

# **Open source** software and **Python** for gamma-ray astronomy

Christoph Deil, Nov 16, 2015 @ PyGamma15



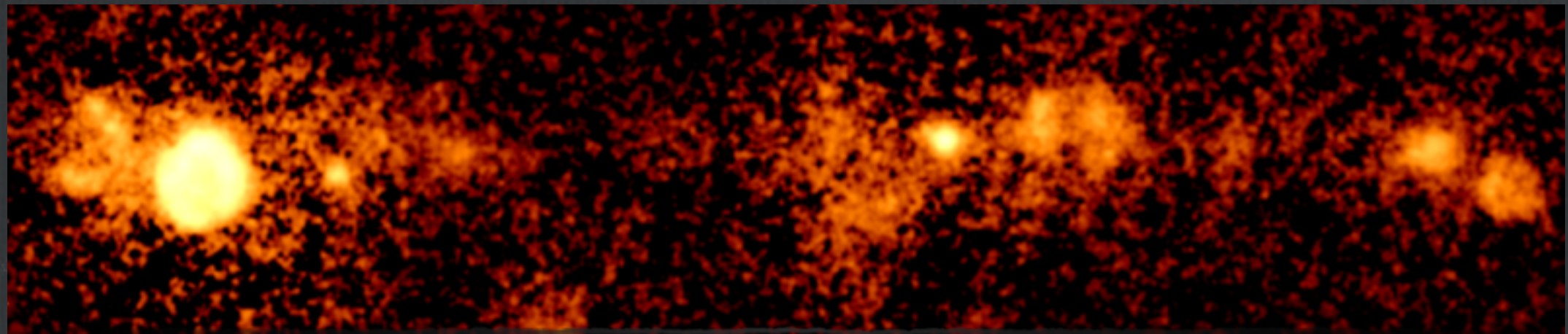
MAX-PLANCK-INSTITUT  
FÜR KERNPHYSIK



# About me

---

- Postdoc at MPIK, working on HESS data and software



- Pascal → C → C++ → Python → ?

**Jake Vanderplas, Astronomer**



SEP 20, 2012

# **Why Python is the Last Language You'll Have To Learn**

<http://jakevdp.github.io/blog/2012/09/20/why-python-is-the-last/>



# **Open source and free software**



# What exactly does “open” and “free” mean?

---

open

See presentation by Axel

☐ **Open development**

Discussion is done in the open and anyone can join in!

Examples: Linux?, ROOT? Python, Scientific Python packages, Astropy, ...

☐ **Open Source / Free Software**

License recognised by the open source initiative and the free software foundation

Examples: ds9, IRAF, CIAO?, Fermi Science Tools?, ERFA

☐ **Source available, but not free**

Codes with non-free license: Numerical recipes, CORSIKA?, IAU SOFA

But also any code without a license!

☐ **Closed source**

Source code not freely available.

Commercial software: Microsoft, Apple, Confluence, Mathematica, IDL, Matlab

Internal software: Banks, NASA, HESS, VERITAS, MAGIC analysis software

closed

# What's a license?

---

- ☐ A software license is a legally-binding agreement which governs the use and redistribution of software.
- ☐ All common open source / free software licences say this:
  - ☐ Users have the right to freely use, copy, share, modify.
  - ☐ As long as you provide attribution back and don't hold the author liable.



# Need a license?

---

- ☐ *“If you stumble across some code with no attached licensing information, copyright laws would have you treat it as ‘all privileges retained’, even if its author in fact was just trying to make it available with no strings attached.”  
— Arto Bendican*
- ☐ **Ask your colleagues to add a license.**
- ☐ **Always license your code when you share it!**
- ☐ **Always use a license that is recognised as “open source” ([opensource.org](http://opensource.org)) and “free” software ([fsf.org](http://fsf.org))!**

<http://www.astrobetter.com/blog/2014/03/10/the-whys-and-hows-of-licensing-scientific-code/>

# Which license?

---

- ☐ **Choosing a license can be hard:**
  - ☐ **What do you want to permit and forbid?**
  - ☐ **What licenses do packages / people use that you want to collaborate with (i.e. share code with)?**



# Let's oversimplify and say there's two classes of licenses

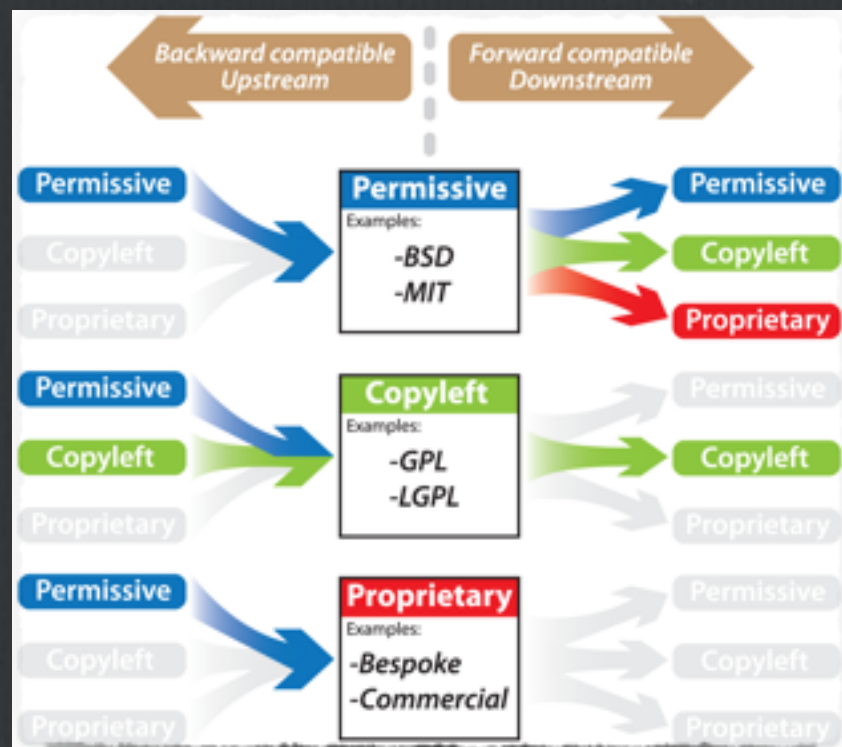
---

- ☐ “Permissive” or “Liberal”  
Examples – MIT, BSD
- ☐ “Copyleft” or “Restrictive”  
Examples – GPL v2 or v3, LGPL
- ☐ All these licenses are “open source” ([opensource.org](http://opensource.org)) and “free” software ([fsf.org](http://fsf.org)) and say this:
  - ☐ Users have the right to freely use, copy, share, modify.
  - ☐ As long as you provide attribution back and don't hold the author liable



**Free Software  
Free Society**

# License compatibility



- ❑ A typical classification of open-source licenses is into “permissive” and “copyleft”.
- ❑ Copyleft licenses requires anyone who distributes your code or a derivative work to make the source available under the same terms.
- ❑ Copying code from a copyleft to a permissive package is forbidden. The other direction is OK.
- ❑ Importing a copyleft package from a permissive package can be OK if it's an optional dependency.



# **Licenses in Python and (gamma-ray) Astronomy**

---

- ☐ **“Permissive” licenses (e.g. MIT, BSD) used by:  
Python, Numpy, Scipy, matplotlib, emcee, Astropy  
Gammapy, Naima, ctapipe**
- ☐ **“Copyleft” licenses (e.g. GPL3, LGPL) used by:  
CFITSIO, WCSLIB, ROOT, LSST, Sherpa, Fermi Science  
tools?, 3ML, Gamera, D3PO, Gammalib, ctools**
- ☐ **Is license compatibility a real issue for us?**

# Python



# Python origin

---

- ❑ Created by Guido van Rossum ~ 1990
- ❑ “My initial goal for Python was to serve as a second language for people who were C or C++ programmers, but who had work where writing a C program was just not effective.”
- ❑ “Bridge the gap between the shell and C.”





# python





# Python Development

---

- ☐ **Guido van Rossum – Benevolent dictator for life (BDFL)**
- ☐ **Python software foundation (PSF)**
- ☐ **Python Enhancement Proposals (PEPs)**
- ☐ **There are other Python implementations besides CPython (IronPython, Jython, PyPy, Pyston), not relevant for us (yet?).**

# Python versions

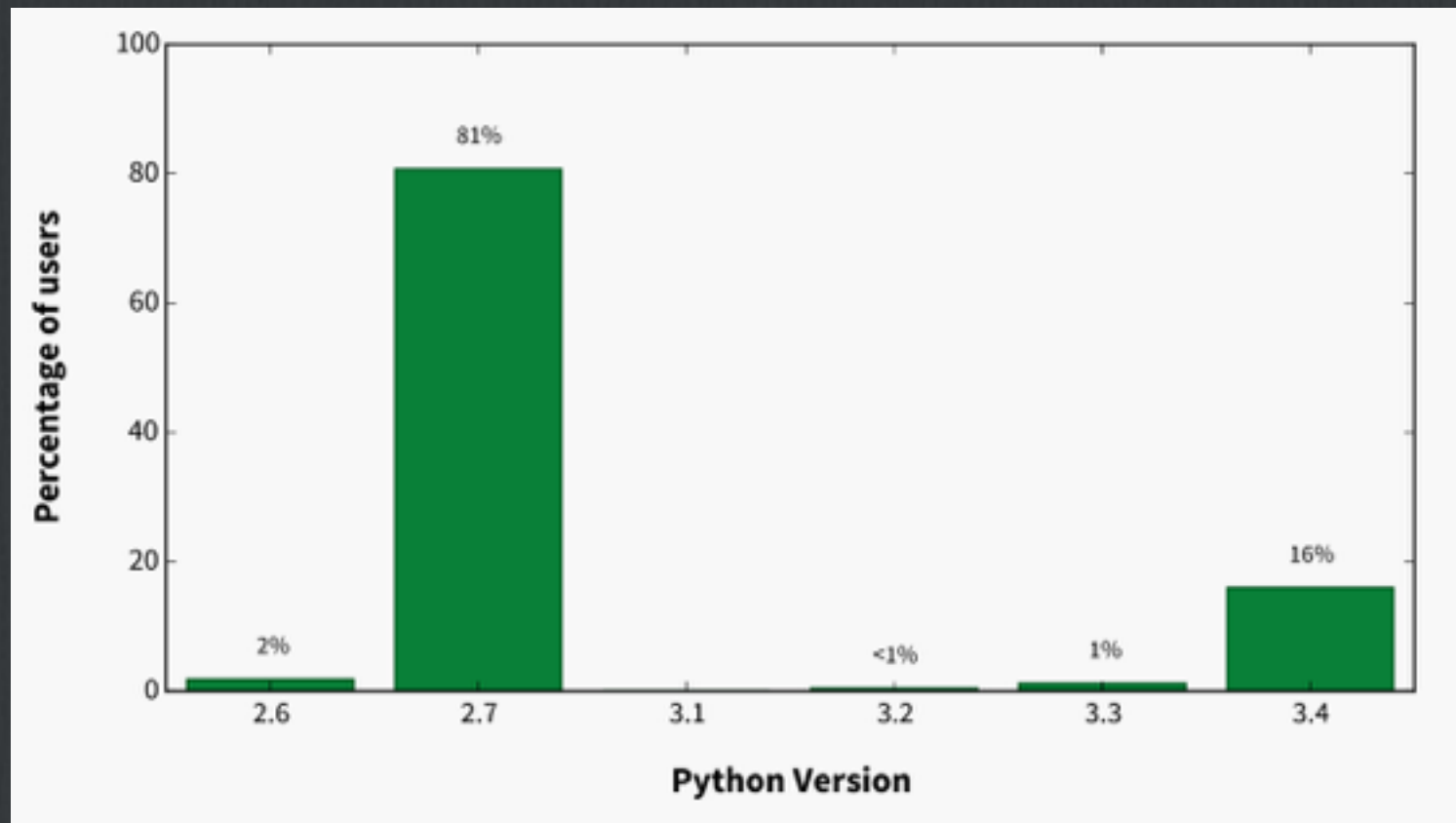
---

- ☐ Python 1.0 (1994), Python 2.0 (2000), Python 3.0 (2008)
- ☐ Python 2.7 is currently the most-used version.
- ☐ For Python 3, most people are on 3.4 (March 2014), Python 3.5 just came out (September 2015)
- ☐ Python itself and many scientific Python packages have dropped Python 2.6 and 3.0 –3.2 support



SAT 09 MAY 2015

# Python 3 in Science: the great migration has begun!



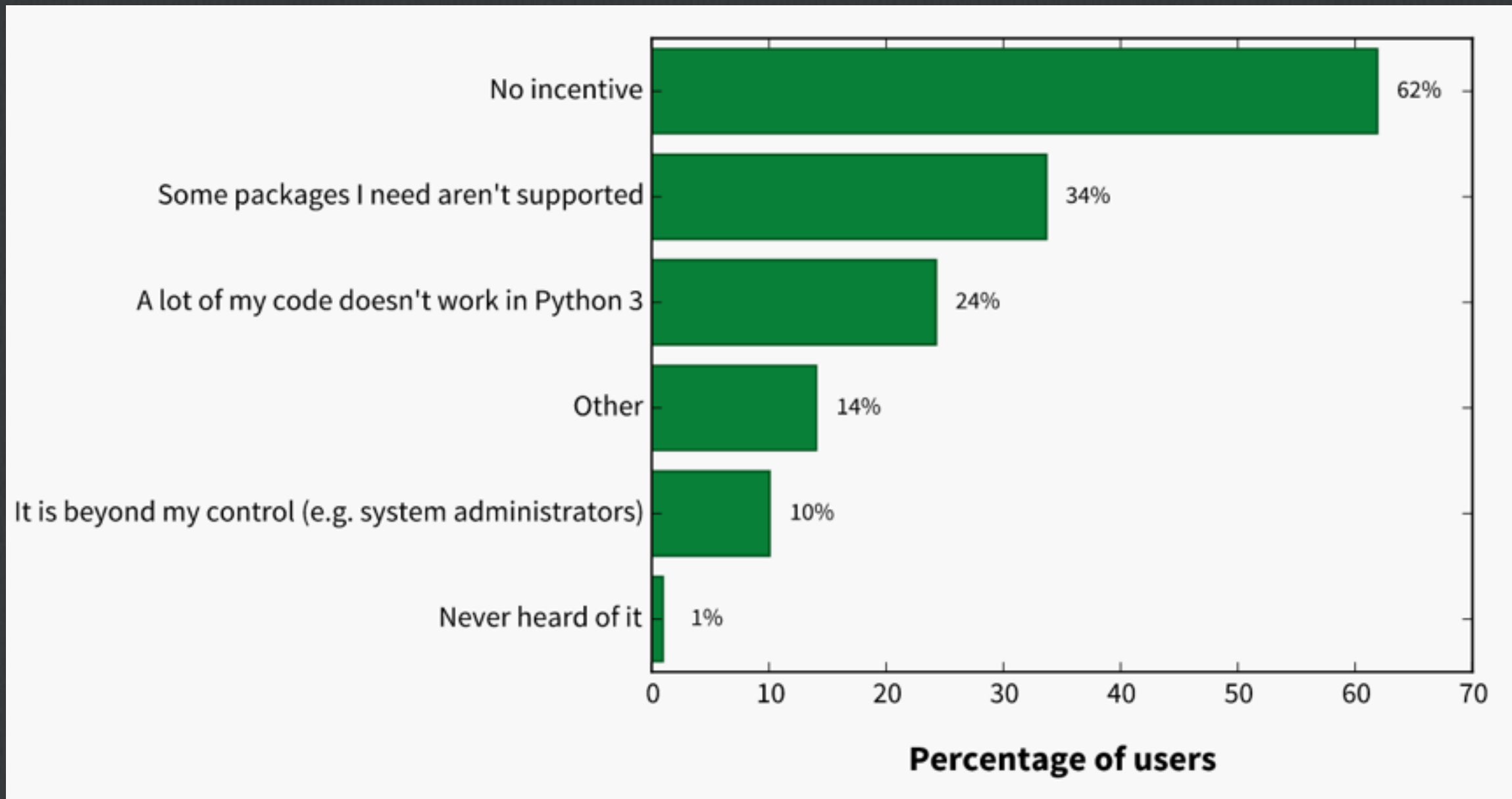
**Thomas Robitaille, MPIA**

Astropy lead, Astronomy in Python conference organiser

<http://astrofrog.github.io/blog/2015/05/09/2015-survey-results/>



# Why aren't you using Python 3?





# Why Python 3?

---

- **Better unicode text handling!**  
to keep Python relevant for web, mobile, ... apps.  
→ required backward-incompatible changes.
- **A better language!**  
→ fix mistakes and use lessons learned from Python 1 & 2.  
Many small improvements, see notes by Nick Coughlan.

# Status Python 2 → 3

---

- ❑ Transition has been going on for ~ 5 years,  
will continue for another ~ 5 years.  
(end of life for RHEL 7 with Python 2.7 is 2024)
- ❑ Most important packages work with Python 2 and 3.  
Python 2 only: Fermi science tools, Sherpa  
Python 3 only: ctapipe
- ❑ There is no doubt that Python 3 is the future.  
Python 3 is default in Fedora 23 (?) and is planned be the default in  
Ubuntu 16.04 LTS and RHEL 7.
- ❑ The migration has been / is painful, but largely successful.  
(note that Perl is dying in part because of the Perl 5 → 6 transition)



# Other things Python

---

- There's many other things to talk about (packaging, distribution, testing, documentation, alternative Python interpreters, JITs, IPython, Scientific Python, ...)
- But I'm out of time ... let's move on ...

# Python is popular

BLOG@CACM

## Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities

At the time of writing (July 2014), [Python](#) is currently the most popular language for teaching introductory computer science courses at top-ranked U.S. departments.

Specifically, eight of the top 10 CS departments (80%), and 27 of the top 39 (69%), teach Python in introductory CS0 or CS1 courses.



Massachusetts  
Institute of  
Technology



HARVARD  
UNIVERSITY



THE UNIVERSITY OF  
**TEXAS**  
— AT AUSTIN —



NYU



Carnegie  
Mellon  
University





# Python is popular

ENTERPRISE

## Python Displacing R As The Programming Language For Data Science

*R remains popular with the PhDs of data science, but as data moves mainstream, Python is taking over.*



Matt Asay on November 25, 2013

<http://readwrite.com/2013/11/25/python-displacing-r-as-the-programming-language-for-data-science>

# Python in astronomy

## SOFTWARE USE IN ASTRONOMY: AN INFORMAL SURVEY

IVELINA MOMCHEVA<sup>1</sup>, ERIK TOLLERUD<sup>1, 2</sup>

*Draft version July 16, 2015*

### ABSTRACT

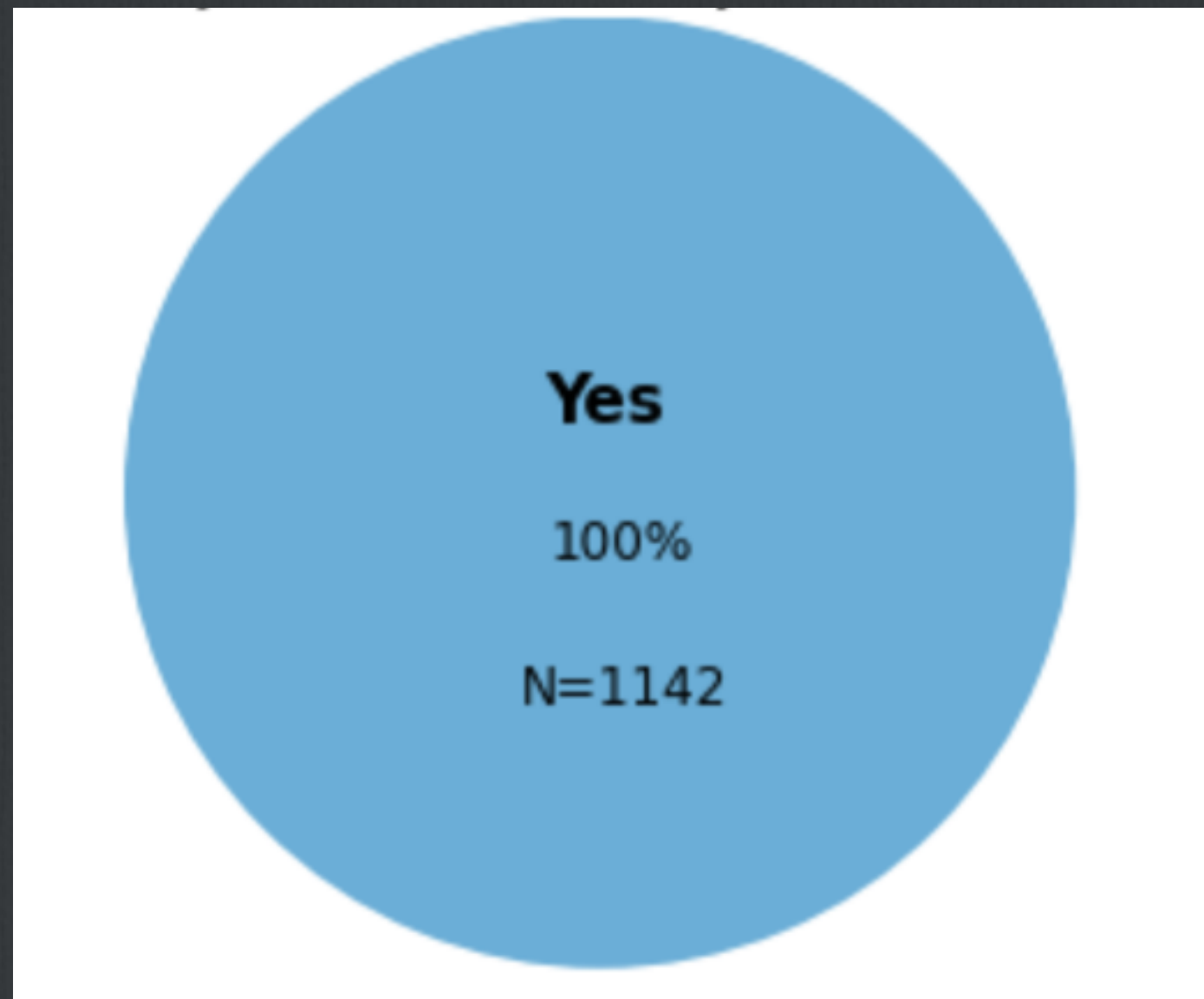
We report on an informal survey about the use of software in the worldwide astronomical community. The survey was carried out between December 2014 and February 2015, collecting responses from 1142 astronomers, spanning all career levels. We find that all participants use software in their research. The vast majority of participants, 90%, write at least some of their own software. Even though writing software is so wide-spread among the survey participants, only 8% of them report that they have received substantial training in software development. Another 49% of the participants have received “little” training. The remaining 43% have received no training. We also find that astronomers’ software stack is fairly narrow. The 10 most popular tools among astronomers are (from most to least popular): Python, shell scripting, IDL, C/C++, Fortran, IRAF, spreadsheets, HTML/CSS, SQL and Supermongo. Across all participants the most common programming language is Python ( $67 \pm 2\%$ ), followed by IDL ( $44 \pm 2\%$ ), C/C++ ( $37 \pm 2\%$ ) and Fortran ( $28 \pm 2\%$ ). IRAF is used frequently by  $24 \pm 1\%$  of participants. We show that all trends are largely independent of career stage, area of research and geographic location.

<http://adsabs.harvard.edu/abs/2015arXiv150703989M>

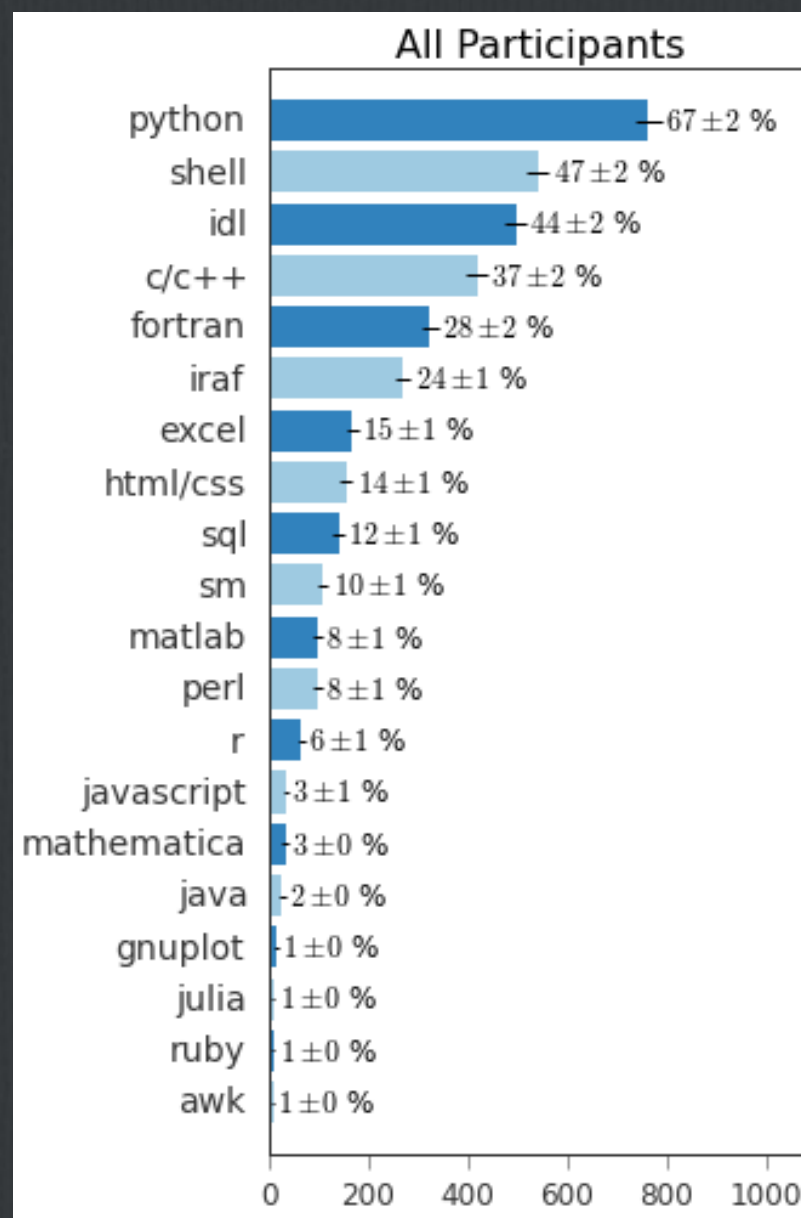


# Do you use software in your research?

---



# What do you use regularly for your research?



☐ Python #1 in astronomy

☐ Responses very similar for PhDs, postdocs, faculty



# Summary

---

- ☐ **Open source is good for science.**  
**License compatibility can be an issue.**
- ☐ **Python is #1 in data science and astronomy.**  
**It has strong and growing community.**
- ☐ **Python 3 is the future.**
- ☐ **Let's build nice Python packages and a community for gamma-ray astronomy!**  
**(this could be the first of a series of workshops over the next years)**