



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

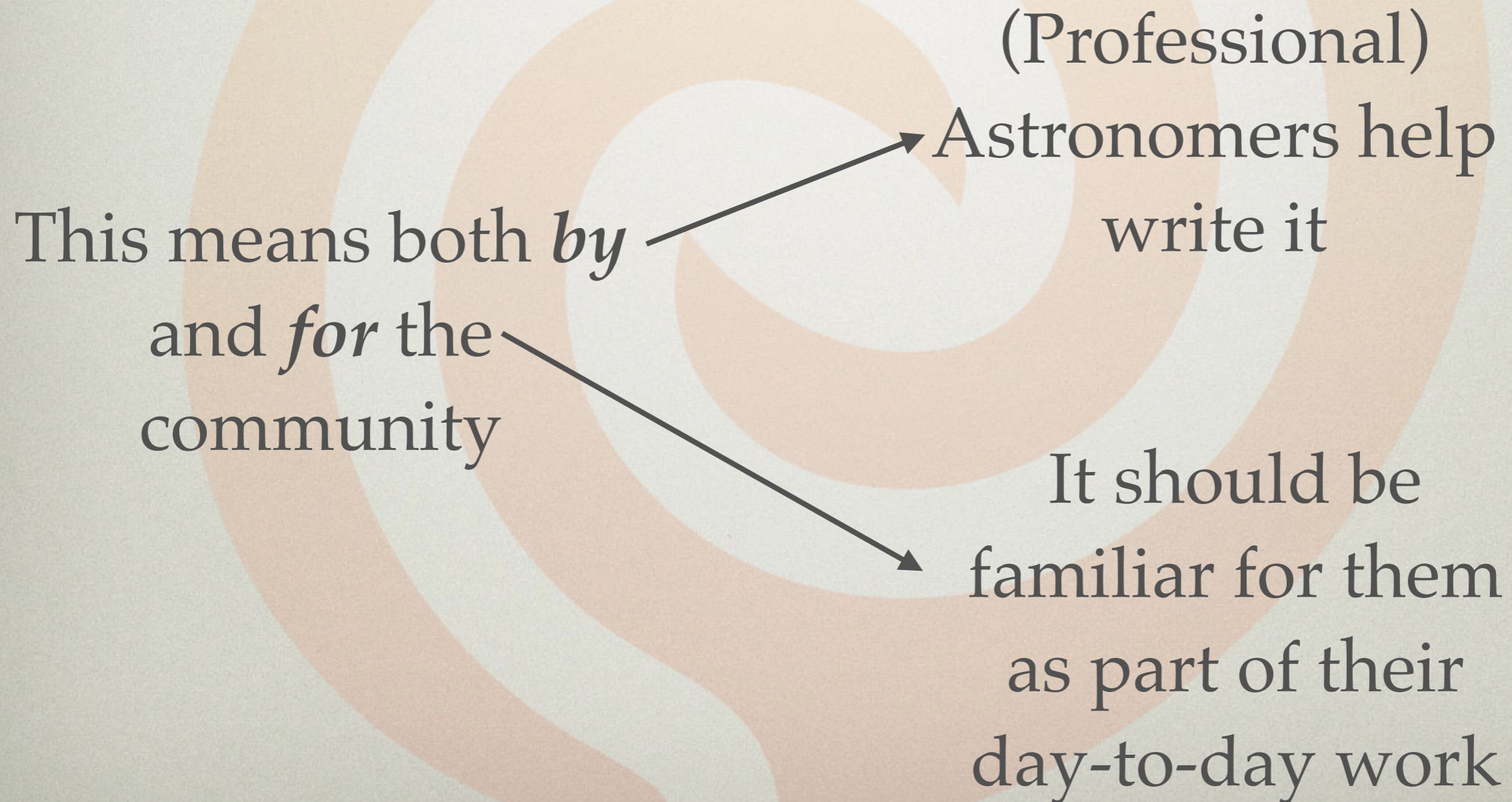
WHAT IS THE PHILOSOPHY BEHIND THE CODE?

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

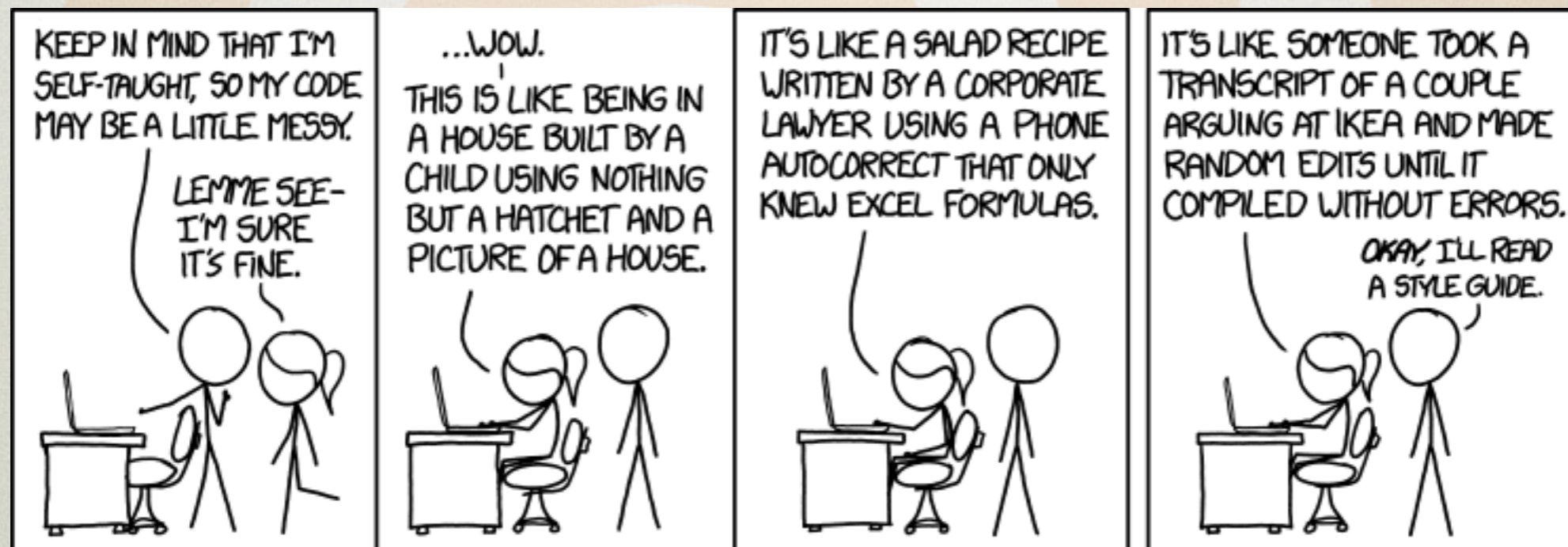
WHAT IS THE PHILOSOPHY BEHIND THE CODE?

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



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

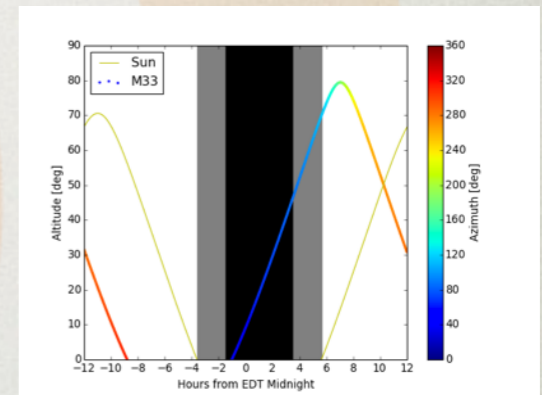
<http://docs.astropy.org>

- 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

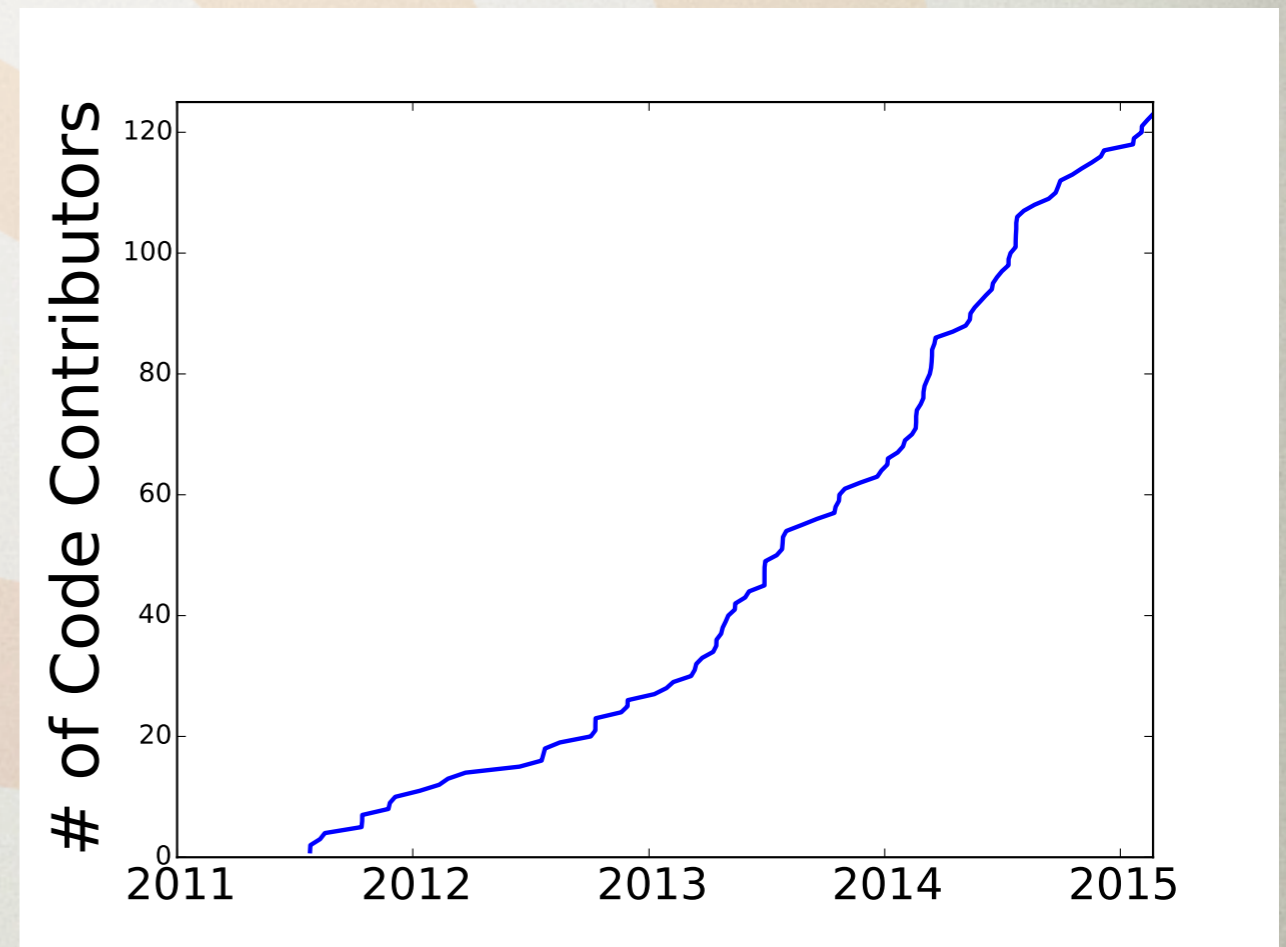
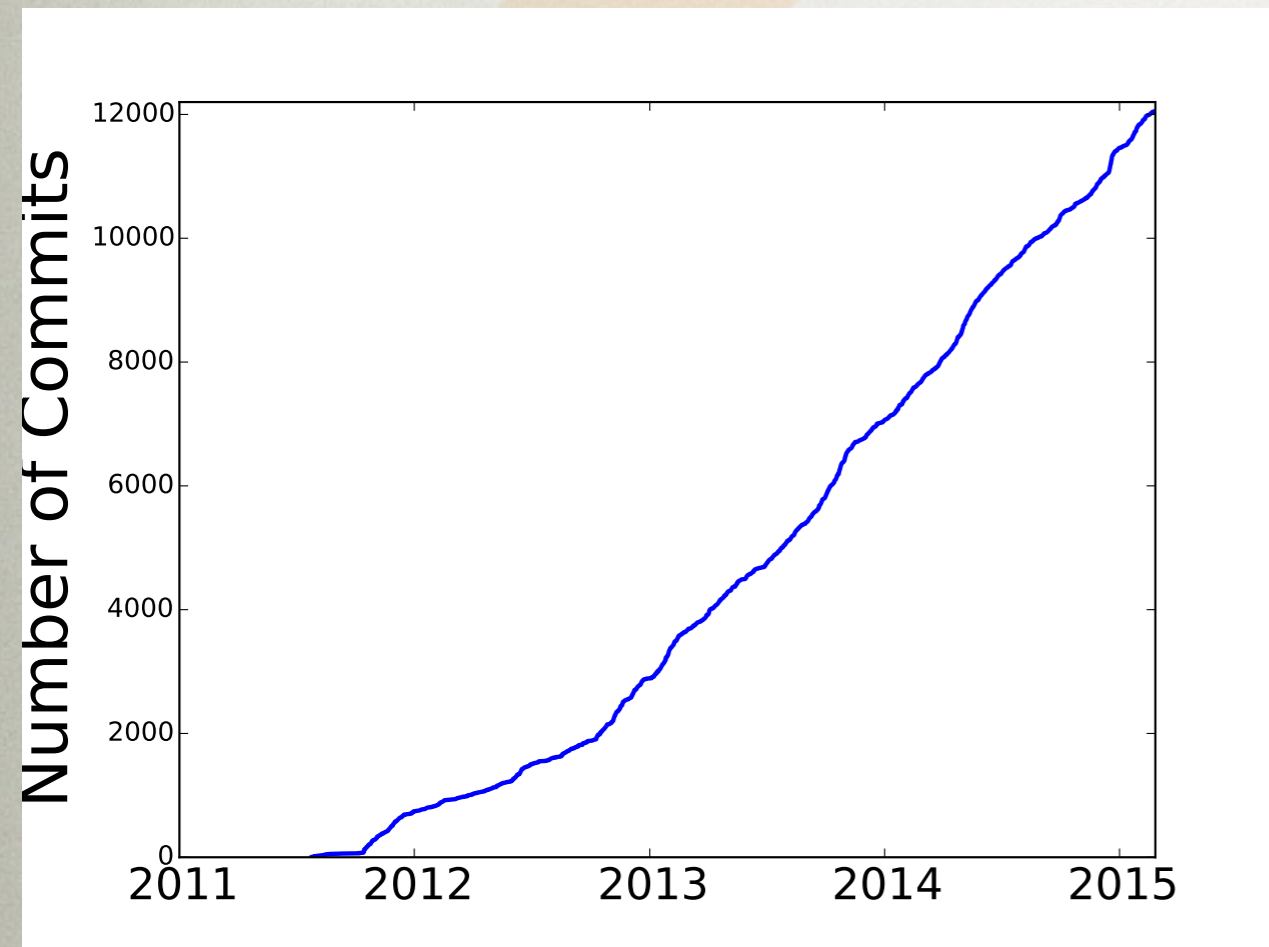
<http://tutorials.astropy.org>

<http://docs.astropy.org>

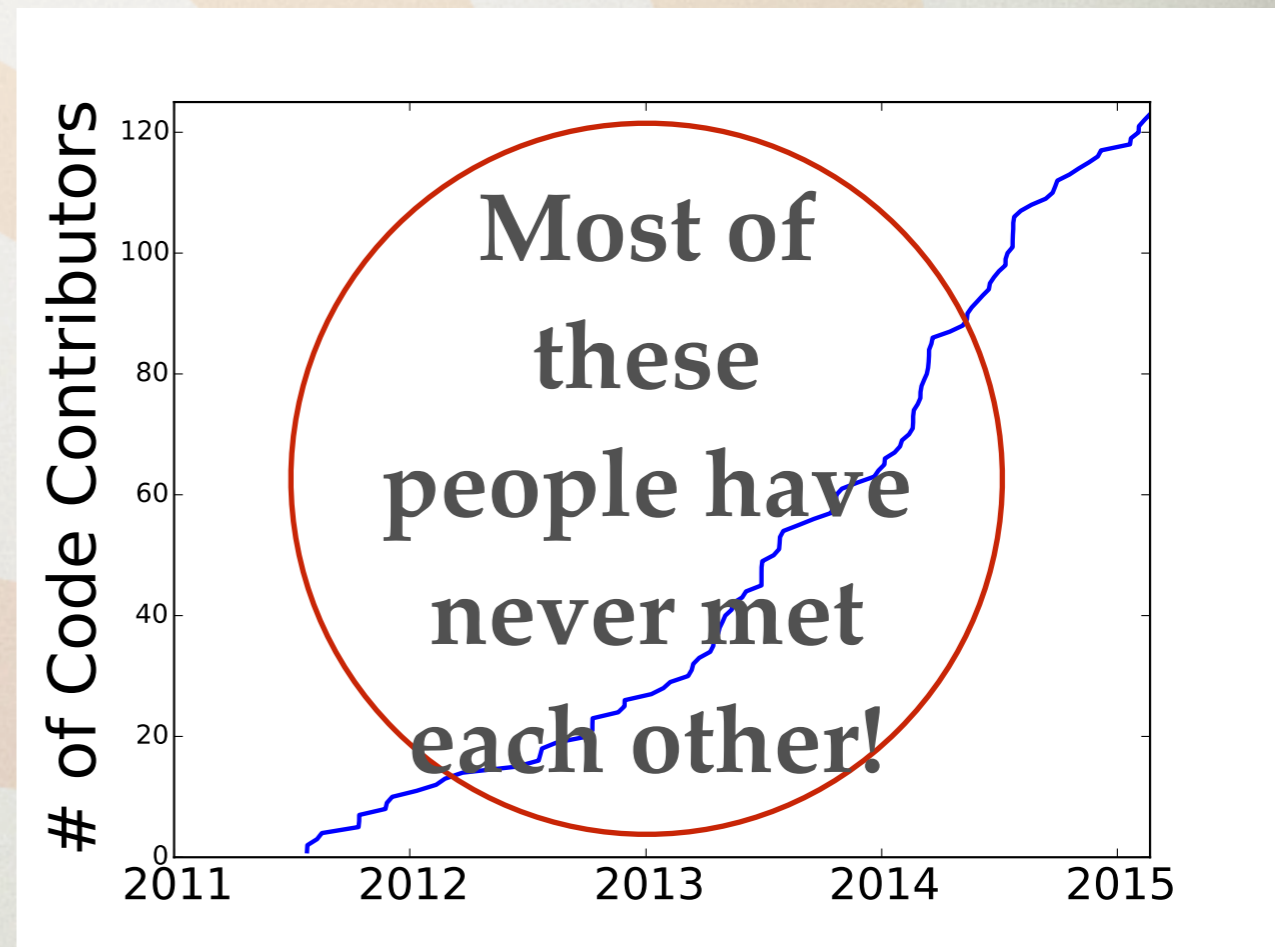
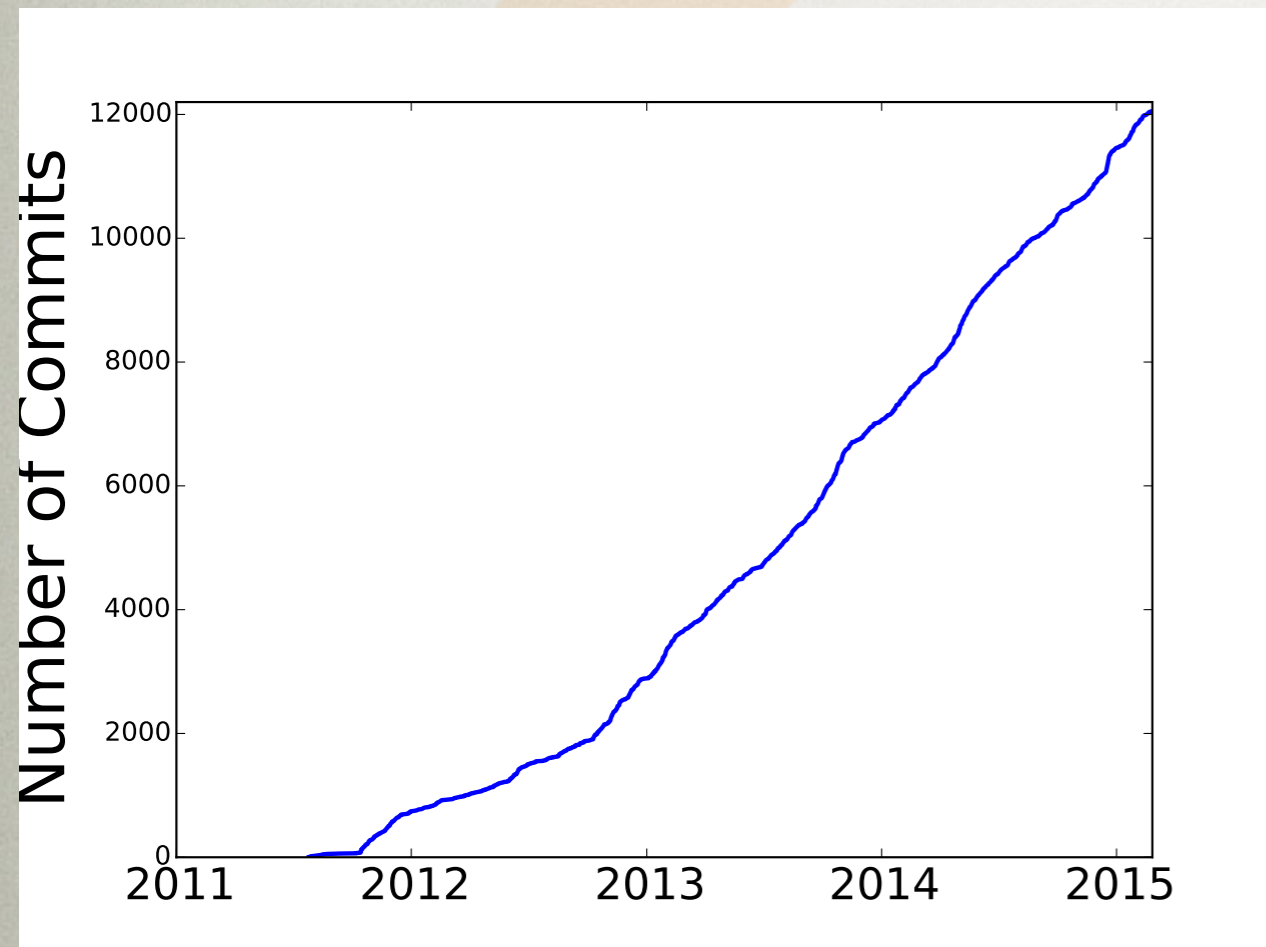
THE “BY” PART

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

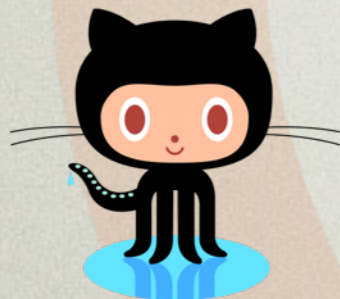



WHAT ARE THE KEY ELEMENTS?








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   

@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

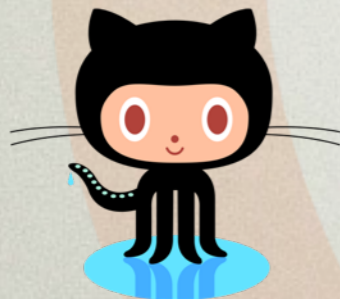
-  Unsubscribe
- You're receiving notifications because you were mentioned.


5 participants

-     

 Lock pull request


WHAT ARE THE KEY ELEMENTS?






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.

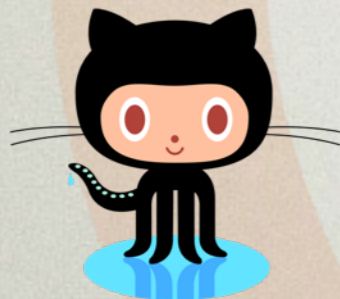
© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#)



[Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

WHAT ARE THE KEY ELEMENTS?

github
SOCIAL CODING



embray commented 26 days ago Collaborator 5 participants

@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




**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?





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.

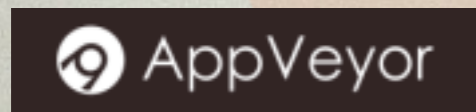
© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#)



[Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

WHAT ARE THE KEY ELEMENTS?

py.test



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.

© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#)  [Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

WHAT ARE THE KEY ELEMENTS?



SPHINX

PYTHON DOCUMENTATION GENERATOR



Read the Docs

Create, host, and browse documentation.

astro.py:docs

astro.py Index Modules Search

Astropy v1.0.1 » Data Tables ([astropy.table](#)) « previous next »

Page Contents

- Data Tables ([astropy.table](#))
- Introduction
- Getting Started
- Using [table](#)
 - Construct table
 - Access table
 - Modify table
 - Table operations
 - Masking
 - I/O with tables
 - Mixin columns
 - Implementation
- Reference/API
 - [astropy.table](#) Module
 - Functions
 - Classes
 - Class Inheritance Diagram

Data Tables ([astropy.table](#))

Introduction

[astropy.table](#) provides functionality for storing and manipulating heterogeneous tables of data in a way that is familiar to [numpy](#) users. A few notable features of this package are:

- Initialize a table from a wide variety of input data structures and types.
- Modify a table by adding or removing columns, changing column names, or adding new rows of data.
- Handle tables containing missing values.
- Include table and column metadata as flexible data structures.
- Specify a description, units and output formatting for columns.
- Interactively scroll through long tables similar to using [more](#).
- Create a new table by selecting rows or columns from a table.
- Perform [Table operations](#) like database joins and concatenation.
- Manipulate multidimensional columns.
- Methods for [Reading and writing Table objects](#) to files
- Hooks for [Subclassing Table](#) and its component classes

Currently [astropy.table](#) is used when reading an ASCII table using [astropy.io.ascii](#). Future releases of AstroPy are expected to use the [Table](#) class for other subpackages such as [astropy.io.votable](#) and [astropy.io.fits](#).

Note

Starting with version 1.0 of astropy the internal implementation of the [Table](#) class changed so that it no longer uses numpy structured arrays as the core table data container. Instead the table is stored as a collection of individual column objects. For most users there is NO CHANGE to the interface and behavior of [Table](#) objects. The page on [Table implementation change in 1.0](#) provides details about the change. This includes discussion of the table architecture, key differences, and benefits of the change.

Getting Started

The basic workflow for creating a table, accessing table elements, and modifying the table is shown below. These examples show a very simple case, while the full [astropy.table](#) documentation is available from the [Using table](#) section.

First create a simple table with three columns of data named `a`, `b`, and `c`. These columns have integer, float, and string values respectively:

```
>>> 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'})
```

v: stable

WHAT ARE THE KEY ELEMENTS?



SPHINX

PYTHON DOCUMENTATION GENERATOR

```
63 class FLRW(Cosmology):
64     """ A class describing an isotropic and homogeneous
65     (Friedmann-Lemaître-Robertson-Walker) cosmology.
66
67     This is an abstract base class -- you can't instantiate
68     examples of this class, but must work with one of its
69     subclasses such as `LambdaCDM` or `wCDM`.
70
71     Parameters
72     ~~~~~
73
74     H0 : float or scalar `~astropy.units.Quantity`
75         Hubble constant at  $z = 0$ . If a float, must be in [km/sec/Mpc]
76
77     Om0 : float
78         Omega matter: density of non-relativistic matter in units of the
79         critical density at  $z=0$ . Note that this does not include
80         massive neutrinos.
81
82     Ode0 : float
83         Omega dark energy: density of dark energy in units of the critical
84         density at  $z=0$ .
85
86     Tcmb0 : float or scalar `~astropy.units.Quantity`
87         Temperature of the CMB  $z=0$ . If a float, must be in [K]. Default: 2.725.
88         Setting this to zero will turn off both photons and neutrinos (even
89         massive ones)
90
91     Neff : float
92         Effective number of Neutrino species. Default 3.04.
93
94     m_nu : `~astropy.units.Quantity`
95         Mass of each neutrino species. If this is a scalar Quantity, then all
96         neutrino species are assumed to have that mass. Otherwise, the mass of
97         each species. The actual number of neutrino species (and hence the
98         number of elements of m_nu if it is not scalar) must be the floor of
99         Neff. Usually this means you must provide three neutrino masses unless
100         you are considering something like a sterile neutrino.
101
102     name : str
103         Optional name for this cosmological object.
104
105     Ob0 : float
106         Omega baryons: density of baryonic matter in units of the critical
107         density at  $z=0$ .
108
109     Notes
110     ~~~~~
111     Class instances are static -- you can't change the values
112     of the parameters. That is, all of the attributes above are
113     read only.
114     """
115     def __init__(self, H0, Om0, Ode0, Tcmb0=2.725, Neff=3.04,
116                  m_nu=u.Quantity(0.0, u.eV), name=None, Ob0=None):
117
118         # all densities are in units of the critical density
119         self._Om0 = float(Om0)
120         if self._Om0 < 0.0:
121             raise ValueError("Matter density can not be negative")
122         self._Ode0 = float(Ode0)
123         if Ob0 is not None:
124             self._Ob0 = float(Ob0)
125             if self._Ob0 < 0.0:
```

```
class astropy.cosmology.FLRW(H0, Om0, Ode0, Tcmb0=2.725, Neff=3.04, m_nu=<Quantity 0.0 eV>,
name=None, Ob0=None) [edit on github][source]
```

Bases: `astropy.cosmology.core.Cosmology`

A class describing an isotropic and homogeneous (Friedmann-Lemaître-Robertson-Walker) cosmology.

This is an abstract base class – you can't instantiate examples of this class, but must work with one of its subclasses such as `LambdaCDM` or `wCDM`.

Parameters: `H0` : float or scalar `Quantity`

Hubble constant at $z = 0$. If a float, must be in [km/sec/Mpc]

`Om0` : float

Omega matter: density of non-relativistic matter in units of the critical density at $z=0$.

`Ode0` : float

Omega dark energy: density of dark energy in units of the critical density at $z=0$.

`Tcmb0` : float or scalar `Quantity`

Temperature of the CMB $z=0$. If a float, must be in [K]. Default: 2.725. Setting this to zero will turn off both photons and neutrinos (even massive ones)

`Neff` : float

Effective number of Neutrino species. Default 3.04.

`m_nu` : `Quantity`

Mass of each neutrino species. If this is a scalar Quantity, then all neutrino species are assumed to have that mass. Otherwise, the mass of each species. The actual number of neutrino species (and hence the number of elements of `m_nu` if it is not scalar) must be the floor of `Neff`. Usually this means you must provide three neutrino masses unless you are considering something like a sterile neutrino.

`name` : str

Optional name for this cosmological object.

`Ob0` : float

Omega baryons: density of baryonic matter in units of the critical density at $z=0$.

Notes

Class instances are static – you can't change the values of the parameters. That is, all of the attributes above are read only.

Attributes Summary

`H0` Return the Hubble constant as an `Quantity` at $z=0$

v: stable

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?



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'. At the top, it indicates 173 commits, 1 branch, 2 releases, and 13 contributors. A pull request #112 from 'eteq/add-badge' is highlighted. Below this, a table lists recent commits with details like the author, message, and time ago. The 'README.rst' file is open, showing the title 'Astropy affiliated package template', a 'powered by AstroPy' badge, and a description of the template's purpose. It also includes links for more information and a 'Status reports for developers' section showing a 'build passing' status.

astropy / package-template

Unwatch 9 Unstar 13 Fork 22

Template for Astropy affiliated packages. Maintainer: @astrofrog — Edit

173 commits 1 branch 2 releases 13 contributors

branch: master package-template / +

Merge pull request #112 from eteq/add-badge

astrofrog authored on Jan 22 latest commit d2395942dc

astrofy_helpers @ 5fd32d0	Updated astropy-helpers to patched v0.4.3	6 months ago
cextern	Clarify purpose of cextern directory.	10 months ago
docs	explain readthedocs	9 months ago
licenses	Simplify package-template usage and updating by requiring fewer files to	a year ago
packagename	Ensure that conftest.py change works with all versions of Astropy	5 months ago
.gitignore	Use setuptools entry_points for command line scripts	3 months ago
.gitmodules	Updating to the actual astropy-helpers	11 months ago
.travis.yml	Drop numpy 1.5 testing and use numpy 1.9 as baseline	3 months ago
MANIFEST.in	Update MANIFEST.in [skip ci]	3 months ago
README.rst	add powered by astropy badge	4 months ago
TEMPLATE_CHANGES.md	Update TEMPLATE_CHANGES [skip ci]	3 months ago
ah_bootstrap.py	Updated astropy-helpers to patched v0.4.3	6 months ago
ez_setup.py	Update to newer ez_setup.py	a year ago
setup.cfg	Update the astropy_helpers version again; use the auto_use feature of...	a year ago
setup.py	Use setuptools entry_points for command line scripts	3 months ago

README.rst

Astropy affiliated package template

powered by AstroPy

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](#)
- [This template's Github code repository](#)

Status reports for developers

build passing

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

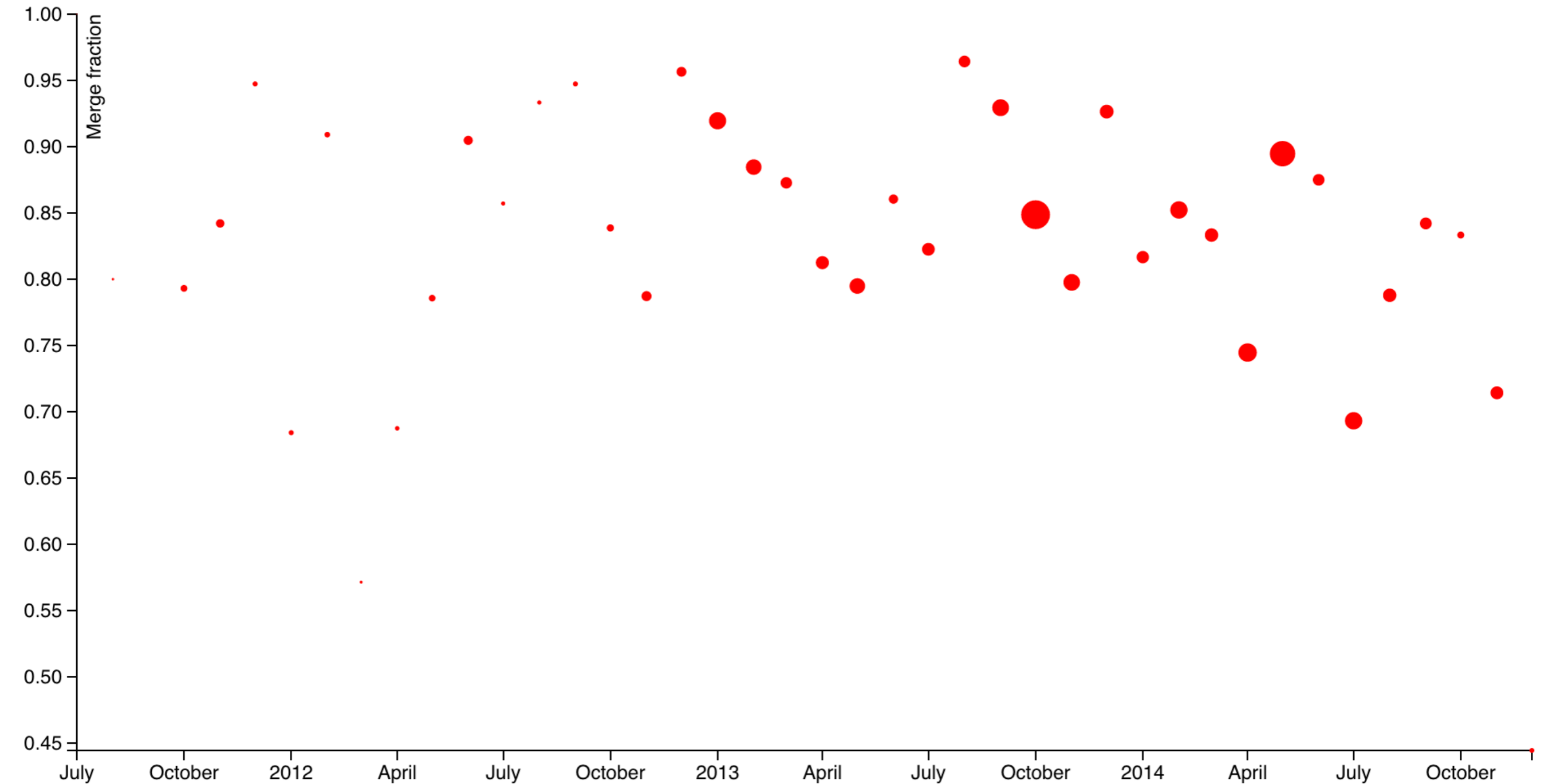
“I don’t see why I should care. My
code doesn’t have units.”

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

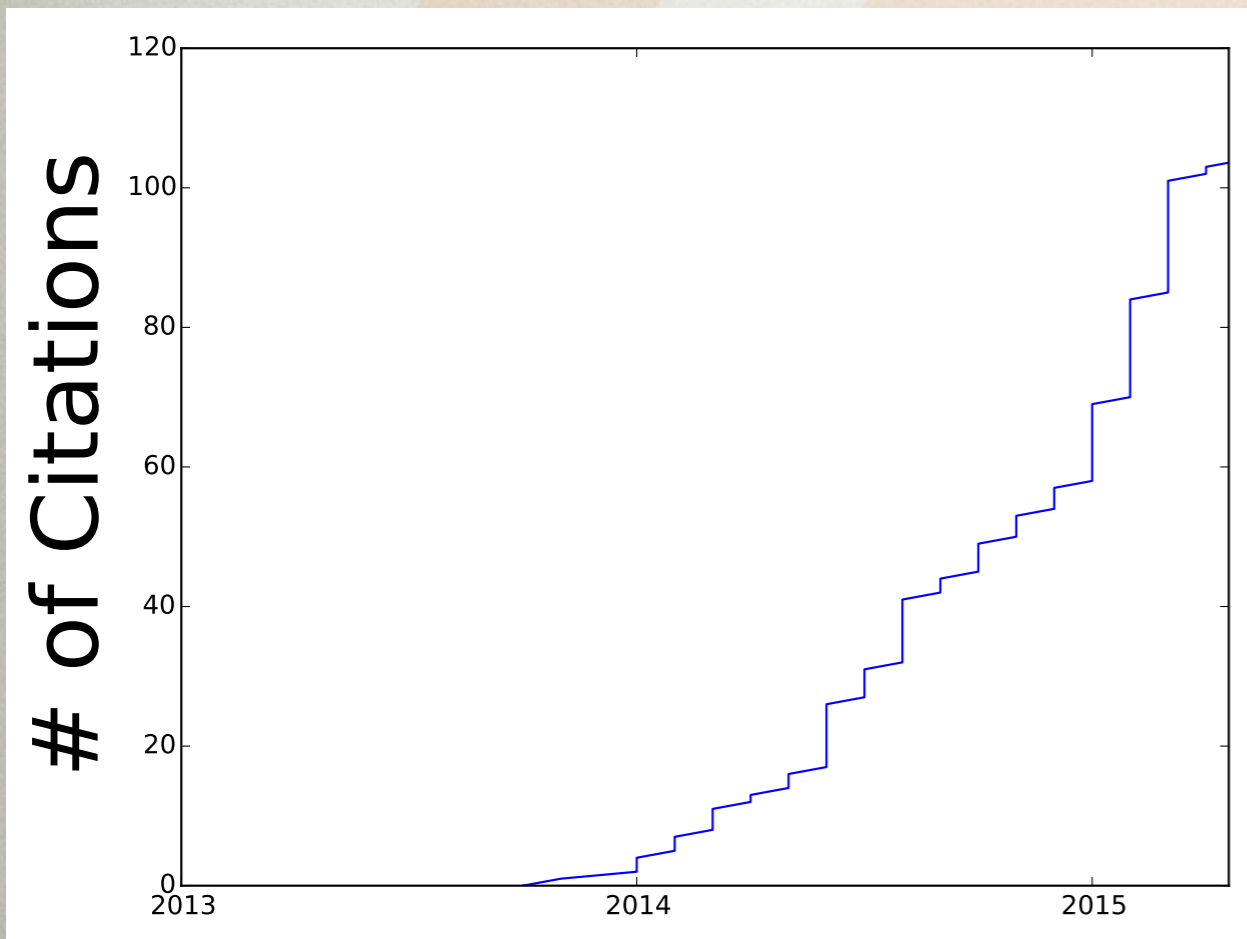
- Shailesh Ahuja
- Tom Aldcroft
- Kyle Barbary
- Geert Barentsen
- Paul Barrett
- Andreas Baumbach
- Chris Beaumont
- Daniel Bell
- Francesco Biscani
- Thompson Le Blanc
- Christopher Bonnett
- Joseph Jon Booker
- Médéric Boquien
- Azalee Bostroem
- Matthew Bourque
- Larry Bradley
- Gustavo Bragança
- 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 Paoro
- 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?)

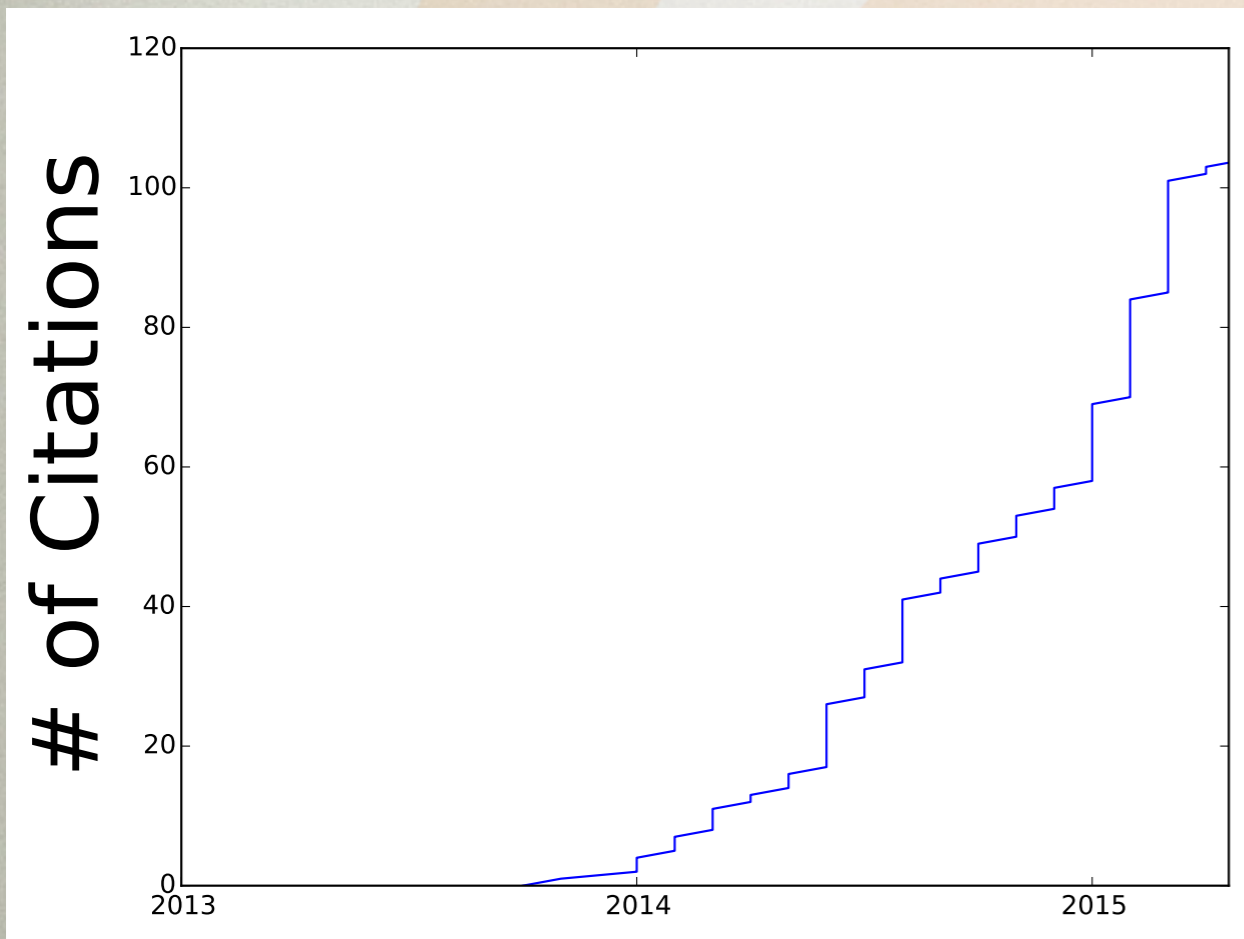
IMPACT OF ASTROPY

Astropy Collaboration et al. 2013
A&A 558 A33



IMPACT OF ASTROPY

Astropy Collaboration et al. 2013
A&A 558 A33

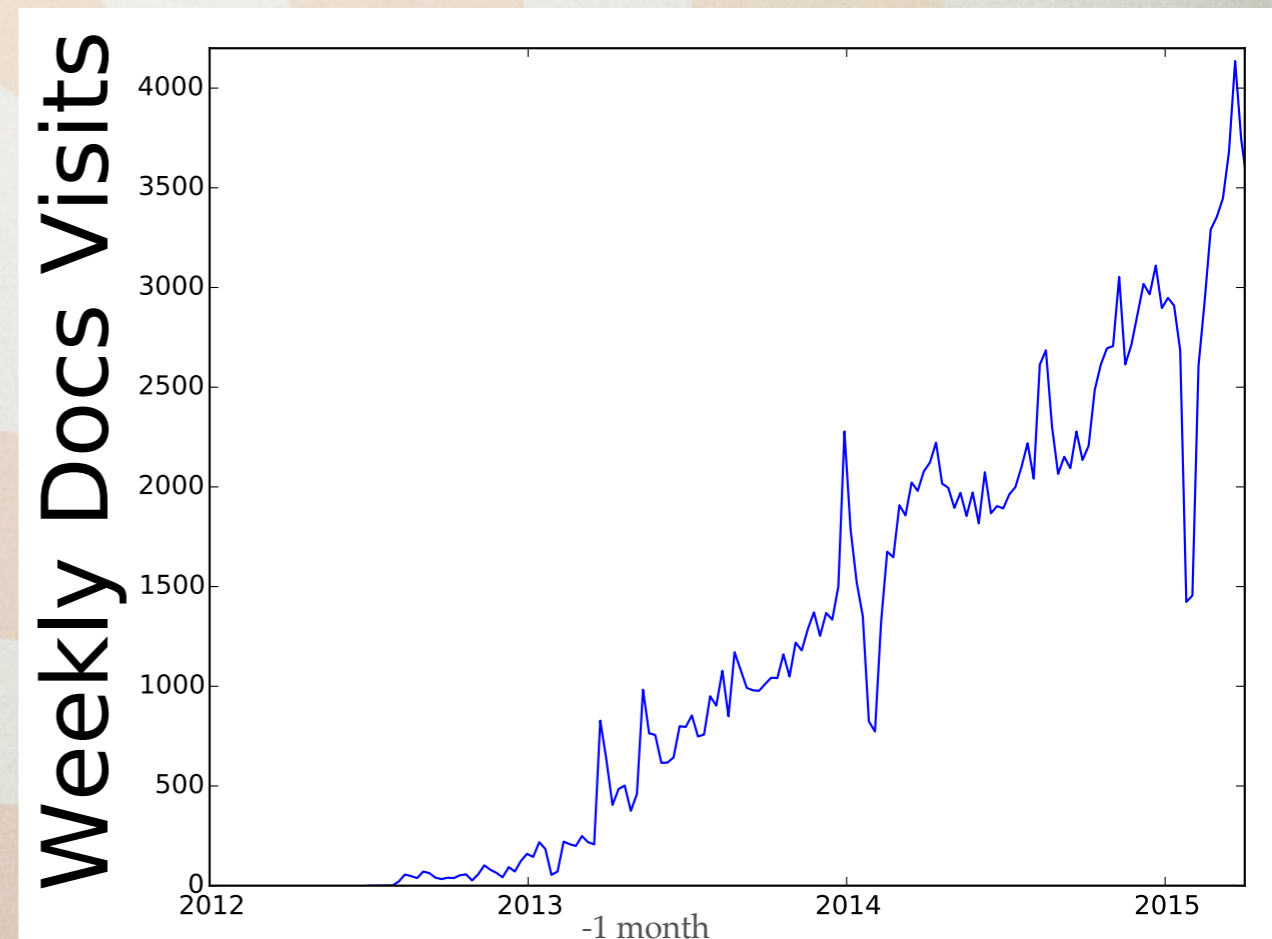
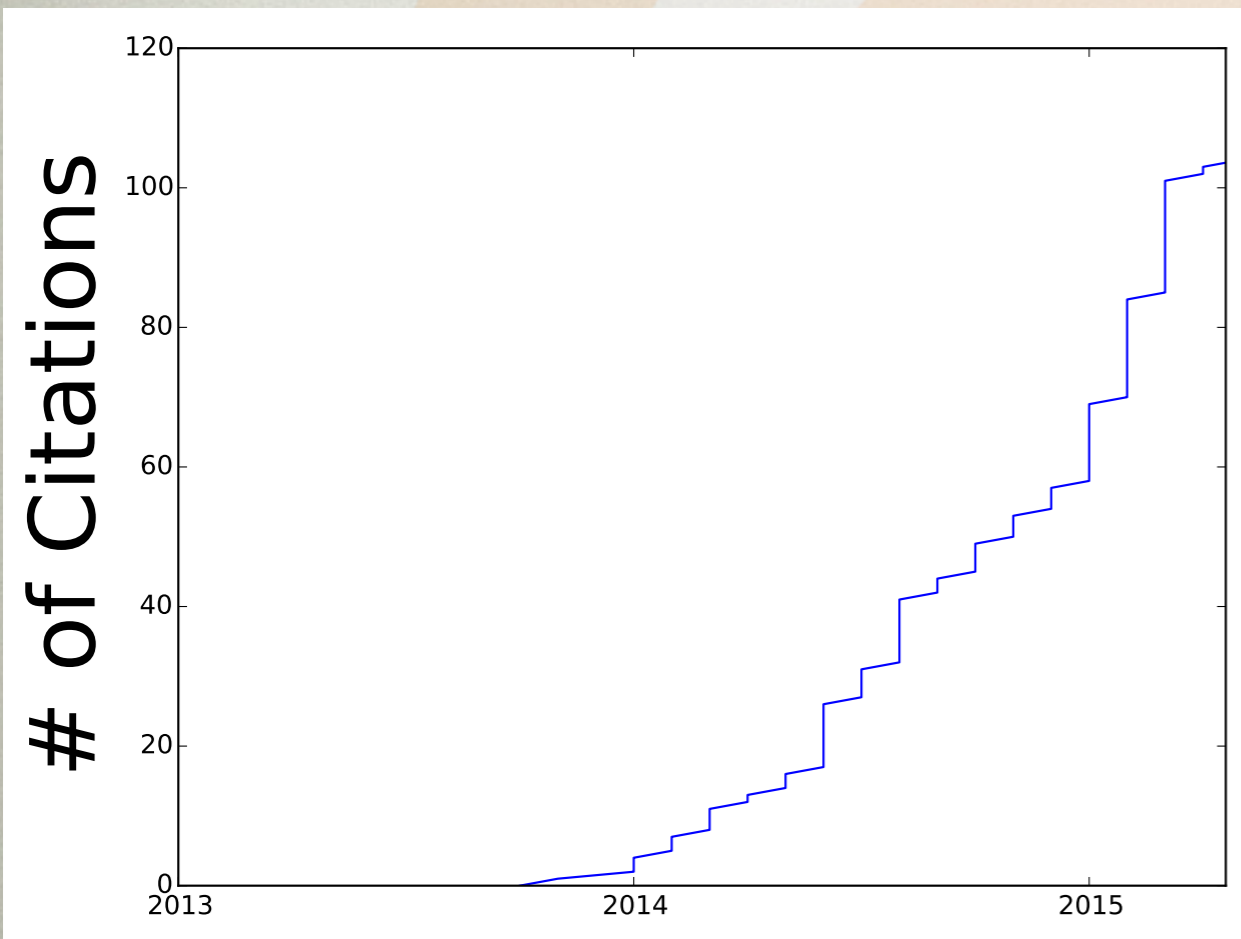


downloads 67.4k/month / 10-100 ?

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