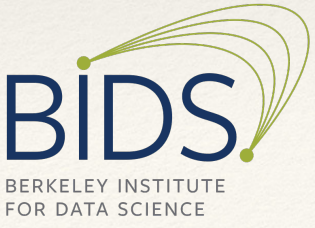
IPython & Project Jupyter

A language-independent architecture for open computing and data science

Fernando Pérez
(@fperez_org & fperez@lbl.gov)

LBL & UC Berkeley



IPython: CU Boulder, 2001

or how to best procrastinate on a Physics dissertation

/bin/bash

```
In [13]: run ~/scratch/error
reps: 5
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/home/fperez/scratch/error.py in <module>()
     70 if __name__ == '__main__':
     71     #explode()
     72
---> 72     main()
     73     g2='another global'

/home/fperez/scratch/error.py in main()
     60     array_num = zeros(size,'d')
     61     for i in xrange(reps):
---> 62         RampNum(array_num, size, 0.0, 1.0)
     63     RNtime = time.clock()-t0
     64     print 'RampNum time:', RNtime

/home/fperez/scratch/error.py in RampNum(result, size, start, end)
     43     tmp = zeros(size+1)
     44     step = (end-start)/(size-1-tmp)
---> 45     result[:] = arange(size)*step + start
     46
     47 def main():

ValueError: shape mismatch: objects cannot be broadcast to a single shape

In [14]:
```

November 2001: "Just an afternoon hack"

- 259 Line Python script.
- `sys.ps1` -> `In [N]`.
- `sys.displayhook` -> `Out[N]`, caches results.
- Plotting, Numeric, etc.

Now (Openhub stats)

- 19,279 commits
- 442 contributors
- Total Lines: 187,326
- Number of Languages : 7 (JS, CSS, HTML, ...)

Real credit goes to whole team

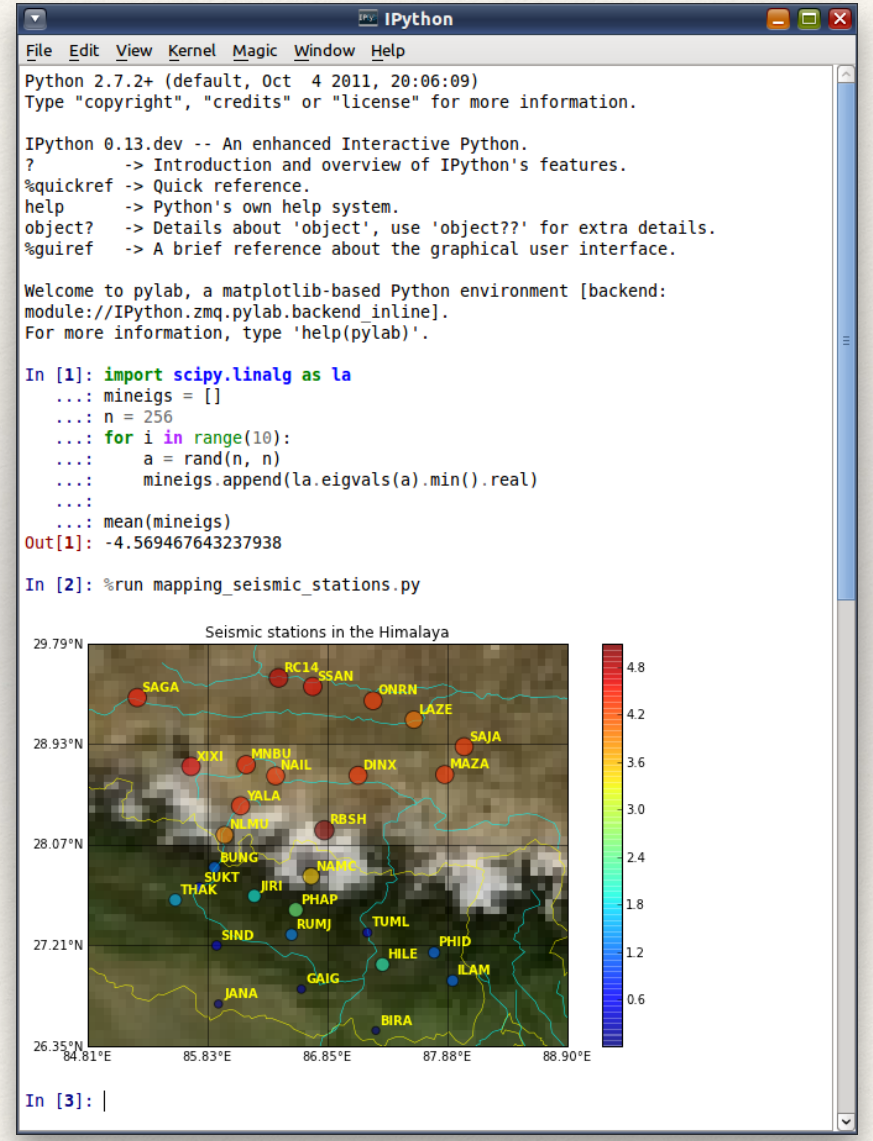
Plus ~ 500 more Open source contributors!

Current and recent funding

Beyond the Terminal…

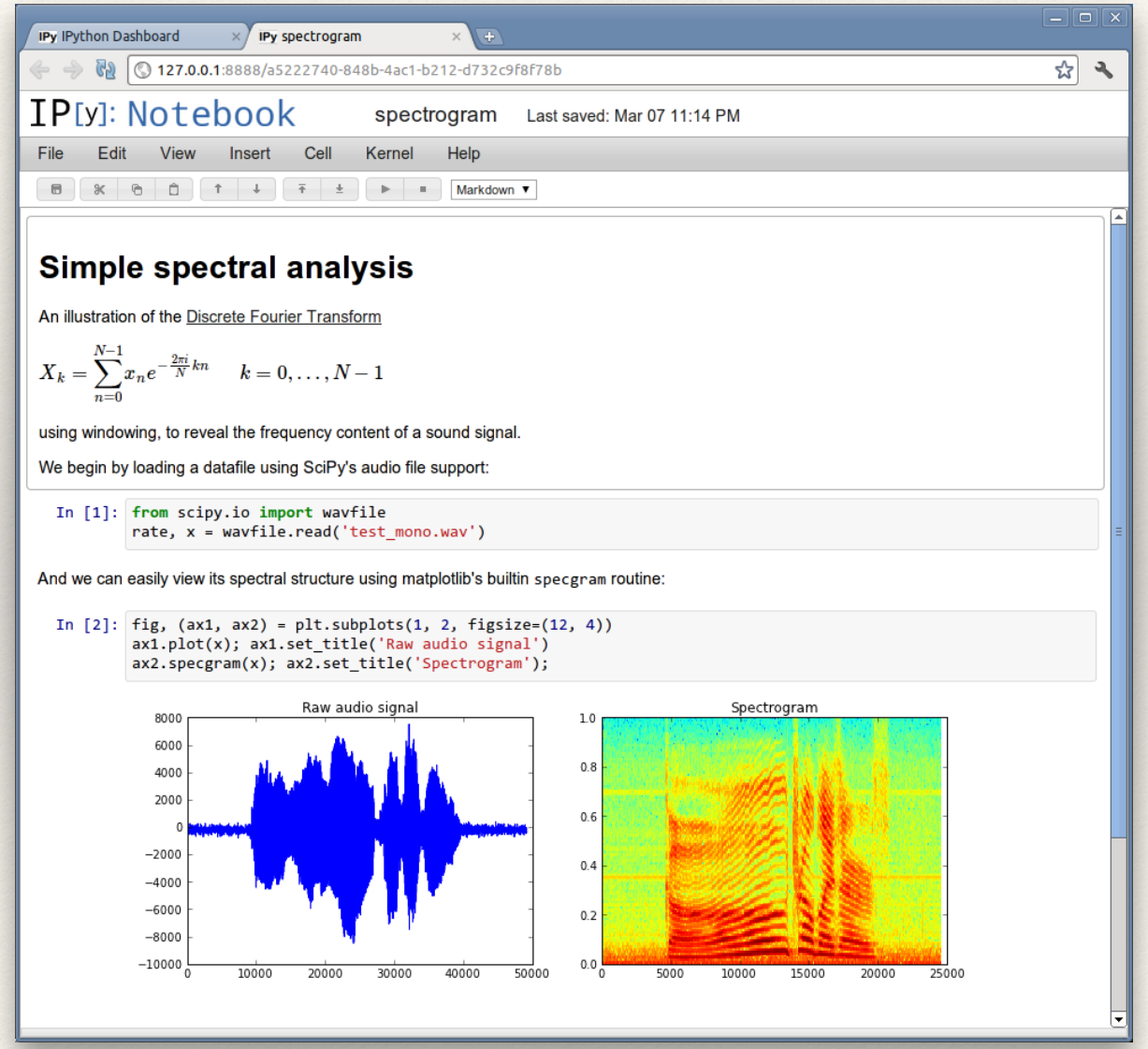
- The REPL as a network protocol
- Kernels
  - execute code
- Clients
  - Read input
  - Present output

Simple abstractions enable rich, sophisticated clients



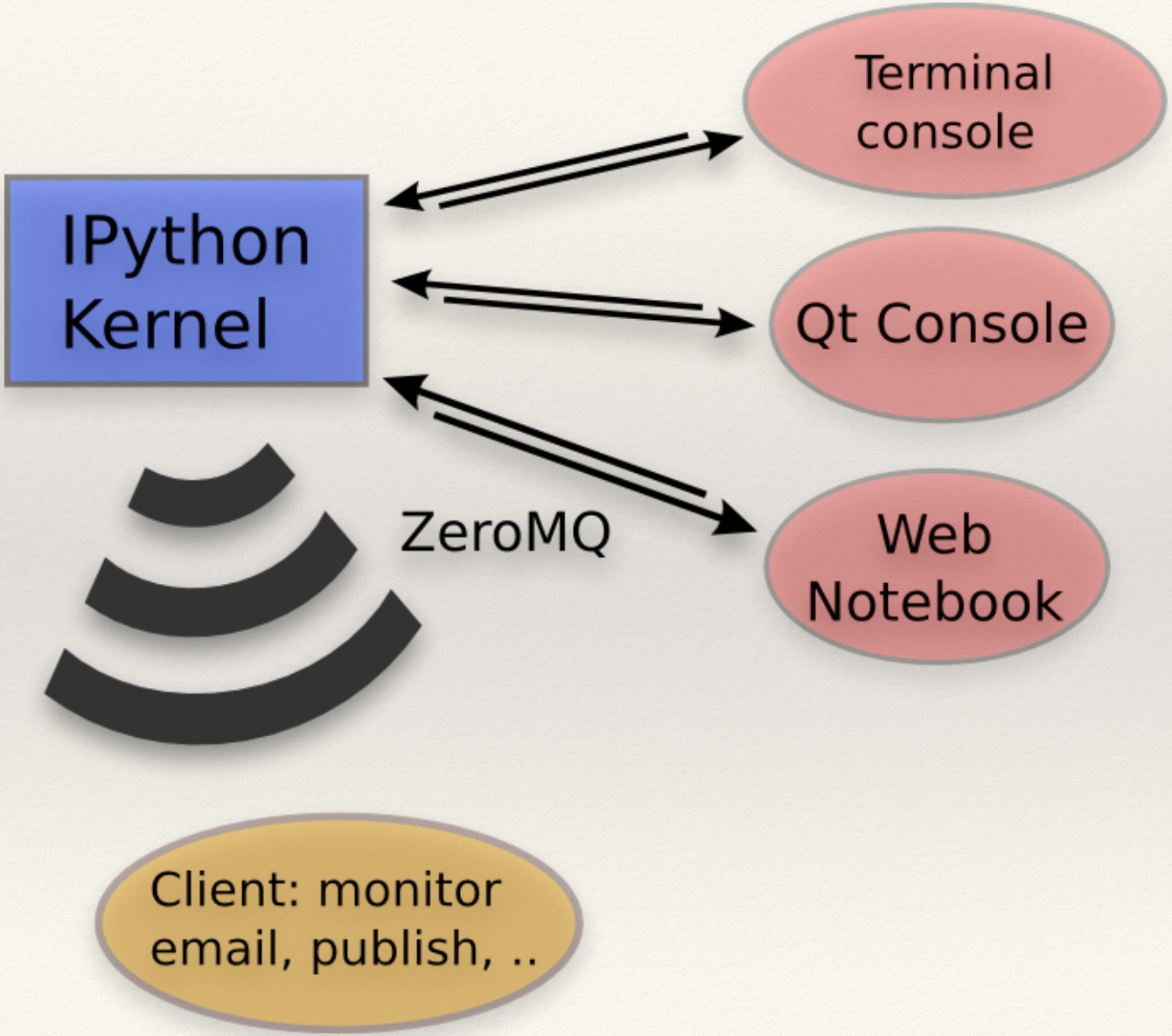
2011: The IPython Notebook

- Rich web client
- Text & math
- Code
- Results
- Share, reproduce.



From IPython to Project Jupyter

A simple and generic architecture



Not just about Python: Kernels in any language

- IPython "Official", we ship it.
- IJulia
- IRKernel
- IHaskell
- IFSharp
- Ruby
- IScala
- IErlang
- **Lots more! ~37 and counting**

“Why is it called IPython,

if it can do Julia, R, Haskell, Ruby, … ?”

IPython

- Interactive Python shell at the terminal
- Kernel for this protocol in Python
- Tools for Interactive Parallel computing
- Network protocol for interactive computing
- Clients for protocol
  - Console
  - Qt Console
  - Notebook
- Notebook file format & tools (nbconvert...)
- Nbviewer

IPython … Jupyter

- Interactive Python shell at the terminal
- Kernel for this protocol in Python
- Tools for Interactive Parallel computing

- Network protocol for interactive computing
- Clients for protocol
  - Console
  - Qt Console
  - Notebook
- Notebook file format & tools (nbconvert...)
- Nbviewer

Language Agnostic

What's in a name?

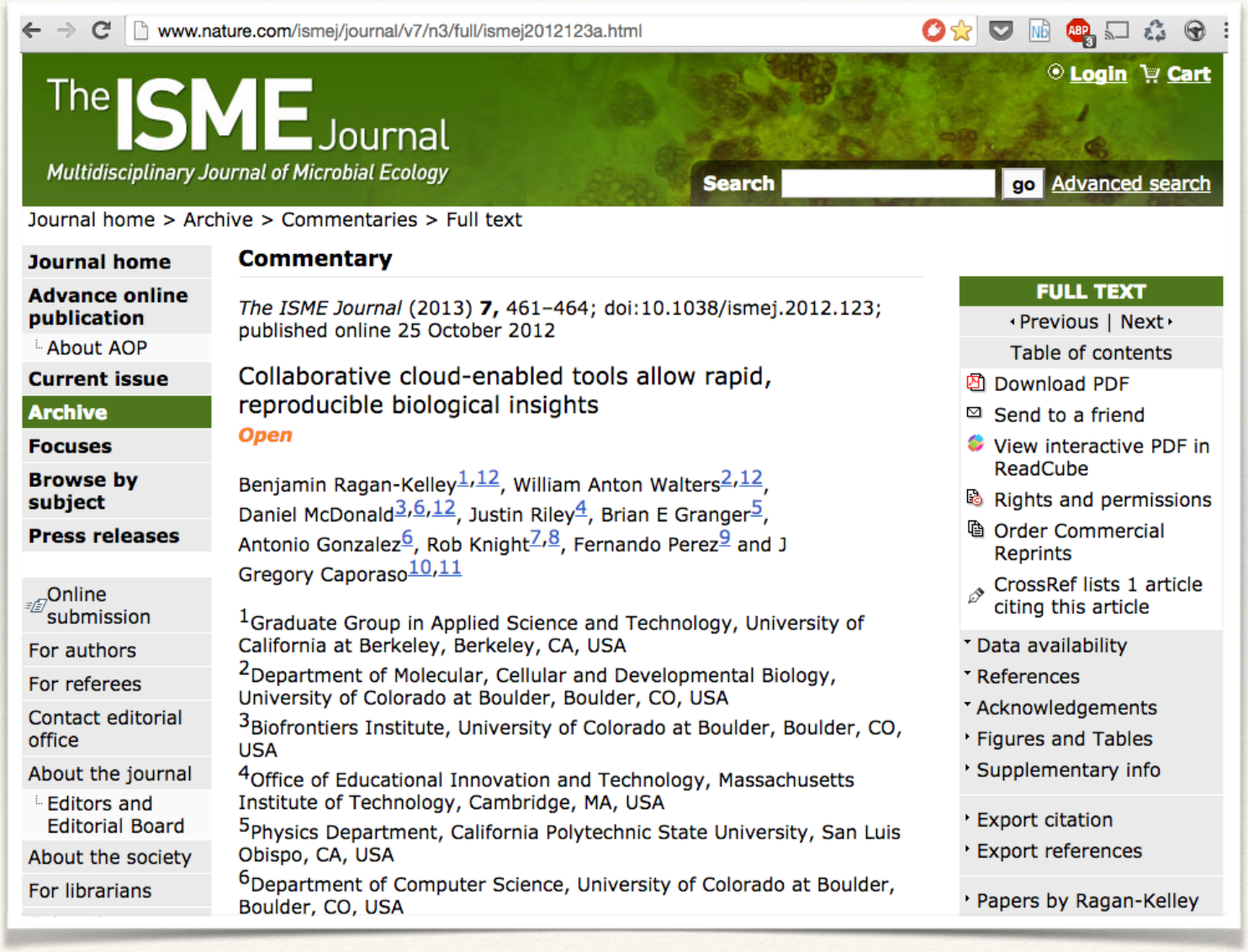
- *Inspired* by the open languages of science:
  - Julia, Python & R
  - *not* an acronym: *all languages* equal class citizens.
- Astronomy and Scientific Python:
  - A long and fruitful collaboration
- Galileo's notebooks:
  - the original, open science, data-and-narrative papers
  - Authorea: "Science was Always meant to be Open"

The Jupyter Notebook Ecosystem

nbviewer: seamless notebook sharing

- Zero-install reading of notebooks
- Just share a URL
- nbviewer.ipython.org

Reproducible Research



http://www.nature.com/ismej/journal/v7/n3/full/ismej2012123a.html

Paper, Notebooks and Virtual Machine

This notebook is intended to calculate the positions of primers in an alignment, using functions from PrimerProspector.

Import the needed functions, and define the primer sequences

In [8]:
```
# Code modified from PrimerProspector library slice_aligned_region.py (development version
)

# Imports and definitions
from string import lower, upper
from operator import itemgetter

from cogent import LoadSeqs, DNA
from cogent.core.alphabet import AlphabetError
from cogent.align.align import make_dna_scoring_dict, local_pairwise
from cogent.parse.fasta import MinimalFastaParser
from cogent.core.moltype import IUPAC_DNA_ambiguities

DNA_CODES = ['A', 'C', 'T', 'G', 'R', 'Y', 'M', 'K',
 'W', 'S', 'B', 'D', 'H', 'V', 'N']


# Note that these are all written 5'->3', the reverse primers are reverse complemented for
 the local alignment

# If one wanted to test different primers, they would be defined here.

# 27f/338r = V2 (also includes V1, but generally just referred to as V2)
# 349f/534r = V3
# 515f/806r = V4
# 967f/1046r = V6
# 1391f/1492r = V9

primer_seqs = {
          '27f':'AGAGTTTGATCMTGGCTCAG',
          '338r':DNA.rc('GCTGCCTCCCGTAGGAGT'),
          '349f':'GYGCASCAGKCGMGAAW',
          '534r':DNA.rc('ATTACCGCGGCTGCTGG'),
          '515f':'GTGCCAGCMGCCGCGGTAA',
          '806r':DNA.rc('GGACTACVSGGGTATCTAAT'),
          '967f':'CAACGCGAAGAACCTTACC',
          '1048r':DNA.rc('CGRCRGCCATGYACCWC'),
          '1391f':'TGYACACACCGCCCGTC',
          '1492r':DNA.rc('GGCTACCTTGTTACGACTT'),
          '1391r':'TGYACACACCGCCCGTC' # Need this rather than forward primer to get prope
r 3' position of reverse version
          }

reference_aligned_file = '/home/ubuntu/qiime_software/gg_otus-4feb2011-release/rep_set/gg_
76_otus_4feb2011_aligned.fasta'
```

Instructions and supporting data for the QIIME/IPython/StarCluster demo at the *2012 NIH Cloud Computing the Microbiome* workshop and our corresponding paper in the ISME Journal.

The analysis made use of the IPython Notebook, QIIME, StarCluster, PyCogent, and PrimerProspector. All of these tools are pre-installed in the `ami-9f69c1f6` public Amazon EC2 instance, which was used in this study.

Supporting Files

The IPython notebooks supporting this study can be viewed here and are available here in PDF format:

- NIH Cloud Demo (Complete)
- NIH Cloud Demo (Fast)
- Timing*
- Variable Region Position Boundaries
- Pearson v Robinson-Foulds Distances
- V3 and V4 Regions Only

* Note that the Timing notebook is for reference as related to the paper only - it will not be directly reproducible on re-runs of the above notebooks as it relies on the semi-manual creation of the `tasks.log` file. The `tasks.log` file used to generate the original timing data is available for download here.

The Greengenes reference OTU collection used in this study is available for download here.

The IPython notebook files (`.ipynb`) are available for download here.

The tree metadata mapping file used in generating the coloring categories in the 3D PCoA plot is available here.

The paper for this analysis, "Collaborative cloud-enabled tools allow rapid, reproducible biological insights", is available here.

Reproducing the analysis

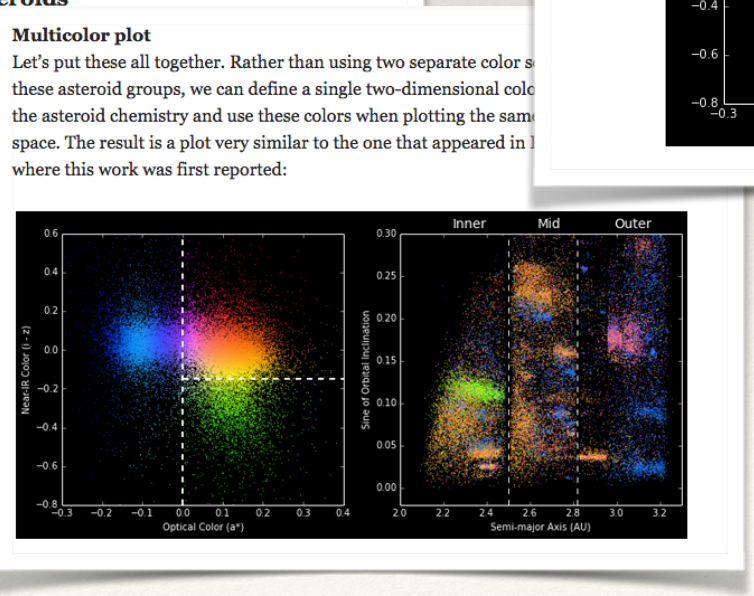
Four `m2.4xlarge` instances were booted using StarCluster to create a 32 core cluster with approximately 280GB of RAM (70GB per 8 core instance). This was used for the full analysis (a more complete analysis then was done during the workshop, where the workshop analysis was optimized to run quickly). To support the large quantity of data that is generated during the analysis, you should create an EBS volume which will be attached to the running instance. A 20 GB volume will be sufficient. The volume used for running these notebooks is available as `snap-75eb8005`.

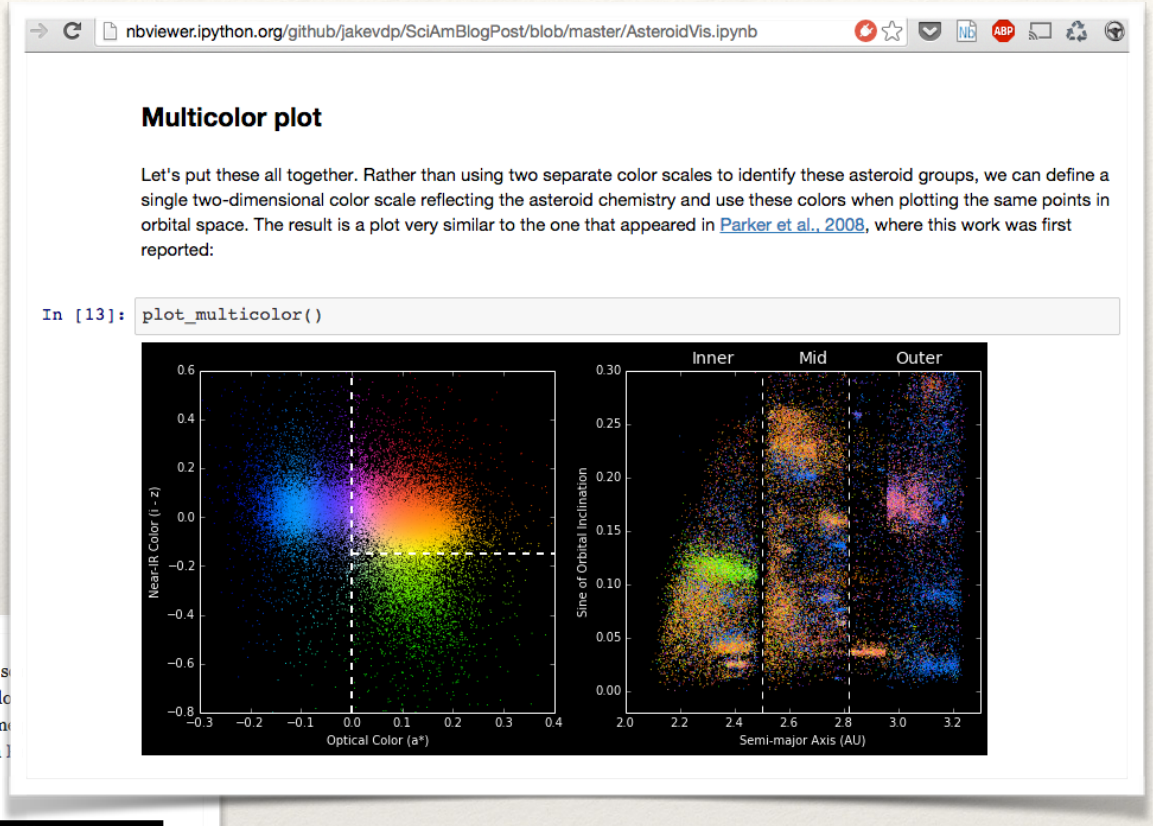
To reproduce the analyses presented in this paper you should install StarCluster locally, and configure it according to the instructions on the StarCluster website. You can then add the following to your `~/.starcluster/config` file:

```
[plugin ipcluster]
setup_class = starcluster.plugins.ipcluster.IPCluster
enable_notebook = true
# If you leave notebook_passwd out, a random password
# will be generated instead.
notebook_passwd = YOUR-PASSWORD

[cluster qiime-ipython]
node_image_id = ami-9f69c1f6
cluster_user = ubuntu
keyname = YOUR-KEY
cluster_size = 4
node_instance_type = m2.4xlarge
plugins = ipcluster
volumes = qiime-ipython-data

[volume qiime-ipython-data]
VOLUME_ID = YOUR-VOLUME-ID
MOUNT_PATH = /home/ubuntu/data
```

http://qiime.org/home_static/nih-cloud-apr2012

Scientific Blogging





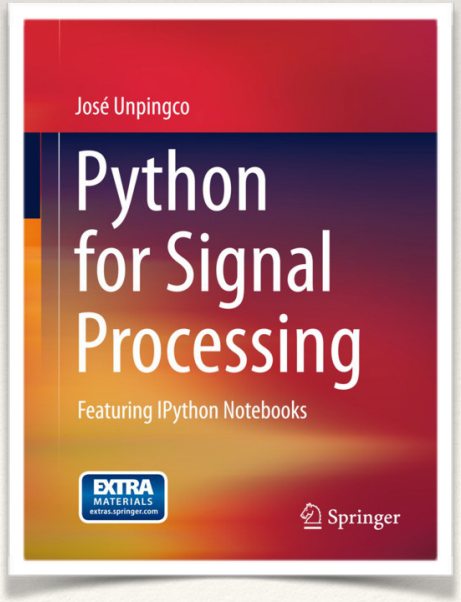
Jake van der Plas @ UW

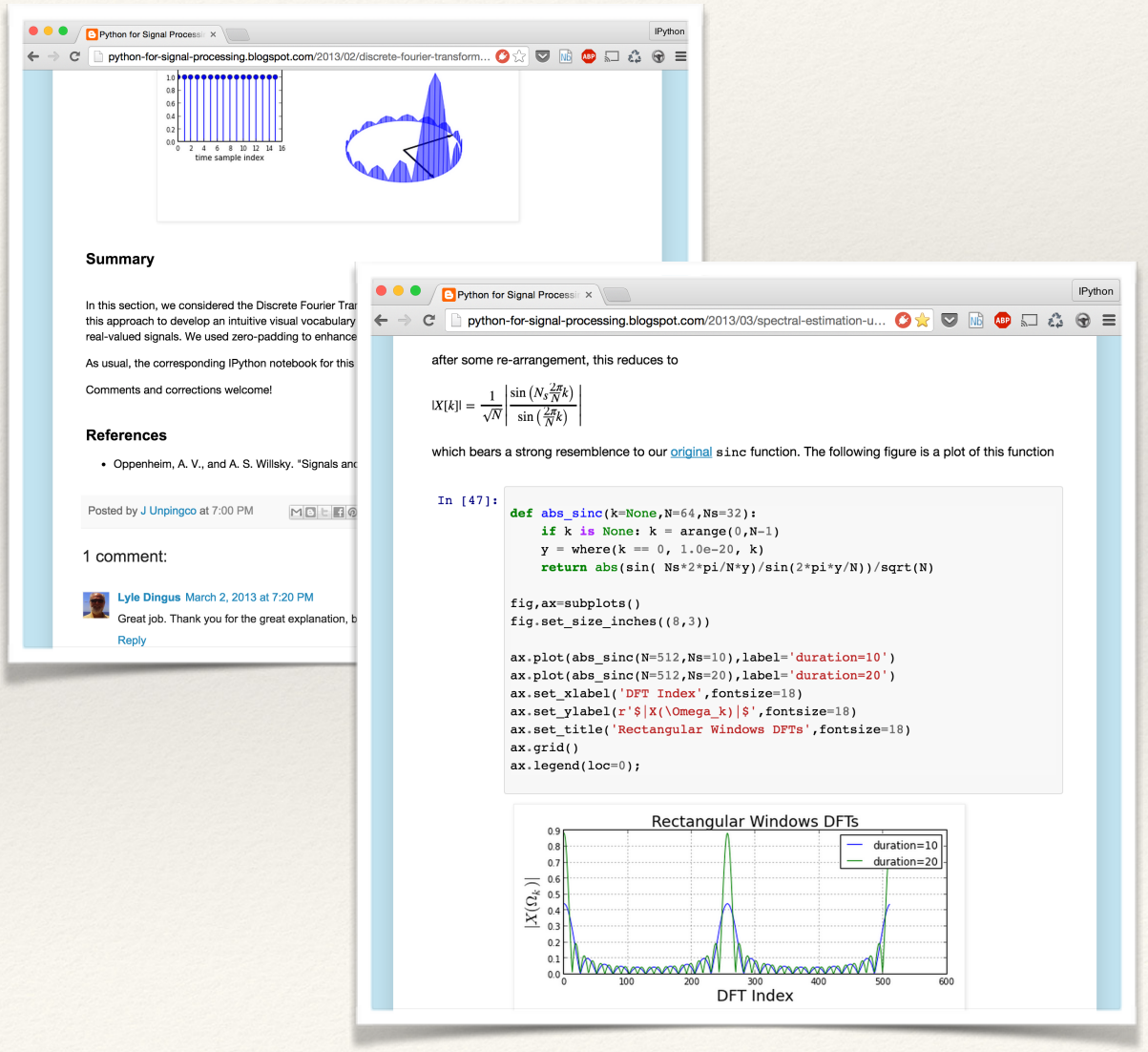
http://blogs.scientificamerican.com/sa-visual/2014/09/16/visualizing-4-dimensional-asteroids

Executable books

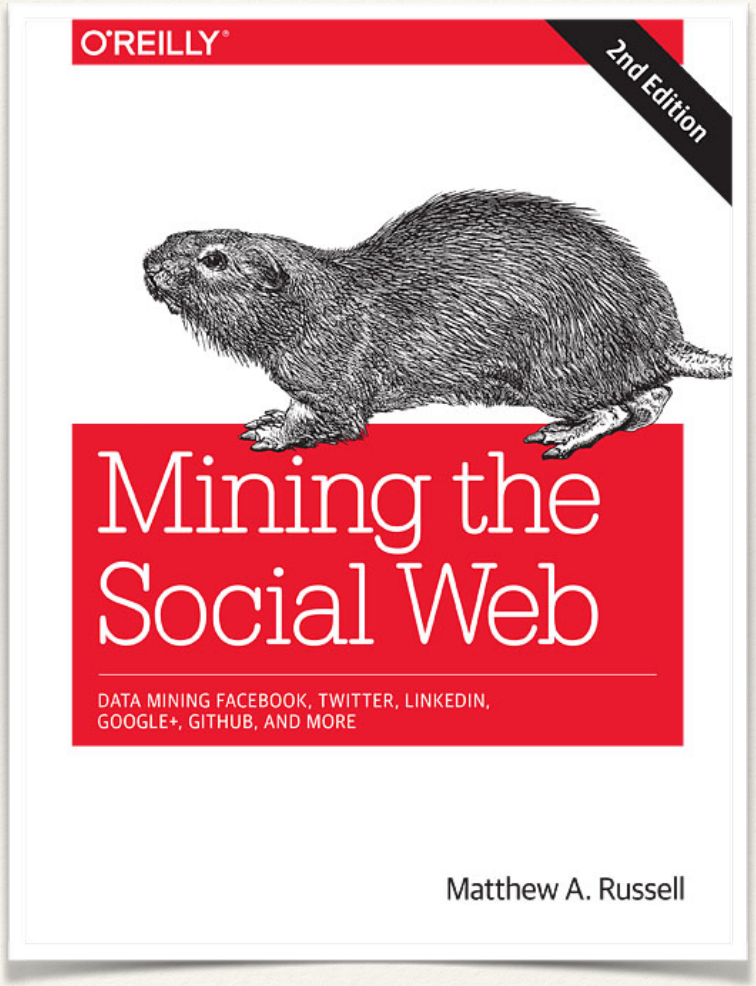
Python for Signal Processing, by José Unpingco

- Springer hardcover book
- Chapters: IPython Notebooks
- Posted as a blog entry
- All available as a Github repo

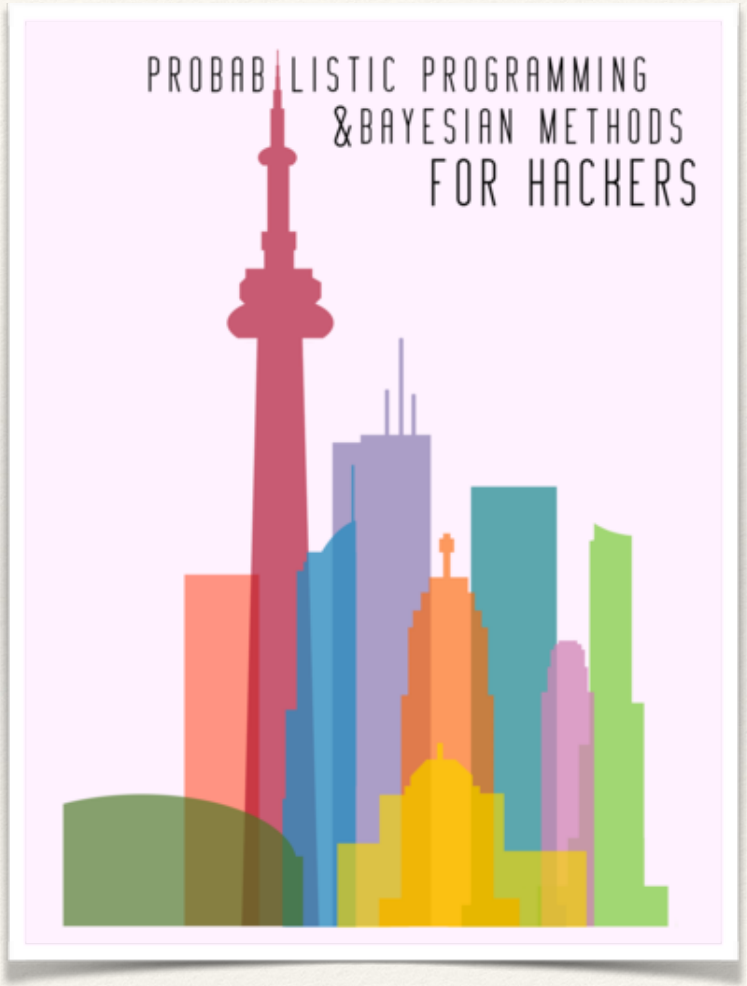




More authors creating books this way



By Matthew Russell



By Cameron Davidson-Pilon

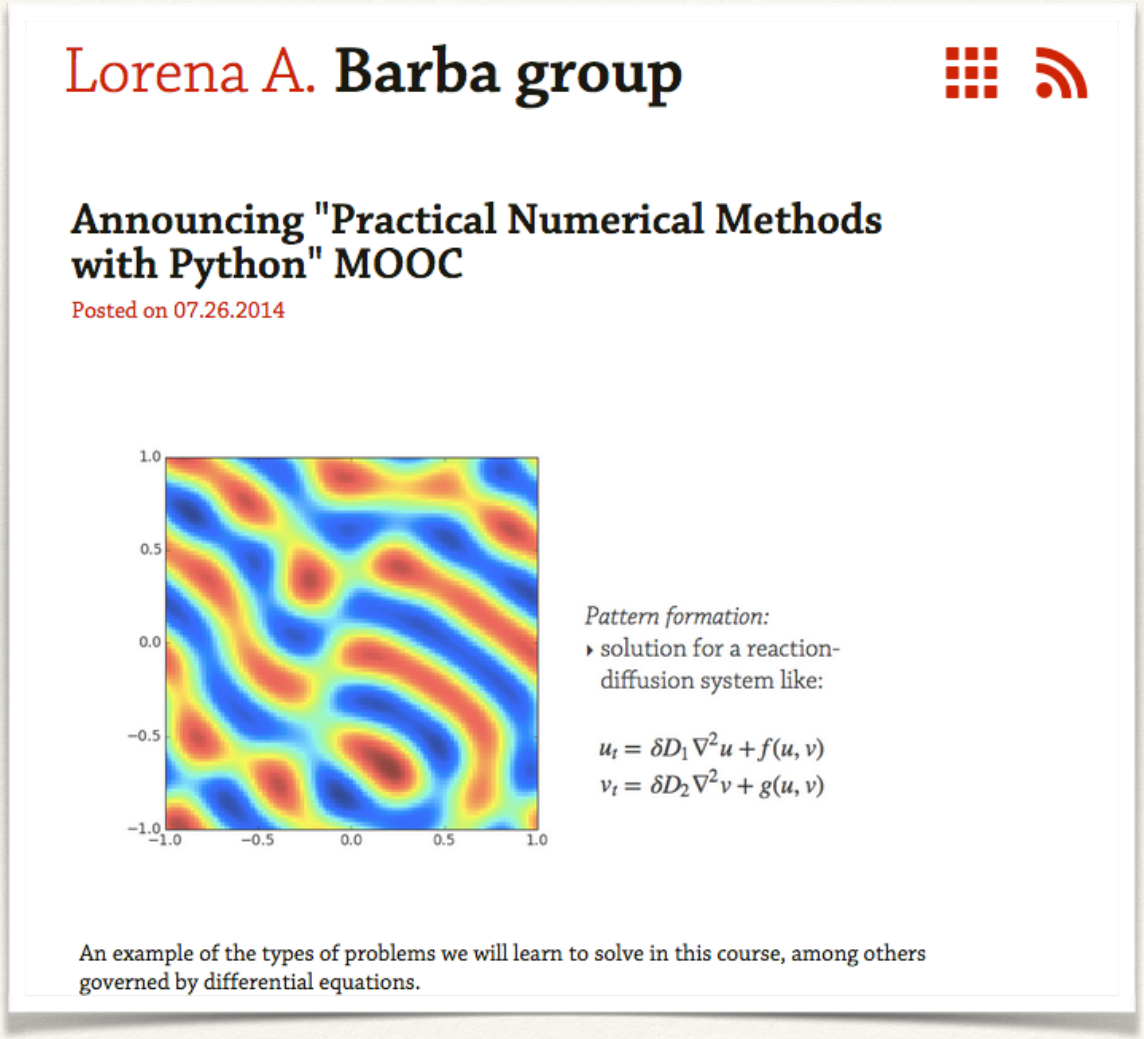
University Courses

	Course	University	Instructor
0	Data Science and Visualization with Python	Santa Clara	Brian Granger
1	Python for Data Science	UC Berkeley	Josh Bloom
2	Introduction to Data Science	UC Berkeley	Michael Franklin
3	Working with Open Data	UC Berkeley	Raymond Yee
4	Introduction to Signal Processing	UC Berkeley	Miki Lustig
5	Data Science (CS 109)	Harvard University	Pfister and Blitzstein
6	Practical Data Science	NYU	Josh Attenberg
7	Scientific Computing (ASTR 599)	University of Washington	Jake Vanderplas
8	Computational Physics	Cal Poly	Jennifer Klay
9	Introduction to Programming	Alaskan High School	Eric Matthes
10	Aerodynamics-Hydrodynamics (MAE 6226)	George Washington University	Lorena Barba
11	HyperPython: hyperbolic conservation laws	KAUST	David Ketcheson
12	Quantitative Economics	NYU	Sargent and Stachurski
13	Practical Numerical Methods with Python	4 separate universities + MOOC	Barba, et al.
14	Data Science: Algorithms	Columbia - Lede Program	Chris Wiggins
15	Data Science: Databases	Columbia - Lede Program	Chris Wiggins
16	Data Science: Foundations	Columbia - Lede Program	Chris Wiggins
17	Data Science: Platforms	Columbia - Lede Program	Chris Wiggins

These are just some we are aware of!

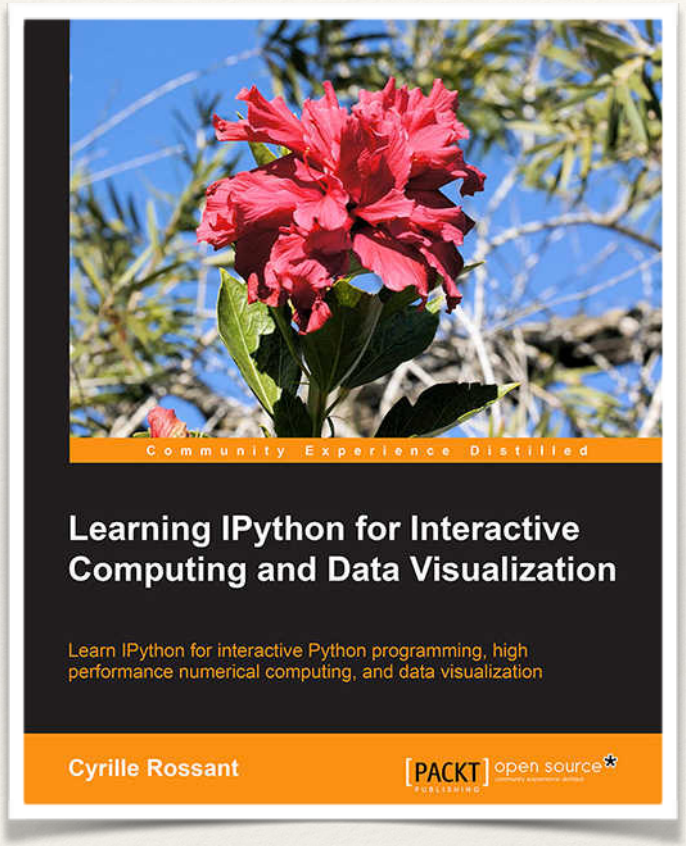
A collaborative MOOC on OpenEdX

- *Lorena Barba* at George Washington University, USA.
- *Ian Hawke* at Southampton, UK
- *Carlos Jerez* at Pontifical Catholic University of Chile.
- All materials on Gihtub.

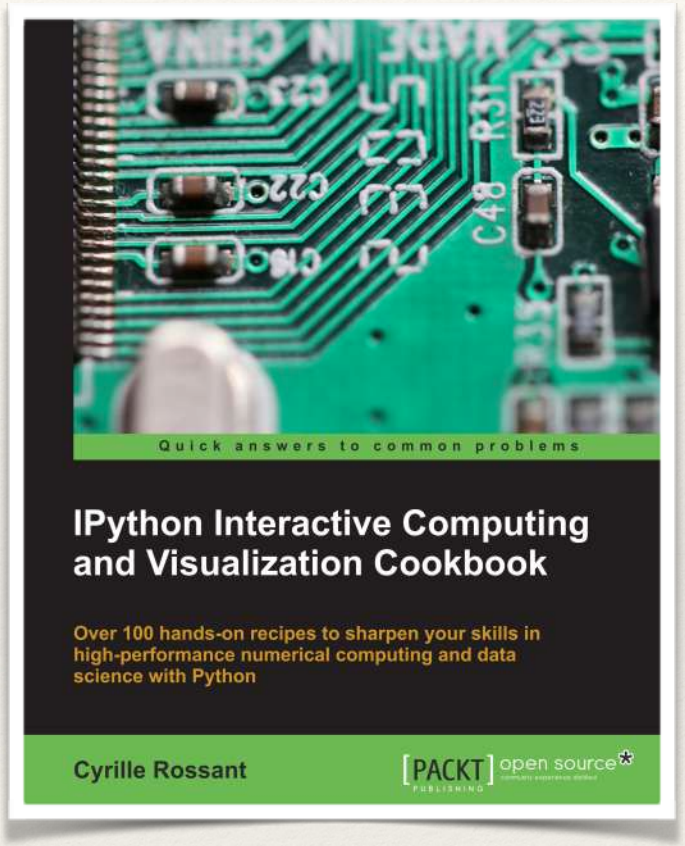


http://lorenabarba.com/news/announcing-practical-numerical-methods-with-python-mooc

Books about IPython



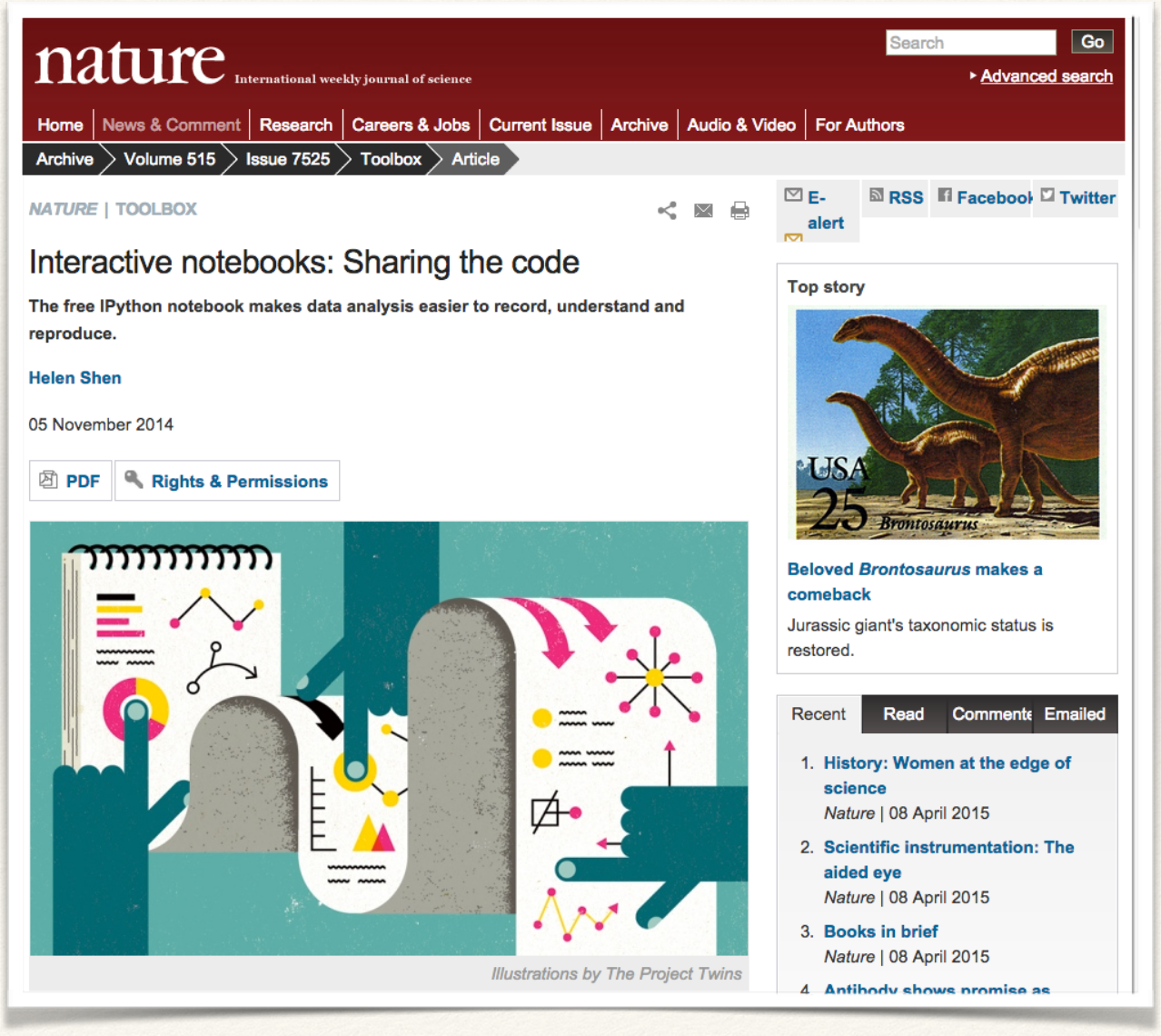
Learning IPython for Interactive Computing and Data Visualization



IPython Interactive Computing and Visualization Cookbook

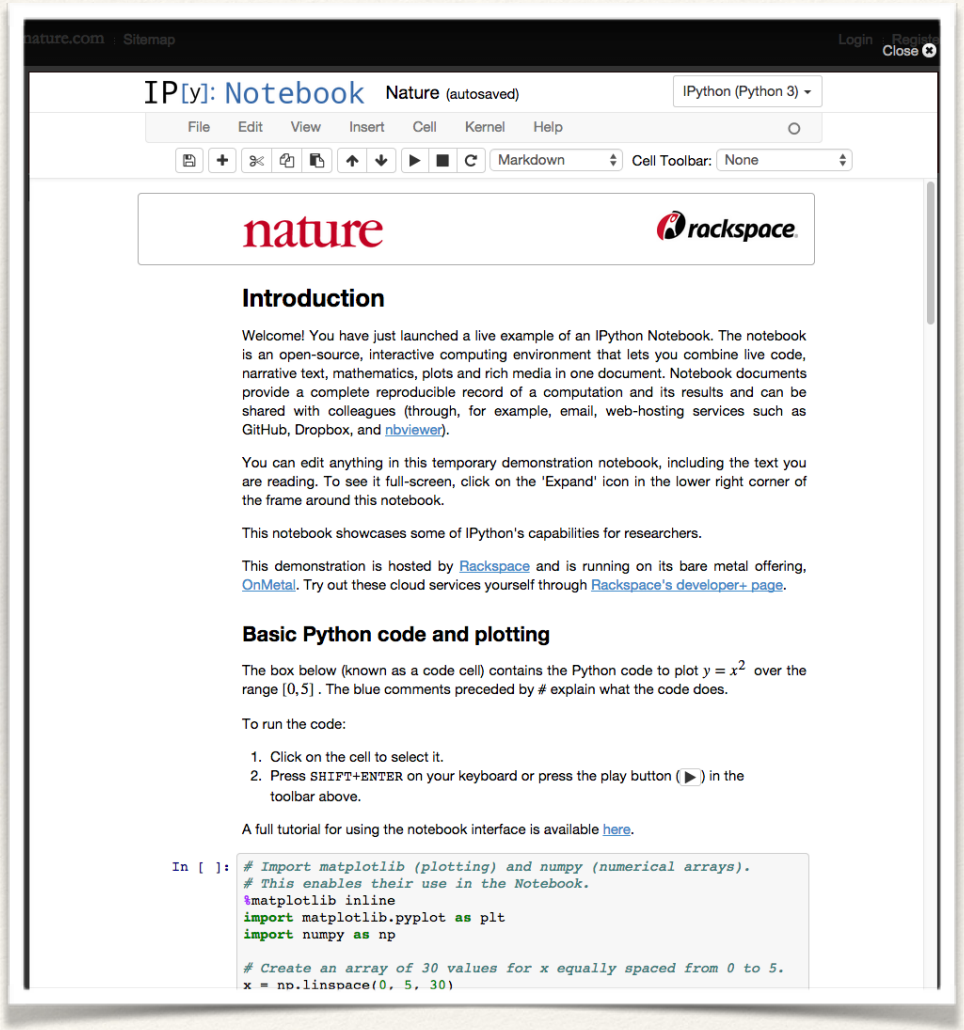
Cyrille Rossant
cyrille.rossant.net

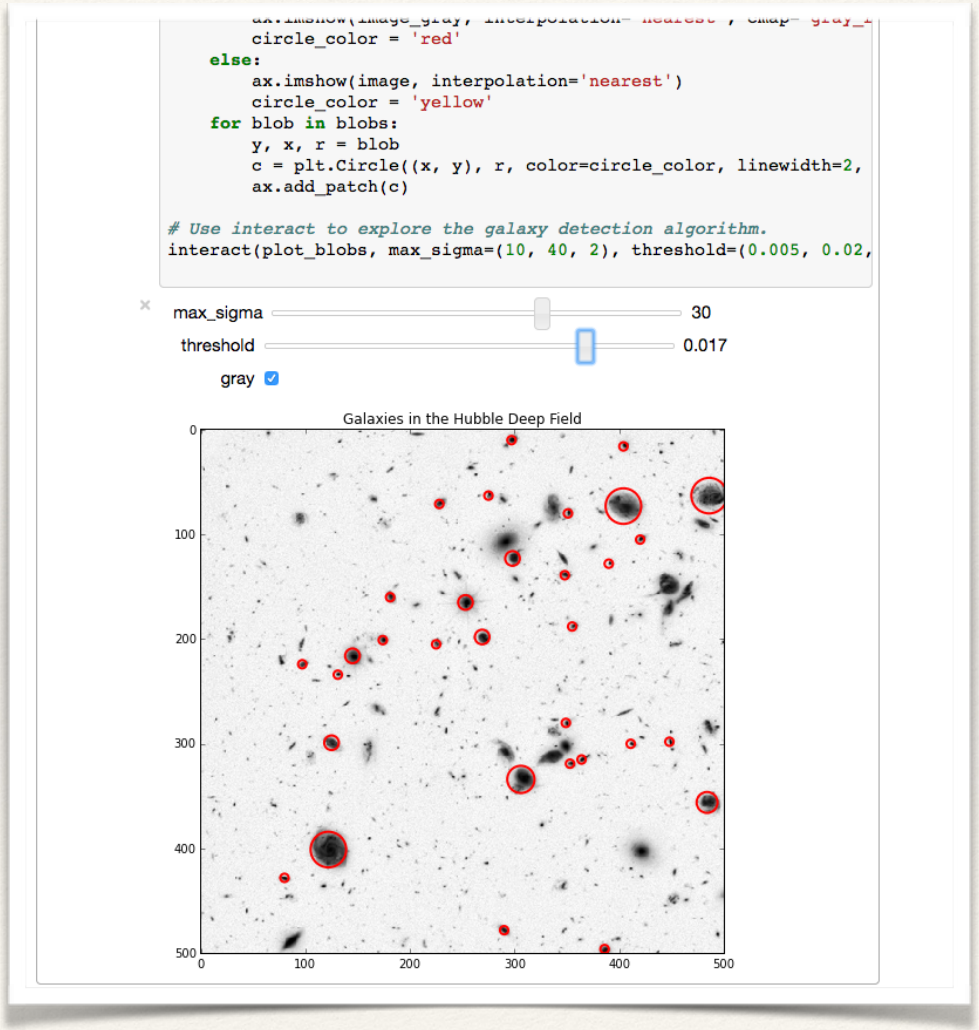
Changing the scientific culture



http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261

Executable papers: the future?





http://www.nature.com/news/ipython-interactive-demo-7.21492?article=1.16261

Notebook Workflows: The Big Picture

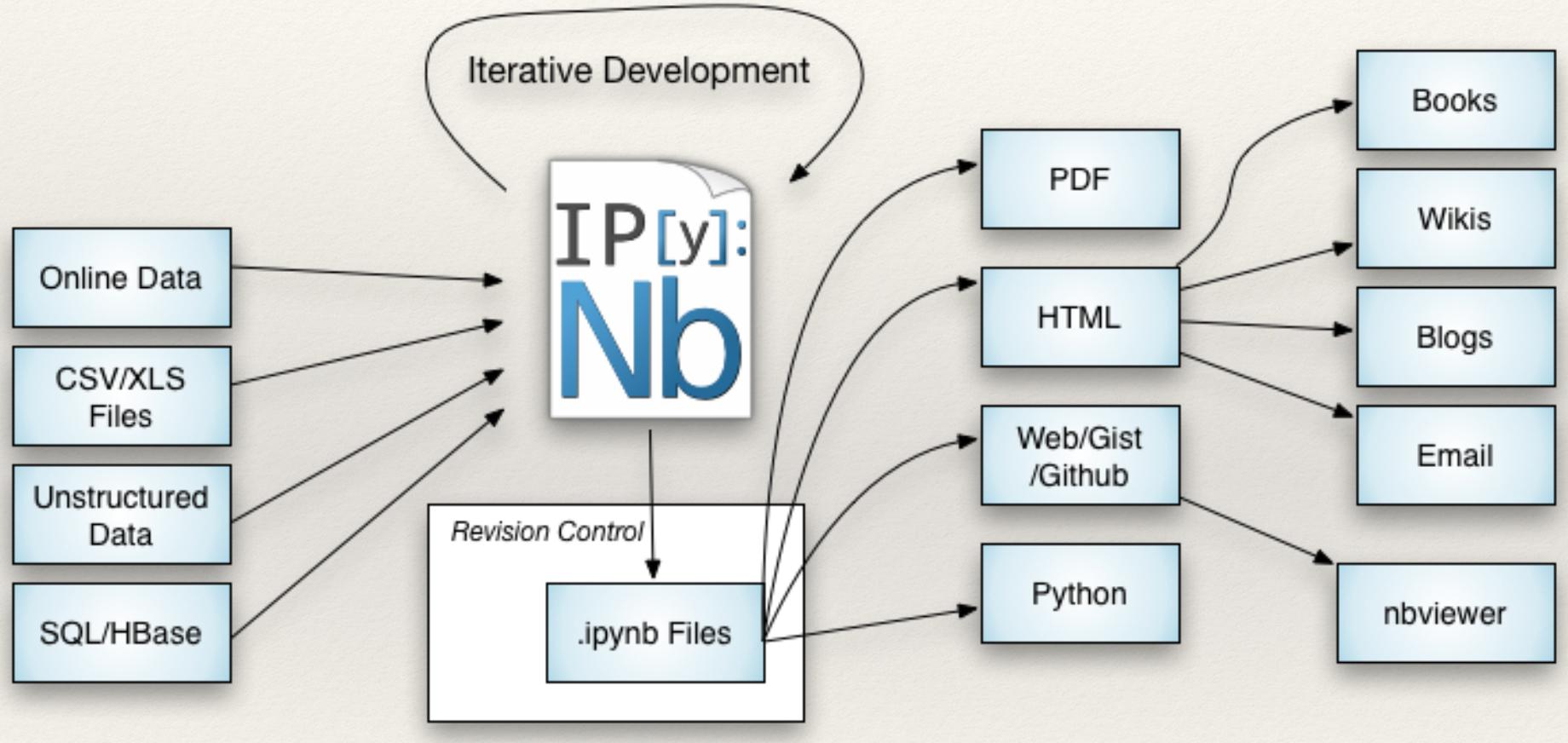


Image credit: Joshua Barratt

Lots more! The IPython Gallery

A gallery of interesting IPython Notebooks

Fernando Perez edited this page 8 days ago · 229 revisions

This page is a curated collection of IPython notebooks that are notable for some reason. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

Important contribution instructions: If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using nbviewer, rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that Matt Davis has conveniently written a set of bookmarklets and extensions to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

Table of Contents

Reproducible academic publications

This section contains academic papers that have been published in the peer-reviewed literature or pre-print sites such as the ArXiv that include one or more notebooks that enable (even if only partially) readers to reproduce the results of the publication. If you include a publication here, please link to the journal article as well as providing the nbviewer notebook link (and any other relevant resources associated with the paper).

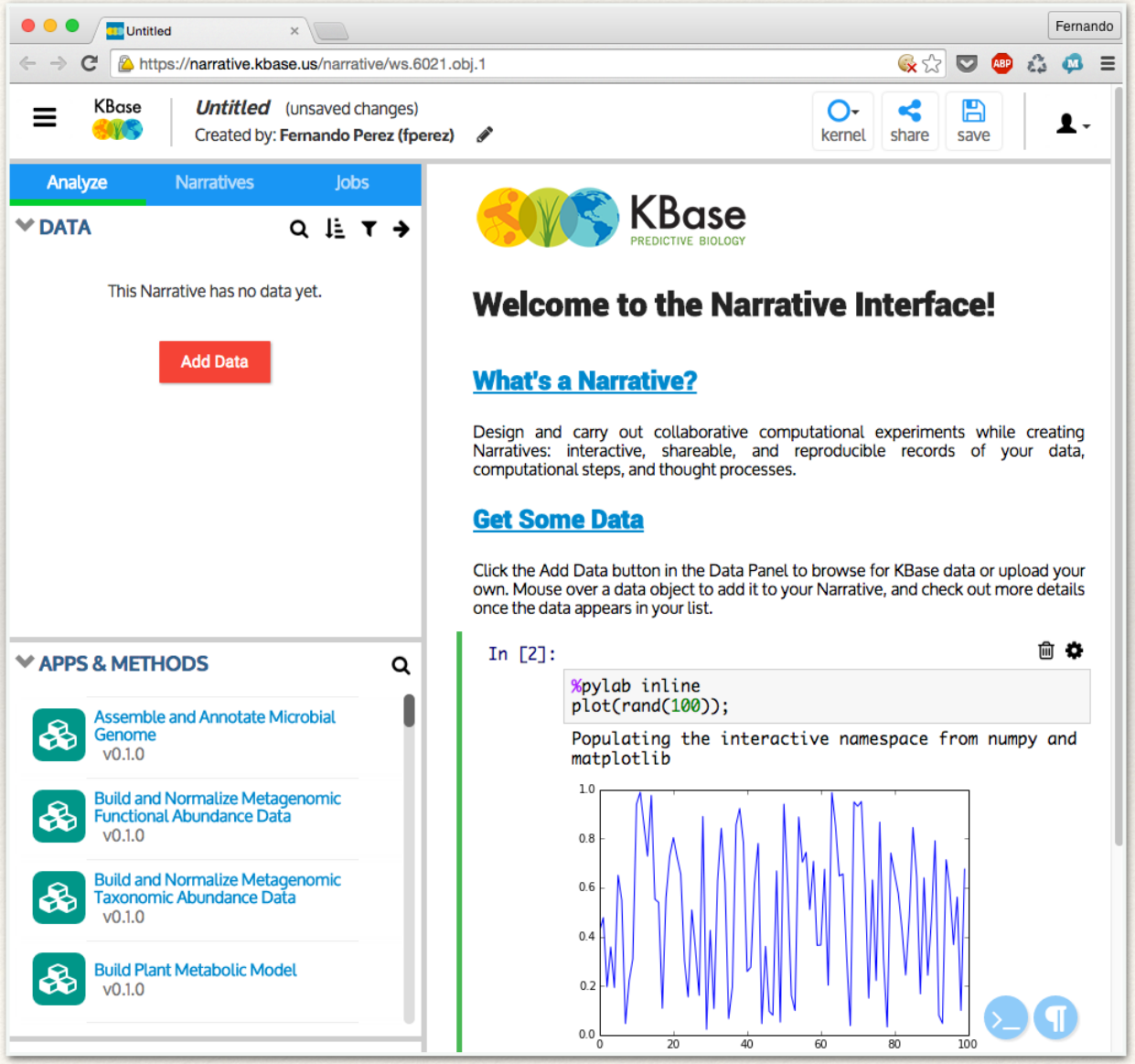
1. Reply to 'Influence of cosmic ray variability on the monsoon rainfall and temperature': a false-positive in the field of solar-terrestrial research by Benjamin Laken, 2015. Reviewed article will appear in JASTP. The IPython notebook reproduces the full analysis and figures exactly as they appear in the article, and is available on Github: link via figshare.
2. The probability of improvement in Fisher's geometric model: a probabilistic approach, by Yoav Ram and Lilach Hadany. (Theoretical Population Biology, 2014). An IPython notebook, allowing figure reproduction, was deposited as a supplementry file.
3. Stress-induced mutagenesis and complex adaptation, by Yoav Ram and Lilach Hadany (Proceedings B, 2014). An IPython notebook, allowing figures reproduction, was deposited as a supplementry file.
4. Automatic segmentation of odor maps in the mouse olfactory bulb using regularized non-negative matrix factorization, by J. Soelter et al. (Neuroimage 2014, Open Access). The notebook allows to reproduce most figures from the paper and provides a deeper look at the data. The full code repository is also available.
5. Multi-tiered genomic analysis of head and neck cancer ties TP53 mutation to 3p loss, by A. Gross et al. (Nature Genetics 2014). The full collection of notebooks to replicate the results.
6. powerlaw: a Python package for analysis of heavy-tailed distributions, by J. Alstott et al.. Notebook of examples in manuscript, ArXiv link and project repository.
7. Collaborative cloud-enabled tools allow rapid, reproducible biological insights, by B. Ragan-Kelley et al.. The main notebook, the full collection of related notebooks and the companion site with the Amazon AMI information for reproducing the full paper.
8. A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data, by C.T. Brown et al.. Full notebook, ArXiv link and project repository.
9. The kinematics of the Local Group in a cosmological context by J.E. Forero-Romero et al.. The Full notebook and also all the data in a github repo.

https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks

Jupyter as Infrastructure

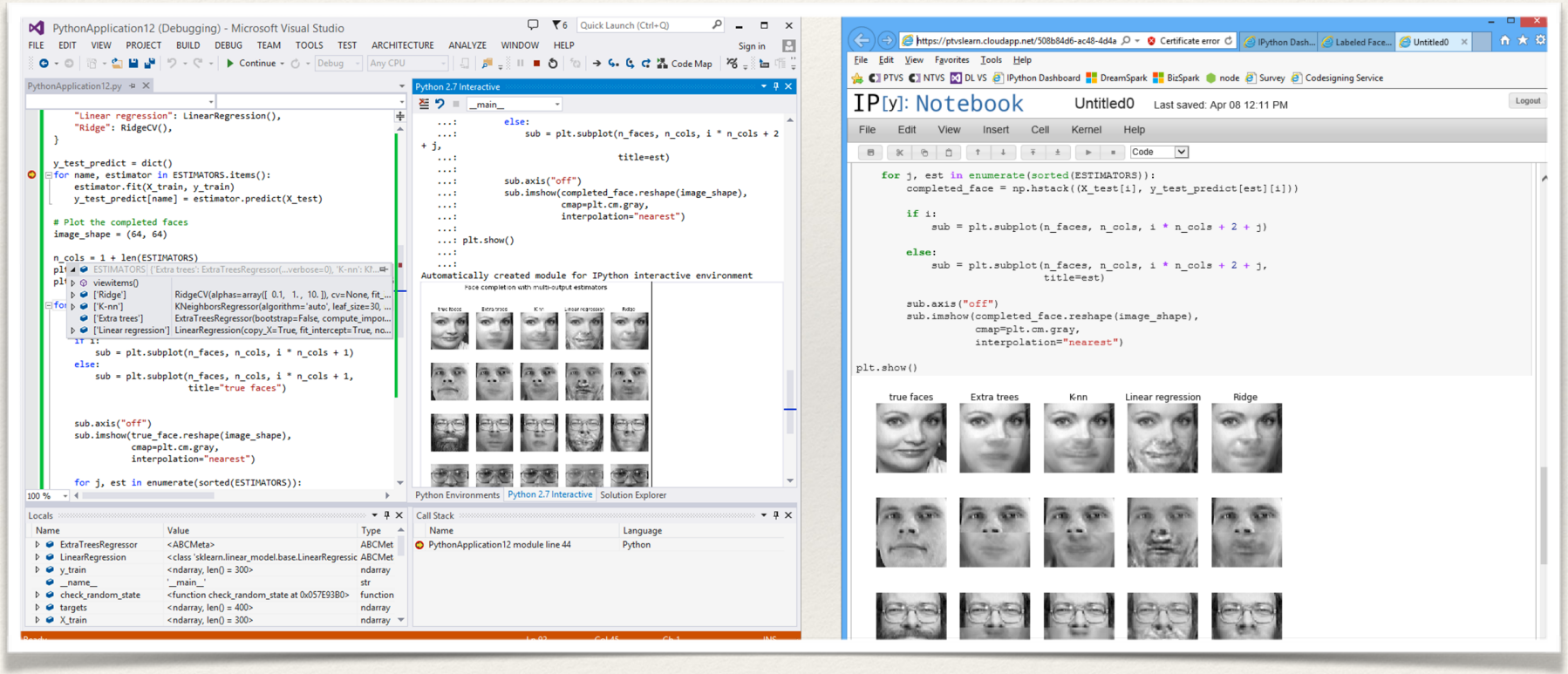
OSS and commercial adoption

The KBase Narrative Interface



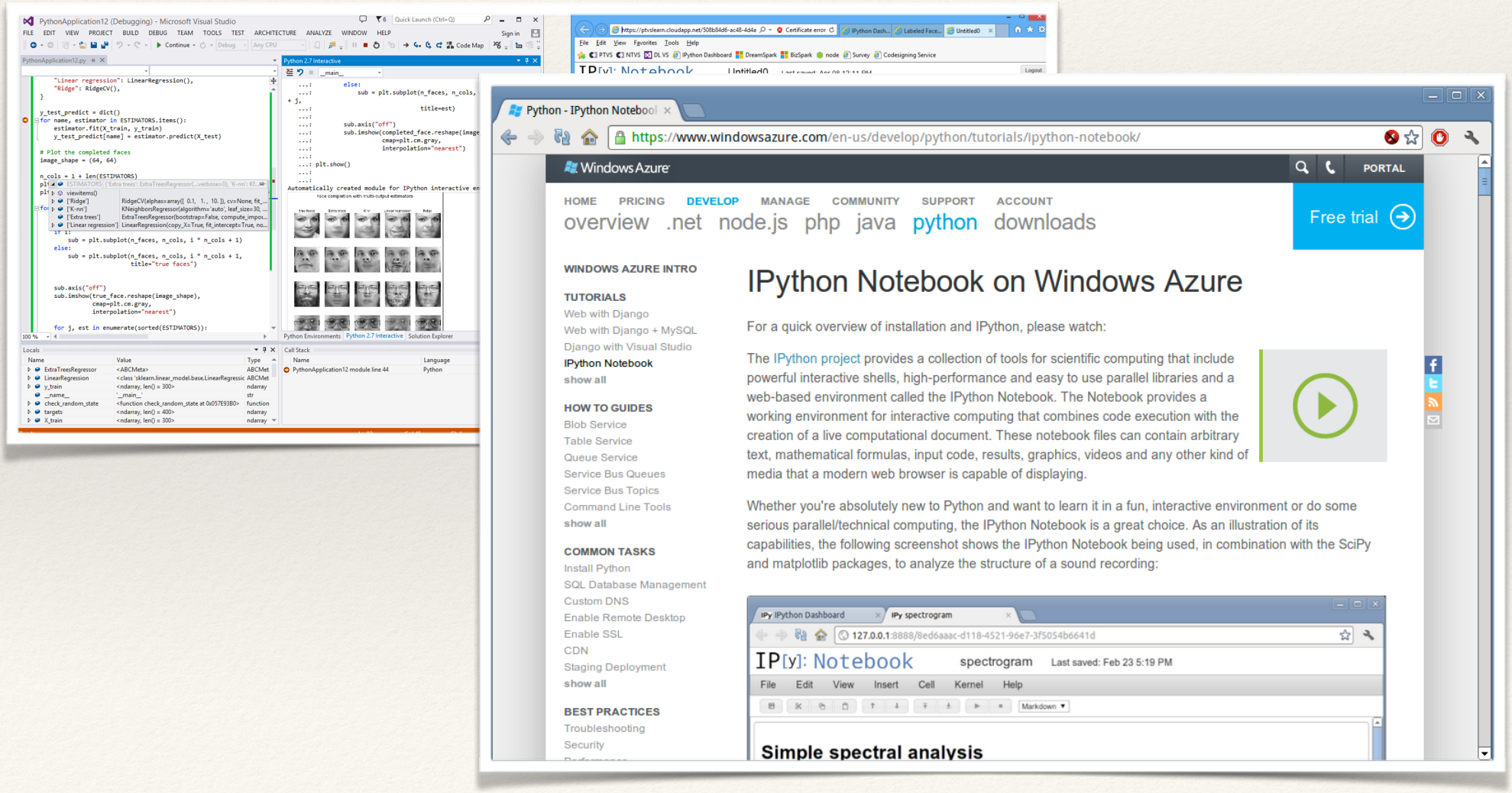
Microsoft: Visual Studio & Azure Cloud

Shahrokh Mortazavi, Dino Viehland, Wenming Ye, Dennis Gannon.



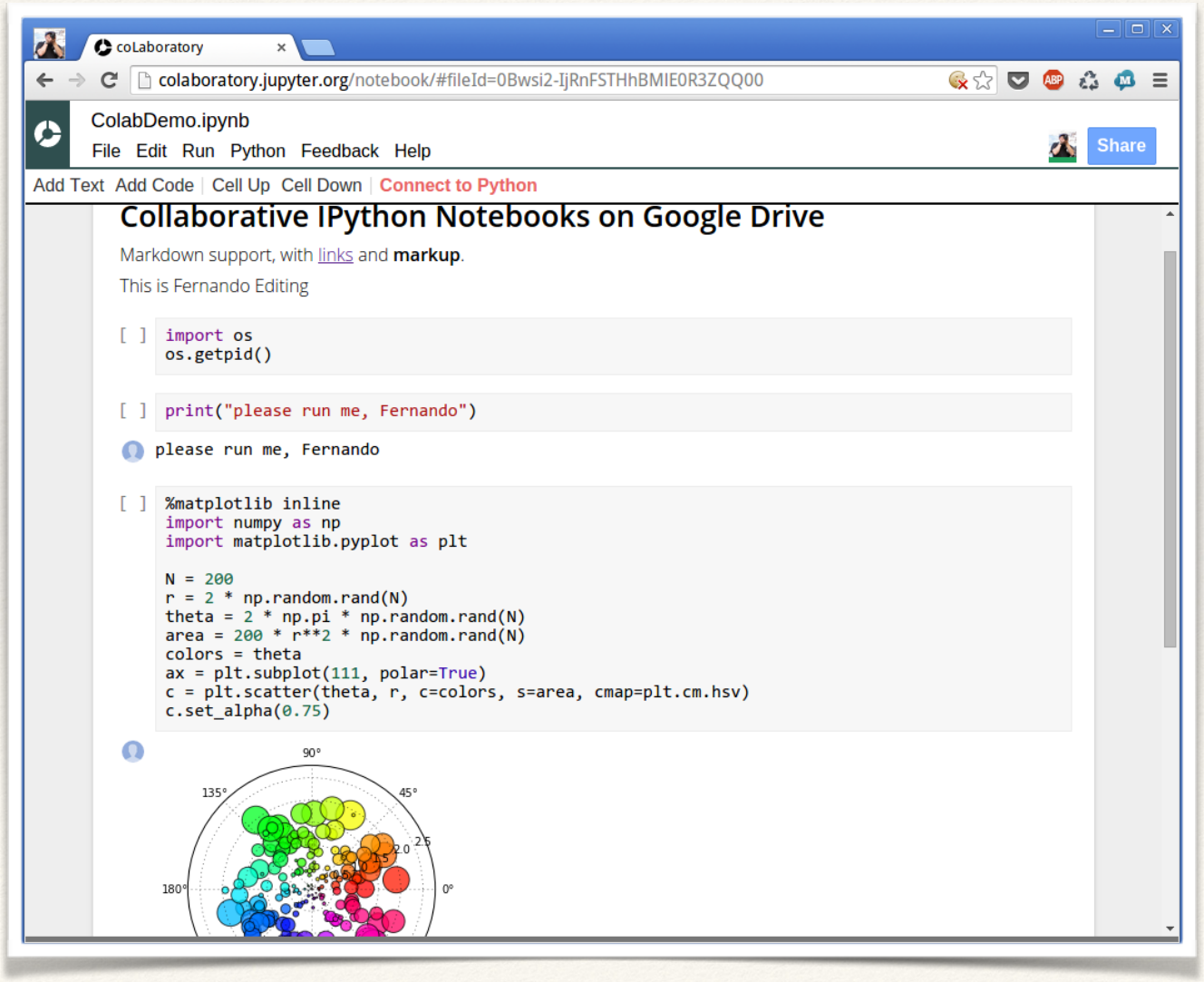
Microsoft: Visual Studio & Azure Cloud

Shahrokh Mortazavi, Dino Viehland, Wenming Ye, Dennis Gannon.

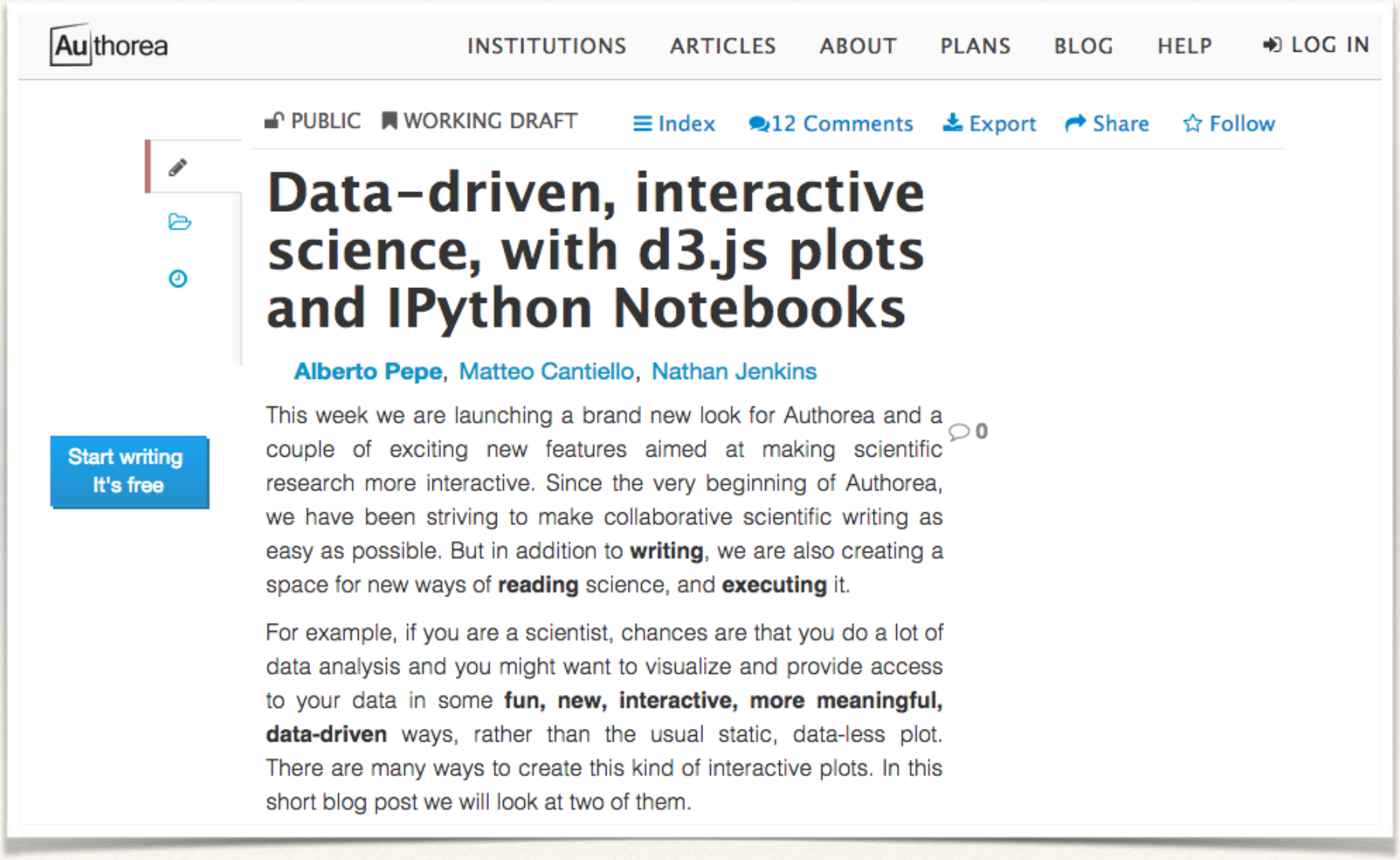


Google CoLaboratory

Kayur Patel, Kester Tong, Mark Sanders, Corinna Cortes @ Google
Matt Turk @ NCSA/UIUC



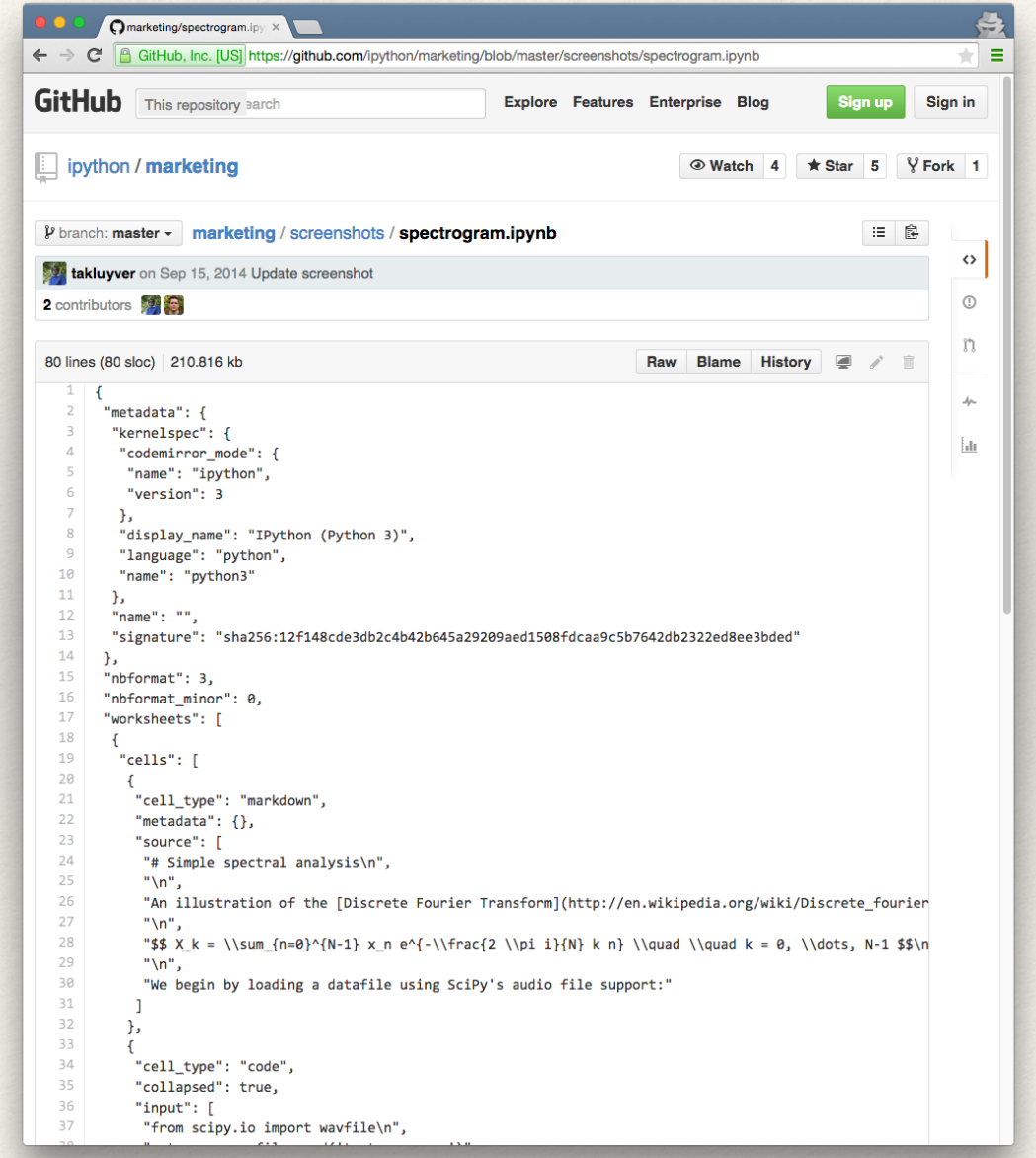
Authorea: notebooks in papers



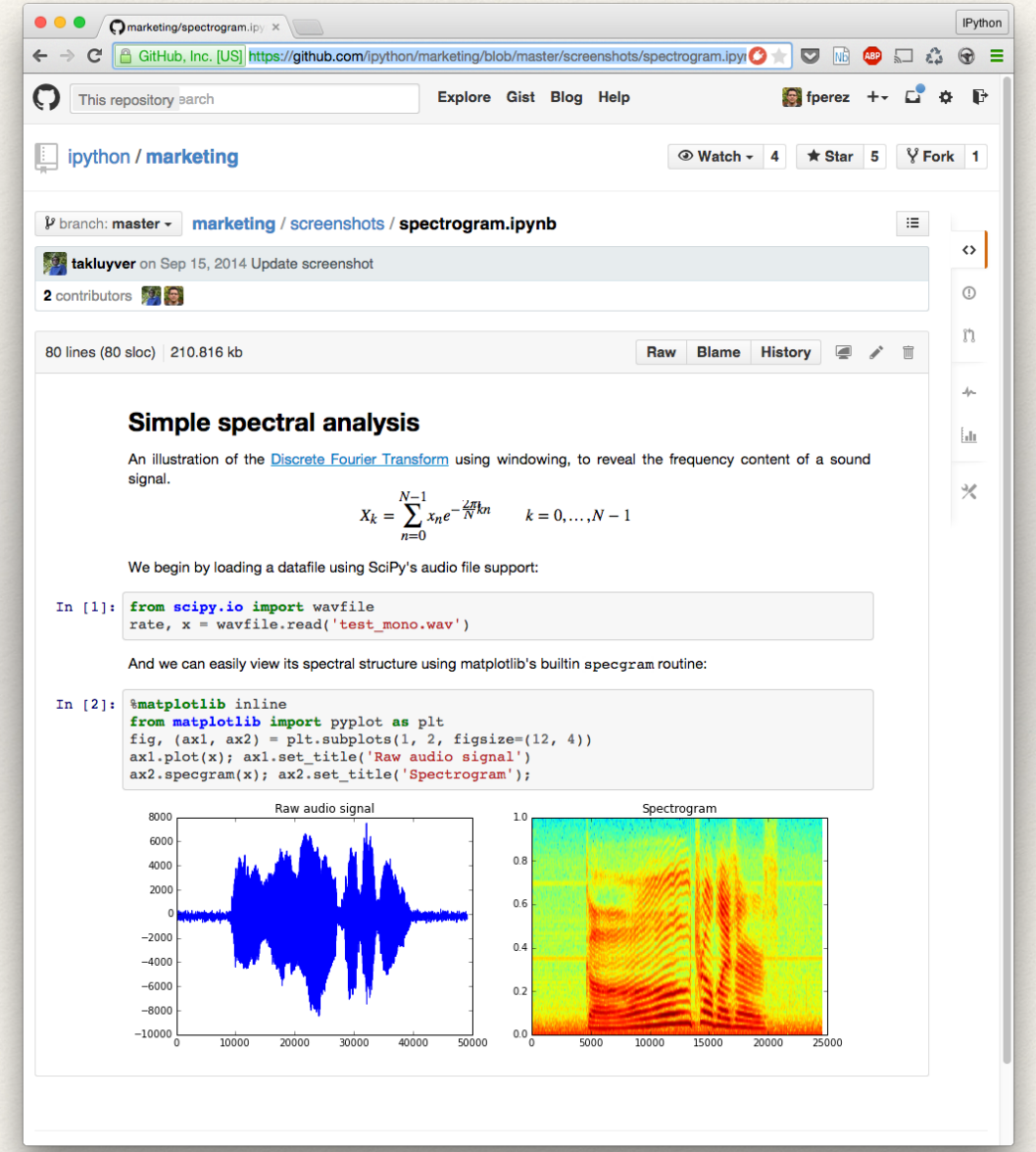
https://www.authorea.com/users/3/articles/3904/_show_article

Recent Developments

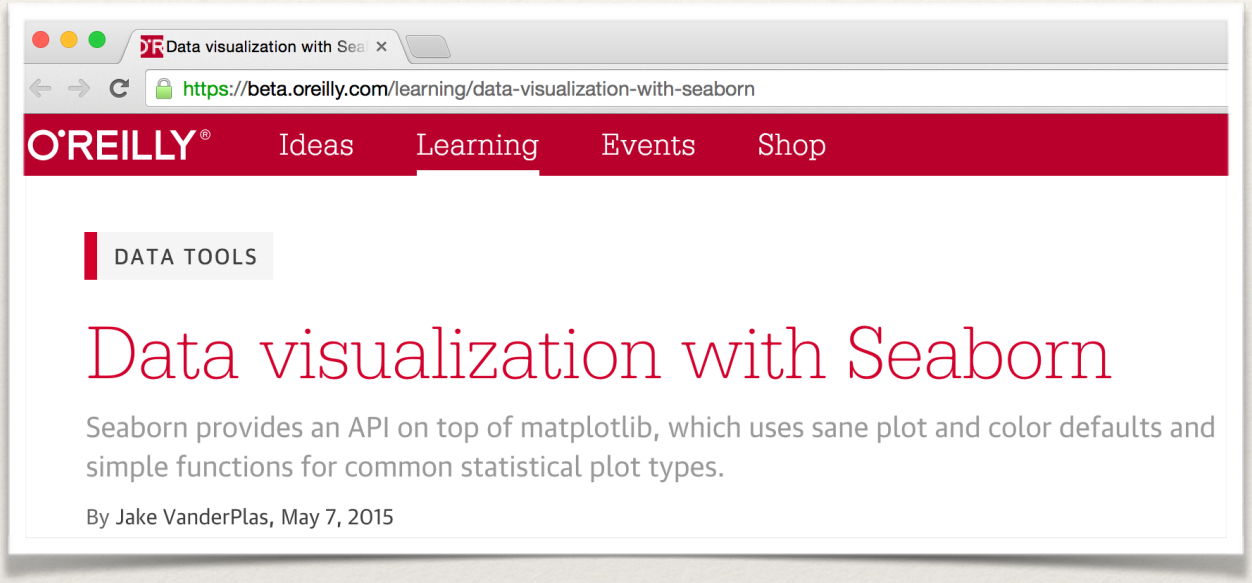
Over 200,000 notebooks on GitHub



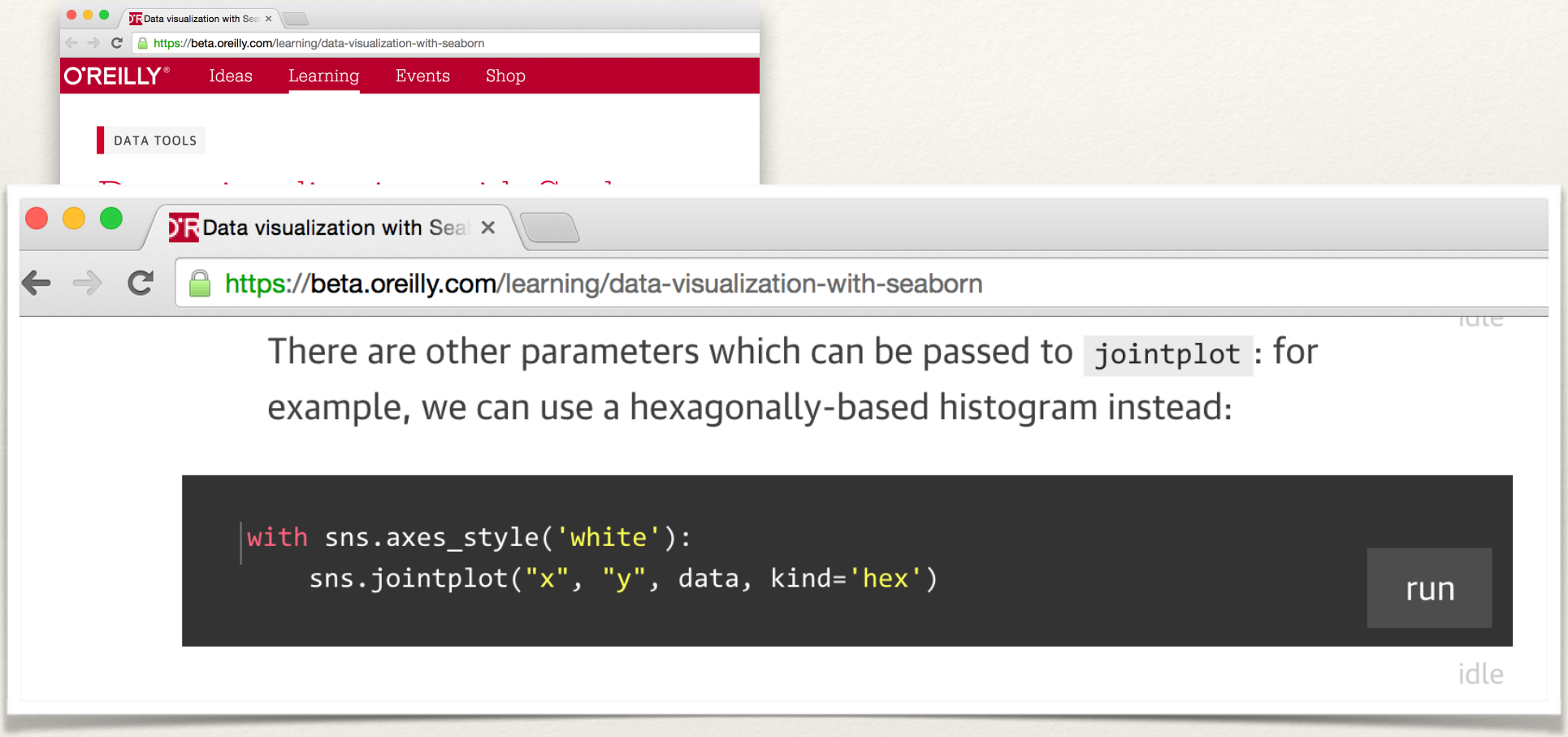
As of May 6, 2015:



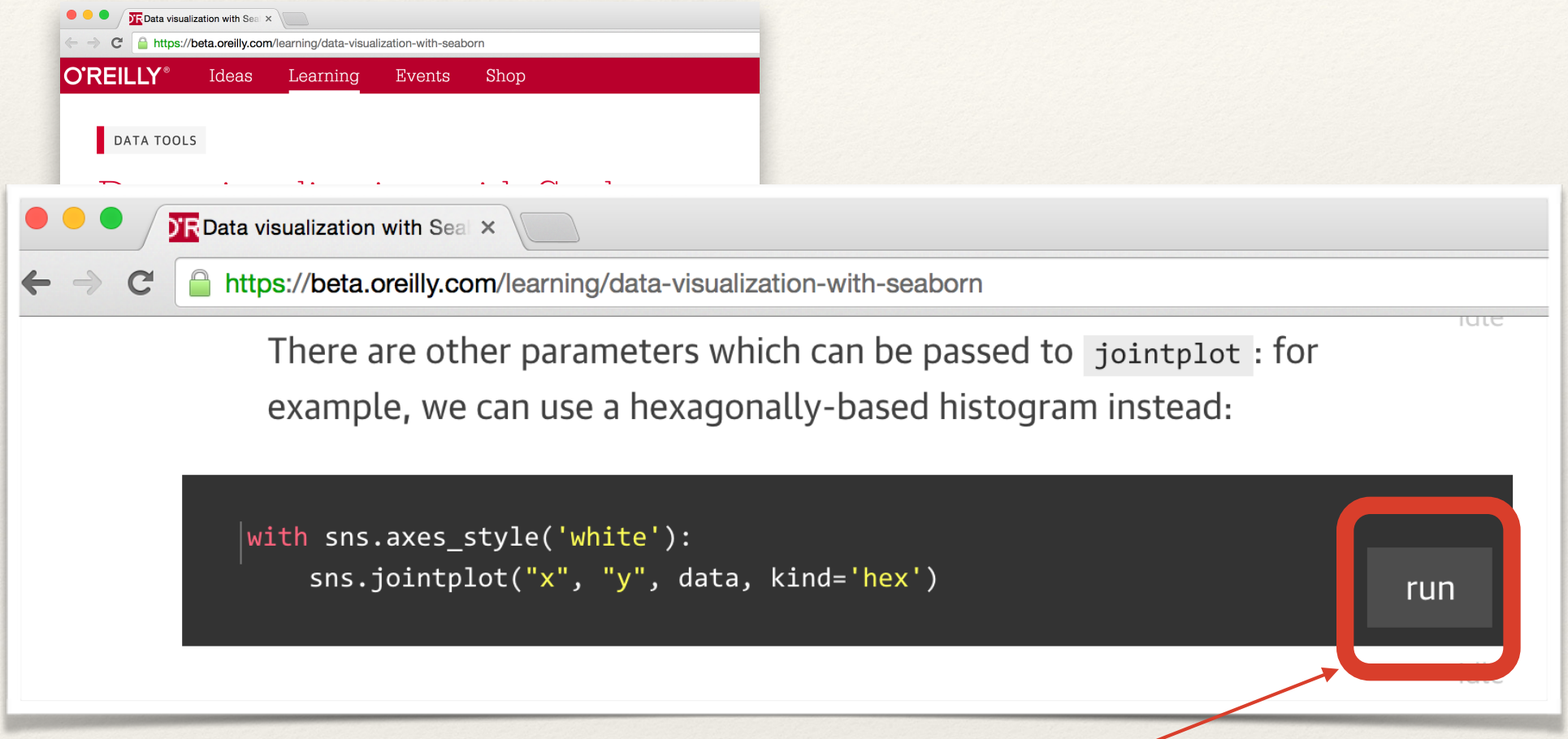
O'Reilly: authoring and delivering executable books



O'Reilly: authoring and delivering executable books

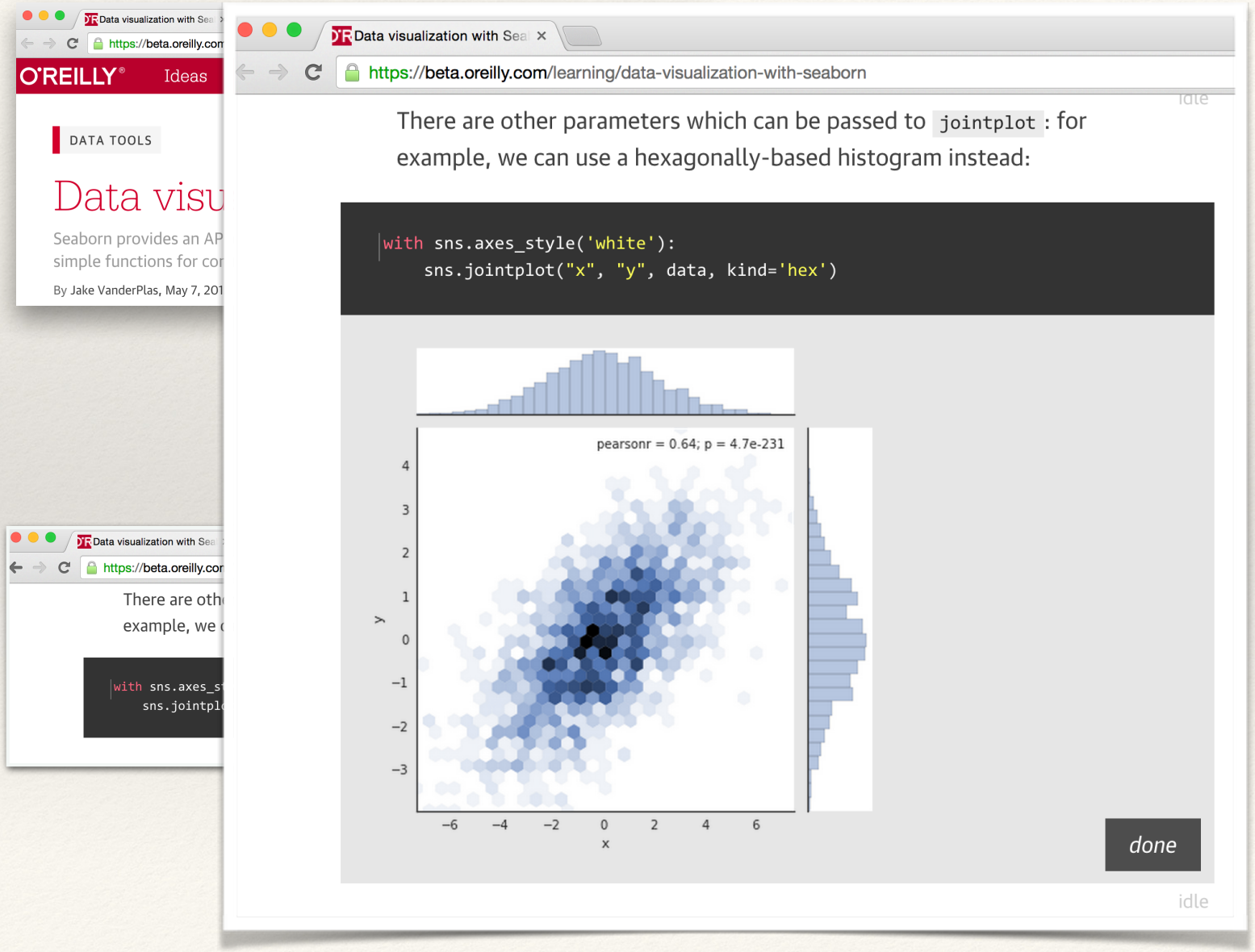


O'Reilly: authoring and delivering executable books



This is a live post!

O'Reilly: authoring and delivering executable books

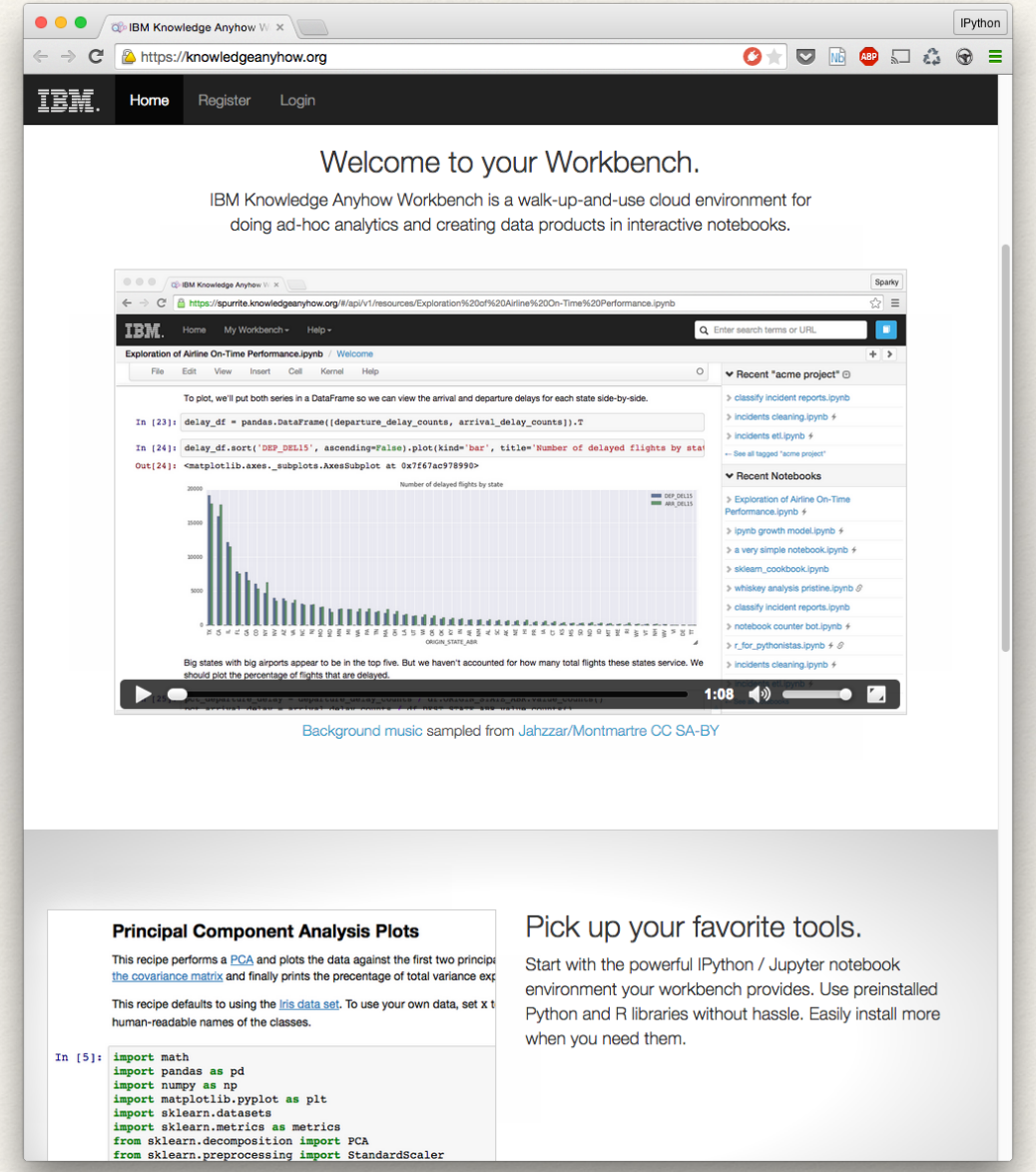


O'Reilly's Thebe: Jupyter Cells As a Service

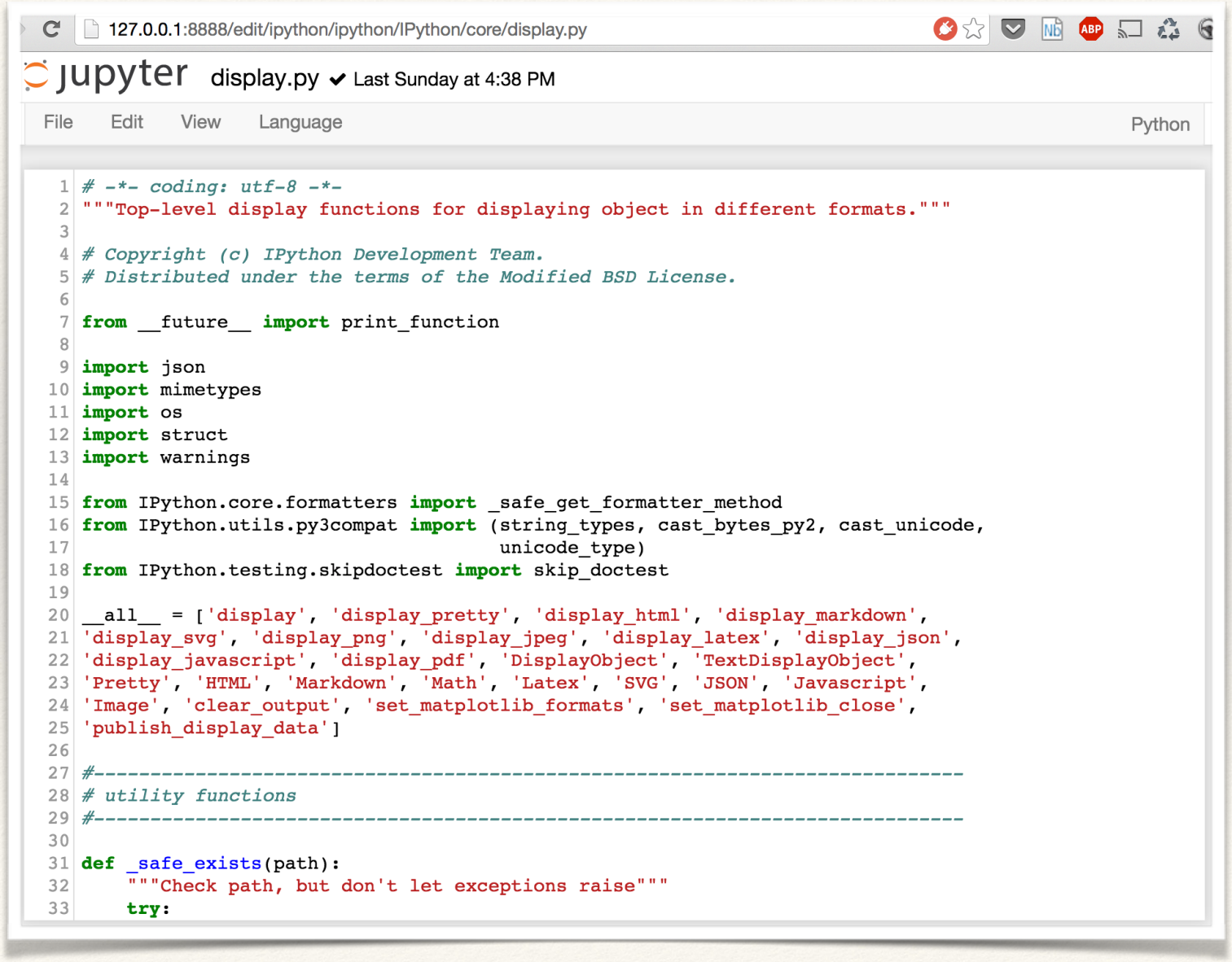
- Jupyter code cells that can be embedded in static web pages.
- Plugin by @zischwartz based on minrk/single cell.
- Supports widgets, plots, and the wide range of Jupyter display protocols.
- Separates serving of static assets from the notebook.

Credit: @odewahn, CTO of O'Reilly

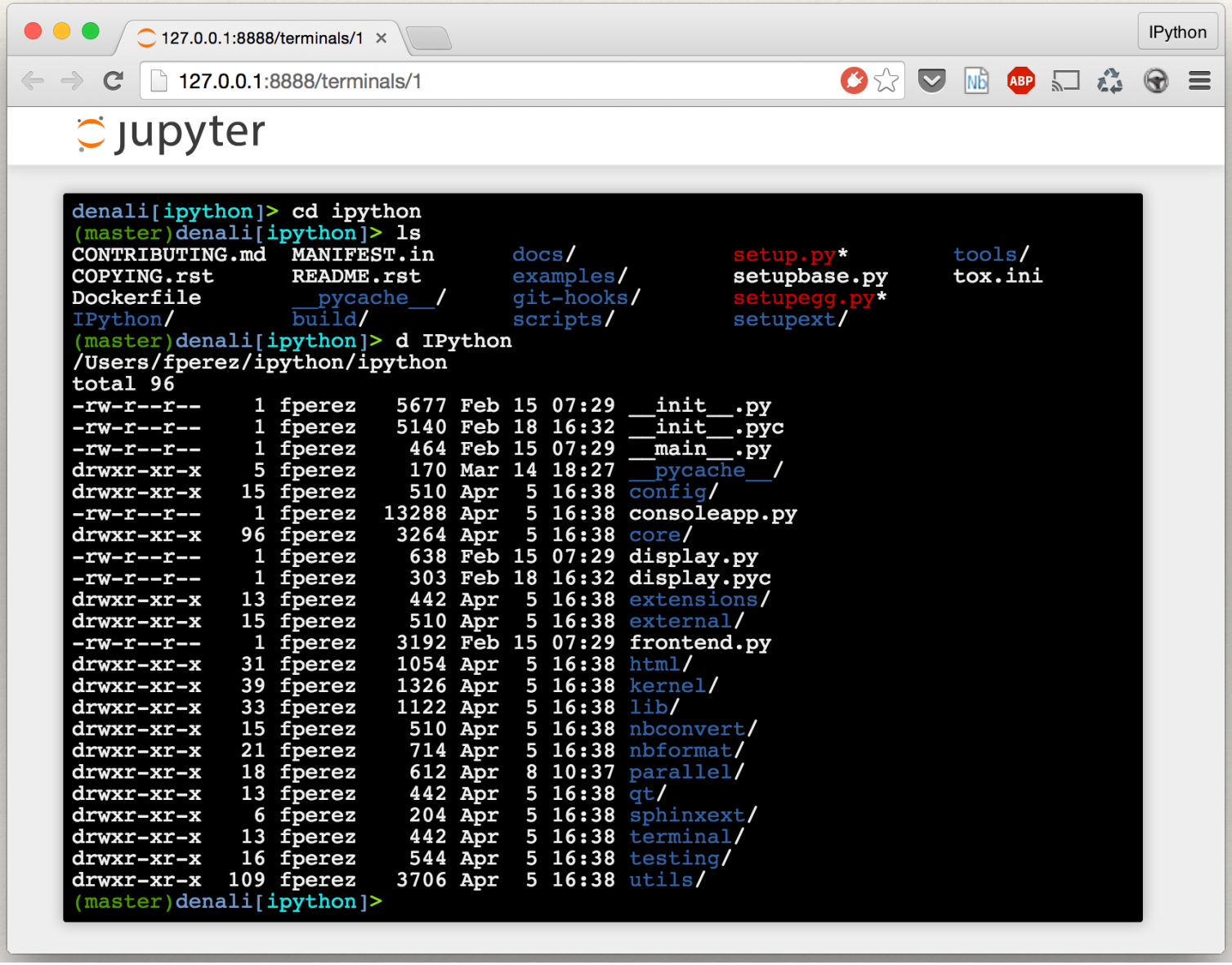
IBM Knowledge Anyhow



Full-page text editor



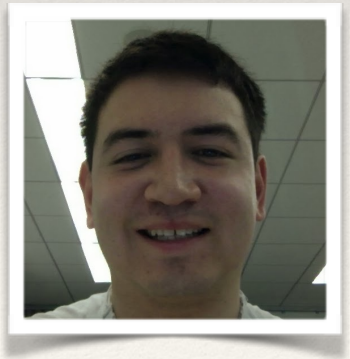
In-browser terminal (real-time sync)



Google CoLab: next steps

- Google Research funding a postdoc @ Berkeley. Thanks!
- Integrate real-time collaboration into Jupyter architecture.
  - First, supported on Google Drive.
  - Then, generalize, support other real-time backends.

Matthias Bussonnier
@ Berkeley



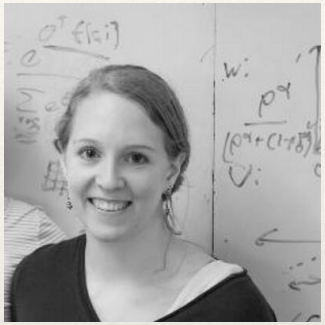
Kester Tong
@ Google

JupyterHub: multiuser support

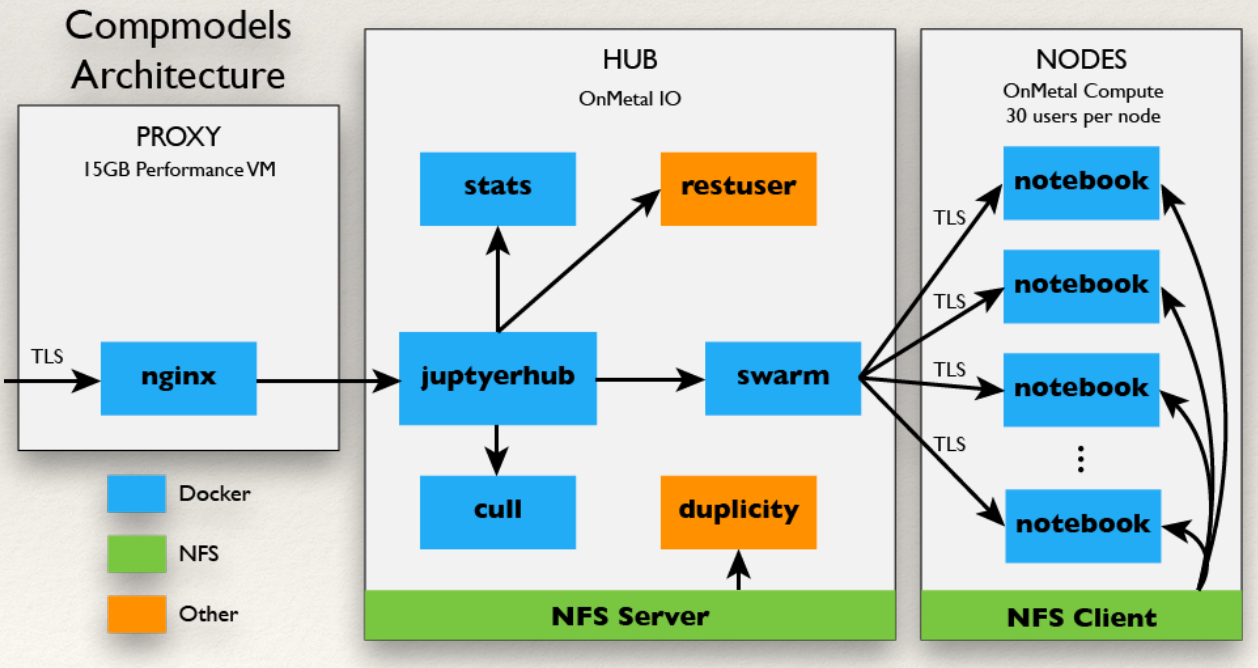
- Out of the box
  - Unix accounts
  - Local single-user notebooks
- Customizable
  - Authentication: OAuth, LDAP, etc.
  - Subprocess control: Docker, VMs, etc.

JupyterHub in Education @ Berkeley

- Computationally intensive course, ~220 students
- Fully hosted environment, zero-install
- Homework management and grading (w B. Granger)



Jess Hamrick @ Cal



K. Kelley
Rackspace

M. Ragan-Kelley
Cal

B. Granger
Cal Poly

POWERED BY rackspace® the open cloud company

https://developer.rackspace.com/blog/deploying-jupyterhub-for-education

Thank You!

@fperez_org fperez@lbl.gov

@ProjectJupyter @IPythonDev

Try it out at

try.jupyter.org