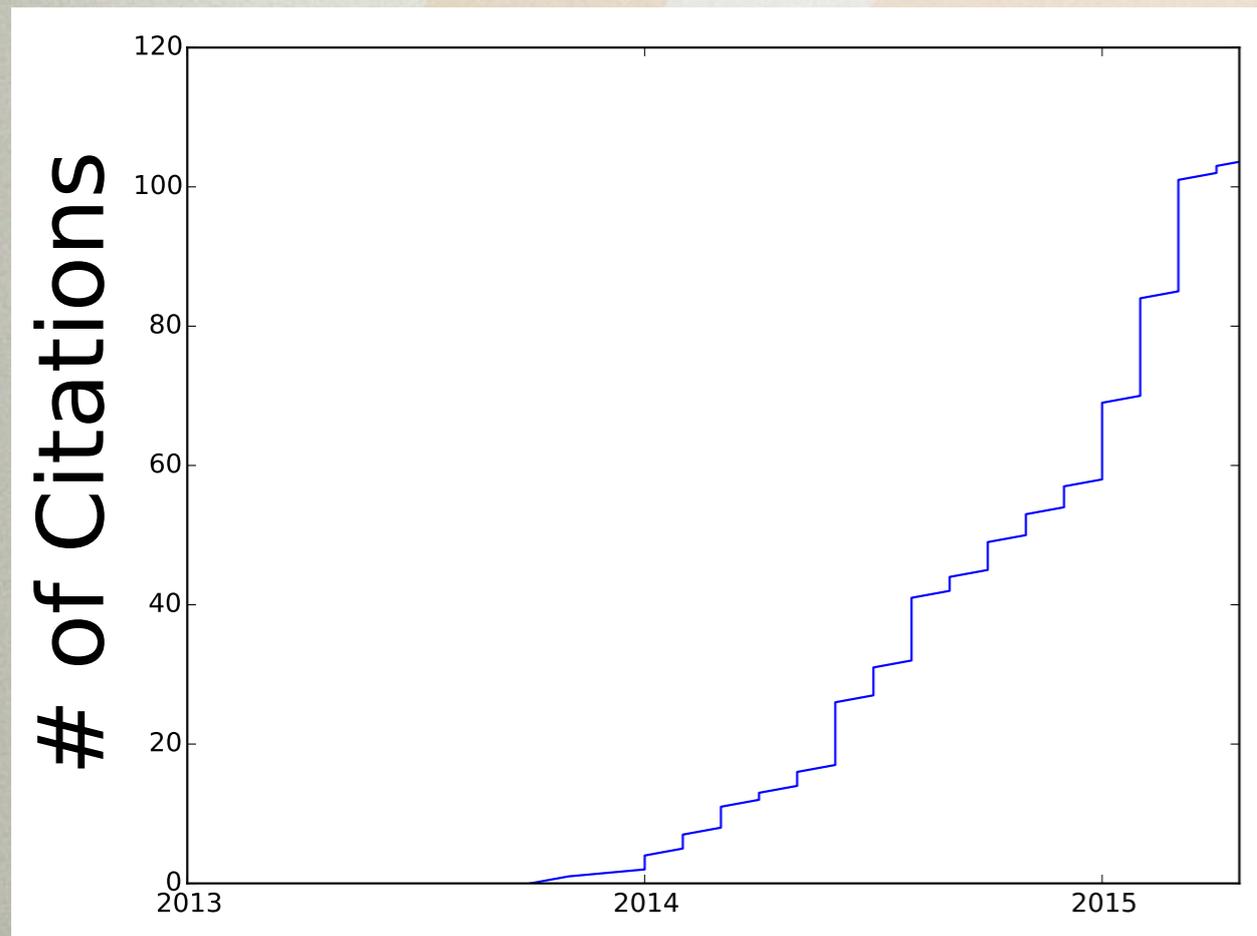
- Shailesh Ahuja
- Tom Aldcroft
- Kyle Barbary
- Geert Barentsen
- Paul Barrett
- Andreas Baumbach
- Chris Beaumont
- Daniel Bell
- Francesco Biscani
- Thompson Le Blanc
- Christopher Bonnett
- Joseph Jon Booker
- Médéric Boquien
- Azalee Bostroem
- Matthew Bourque
- Larry Bradley
- Gustavo Bragança
- Erik M. Bray
- Eli Bressert
- Hugo Buddelmeijer
- Mihai Cara
- Mabry Cervin
- Pritish Chakraborty
- Alex Conley
- Jean Connelly
- Simon Conseil
- Ryan Cooke
- Matthew Craig
- Steven Crawford
- Neil Crighton
- Kelle Cruz
- Daniel Datsev
- Matt Davis
- Christoph Deil
- Nadia Dencheva
- Jörg Dietrich
- Axel Donath
- Michael Droettboom
- Jonathan Eisenhamer
- Zach Edwards
- Thomas Erben
- Henry Ferguson
- Jonathan Foster
- Ryan Fox
- Lehman Garrison
- Simon Gibbons
- Adam Ginsburg
- Christoph Gohlke
- Perry Greenfield
- Dylan Gregersen
- Frédéric Grollier
- Karan Grover
- Kevin Gullikson
- Hans Moritz Günther
- Alex Hagen
- Michael Hoenig
- Emma Hogan
- Chris Hanley
- JC Hsu
- Anthony Horton
- Eric Jeschke
- Sarah Kendrew
- Marten van Kerkwijk
- Wolfgang Kerzendorf
- Lennard Kiehl
- Rashid Khan
- Dominik Klaes
- Kacper Kowalik
- Roban Hultman  
Kramer
- Arne de Laat
- Antony Lee
- Simon Liedtke
- Pey Lian Lim
- Joseph Long
- Aaron Meisner
- Serge Montagnac
- José Sabater Montes
- Michael Mueller
- Stuart Mumford
- Demitri Muna
- Prasanth Nair
- Bogdan Nicula
- Joe Philip Ninan
- Bryce Nordgren
- Miruna Oprescu
- Luigi Paioro
- Asish Panda
- Madhura Parikh
- Sergio Pascual
- Rohit Patil
- David Perez-Suarez
- Ray Plante
- Adrian Price-Whelan
- Xavier Prochaska
- David Pérez-Suárez
- QuanTakeuchi
- Tanuj Rastogi
- Thomas Robitaille
- Juan Luis Cano  
Rodríguez
- Evert Rol
- Alex Rudy
- Joseph Ryan
- Eloy Salinas
- Gerrit Schellenberger
- David Shiga
- David Shupe
- Jonathan Sick
- Leo Singer
- Brigitta Sipocz
- Shivan Sornarajah
- Shantanu Srivastava
- Ole Streicher
- Bernardo Sulzbach
- James Taylor
- Jeff Taylor
- Kirill Tchernyshyov
- Víctor Terrón
- Erik Tollerud
- James Turner
- Miguel de Val-Borro
- Jonathan Whitmore
- Lisa Walter
- Benjamin Alan  
Weaver
- Jonathan Whitmore
- Julien Woillez
- Víctor Zabalza

# GIVING JOBS WHERE JOBS ARE DUE

- We are facing a possible brain drain of pro-social software workers from astro (data science = \$\$\$)
- Is the solution within the current system? E.g., Astropy Papers? Calling us “electronic instrumentalists”?
- Or more drastic? E.g., Formal altmetrics, new job tracks, accept perma-soft-money
- (A worthy unconference topic?)

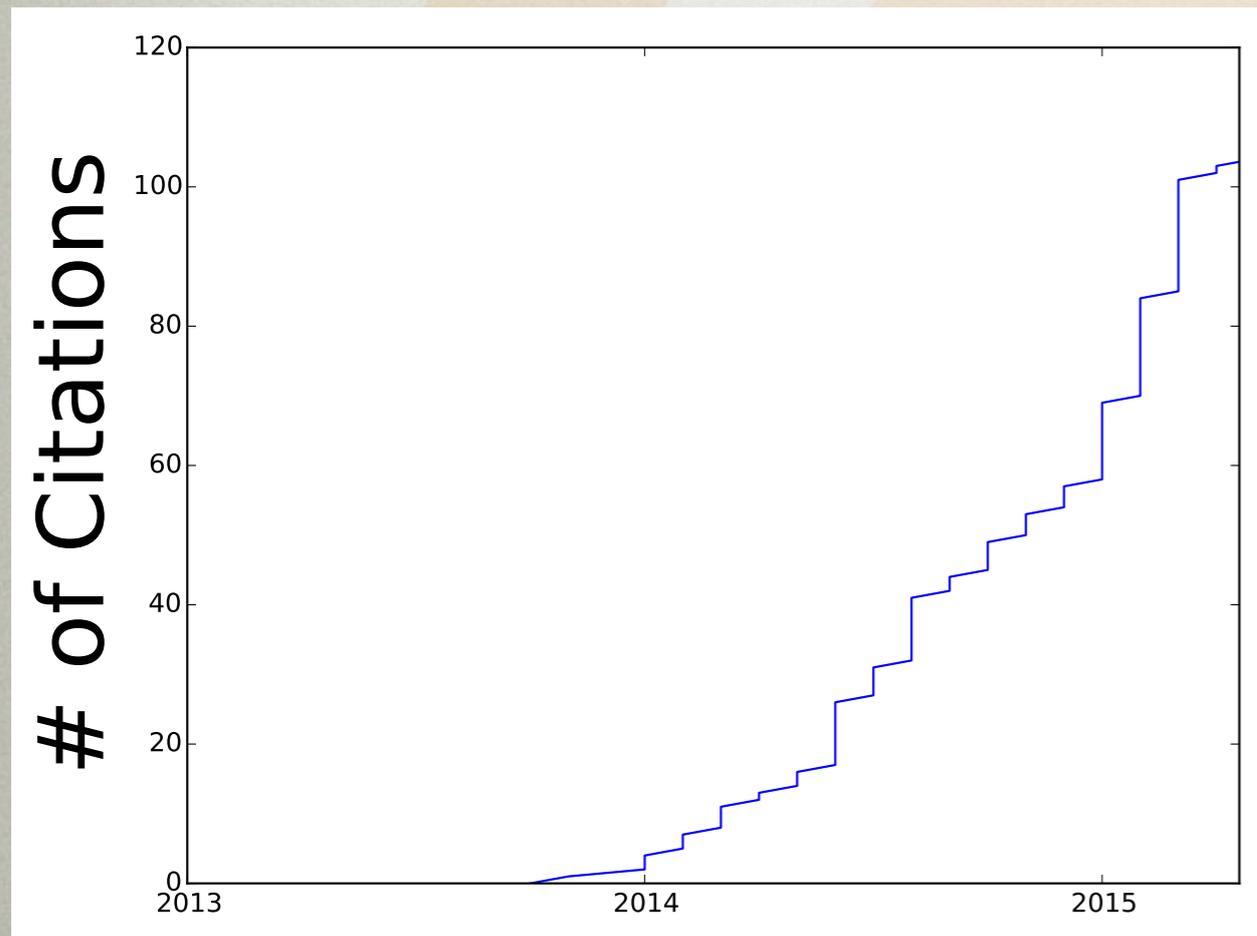
# IMPACT OF ASTROPY

Astropy Collaboration et al. 2013  
A&A 558 A33



# IMPACT OF ASTROPY

Astropy Collaboration et al. 2013  
A&A 558 A33

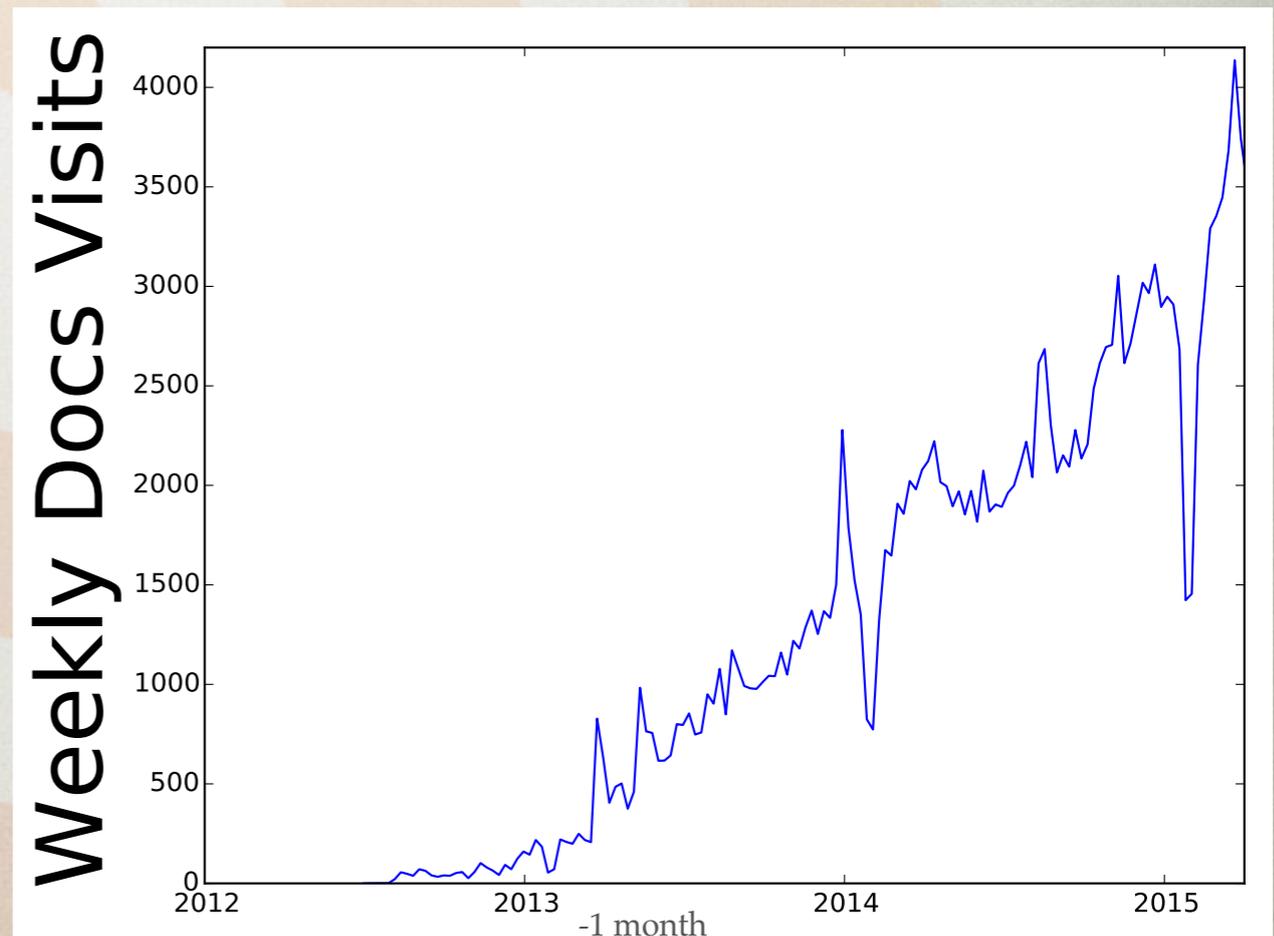
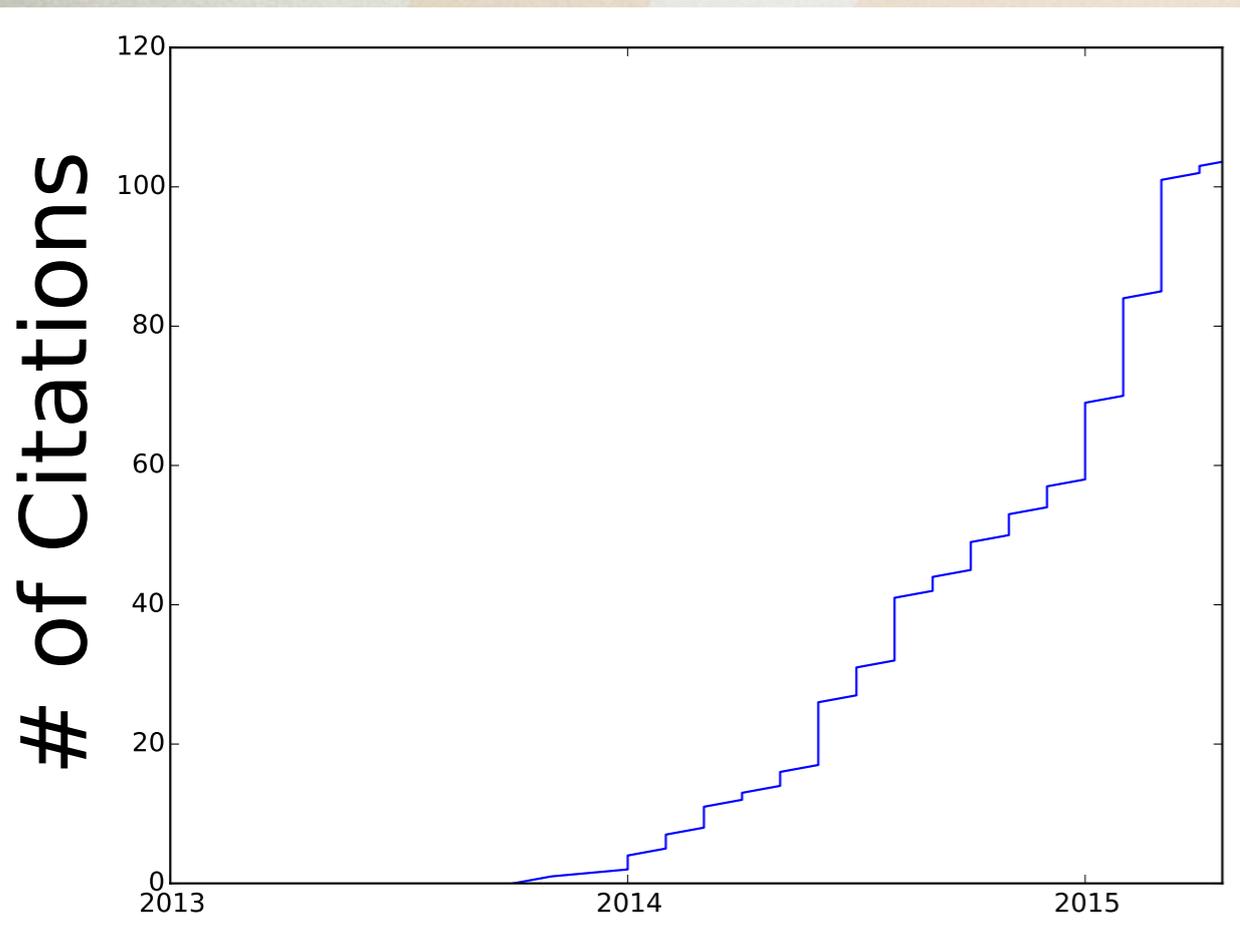


downloads 67.4k/month / 10-100 ?

# IMPACT OF ASTROPY

Astropy Collaboration et al. 2013  
A&A 558 A33

<http://docs.astropy.org>



downloads 67.4k/month / 10-100 ?

# ASTROPY'S DEVELOPMENT

Monday, 25 July, 2011 17:01:42