Interactive computing and Data Science with the Jupyter platform

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

LBL & UC Berkeley

Mandatory COI disclosure

“The purpose of computing is insight, not numbers”

–Hamming’62

The Lifecycle of a Scientific Idea (schematically)

1. Individual exploratory work
2. Collaborative development
3. Parallel production runs (HPC, cloud, ...)
4. Publication & communication (reproducibly!)
5. Education
6. Goto 1

The Lifecycle of a Scientific Idea (schematically)

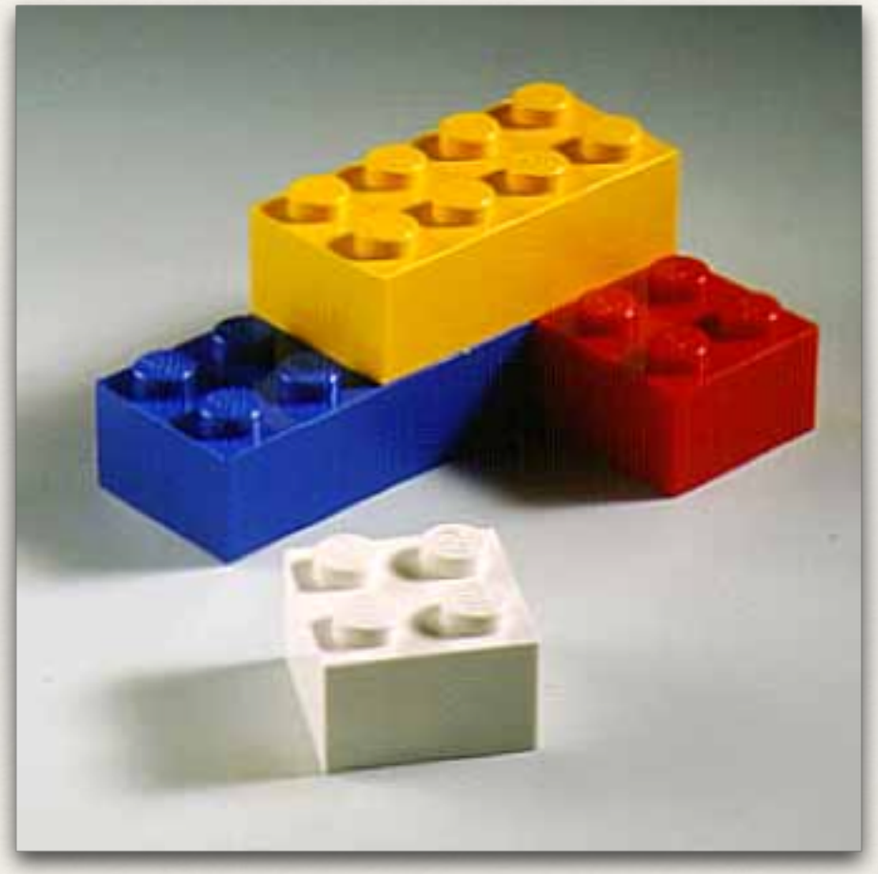
1. Individual exploratory work
2. Collaborative development
3. Parallel production runs (HPC, cloud, ...)
4. Publication & communication (reproducibly!)
5. Education
6. Goto 1

We treat this as a single, coherent problem

Interactive Computing?

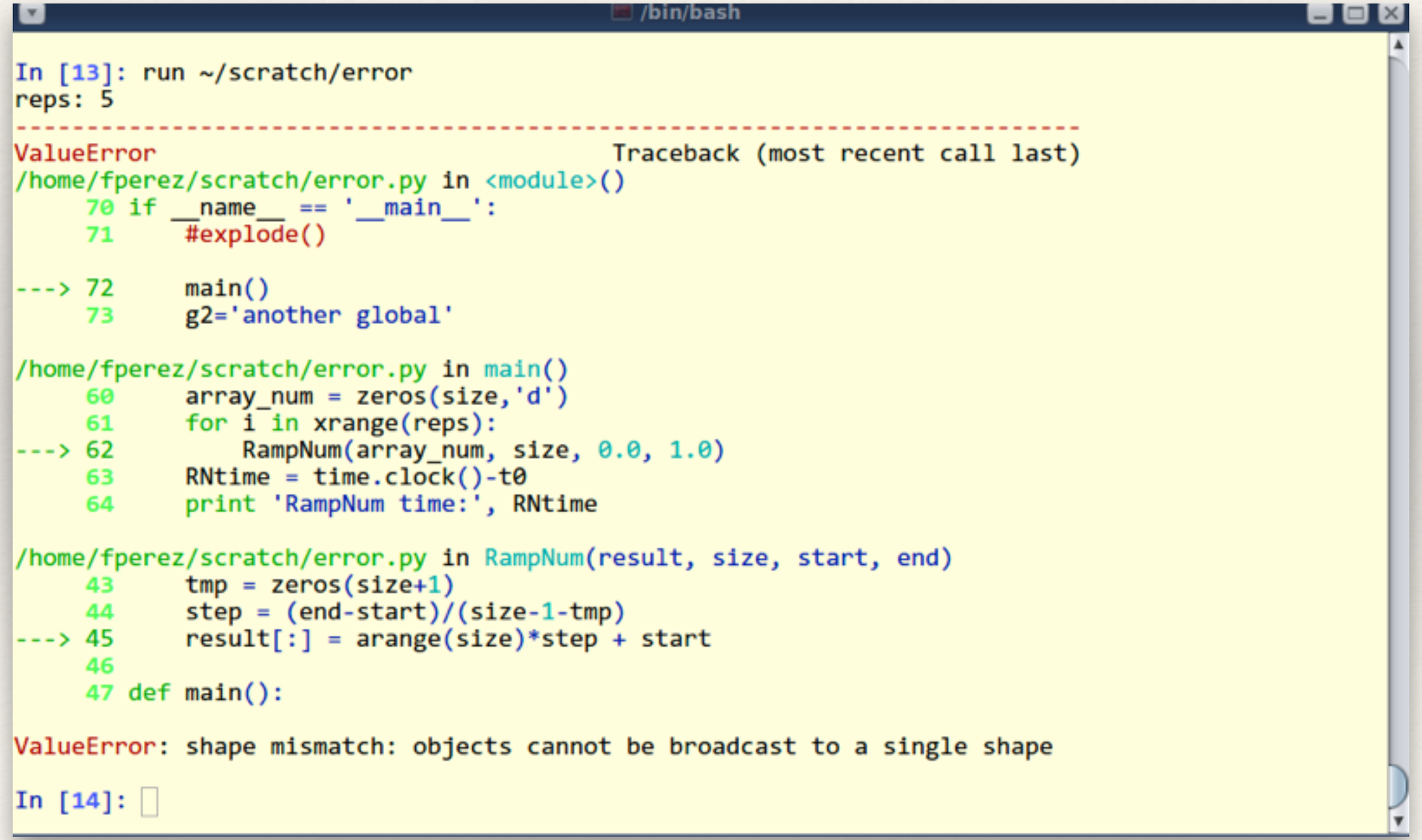
- The venerable REPL idea
  - Read, Eval, Print, Loop
- From the Unix command line to Matlab or Mathematica

Why a REPL?



IPython: CU Boulder, 2001

or how to best procrastinate on a Physics dissertation



November 2001: "Just an afternoon hack"

- 259 Line Python script.
- Plotting, Numerics, interactive workflow.

Today

- > 500 contributors
- Multiple GitHub orgs, > 120 repos
- Hundreds of thousands of LOC, multiple languages
- At least 4 million users
- ~ $ 8M in direct funding
- ~25 full-time developers (academia, industry, non-profits)

Team today: where *all the credit* goes

Plus ~ 500 more Open Source contributors!

Funding and partnerships

ALFRED P. SLOAN FOUNDATION

THE LEONA M. AND HARRY B. HELMSLEY CHARITABLE TRUST

GORDON AND BETTY MOORE FOUNDATION

U.S. DEPARTMENT OF ENERGY

NEW! I'm hiring!

SIMONS FOUNDATION

CONTINUUM ANALYTICS

IBM

POWERED BY rackspace the open cloud company

Microsoft

ENTHOUGHT SCIENTIFIC COMPUTING SOLUTIONS

Google

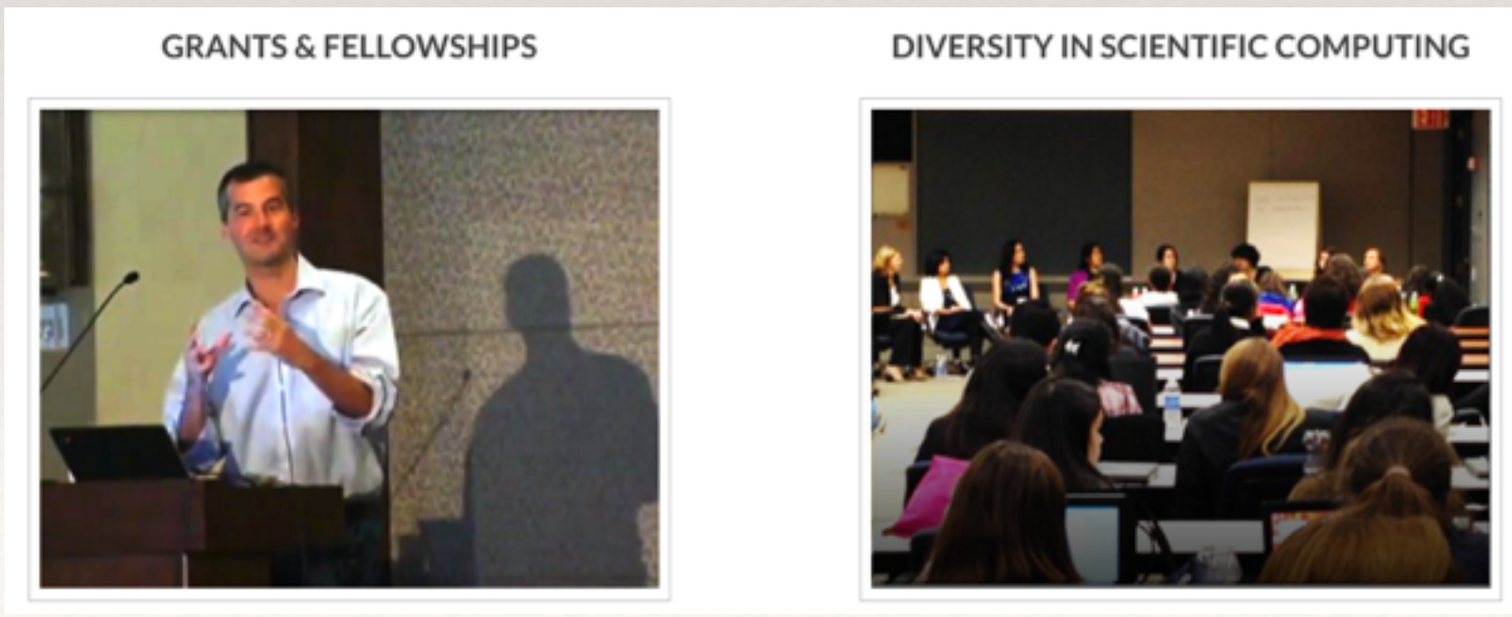
Bloomberg

NumFOCUS: *Scientific* OSS Foundation

NUMFOCUS

OPEN CODE = BETTER SCIENCE

- 501(c)3, scientific focus
- NOT Python Specific
- Bridges between
  - Open community
  - Academic research
  - Industry
  - Private philanthropy
  - Government funding
- **Sustainability**



Generalizing and abstracting interactive computing

Abstracting the REPL

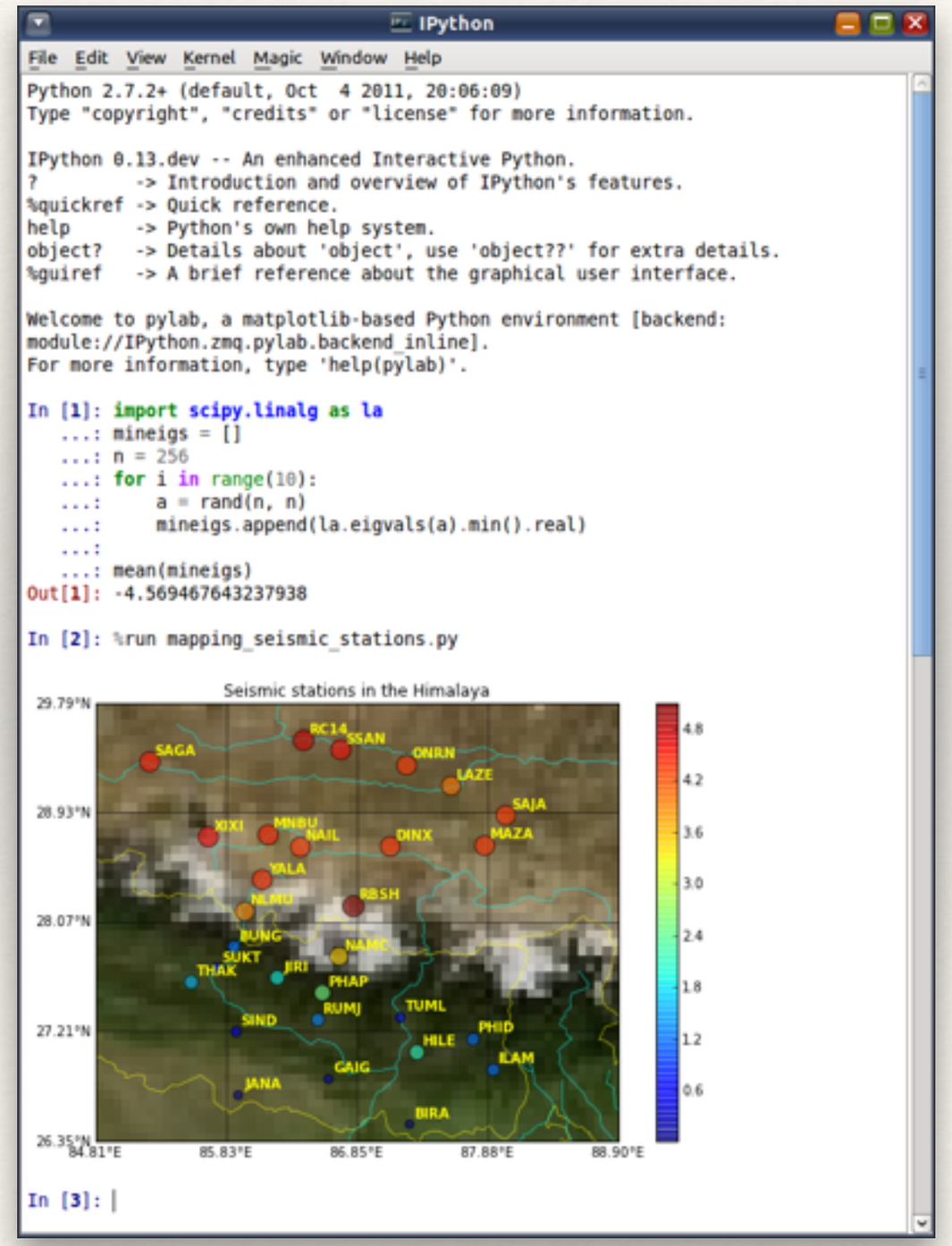
- Pair programming (Brian Granger/FP), 24 hours of design/code, less than 500 LOC total.
- Live, over-the-wire developing and debugging.
- Designed protocol, client and server together.
  - 1st terminal: `./kernel.py`
  - 2nd terminal: `./frontend.py`

Code: https://github.com/fperez/zmq-pykernel

Beyond the Terminal…

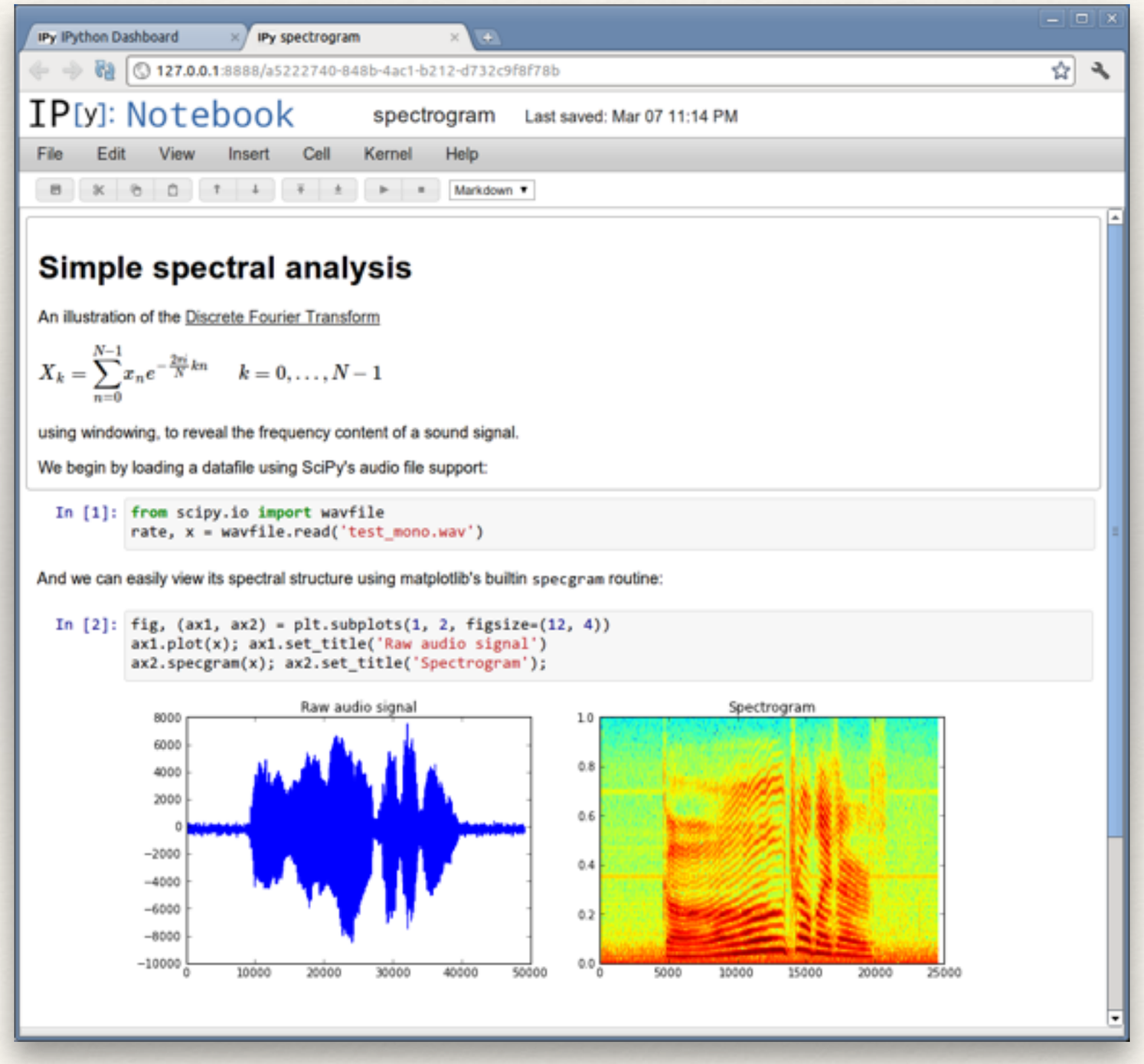
- The REPL as a network protocol
- Kernels
  - execute code
- Clients
  - Read input
  - Present output

Simple abstractions enable rich, sophisticated clients



2011: The IPython Notebook

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



Notebooks: “Literate Computing”

Computational Narratives

- Computers deal with *code and data*.
- Humans deal with narratives that *communicate*.

Literate Computing (*not Knuth's* Literate Programming)

narratives anchored in a live computation, that communicate a story based on data and results.

Cf: Mathematica, Maple, MuPad, Sage…

From IPython to Project Jupyter

IP[y]:
IPython

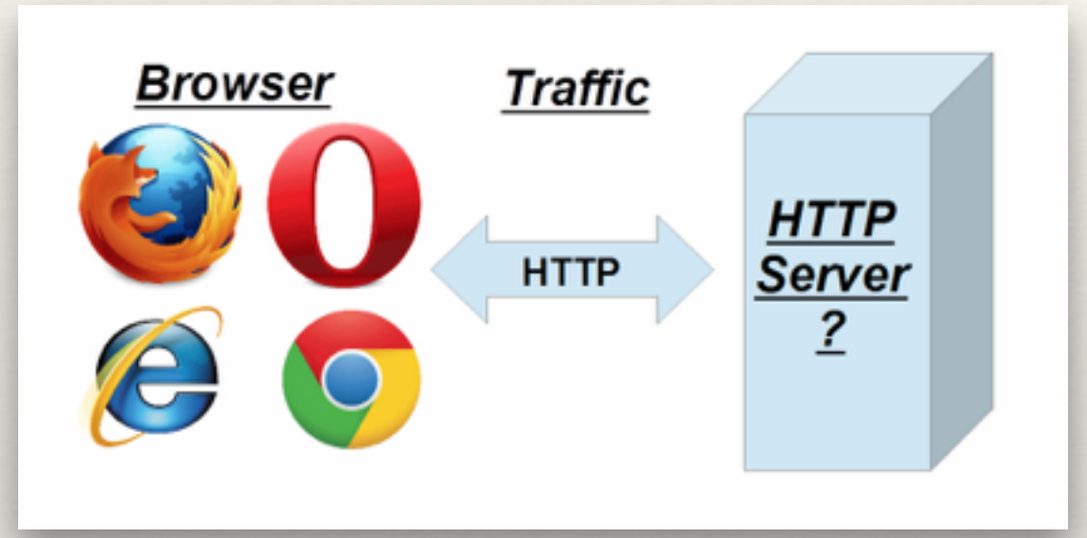
IPython … Jupyter

- 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

Core ideas of the web: HTTP & HTML



HTTP: protocol to connect clients and servers
HyperText Transport Protocol

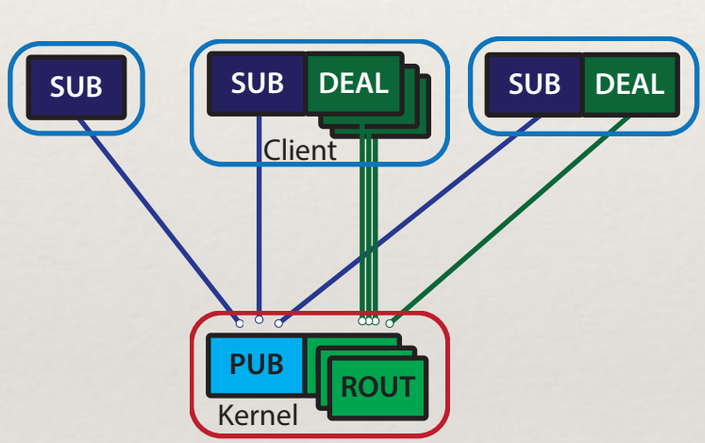
Image credit: eviltester.com

```
    <a href="/" rel="home" title="University of Colorado Boulder " class="custom-logo-link
active"><img class="custom-logo custom-logo-white"
src="http://www.colorado.edu/sites/default/files/custom_logo/beboulder-logo-2x.png"
alt="University of Colorado Boulder  logo" /></a><a href="/" rel="home" title="University of
Colorado Boulder " class="custom-logo-link active"><img class="custom-logo custom-logo-
black" src="http://www.colorado.edu/sites/default/files/custom_logo/beboulder-logo-2x_0.png"
alt="University of Colorado Boulder  logo" /></a>  <div class="element-invisible">
          <div class="header__name-and-slogan" id="name-and-slogan">
                <h1 class="header__site-name" id="site-name">
            <a href="/" title="Home" class="header__site-link" rel="home"><span>University
of Colorado Boulder </span></a>
          </h1>
```

HTML: format to represent content
HyperText Markup Language

Core ideas of Jupyter

Interactive Computing Protocol



ØMQ + JSON

Document Format

We have already computed $P(X|A)$ above. On the other hand, $P(X| \sim A)$ is subjective: our code can pass tests but still have a bug in it, though the probability there is a bug present is reduced. Note this is dependent on the number of tests performed, the degree of complication in the tests, etc. Let's be conservative and assign $P(X| \sim A) = 0.5$. Then

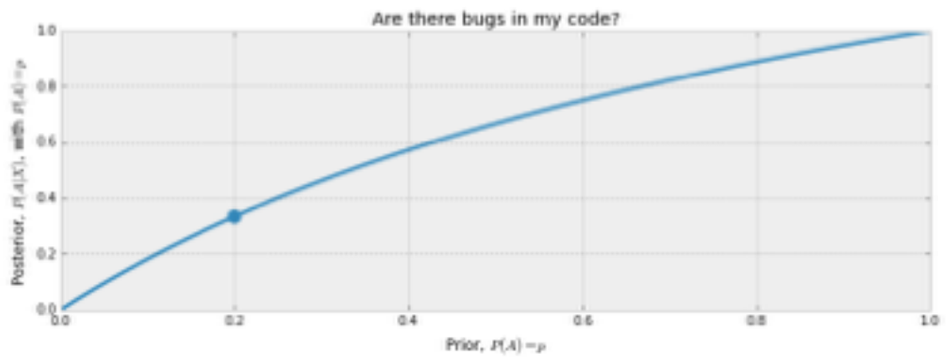
$$P(A|X) = \frac{1 \cdot p}{1 \cdot p + 0.5(1-p)}$$

$$= \frac{2p}{1+p}$$

This is the posterior probability. What does it look like as a function of our prior, $p \in [0, 1]$?

```
figsize(12.5, 4)
p = np.linspace(0, 1, 50)
plt.plot(p, 2 * p / (1 + p), color="#348ABD", lw=3)
# plt.fill_between(p, 2*p/(1+p), alpha=.5, facecolor=["#A60628"])
plt.scatter(0.2, 2 * (0.2) / 1.2, s=140, c="#348ABD")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel("Prior, $P(A) = p$")
plt.ylabel("Posterior, $P(A|X)$, with $P(A) = p$")
plt.title("Are there bugs in my code?")
```

```
<matplotlib.text.Text at 0x1051de650>
```



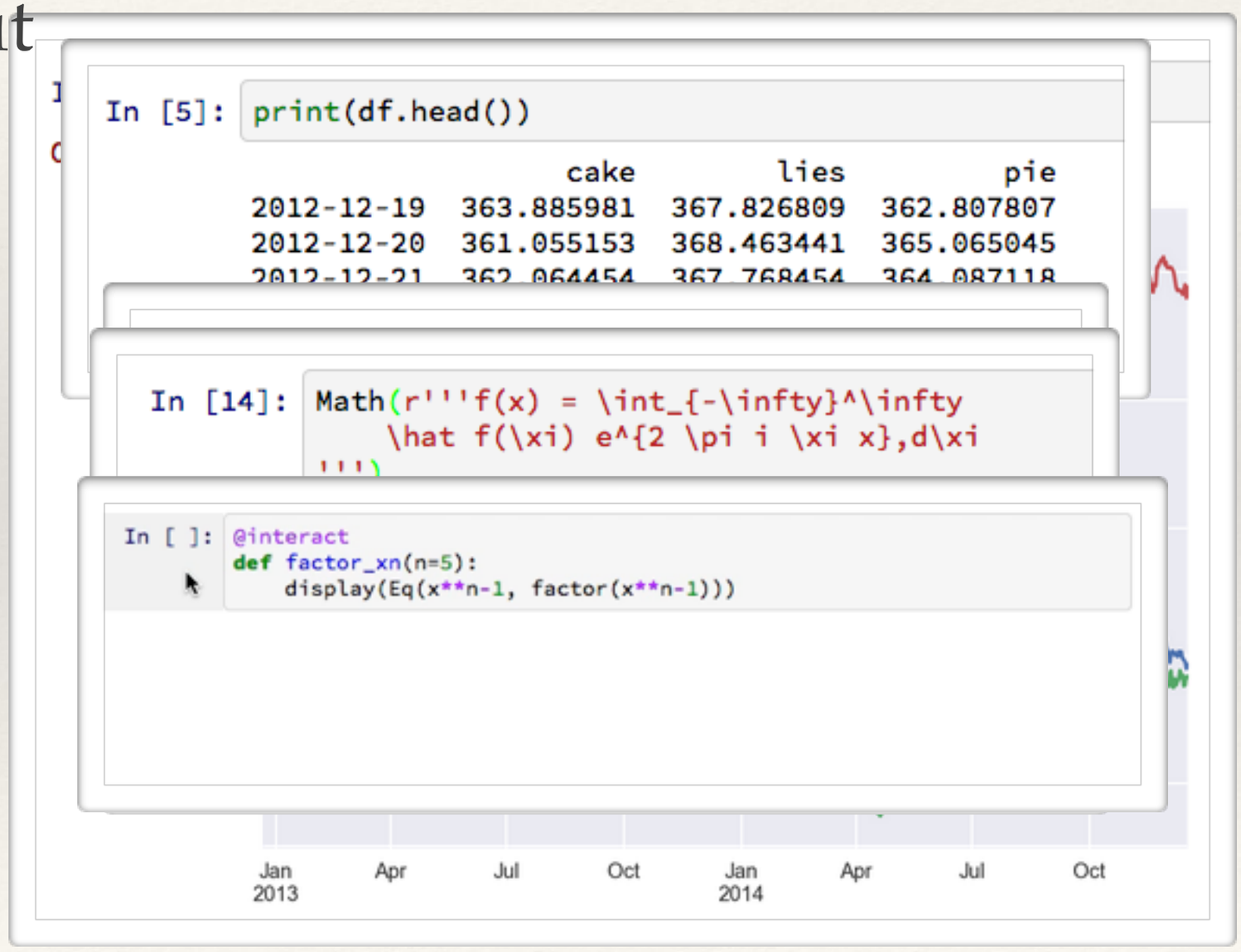
https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers

Jupyter Protocol

capture the process of interactive computing

any mime-type output

- text
- svg, png, jpeg
- latex, pdf
- html, javascript
- interactive widgets



Jupyter Protocol
is language agnostic

~75 different kernels: https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages

Reproducible research: from cancer and microbes…

Reproducible Research

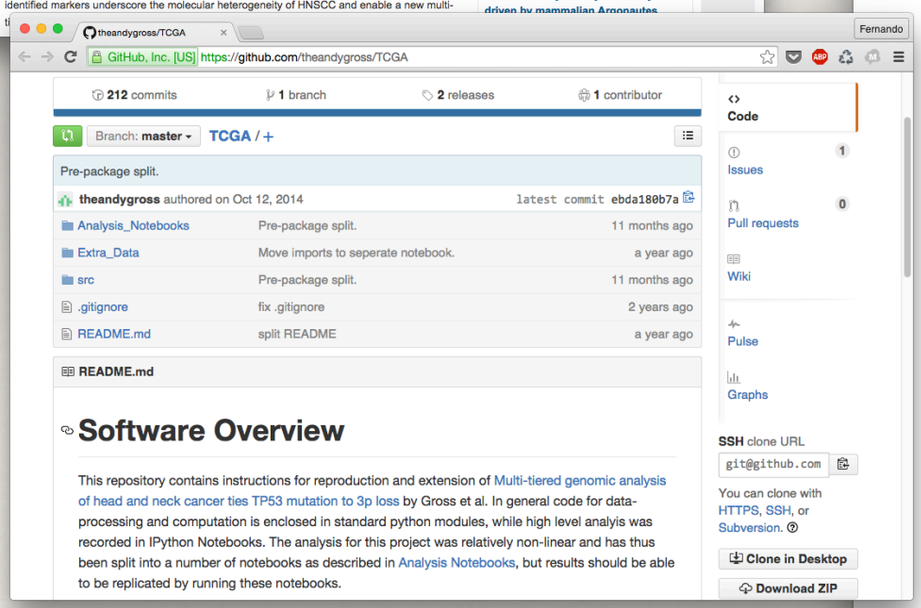
An article about computational science in a scientific publication is **not** the scholarship itself, it is merely **advertising** of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.

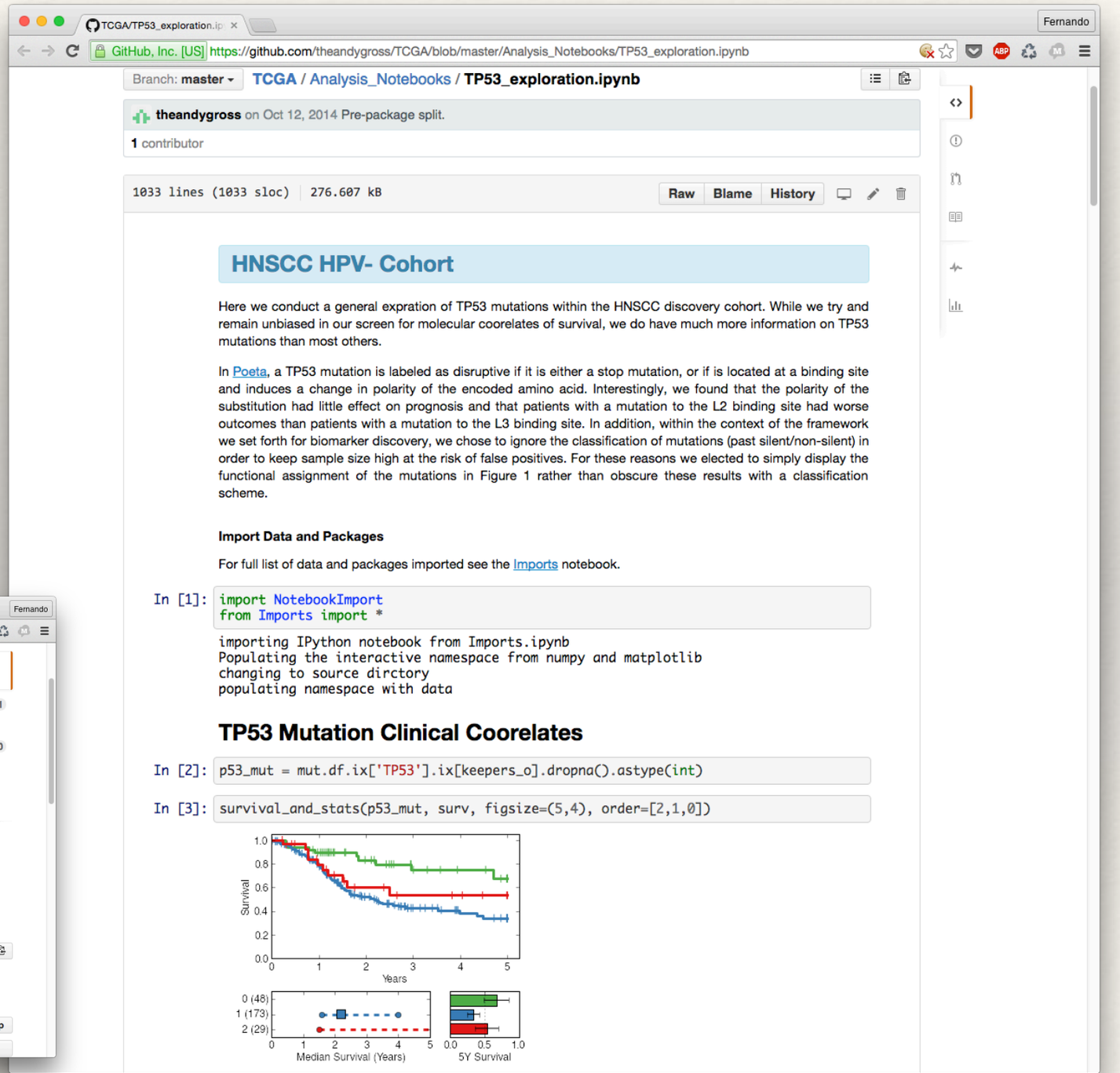
*Buckheit and Donoho, WaveLab and Reproducible Research, **1995***

Nature: “the advertising”

Gross, Andrew M., et al. Nature genetics 46.9 (2014): 939-943.

Notebooks on Github: the “actual scolarship”



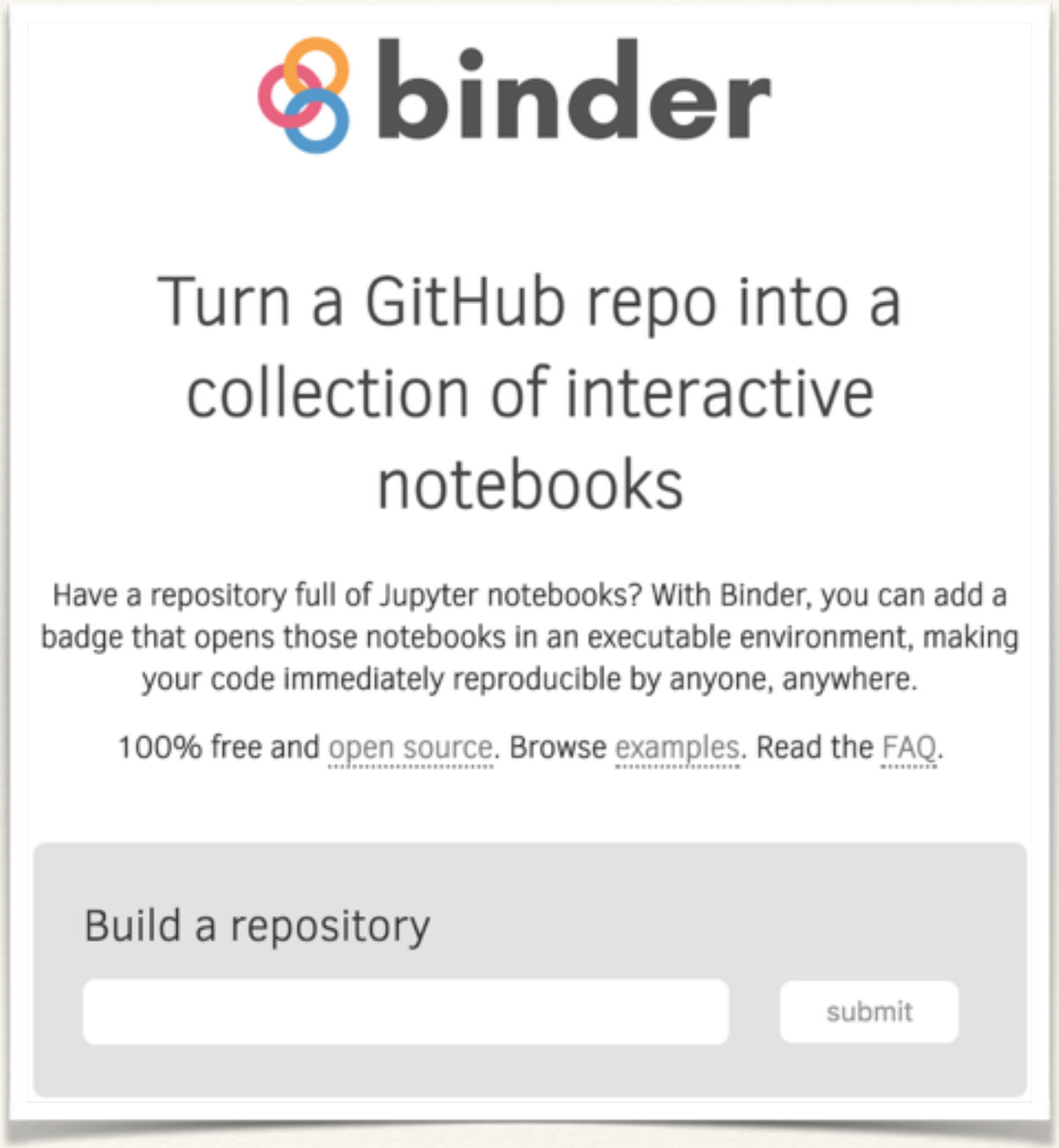


Reproducible Research (2012):
Paper, Notebooks and Virtual Machine

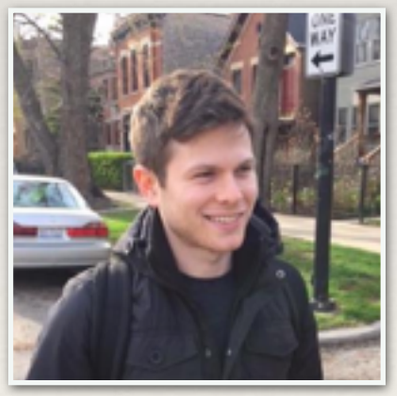
http://www.nature.com/ismej/journal/v7/n3/full/ismej2012123a.html

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

Today: mybinder.org



github.com/freeman-lab

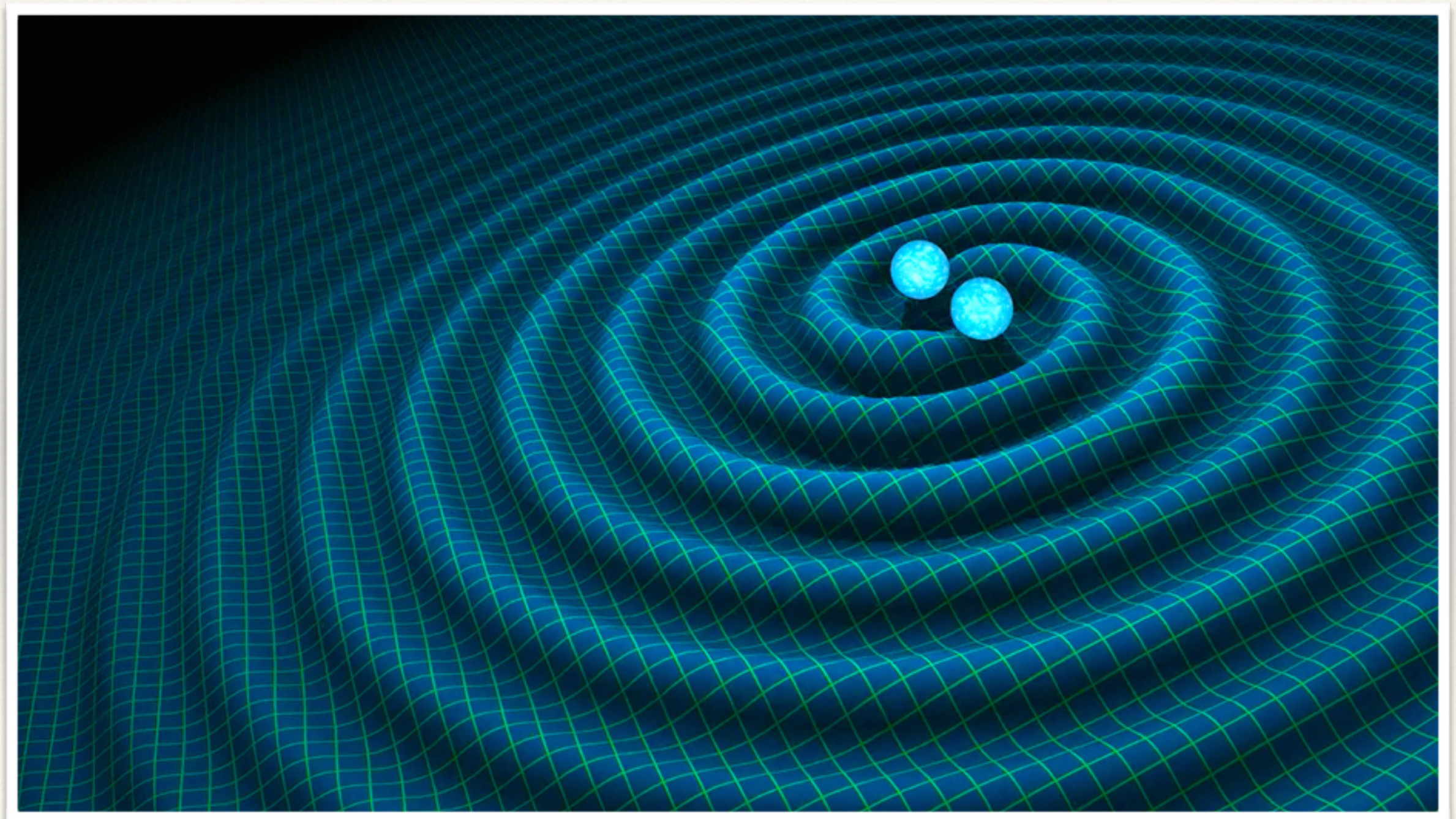


github.com/andrewosh

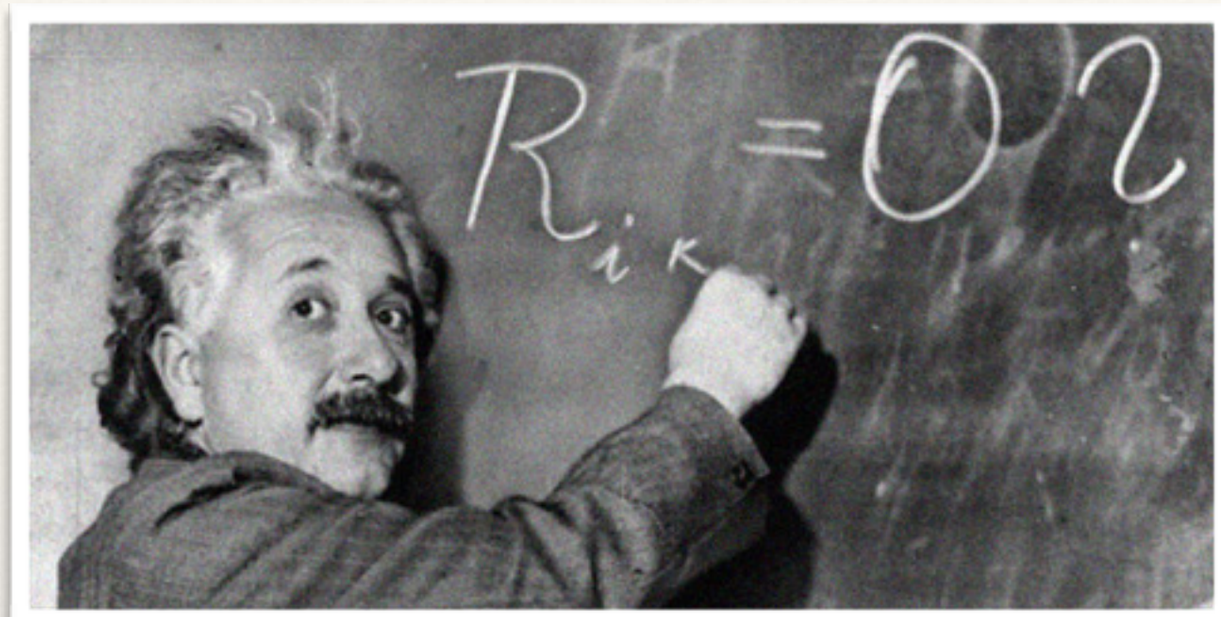
Andrew Osheroff's SciPy'16 talk:
https://www.youtube.com/watch?v=OK6M4w7LYIc

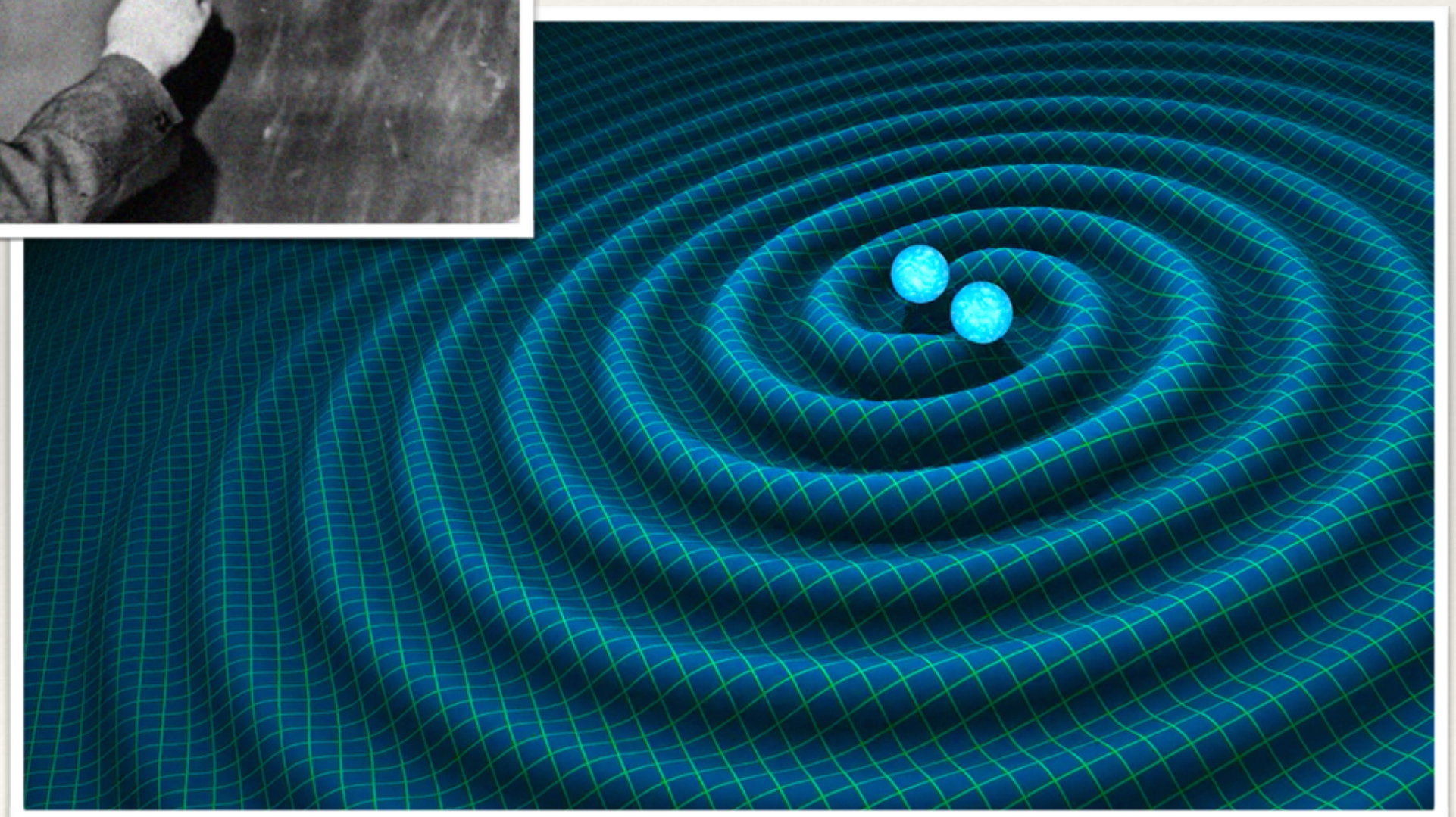
… to black holes and deep learning

A long time ago in a galaxy far, far away…



A long time ago in a galaxy far, far away…





Two identical detectors: Hanford, WA and Livingston, LA

LIGO: a feat of science & engineering

Detection problem:

- ~ 1/1000 proton over 4 km.
- Sensitivity ~ 1e-21
- Milky Way: 1e+21m across!

September 14, 2015

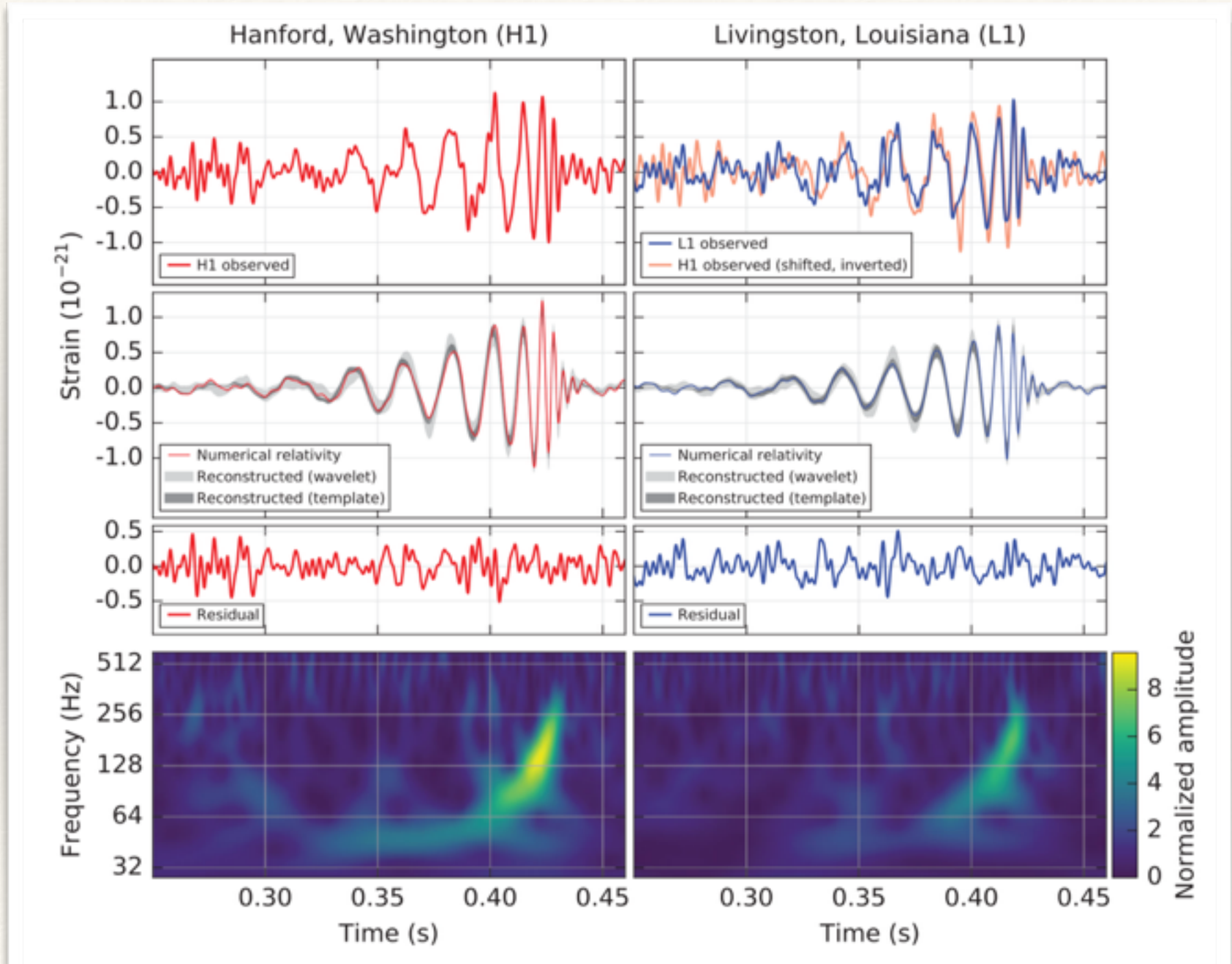
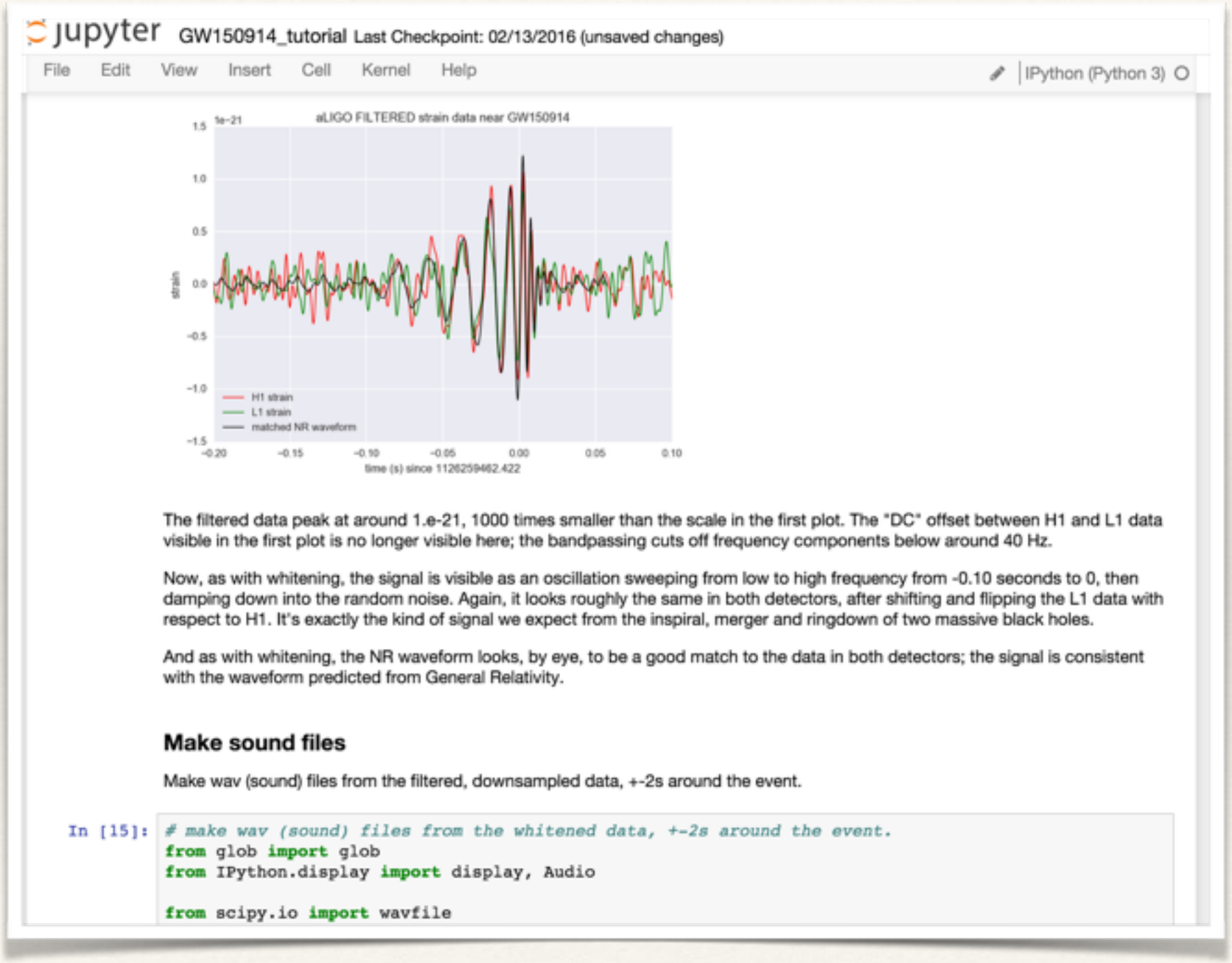


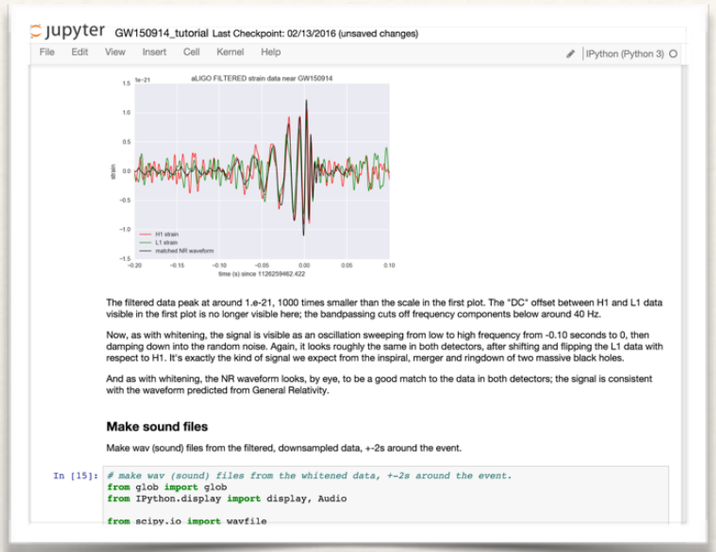
FIG. 1. The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress large fluctuations outside the detectors' most sensitive frequency band, and band-reject

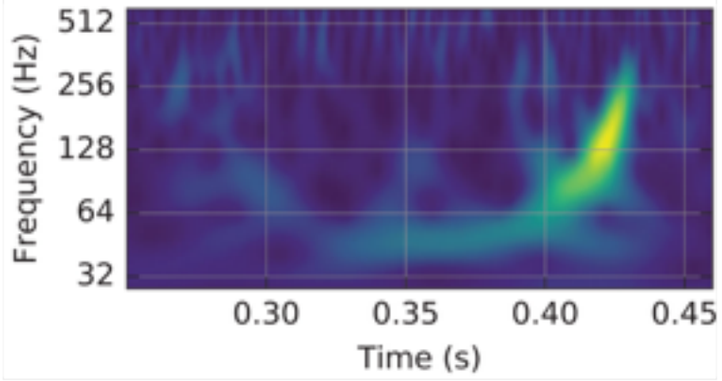
Gravitational waves detected on Jupyter!



From LIGO Open Science Center, binder-ified: github.com/minrk/ligo-binder

The song of the universe





Make sound files

Make wav (sound) files from the filtered, downsampled data, +-2s around the event.

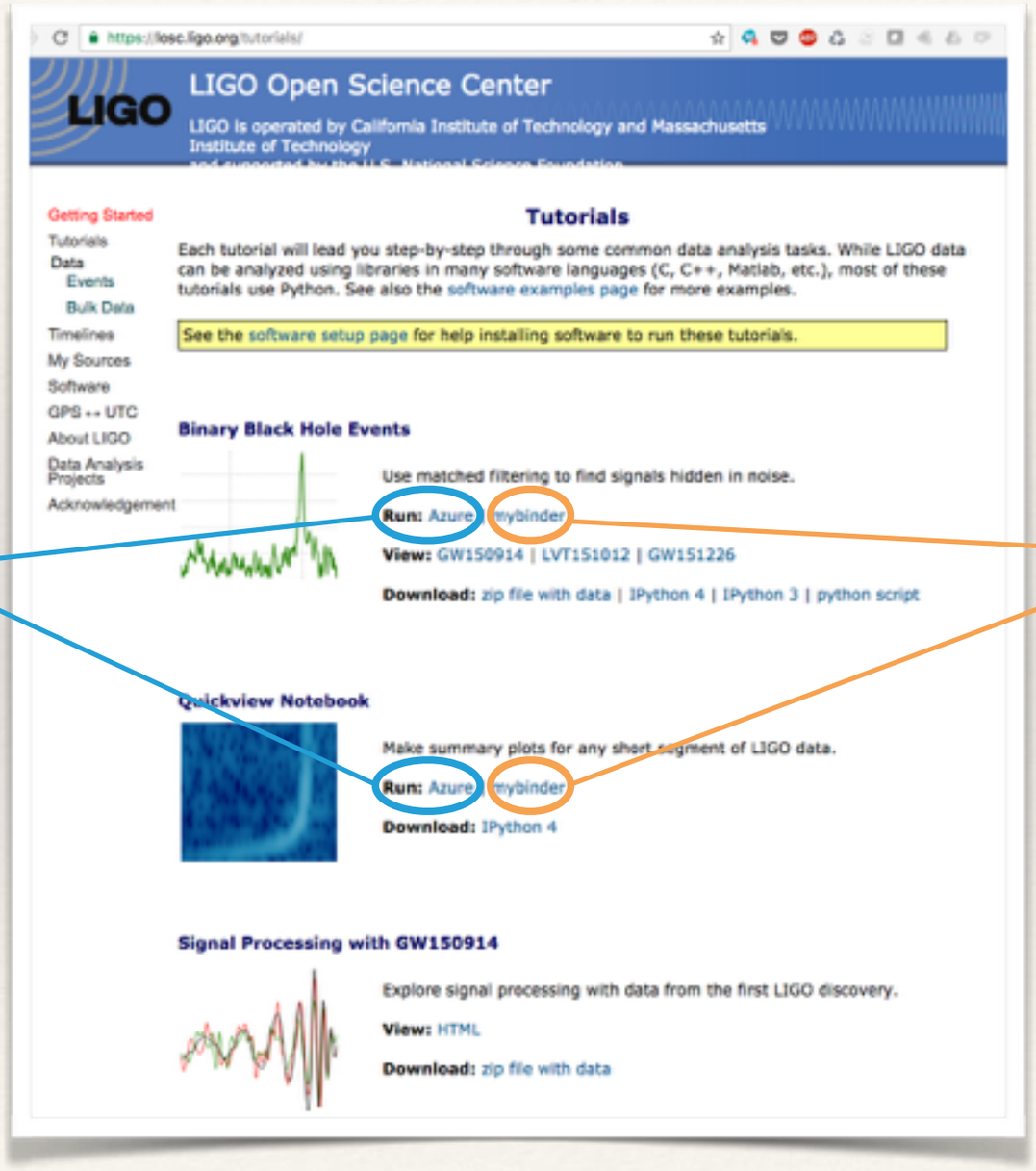
```
# make wav (sound) files from the whitened data, +-2s around the event.
from glob import glob
from IPython.display import display, Audio

from scipy.io import wavfile

# function to keep the data within integer limits, and write to wavfile:
def write_wavfile(filename,fs,data):
    d = np.int16(data/np.max(np.abs(data)) * 32767 * 0.9)
    wavfile.write(filename,int(fs), d)

tevent = 1126259462.422           # Mon Sep 14 09:50:45 GMT 2015
deltat = 2.                       # seconds around the event

# index into the strain time series for this time interval:
indxt = np.where((time >= tevent-deltat) & (time < tevent+deltat))

# write the files:
write_wavfile("GW150914_H1_whitenbp.wav",int(fs), strain_H1_whitenbp[indxt])
write_wavfile("GW150914_L1_whitenbp.wav",int(fs), strain_L1_whitenbp[indxt])
write_wavfile("GW150914_NR_whitenbp.wav",int(fs), NR_H1_whitenbp)

for wav in glob('*whitenbp.wav'):
    display(wav)
    display(Audio(filename=wav))
```

```
'GW150914_H1_whitenbp.wav'
```

0:00

Using the IPython.display.Audio object

LIGO: Open Science with Jupyter



an anecdote from a recent trip…

A recent paper: a physics look at deep learning


MIT Technology Review

Topics+ Top Stories Magazine

Computing

The Extraordinary Link Between Deep Neural Networks and the Nature of the Universe

Nobody understands why deep neural networks are so good at solving complex problems. Now physicists say the secret is buried in the laws of physics.

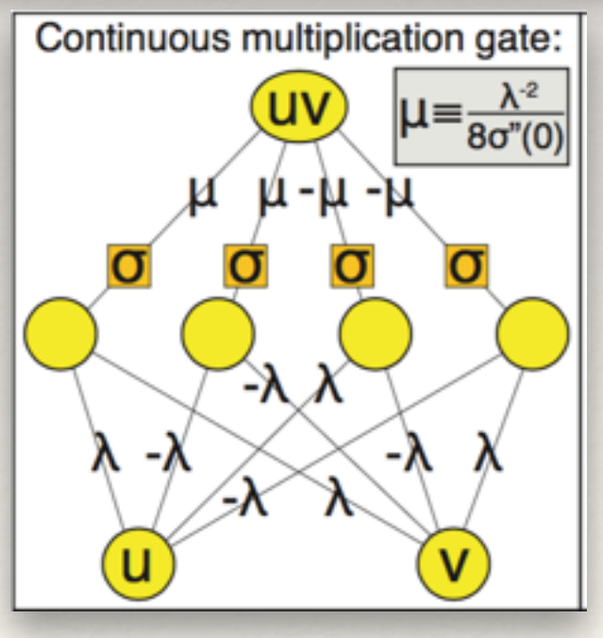
by Emerging Technology from the arXiv September 9, 2016


Why does deep and cheap learning work so well?

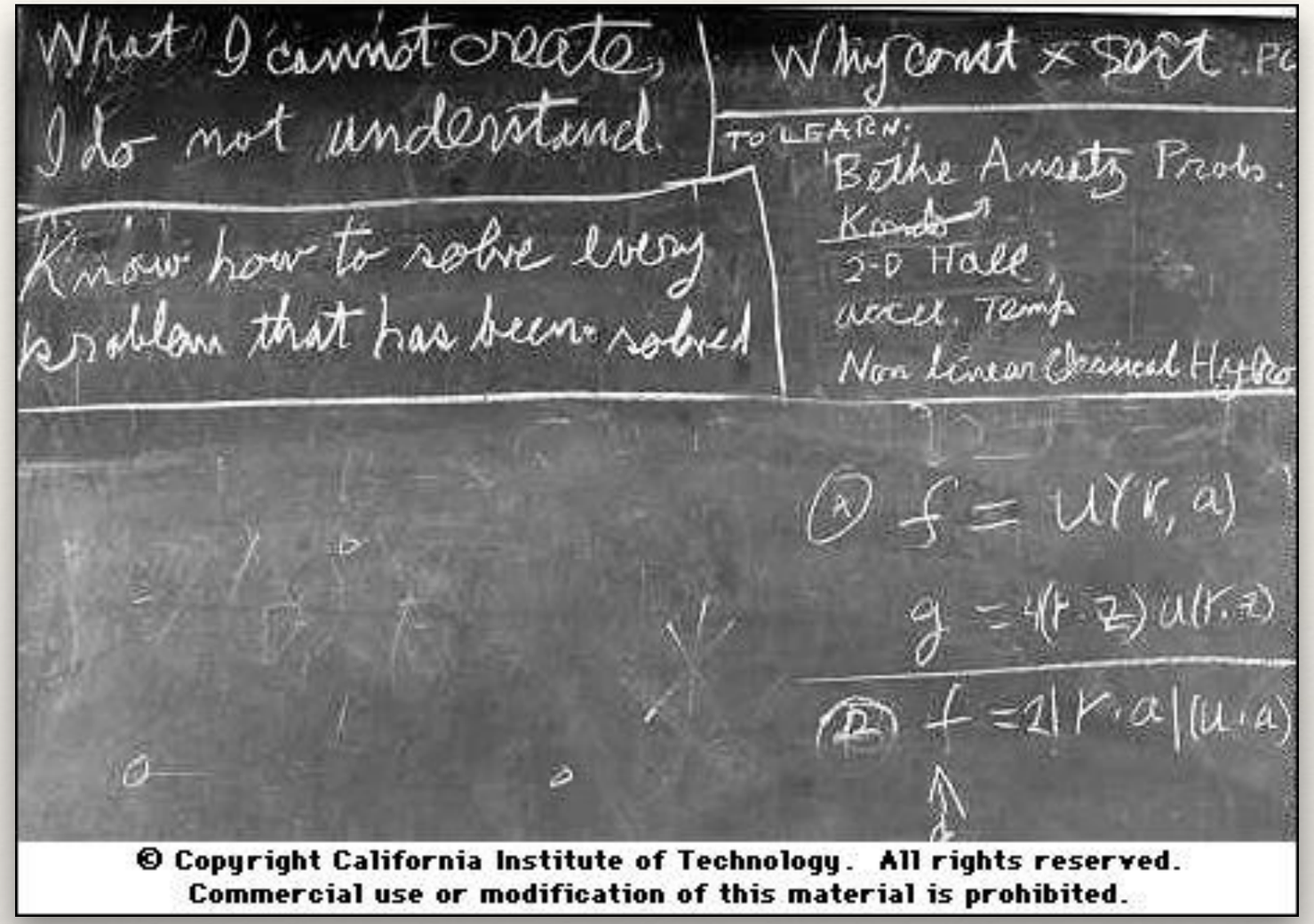
Henry W. Lin and Max Tegmark
Dept. of Physics, Harvard University, Cambridge, MA 02138 and
Dept. of Physics & MIT Kavli Institute, Massachusetts Institute of Technology, Cambridge, MA 02139

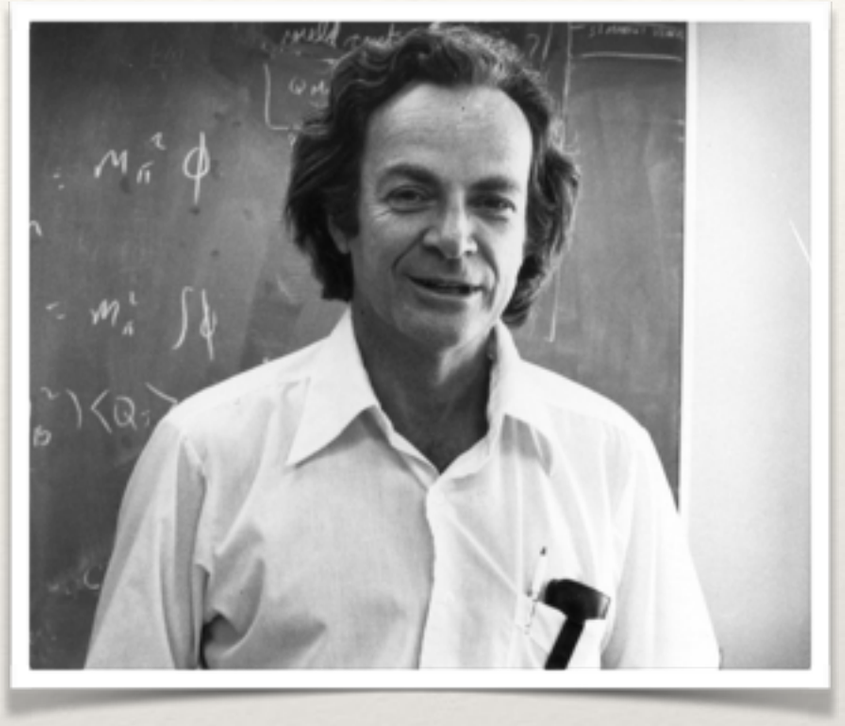
(Dated: August 31, 2016)

Theorem: Let $\mathbf{f}$ be a neural network of the form $\mathbf{f} = \mathbf{A}_2\boldsymbol{\sigma}\mathbf{A}_1$, where $\boldsymbol{\sigma}$ acts elementwise by applying some smooth non-linear function σ to each element. Let the input layer, hidden layer and output layer have sizes 2, 4 and 1, respectively. Then $\mathbf{f}$ can approximate a multiplication gate arbitrarily well.



https://arxiv.org/abs/1608.08225

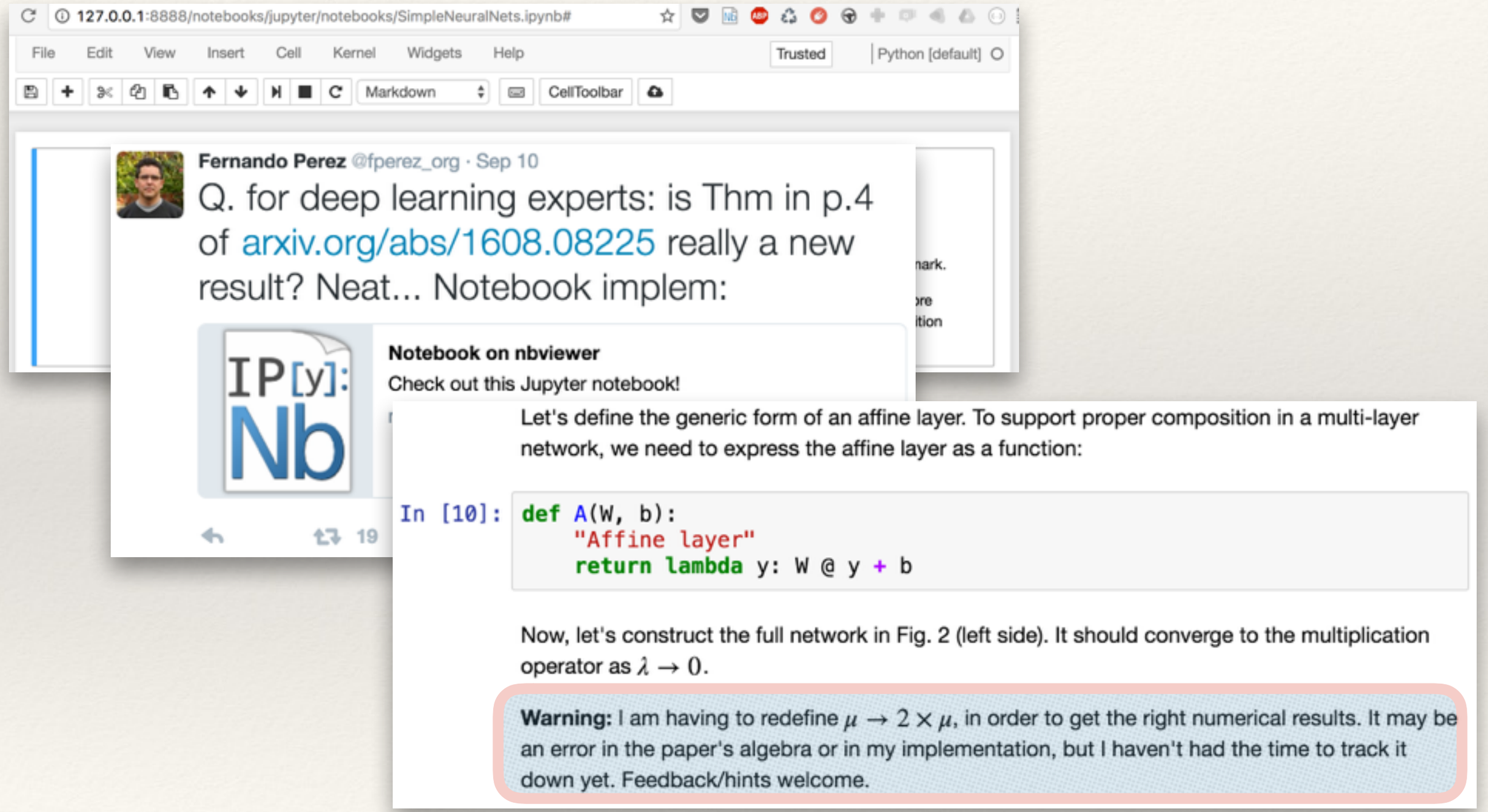




What I can not create, I do not understand

Richard Feynman

Let's implement it!



https://gist.github.com/fperez/c7b1cb4810f9d0935e893f34c41f0c62

Five hours later, Lin (author) responds

hwlin76 commented 3 days ago

Thanks for the interest in our paper: you are right about the factor of 2! It is correct in equation (11) but incorrect in the figure. We are correcting it in the next draft and will acknowledge you appropriately!

Matteo Visconti dOC @contematto · 5h

What is just happening on this gist is how science should work in the future. No, not the future, now!

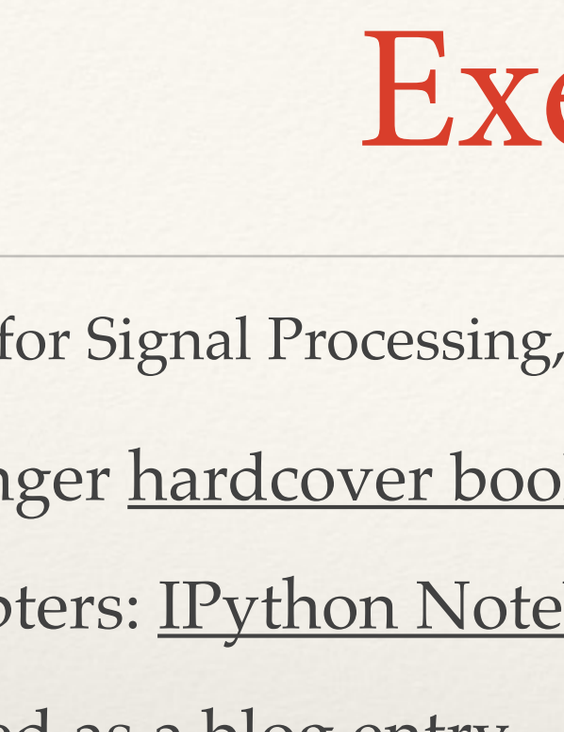
> **Fernando Perez** @fperez_org
> Delighted to see one of the authors (Lin) respond and clarify the mystery factor of 2 was an error in the figure: gist.github.com/fperez/c7b1cb4…

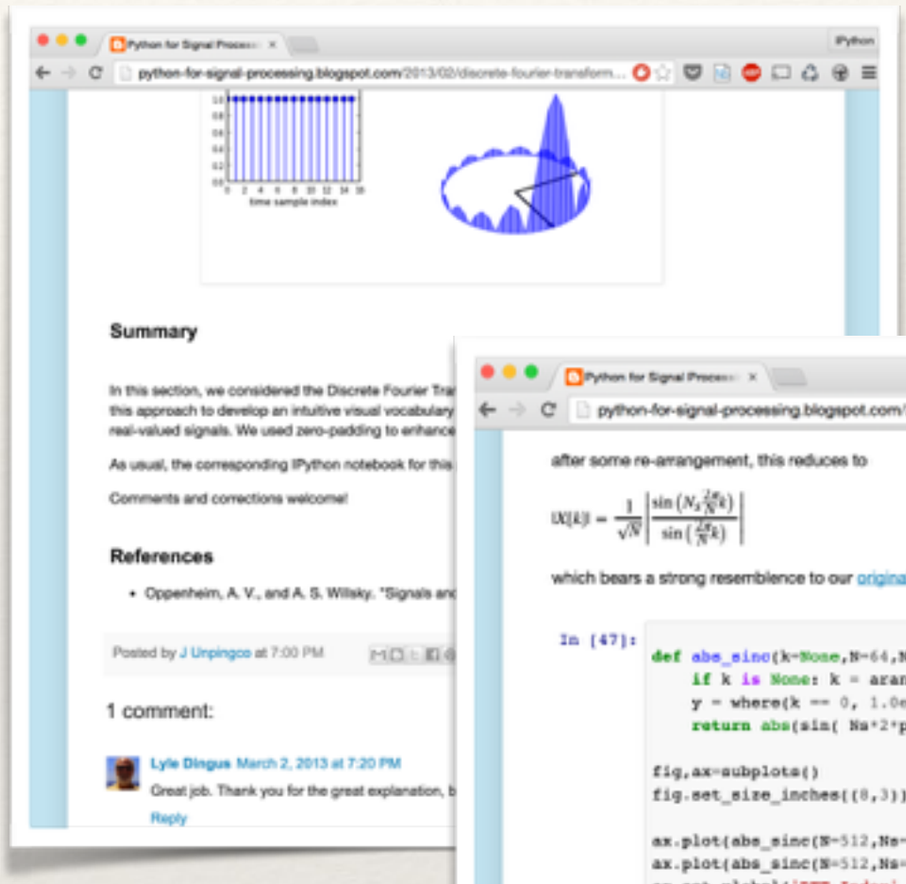
6 11

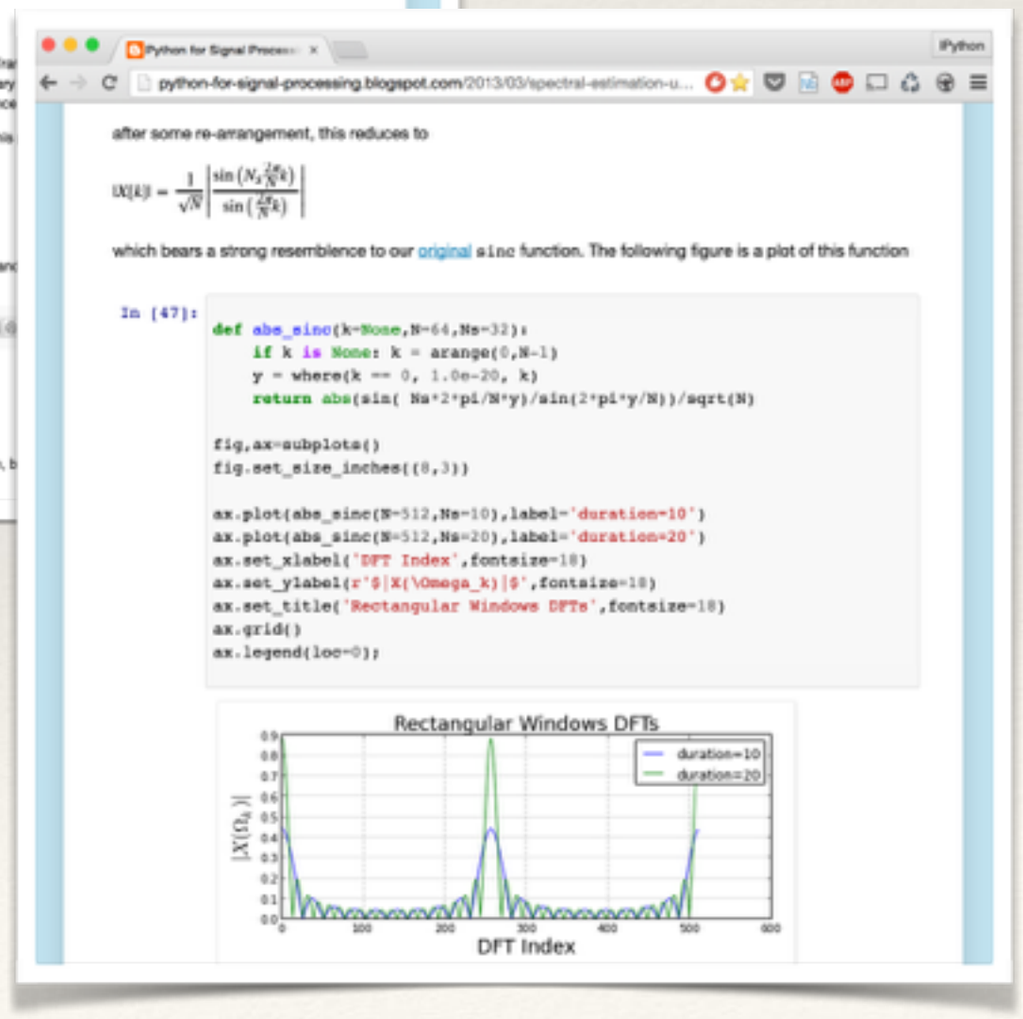
Executable books

Python for Signal Processing, by José Unpingco

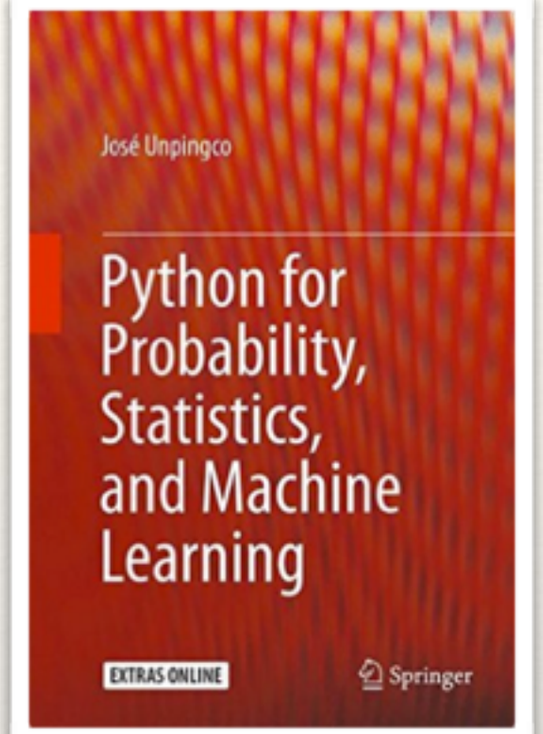
- Springer hardcover book
- Chapters: IPython Notebooks
- Posted as a blog entry
- All available as a Github repo



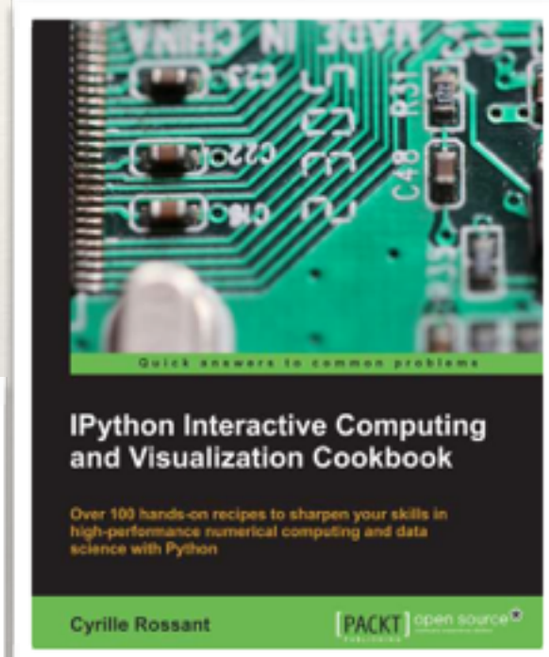




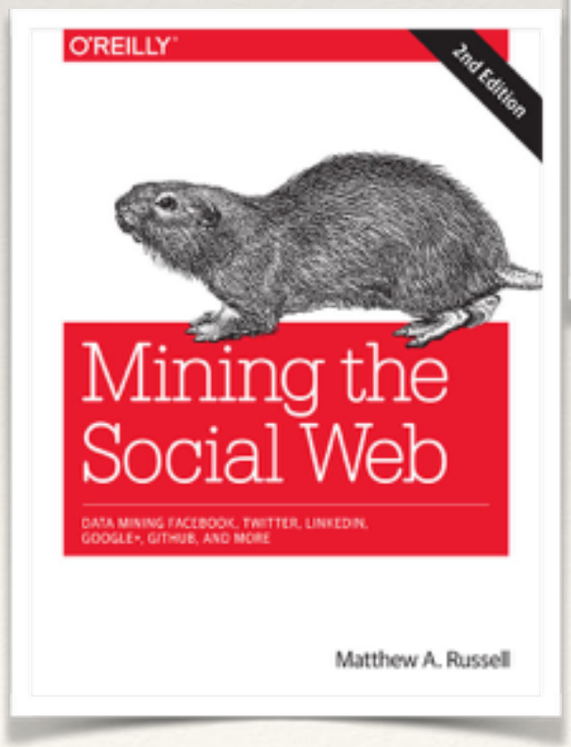
Not the only book



By Jose Unpingco



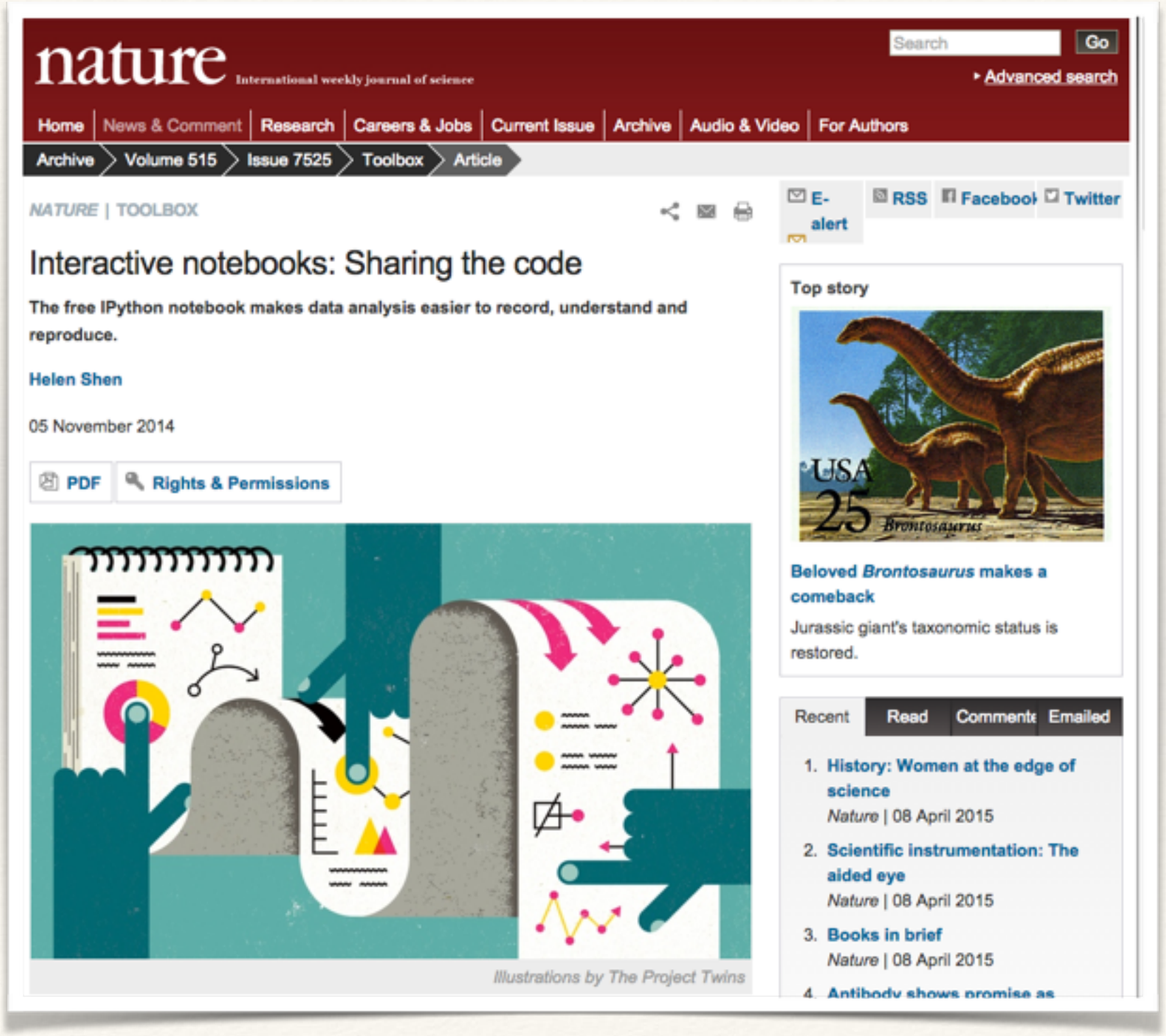
By Cyrille Rossant



By Matthew Russell

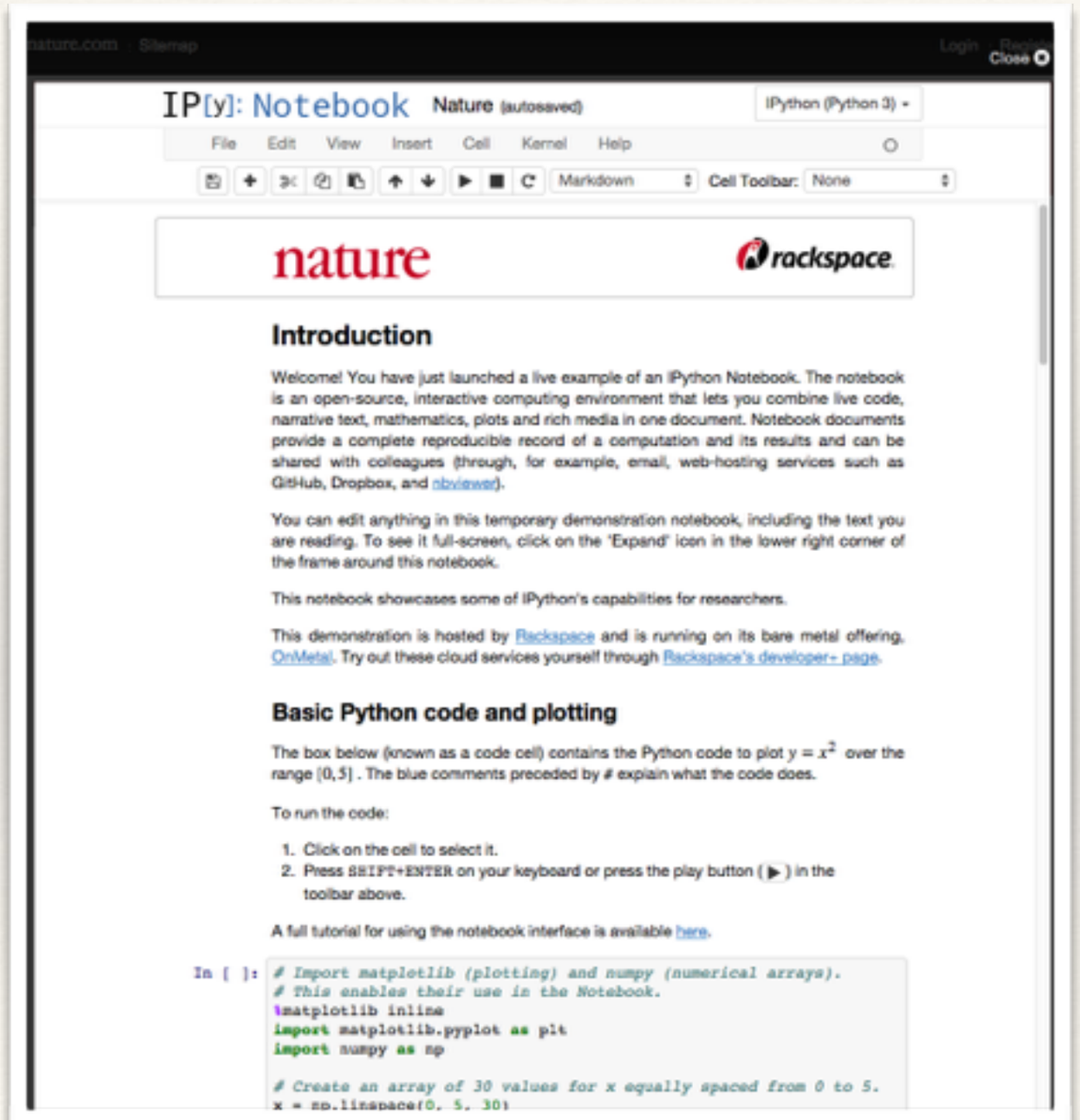
By Cameron Davidson-Pilon

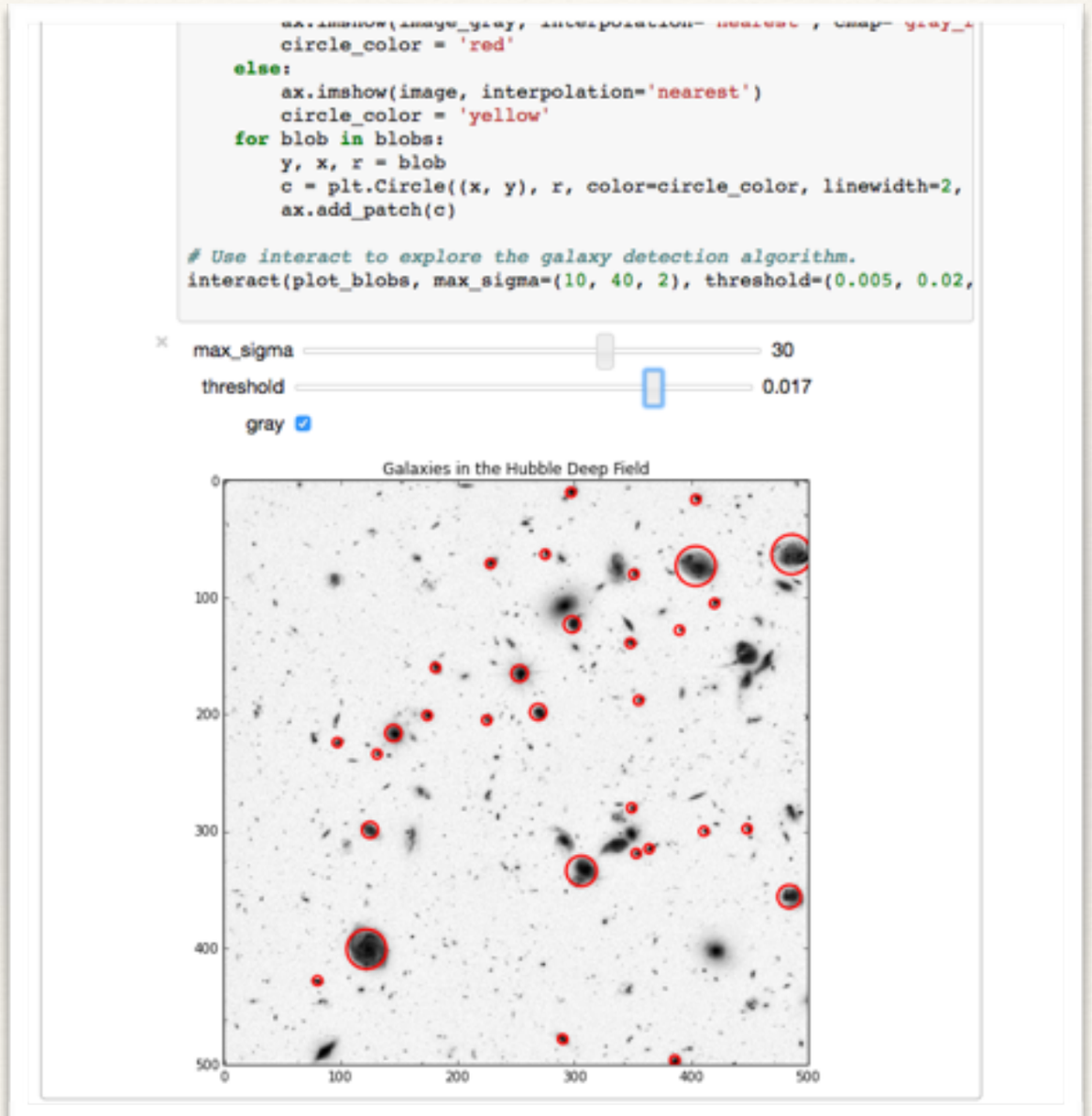
Changing the scientific culture



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

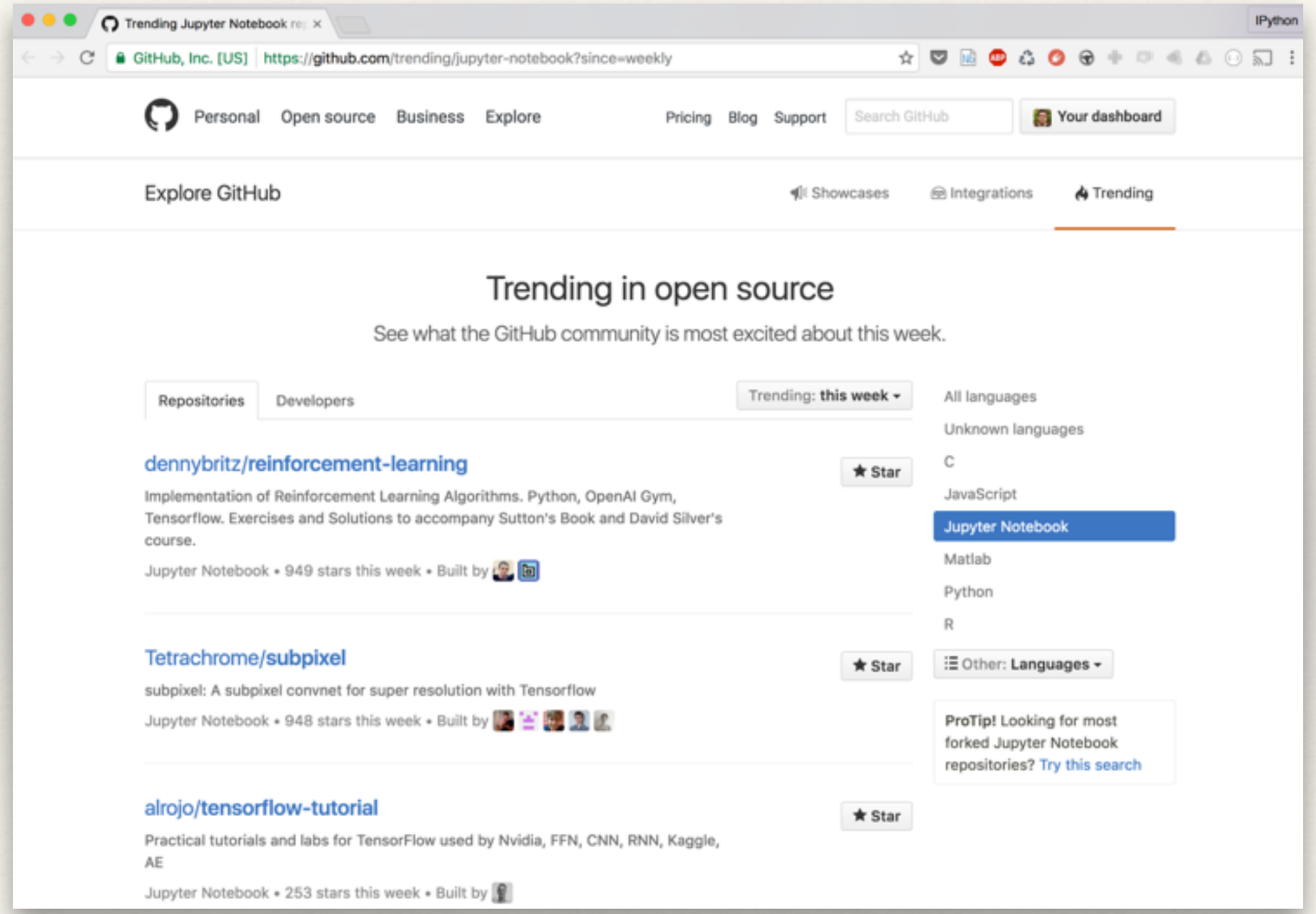
Executable papers: the future?





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

Over 500,000 notebooks on Github



https://github.com/trending/jupyter-notebook?since=weekly

Notebook Workflows: The Big Picture

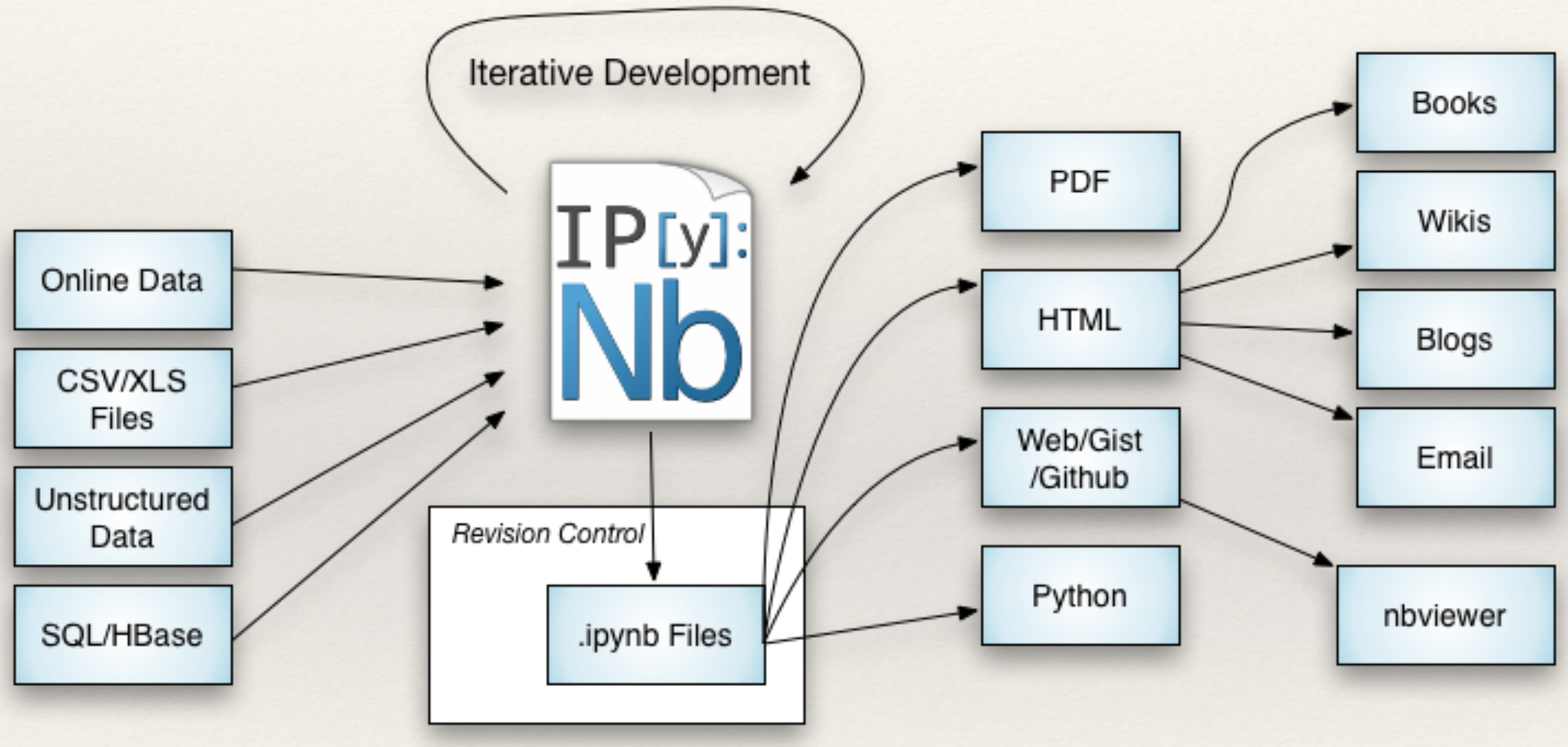


Image credit: Joshua Barratt

These foundations have become infrastructure

JupyterHub: multiuser support

Jupyter for Organizations

JupyterHub is a multiuser version of the notebook designed for centralized deployments in companies, university classrooms and research labs.

Pluggable authentication

Manage users and authentication with PAM, OAuth or integrate with your own directory service system. Collaborate with others through the Linux permission model.

Centralized deployment

Deploy the Jupyter Notebook to all users in your organization on centralized servers on- or off-site.

Container friendly

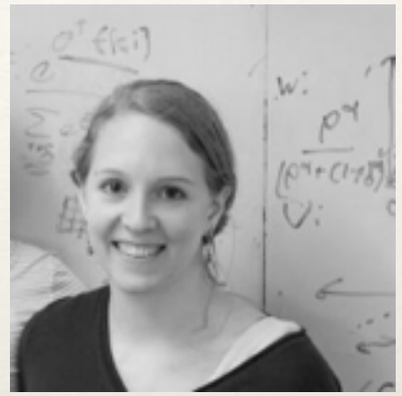
Use Docker containers to scale your deployment and isolate user processes using a growing ecosystem of prebuilt Docker containers.

Code meets data

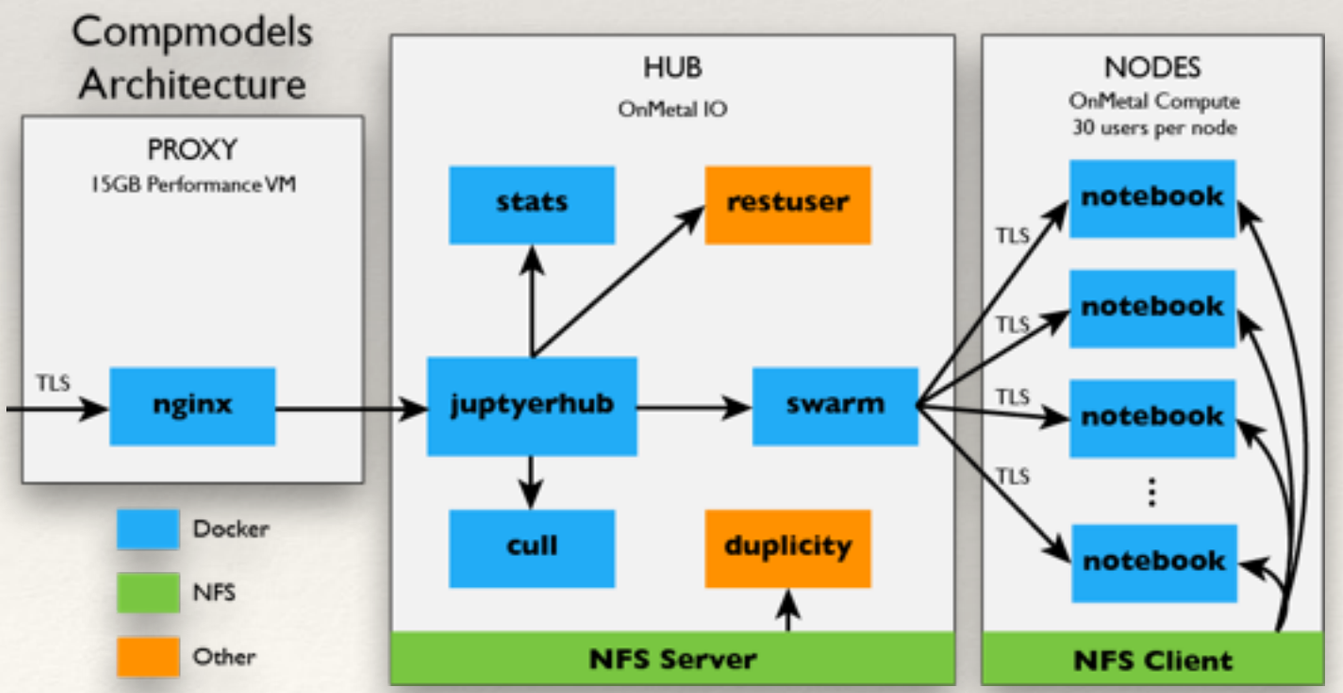
Deploy the Notebook next to your data to provide unified software management and data access within your organization.

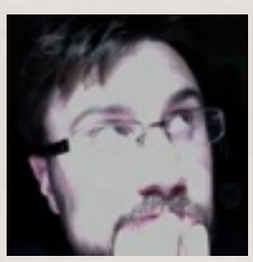
JupyterHub in Education @ Berkeley

- Computationally intensive course, ~220 students
- Fully hosted environment, zero-install, spring 2015.
- Homework management and grading (w B. Granger)
- Now powers data8.org - Cal's new *Foundations of Data Science*, (fall 2015).

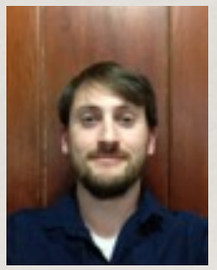


Jess Hamrick @ Cal





K. Kelley
Rackspace



M. Ragan-Kelley
Cal

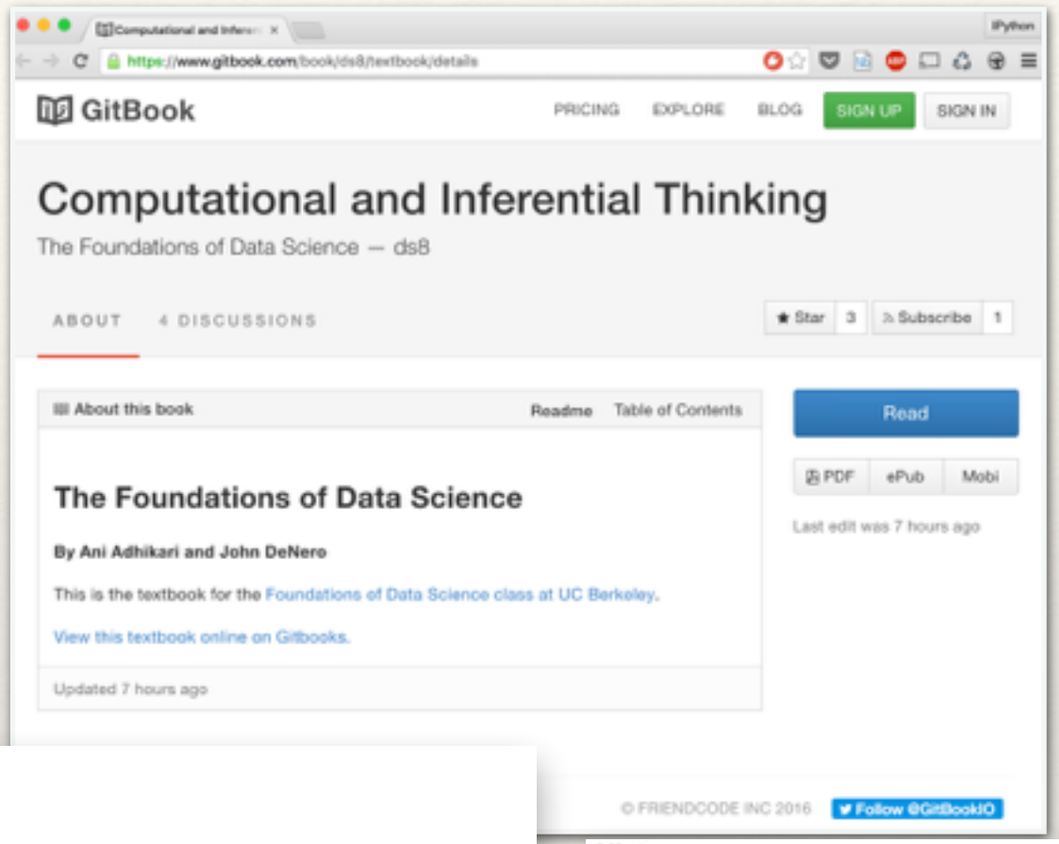
B. Granger
Cal Poly

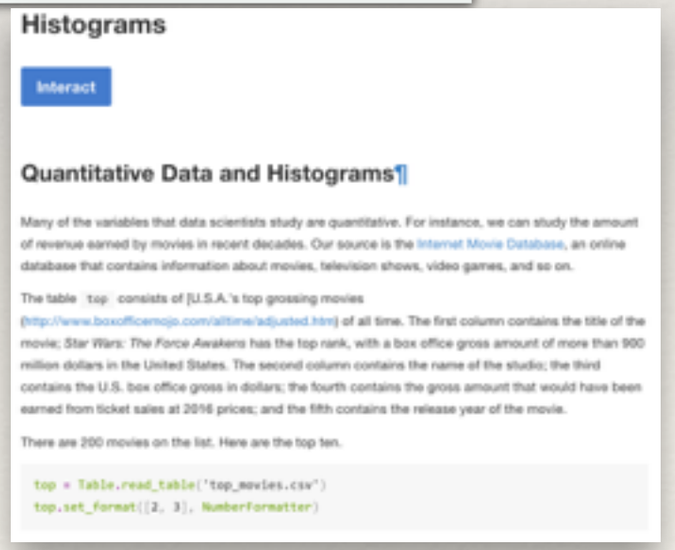
POWERED BY rackspace® the open cloud company

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

Berkeley's *Foundations of Data Science*

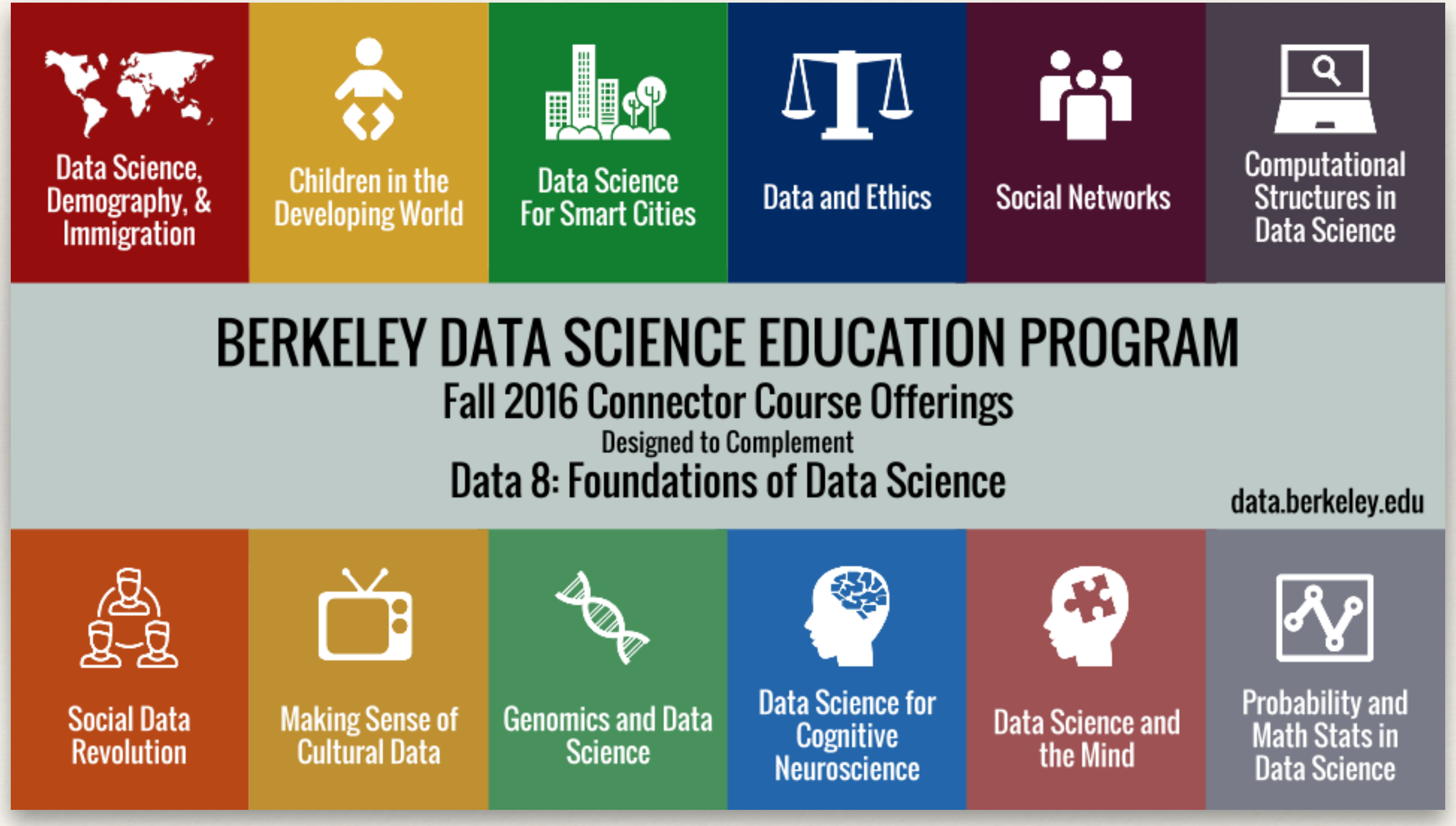
- New curriculum aimed at all freshmen at UC Berkeley
- Interactive textbook is Jupyter Notebooks
- Course deployment is JupyterHub
  - Off Jess Hamrick's work





data.berkeley.edu, data8.org

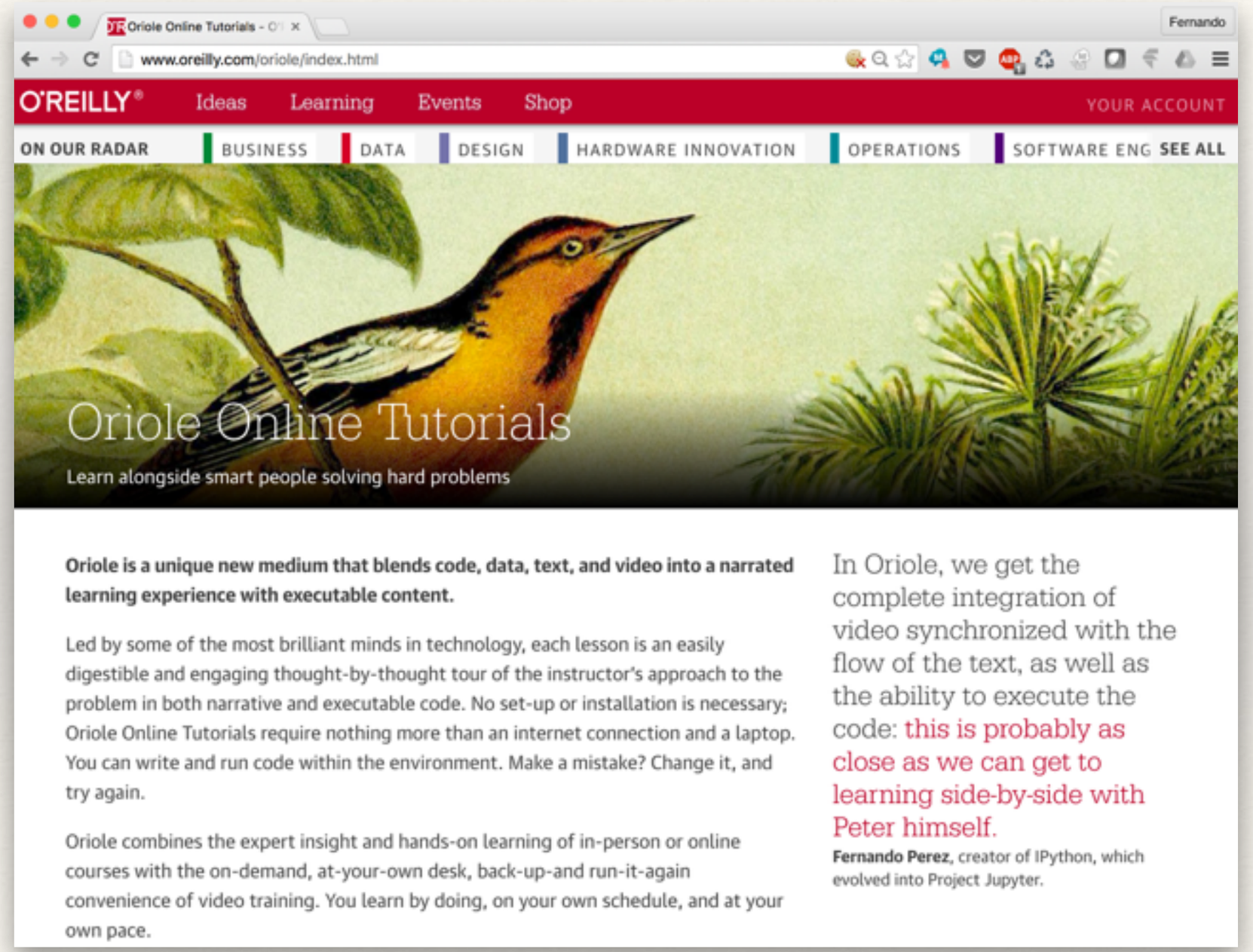
Data Science: *Connector Courses*



Other JupyterHub Deployments

- Educational:
  - Cal Poly, Bryn Mawr, U. Sheffield, ETH Zurich, …
- NERSC (DOE open science HPC center @ LBNL)
- San Diego Supercomputing Center, Minnesota, CU Boulder
- CERN
- Wikimedia Foundation
- Danish e-Infrastructure cooperation.

O'Reilly's Oriole



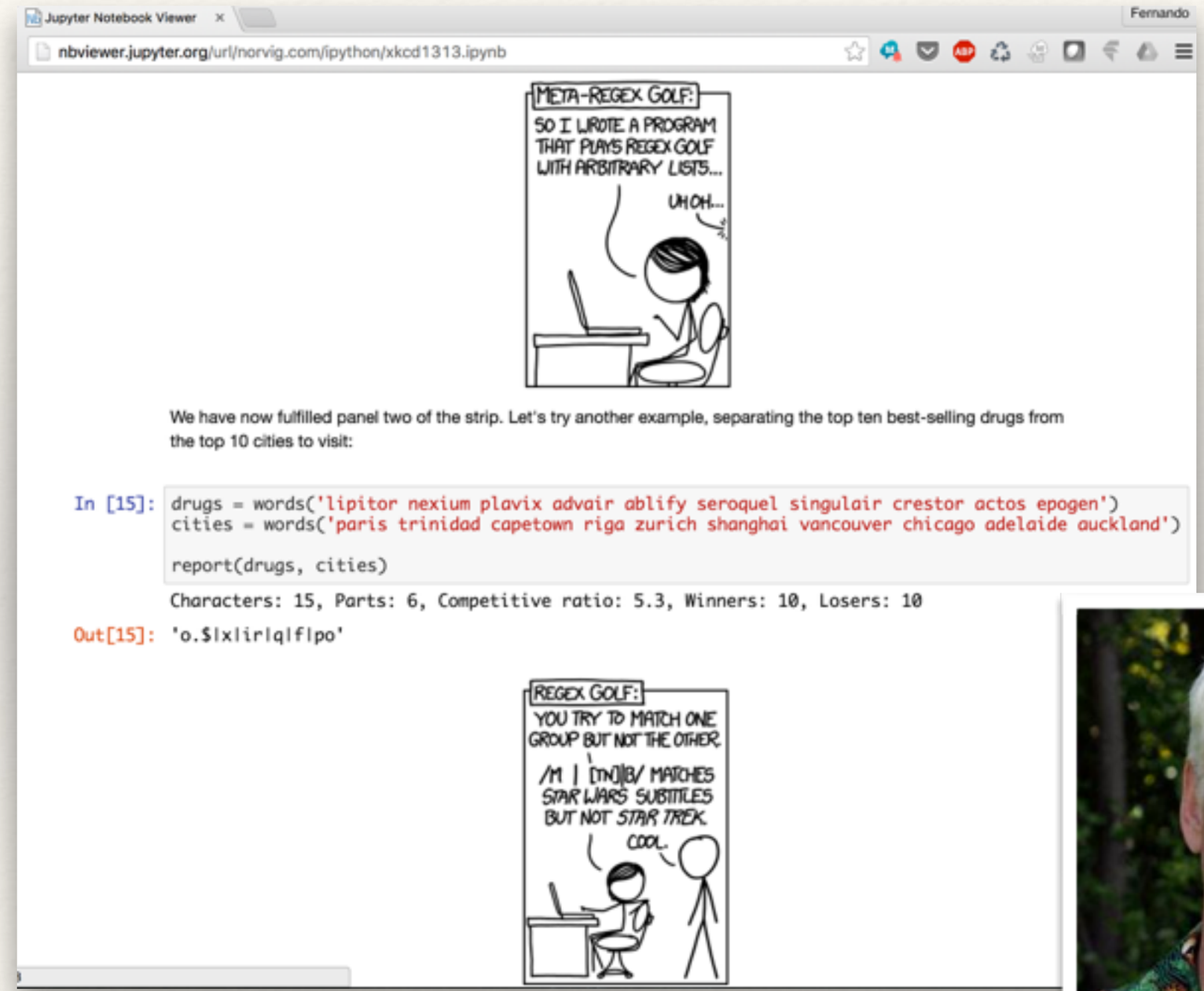
Paco Nathan

Taylor Martin

Andrew Odewahn

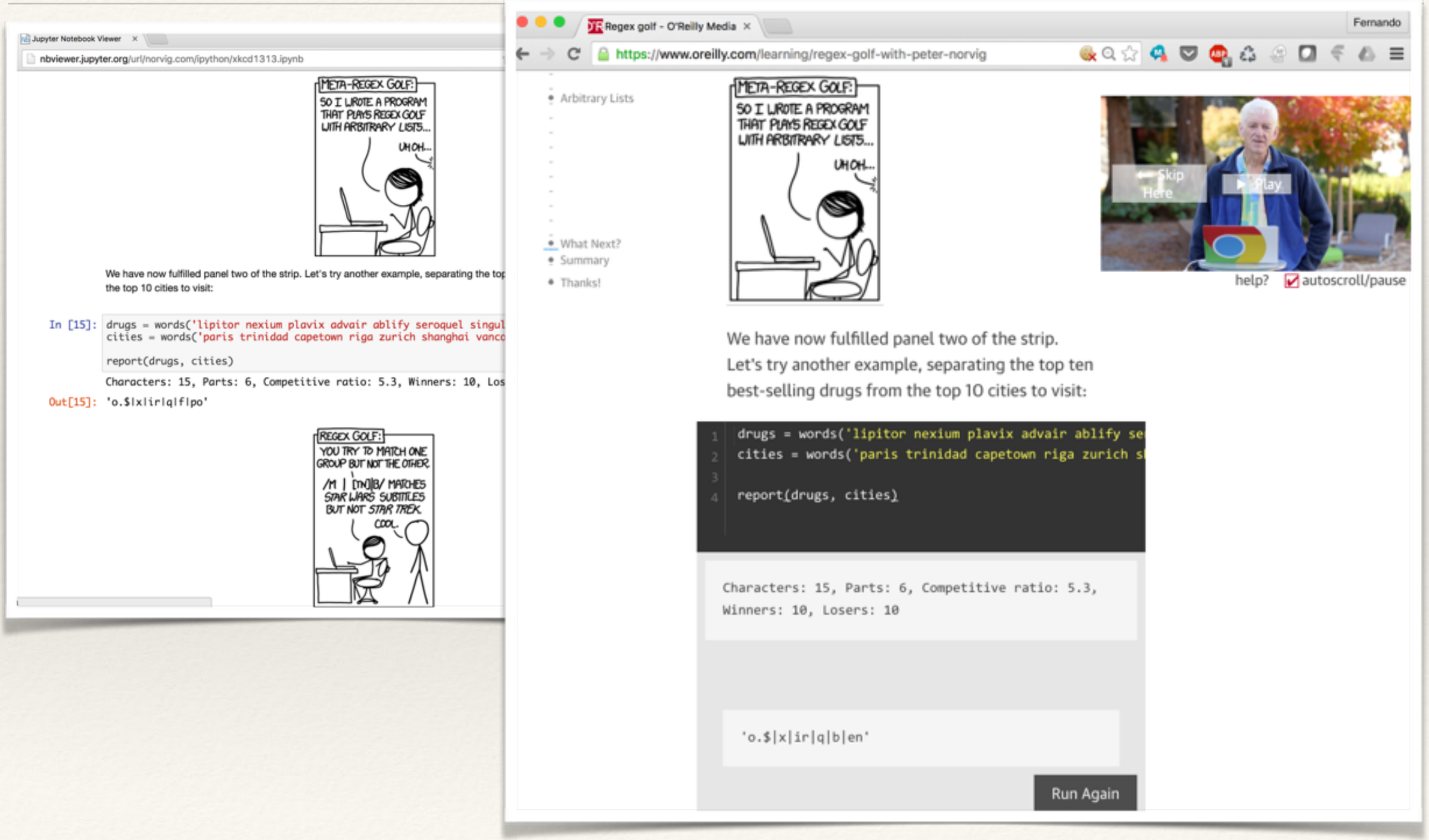
Launched 3/21/2016: oreilly.com/oriole

Oriole: from static notebook…



nbviewer.jupyter.org/url/norvig.com/ipython/xkcd1313.ipynb

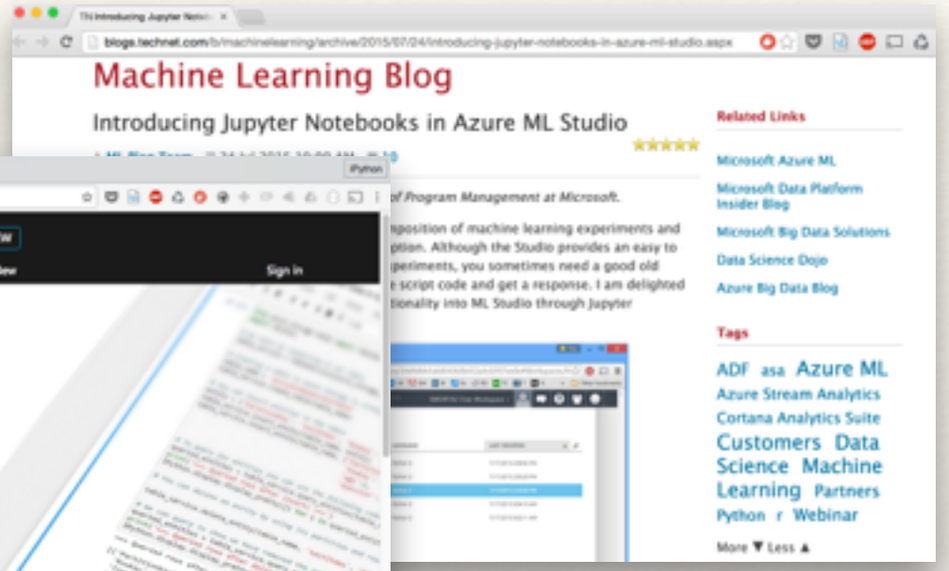
… to executable, video-narrated tutorial:

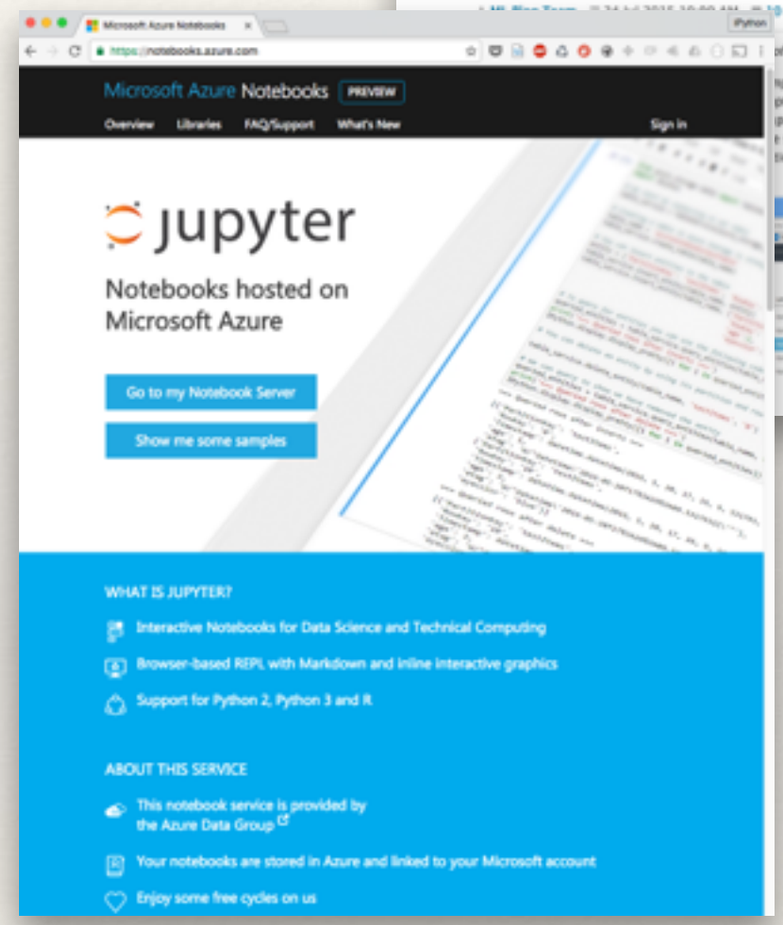


oreilly.com/learning/regex-golf-with-peter-norvig

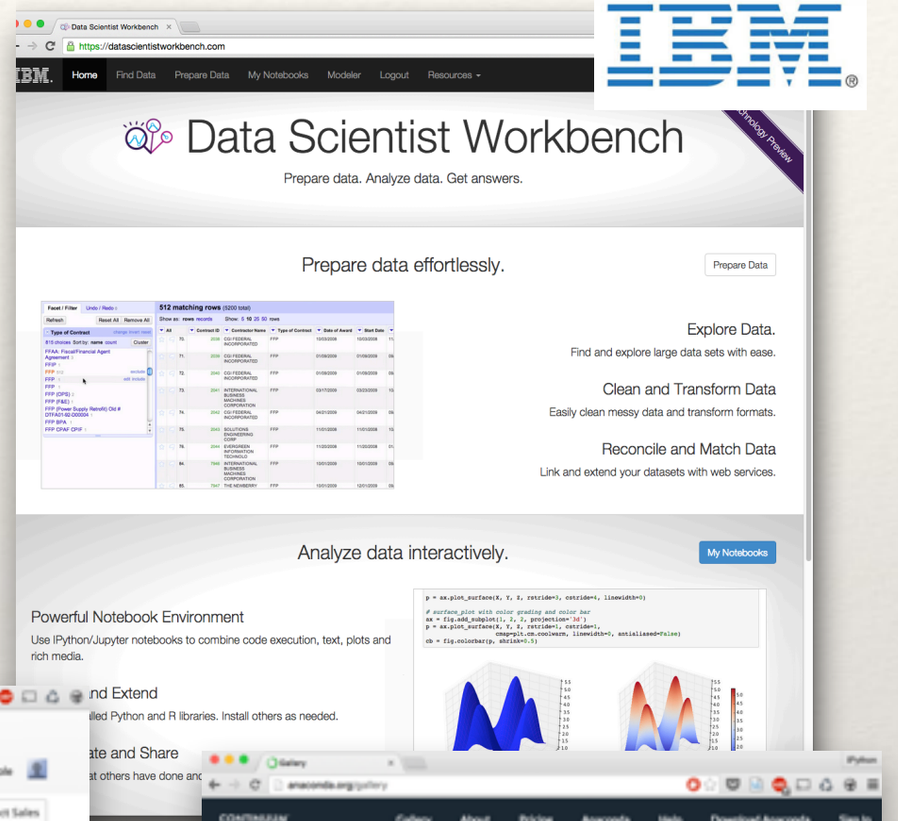
Microsoft, IBM, Google, Continuum…

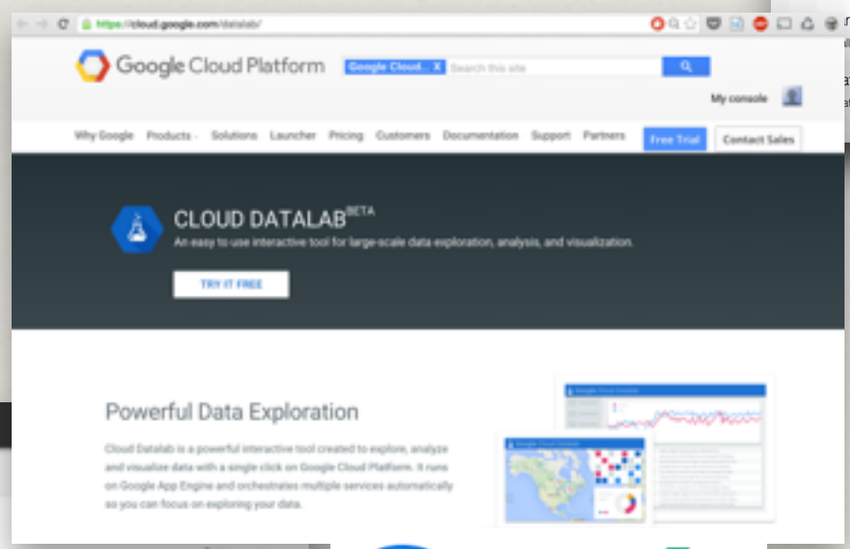
Microsoft

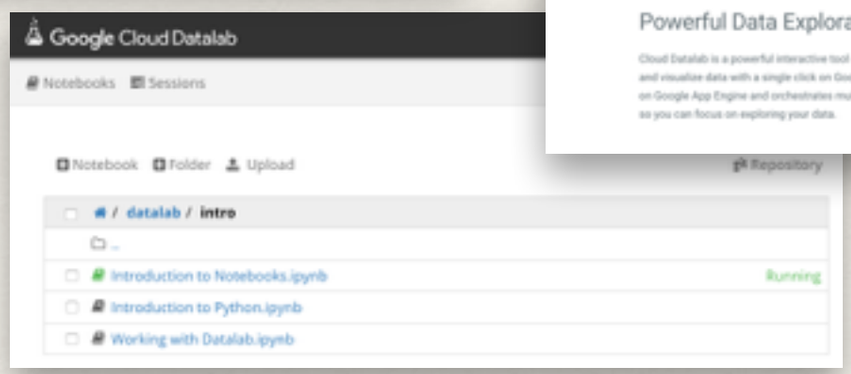




IBM





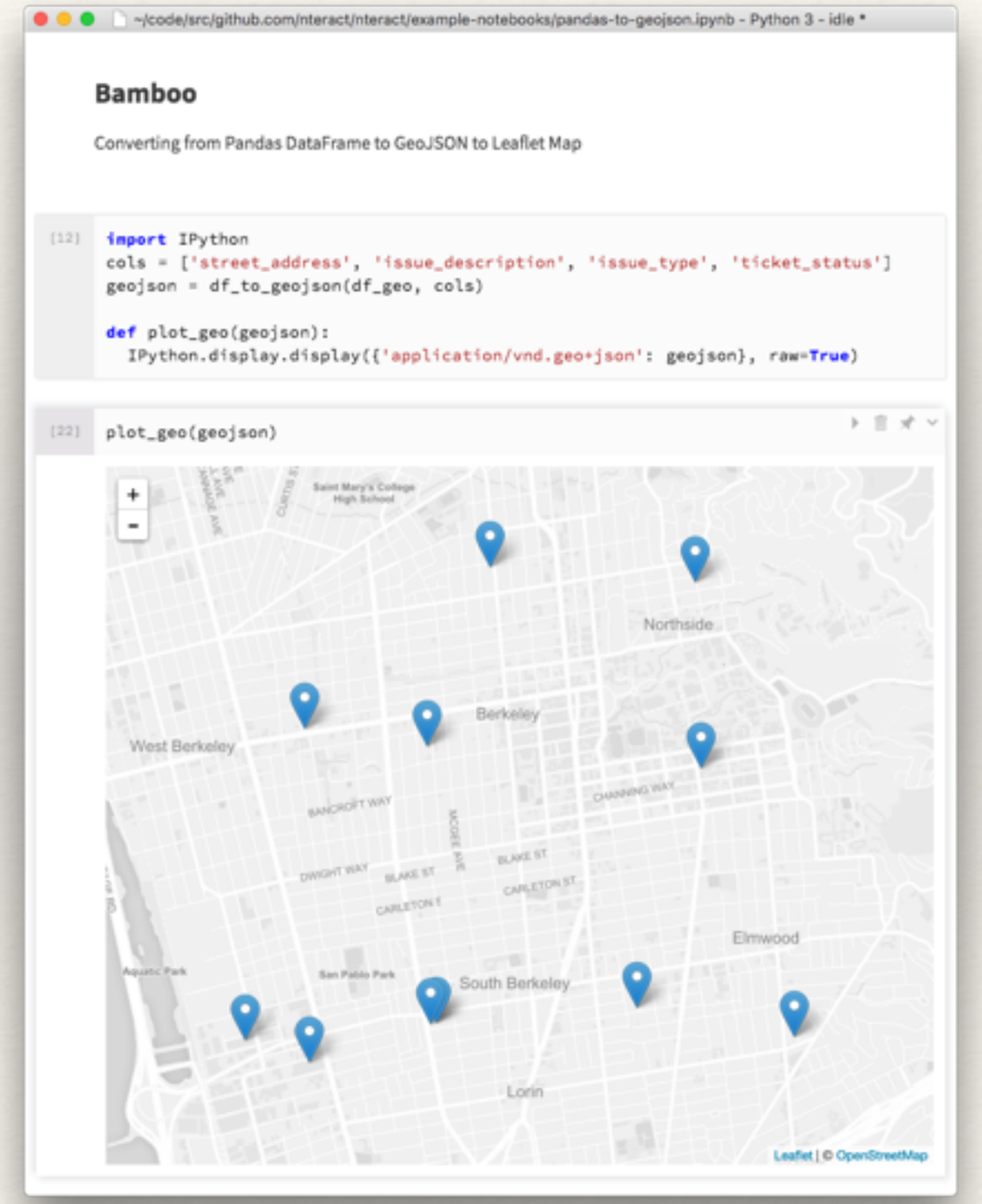


Google

CONTINUUM ANALYTICS

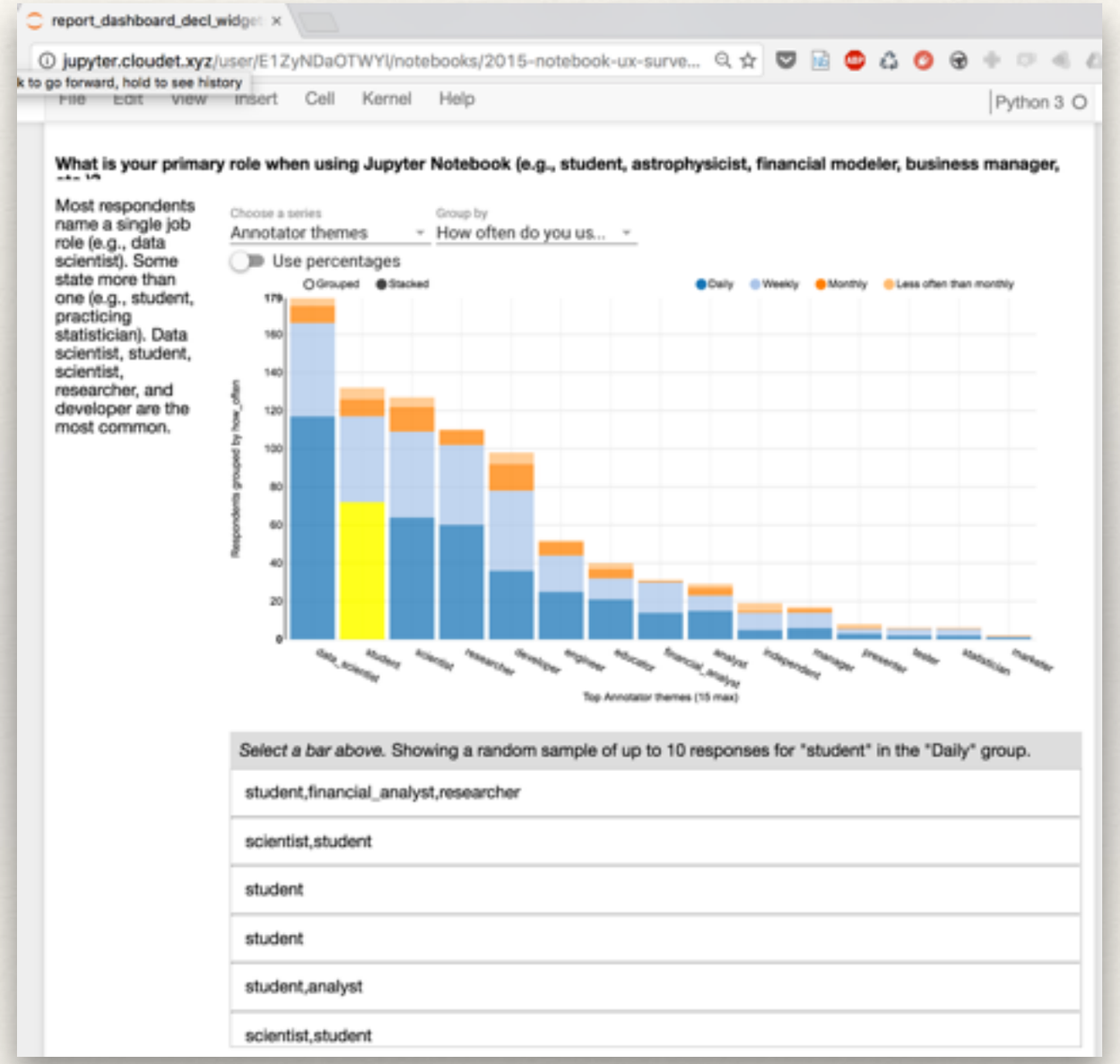
Alternate clients: nteract

- Local desktop application
- Written in node.js (uses React)
- Uses:
  - Jupyter messaging protocols
  - Notebook file format.
- https://github.com/nteract/nteract



JupyterLab: the notebook, evolved…

Estimated user base: at least ~3M



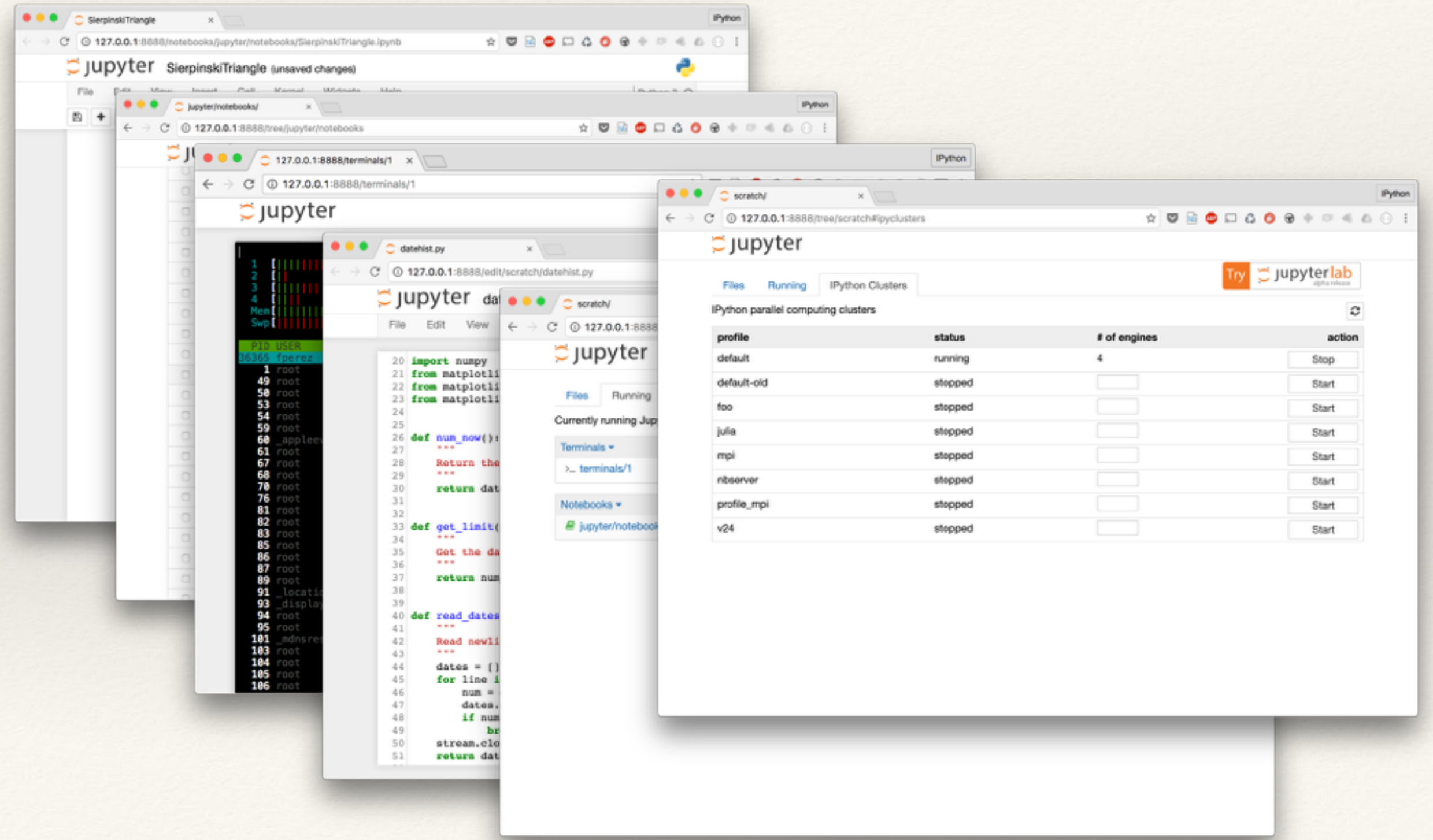
IBM UX User Survey: a live, interactive dashboard

https://github.com/jupyter/design/blob/master/surveys/2015-notebook-ux/analysis/report_dashboard.ipynb

IBM-led 2015 UX Survey

- Mostly daily / weekly users
- Love the notebook workflow and user experience
- Top needs:
  - Integration with version control systems (Git, GitHub)
  - Improved code / text editing
  - Flexible layout and integration between the building blocks
  - Debugger, profiler, variable inspector, etc.

The “Notebook”?



Building Blocks

File Browser

Notebooks

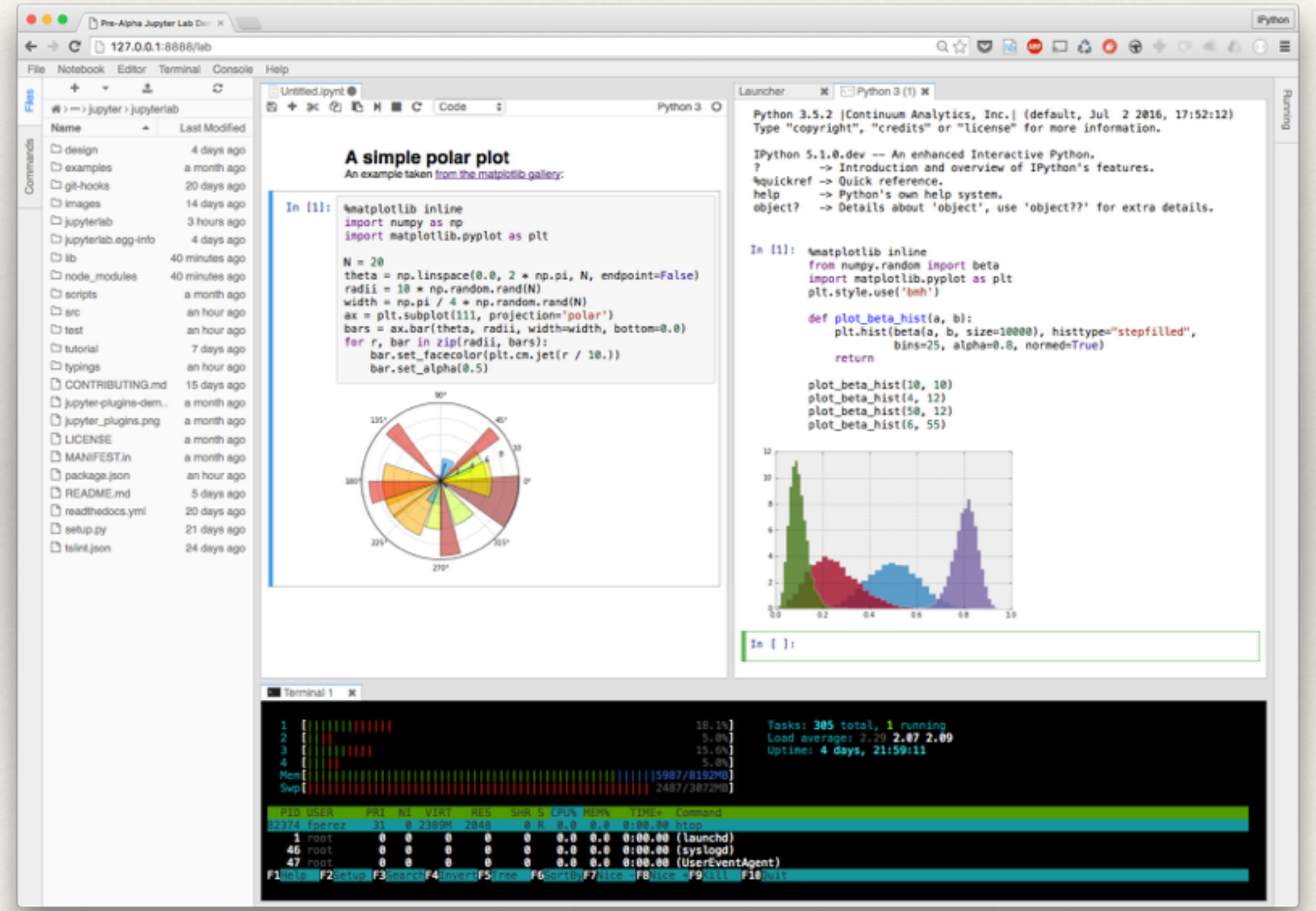
Terminal

Text Editor

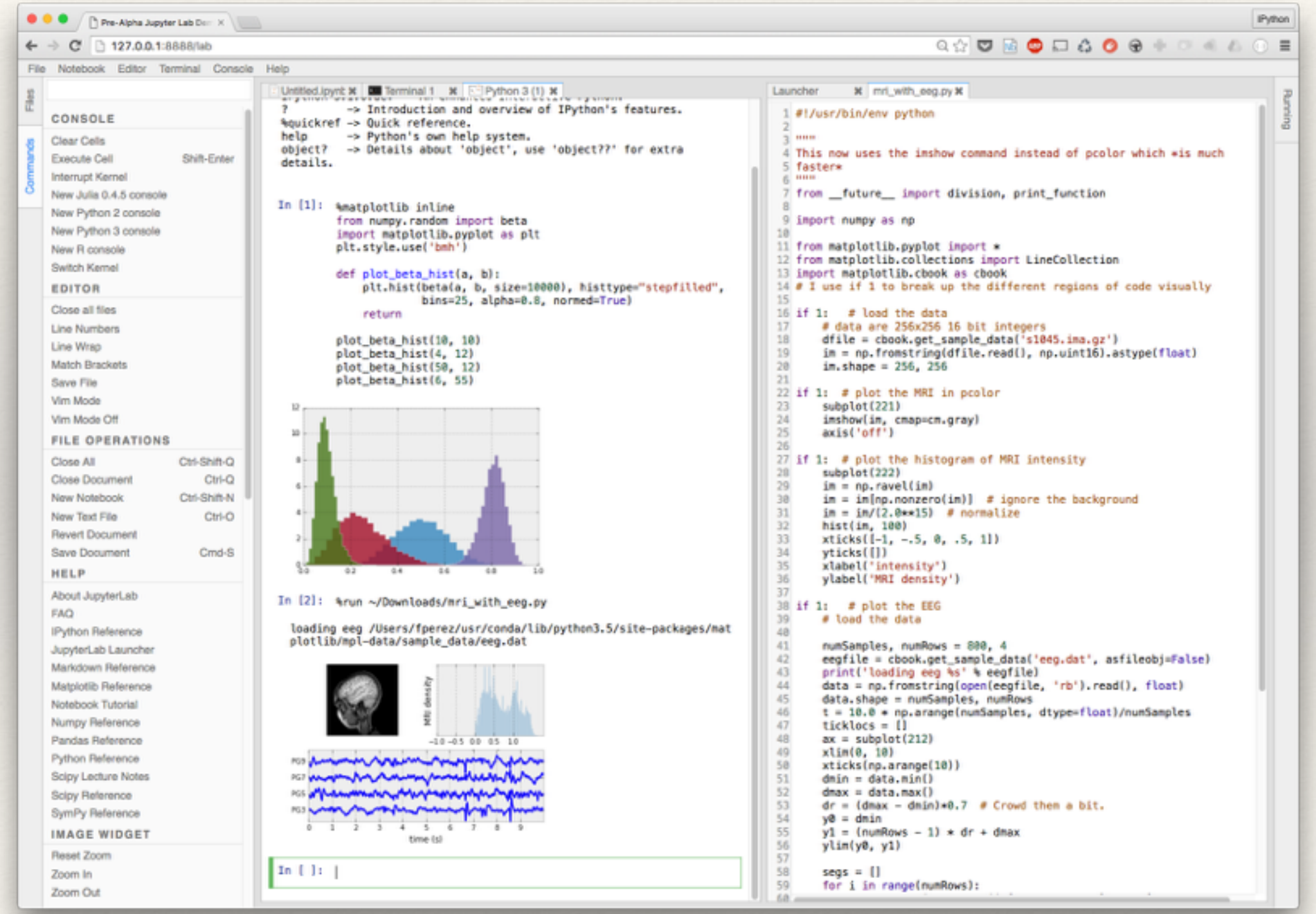
Kernels

Output

JupyterLab: unifying these ideas



A modular, plugin-based architecture



A collaboration with Bloomberg and Continuum

jupyter

Bloomberg

Roadmap

- Today JupyterLab is an early preview only
- Not suggested for general usage:
  - Design, UI, UX, interactions, code all changing rapidly!
- Phases:

1. Series of alpha/beta releases of JupyterLab available as an alternative UI alongside the classic notebook
2. JupyterLab 1.0 = feature parity with classic notebook + small number of new features
3. JupyterLab default UI, classic notebook still available
4. Classic notebook only available as separate download

Thank You!

@fperez_org fperez@lbl.gov

@ProjectJupyter @IPythonDev

Try it out at

try.jupyter.org