

IP[y]:
IPython



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)

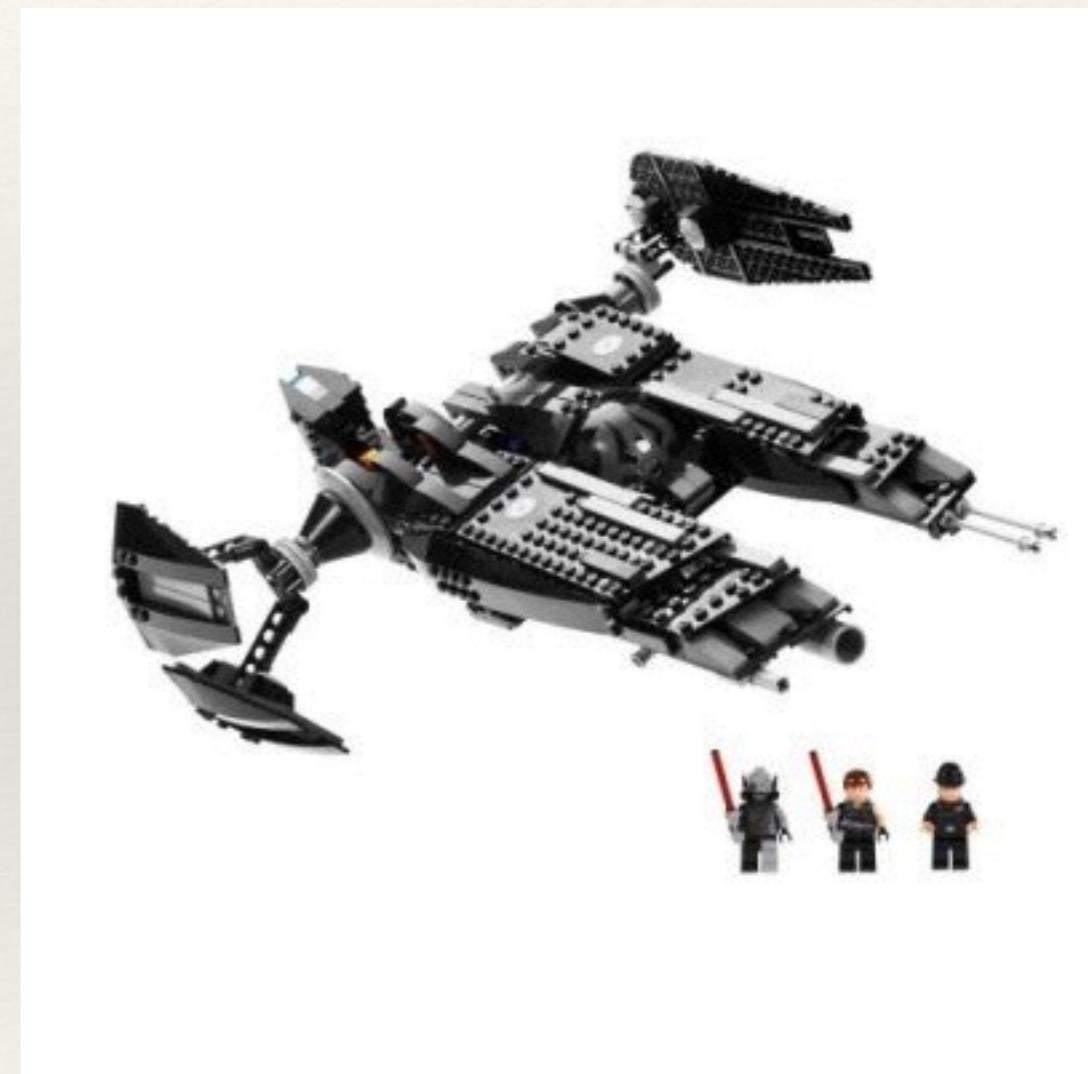
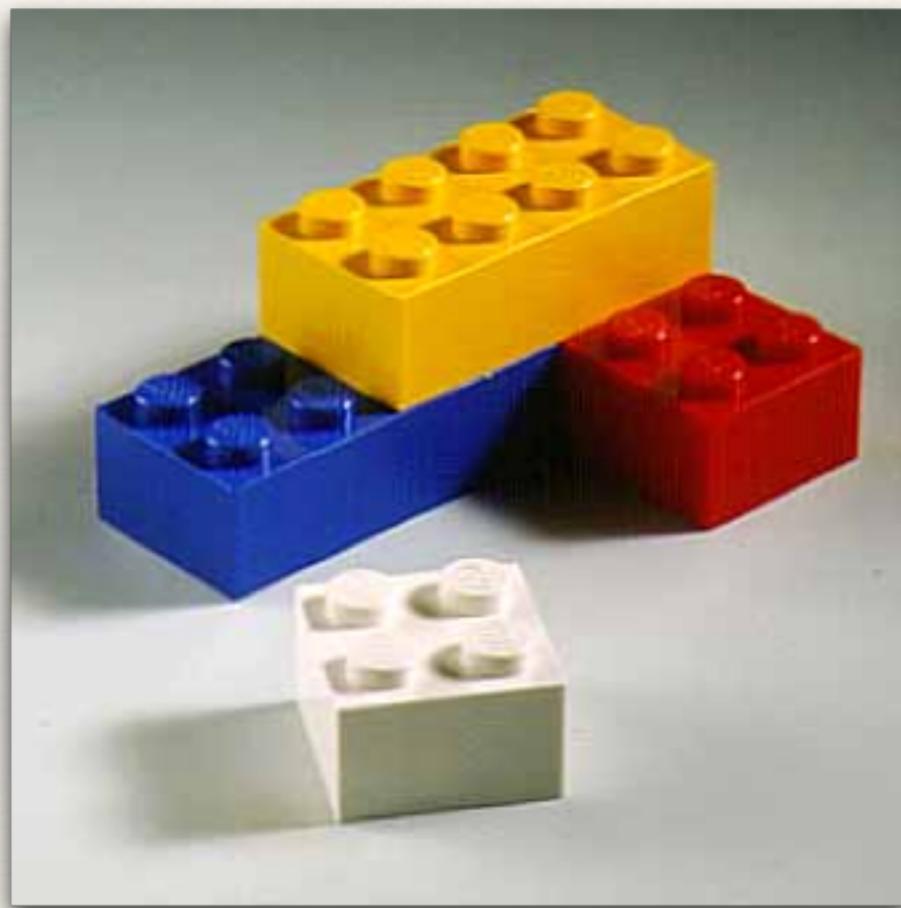
1. Individual exploratory work
2. Collaborative development
3. Parallel production runs (HPC, cloud, ...)
4. Publication & communication (reproducibly!)
5. Education
6. Goto 1

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

```
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     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.
- ❖ 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



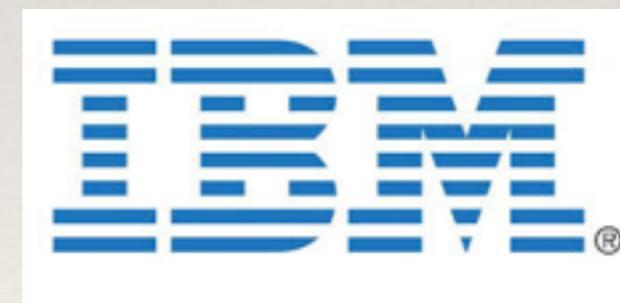
ALFRED P. SLOAN
FOUNDATION



GORDON AND BETTY
MOORE
FOUNDATION



CONTINUUM
ANALYTICS



Google

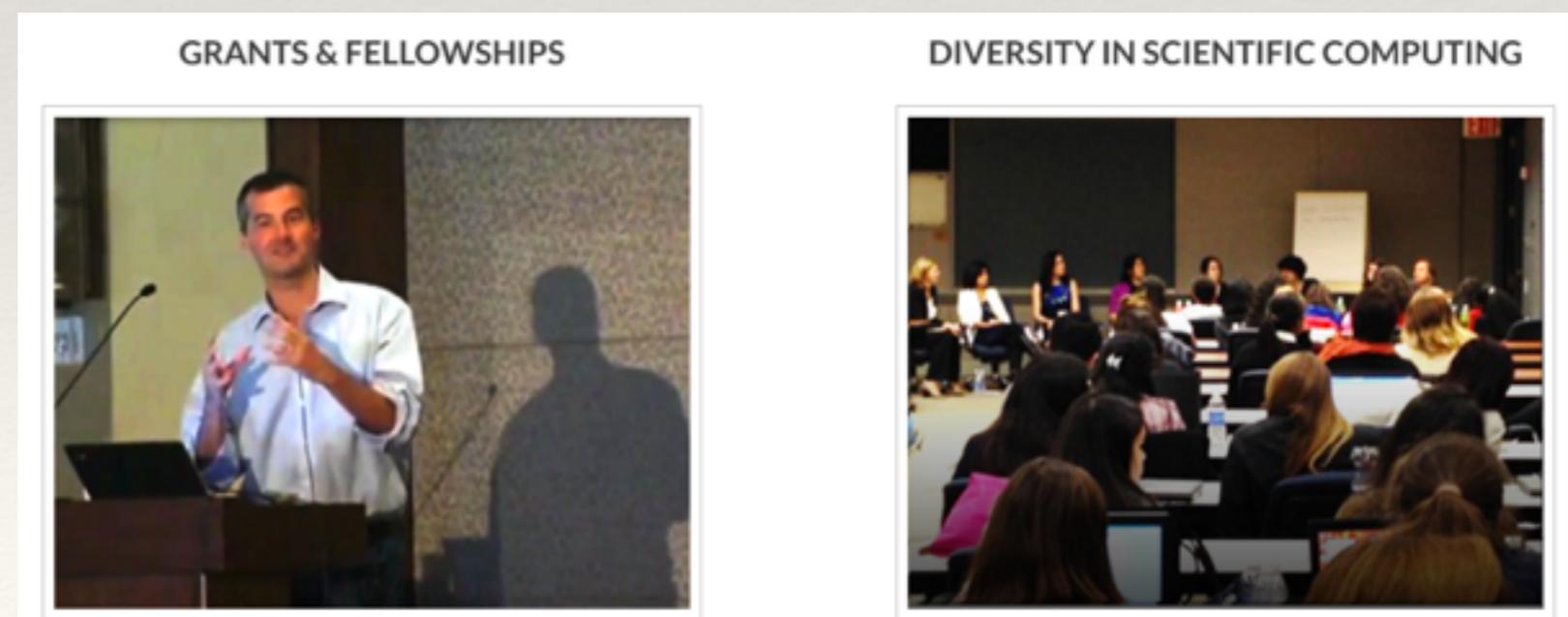
POWERED BY
rackspace®
the open cloud company
 Microsoft
 Bloomberg

ENTHOUGHT
SCIENTIFIC COMPUTING SOLUTIONS

NumFOCUS: *Scientific* OSS Foundation



- 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

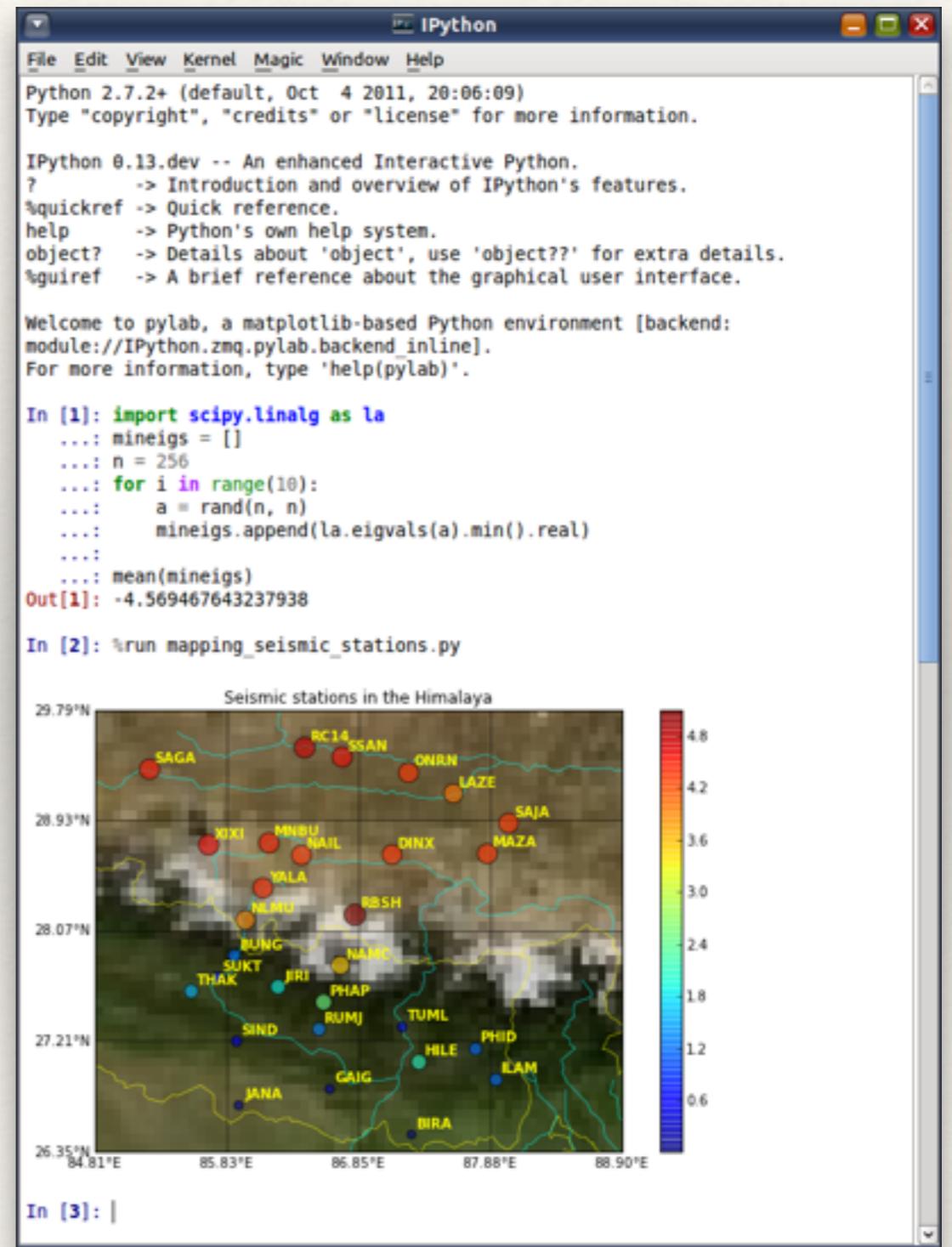
- ❖ Pair programming (Brian Granger/FP), 24 hours of design/code, less than 500 LOC total.
- ❖ Live, over-the-wire developing and debugging.
- ❖ Designed protocol, client and server together.
 - ❖ 1st terminal: `./kernel.py`
 - ❖ 2nd terminal: `./frontend.py`

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

Beyond the Terminal...

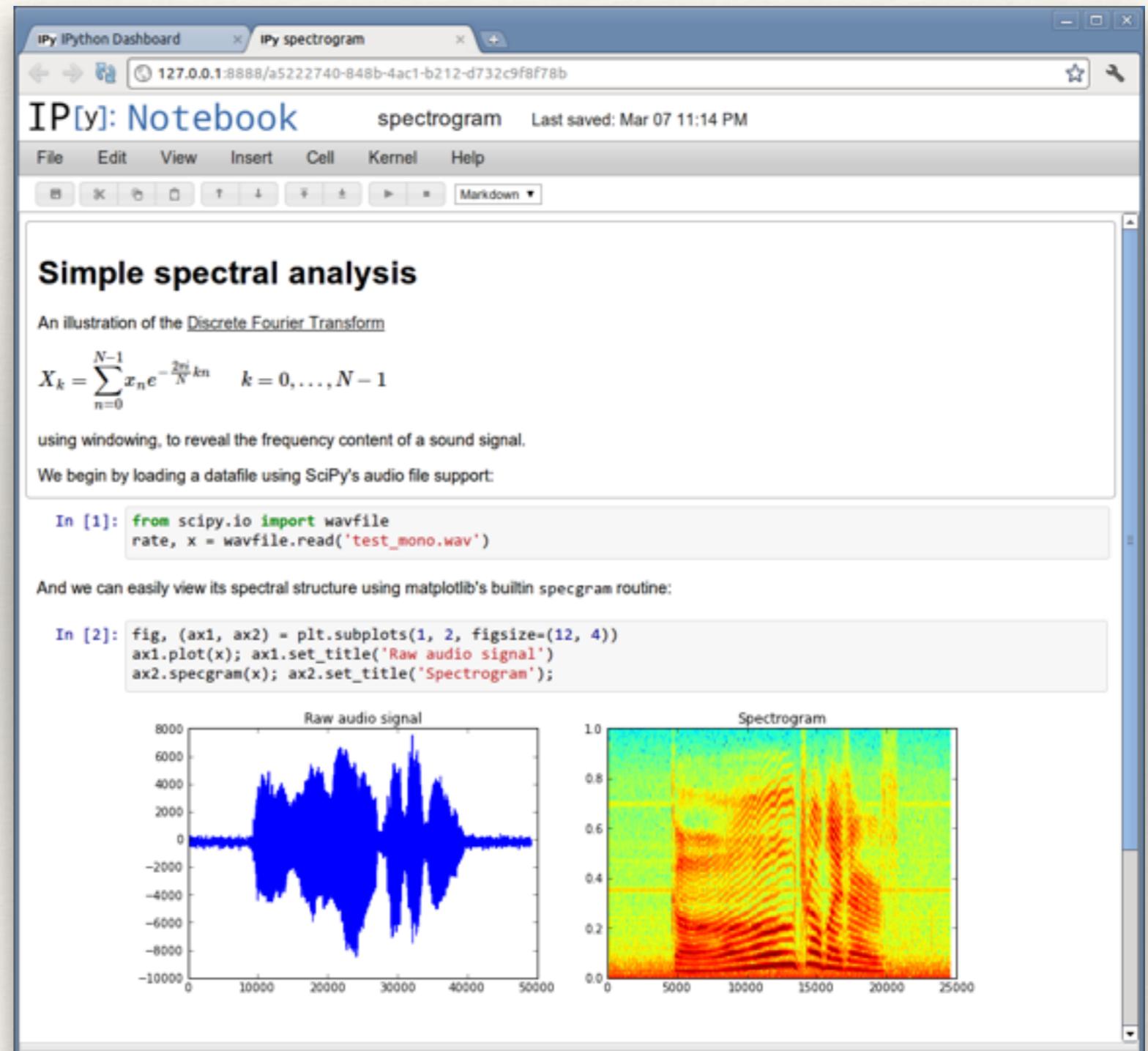
- ❖ The REPL as a network protocol
- ❖ Kernels
 - ❖ execute code
- ❖ Clients
 - ❖ Read input
 - ❖ Present output

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

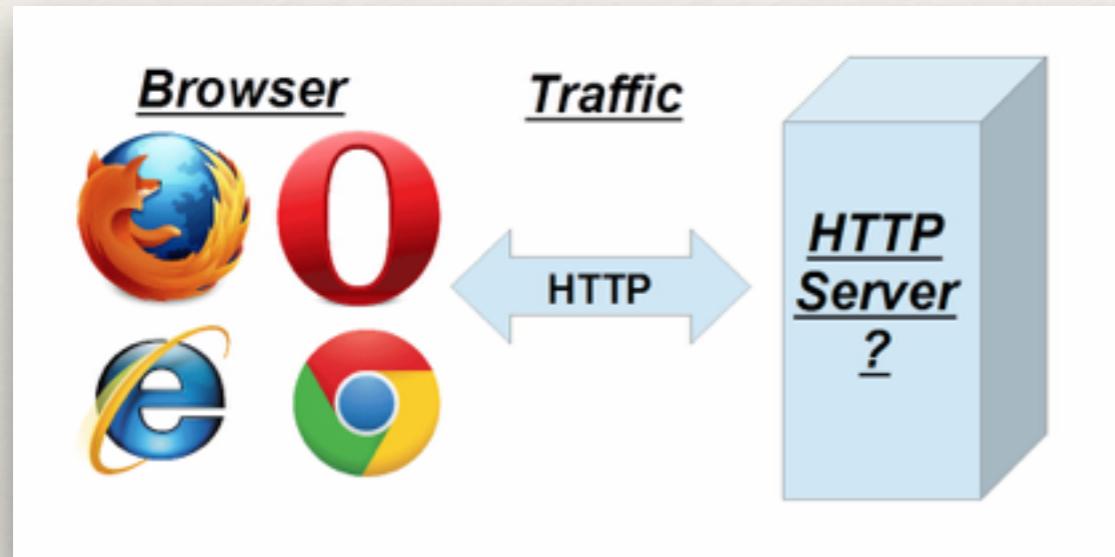
- ❖ Interactive Python shell at the terminal
- ❖ Kernel for this protocol in Python
- ❖ Tools for Interactive Parallel computing

- ❖ 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

```
<a href="/" rel="home" title="University of Colorado Boulder " class="custom-logo-link active"></a><a href="/" rel="home" title="University of Colorado Boulder " class="custom-logo-link active"></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>
```

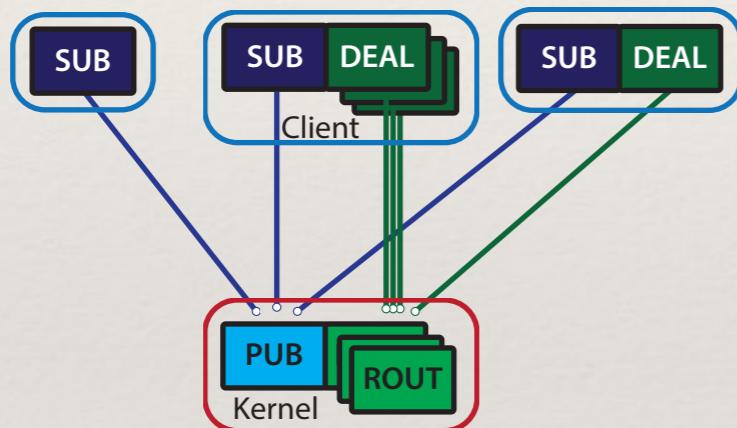


HTML: format to represent content

HyperText Markup Language

Core ideas of Jupyter

Interactive Computing Protocol



ØMQ + JSON

Document Format

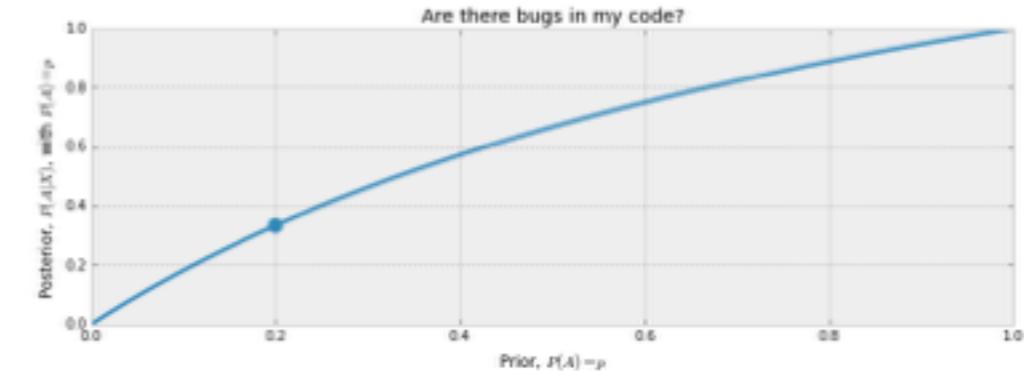
We have already computed $P(X|A)$ above. On the other hand, $P(X| \sim A)$ is subjective: our code can pass tests but still have a bug in it, though the probability there is a bug present is reduced. Note this is dependent on the number of tests performed, the degree of complication in the tests, etc.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

$$\begin{aligned} P(A|X) &= \frac{1 \cdot p}{1 \cdot p + 0.5(1 - p)} \\ &= \frac{2p}{1 + p} \end{aligned}$$

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>

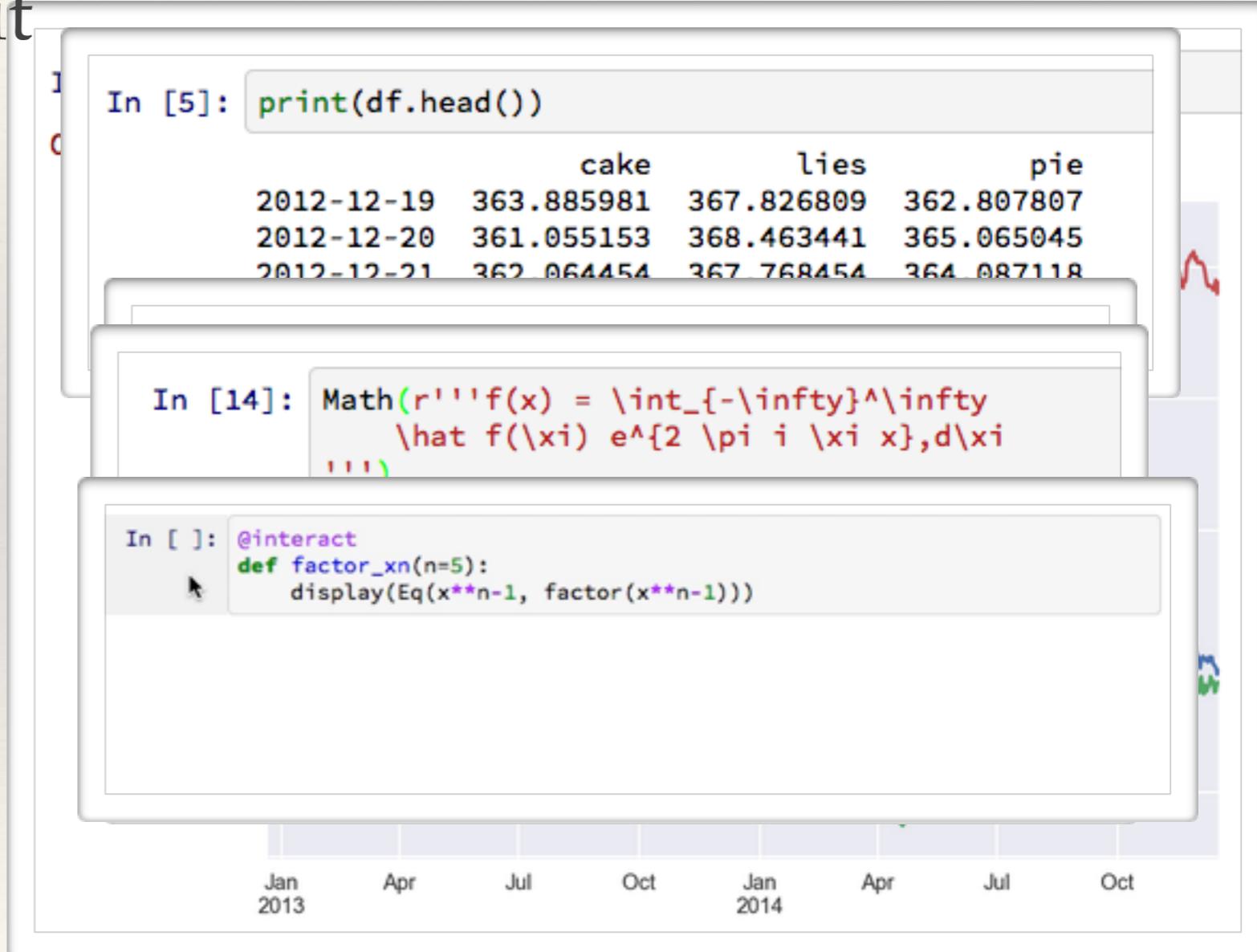


Jupyter Protocol

capture the process of interactive computing

any mime-type output

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



The image shows a Jupyter Notebook interface with three code cells and their corresponding outputs. The first cell (In [5]) contains the code `print(df.head())` and its output is a table with three rows of data:

		cake	lies	pie
2012-12-19	363.885981	367.826809	362.807807	
2012-12-20	361.055153	368.463441	365.065045	
2012-12-21	362.064454	367.768454	364.087118	

The second cell (In [14]) contains the code `Math(r'''f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi'''')`. The third cell (In []) contains the code `@interact`, `def factor_xn(n=5):`, and `display(Eq(x**n-1, factor(x**n-1)))`. The notebook has a timeline at the bottom showing months from Jan 2013 to Oct 2014.

Jupyter Protocol is language agnostic



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

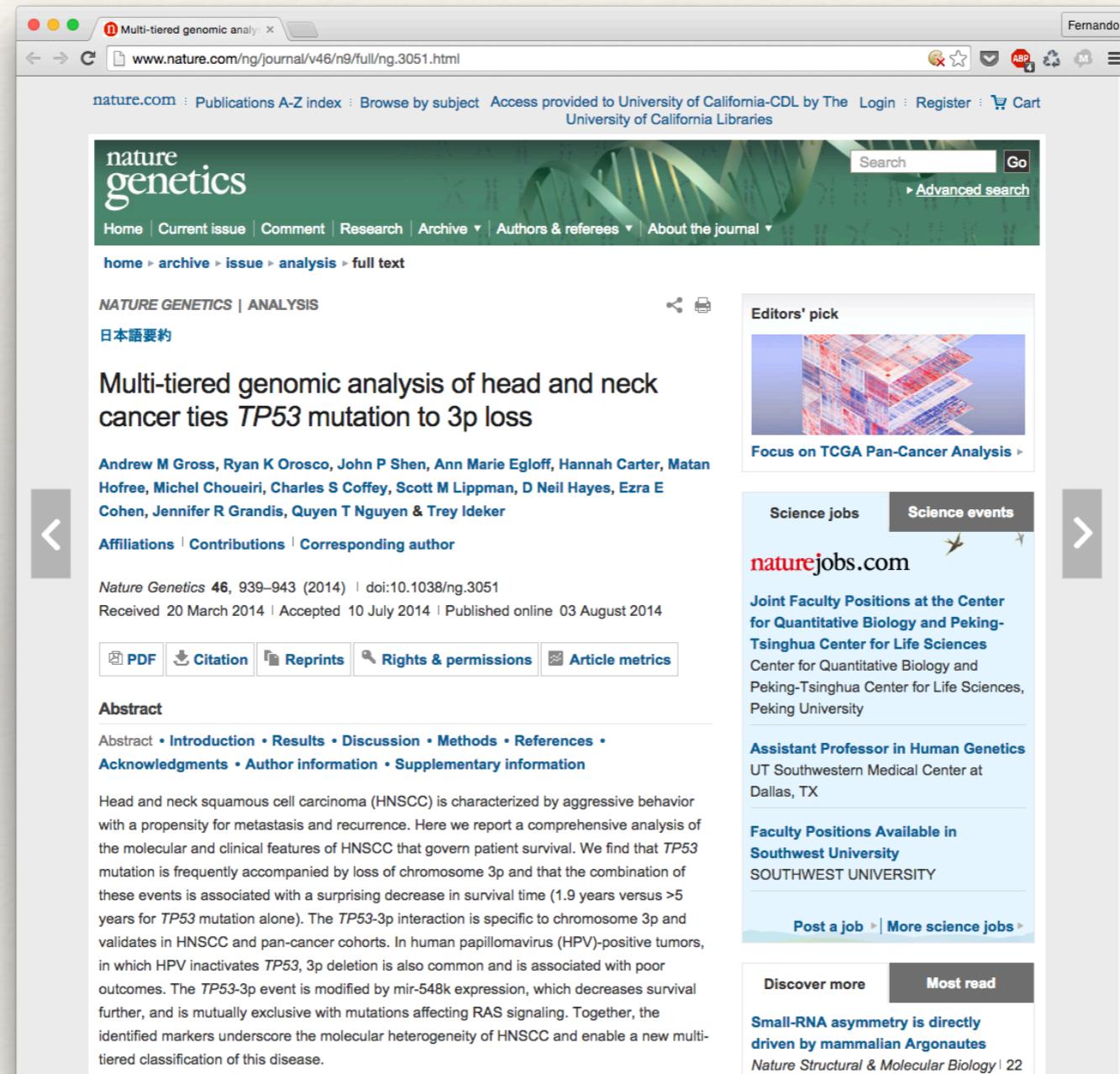
Reproducible research:
from cancer and microbes...

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”



Multi-tiered genomic analysis of head and neck cancer ties *TP53* mutation to 3p loss

Andrew M Gross, Ryan K Orosco, John P Shen, Ann Marie Egloff, Hannah Carter, Matan Hofree, Michel Choueiri, Charles S Coffey, Scott M Lippman, D Neil Hayes, Ezra E Cohen, Jennifer R Grandis, Quyen T Nguyen & Trey Ideker

Affiliations | Contributions | Corresponding author

Nature Genetics 46, 939–943 (2014) | doi:10.1038/ng.3051
Received 20 March 2014 | Accepted 10 July 2014 | Published online 03 August 2014

PDF Citation Reprints Rights & permissions Article metrics

Abstract

Abstract • Introduction • Results • Discussion • Methods • References • Acknowledgments • Author information • Supplementary information

Head and neck squamous cell carcinoma (HNSCC) is characterized by aggressive behavior with a propensity for metastasis and recurrence. Here we report a comprehensive analysis of the molecular and clinical features of HNSCC that govern patient survival. We find that *TP53* mutation is frequently accompanied by loss of chromosome 3p and that the combination of these events is associated with a surprising decrease in survival time (1.9 years versus >5 years for *TP53* mutation alone). The *TP53*-3p interaction is specific to chromosome 3p and validates in HNSCC and pan-cancer cohorts. In human papillomavirus (HPV)-positive tumors, in which HPV inactivates *TP53*, 3p deletion is also common and is associated with poor outcomes. The *TP53*-3p event is modified by mir-548k expression, which decreases survival further, and is mutually exclusive with mutations affecting RAS signaling. Together, the identified markers underscore the molecular heterogeneity of HNSCC and enable a new multi-tiered classification of this disease.

Editors' pick

Focus on TCGA Pan-Cancer Analysis >

Science jobs Science events

naturejobs.com

Joint Faculty Positions at the Center for Quantitative Biology and Peking-Tsinghua Center for Life Sciences

Center for Quantitative Biology and Peking-Tsinghua Center for Life Sciences, Peking University

Assistant Professor in Human Genetics

UT Southwestern Medical Center at Dallas, TX

Faculty Positions Available in Southwest University

SOUTHWEST UNIVERSITY

Post a job | More science jobs >

Discover more Most read

Small-RNA asymmetry is directly driven by mammalian Argonautes

Nature Structural & Molecular Biology | 22

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

Notebooks on Github: the “actual scolarship”

Top Screenshot: Nature Genetics Article

Multi-tiered genomic analysis of head and neck cancer ties TP53 mutation to 3p loss

Abstract

Head and neck squamous cell carcinoma (HNSCC) is characterized by aggressive behavior with a propensity for metastasis and recurrence. Here we report a comprehensive analysis of the molecular and clinical features of HNSCC that govern patient survival. We find that TP53 mutation is frequently accompanied by loss of chromosome 3p and that the combination of these events is associated with a surprising decrease in survival time (1.9 years versus >5 years for TP53 mutation alone). The TP53-3p interaction is specific to chromosome 3p and validates in HNSCC and pan-cancer cohorts. In human papillomavirus (HPV)-positive tumors, in which HPV inactivates TP53, 3p deletion is also common and is associated with poor outcomes. The TP53-3p event is modified by miR-548k expression, which decreases survival further, and is mutually exclusive with mutations affecting RAS signaling. Together, the identified markers underscore the molecular heterogeneity of HNSCC and enable a new multi-tiered genomic analysis.

Middle Screenshot: GitHub Repository Page

Branch: master | TCGA / Analysis_Notebooks / TP53_exploration.ipynb

theandygross on Oct 12, 2014 Pre-package split.

1 contributor

1033 lines (1033 sloc) | 276.607 kB

Bottom Screenshot: IPython Notebook Content

HNSCC HPV- Cohort

Here we conduct a general exploration of TP53 mutations within the HNSCC discovery cohort. While we try and remain unbiased in our screen for molecular correlates of survival, we do have much more information on TP53 mutations than most others.

In `Poeta`, a TP53 mutation is labeled as disruptive if it is either a stop mutation, or if is located at a binding site and induces a change in polarity of the encoded amino acid. Interestingly, we found that the polarity of the substitution had little effect on prognosis and that patients with a mutation to the L2 binding site had worse outcomes than patients with a mutation to the L3 binding site. In addition, within the context of the framework we set forth for biomarker discovery, we chose to ignore the classification of mutations (past silent/non-silent) in order to keep sample size high at the risk of false positives. For these reasons we elected to simply display the functional assignment of the mutations in Figure 1 rather than obscure these results with a classification scheme.

Import Data and Packages

For full list of data and packages imported see the [Imports](#) notebook.

```
In [1]: import NotebookImport  
from Imports import *
```

importing IPython notebook from Imports.ipynb
Populating the interactive namespace from numpy and matplotlib
changing to source directory
populating namespace with data

TP53 Mutation Clinical Correlates

```
In [2]: p53_mut = mut.df.ix['TP53'].ix[keepers_o].dropna().astype(int)
```

```
In [3]: survival_and_stats(p53_mut, surv, figsize=(5,4), order=[2,1,0])
```

Survival

Years

Median Survival (Years)

5Y Survival

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



The screenshot shows a web browser displaying the **The ISME Journal** website. The main content is a commentary titled "Collaborative cloud-enabled tools allow rapid, reproducible biological insights" by several authors. To the left, a sidebar shows a Python notebook titled "slice_aligned_region.py" with code for primer calculation. The notebook includes comments explaining the code and its execution environment. On the right, there are sections for "FULL TEXT" (with links to previous/next pages, table of contents, and download options), "Supporting Files" (with links to various datasets and tools), and "Instructions and supporting data for the QIIME/Python/StarCluster demo at the 2012 NIH Cloud Computing the Microbiome workshop and our corresponding paper in the ISME Journal".

Journal home > Archive > Commentaries > Full text

Commentary

The ISME Journal (2013) 7, 461–464; doi:10.1038/ismej.2012.123; published online 25 October 2012

Collaborative cloud-enabled tools allow rapid, reproducible biological insights

Agan-Kelley^{1,12}, William Anton Walters^{2,12}, Oswald^{3,6,12}, Justin Riley⁴, Brian E Granger⁵, Gonzalez⁶, Rob Knight^{7,8}, Fernando Perez⁹ and J poraso^{10,11}

Group in Applied Science and Technology, University of California Berkeley, Berkeley, CA, USA
Department of Molecular, Cellular and Developmental Biology, University of Colorado at Boulder, Boulder, CO, USA
National Institute of Computer Science, University of Colorado at Boulder, Boulder, CO, USA
Educational Innovation and Technology, Massachusetts Institute of Technology, Cambridge, MA, USA
Department of Computer Science, University of Colorado at Boulder, Boulder, CO, USA

FULL TEXT

• Previous | Next •
Table of contents
Download PDF
Send to a friend
View interactive PDF in

Instructions and supporting data for the QIIME/Python/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-9f69e1f6](#) 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 Distributions](#)
- [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 than 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 [asap-714b8005](#).

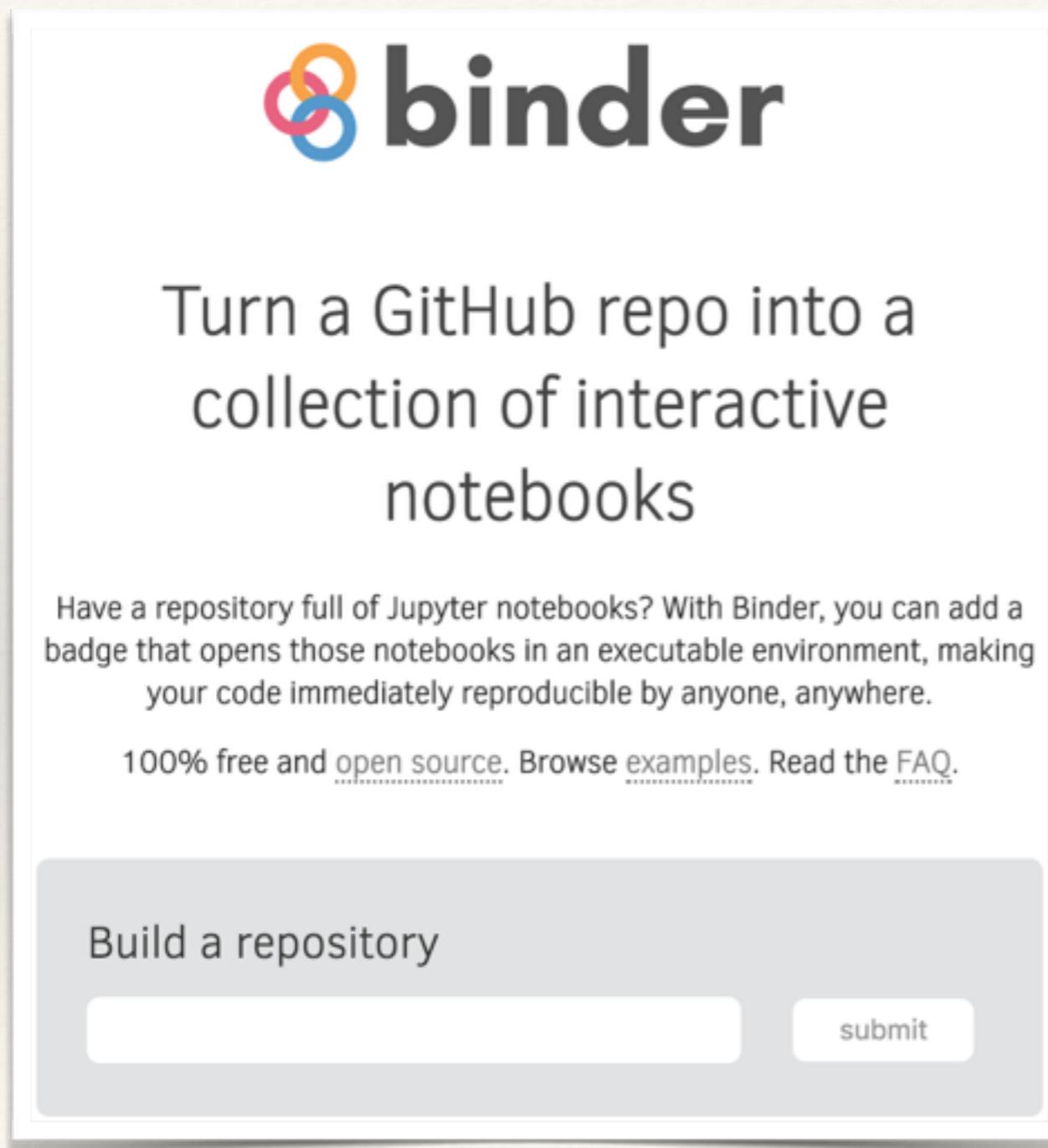
To reproduce the analyses presented in this paper you should install StarCluster locally, and configure it according to the [instructions on the StarCluster website](#). You can then add the following to your `~/starcluster/config` file:

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

[cluster_qiime-ipython]
node_image_id = ami-9f69e1f6
cluster_user = ubuntu
keyname = YOUR-KEY
cluster_size = 4
node_instance_type = m2.4xlarge
plugins = ipcluster
volume = qiime-ipython-data

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

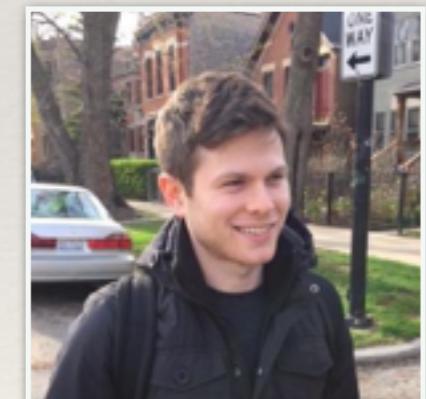
Today: mybinder.org



The screenshot shows the homepage of mybinder.org. At the top is the "binder" logo, which consists of three overlapping circles in orange, red, and blue. Below the logo, the word "binder" is written in a large, dark, sans-serif font. Underneath this, the text "Turn a GitHub repo into a collection of interactive notebooks" is displayed in a large, dark, sans-serif font. Below this text is a paragraph of smaller, dark, sans-serif font: "Have a repository full of Jupyter notebooks? With Binder, you can add a badge that opens those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere." At the bottom of the page is a large, light-grey button with the text "Build a repository" in a dark, sans-serif font. To the right of this button is a small "submit" button with the word "submit" in a small, dark, sans-serif font.



github.com/freeman-lab

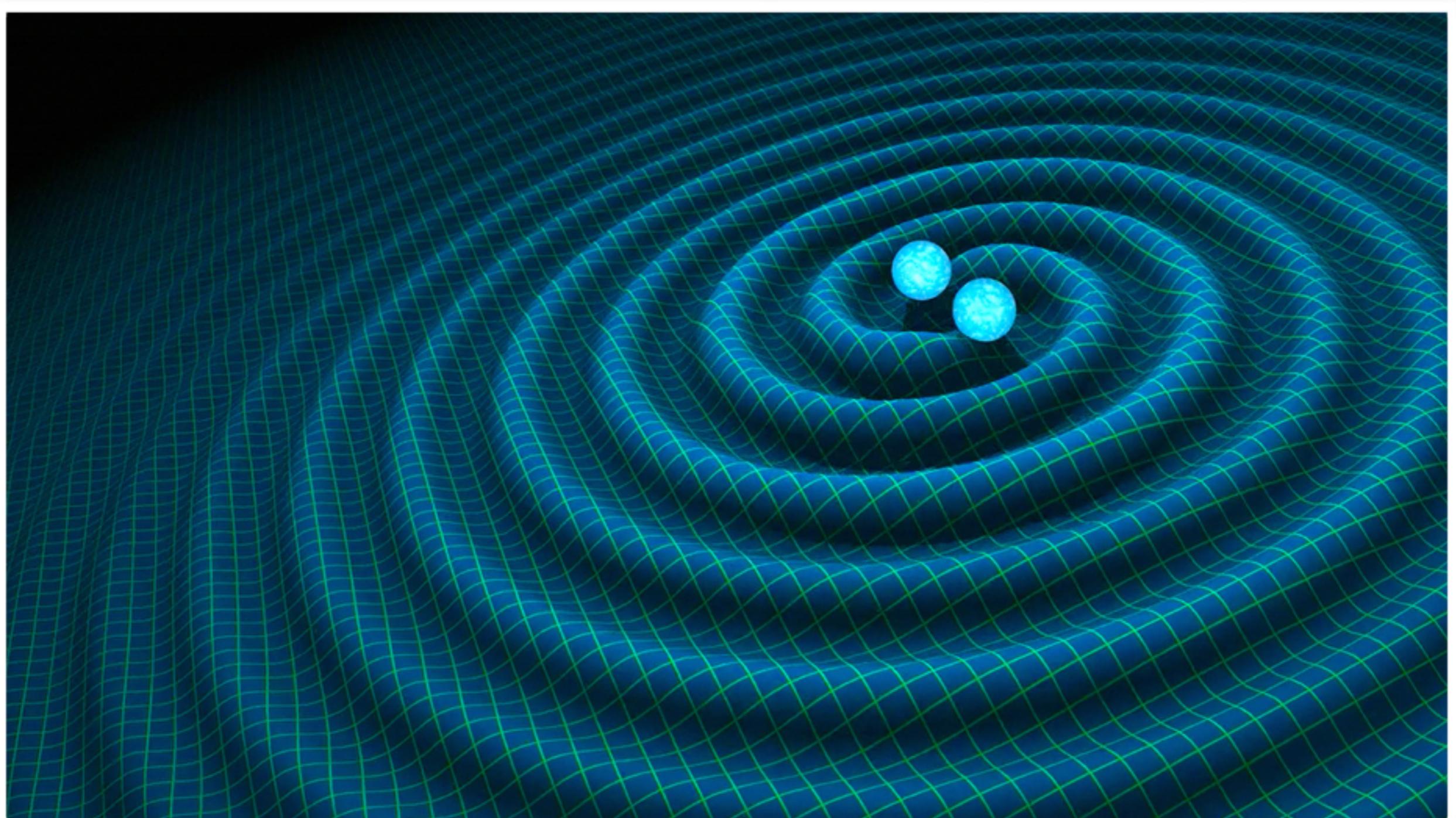


github.com/andrewosh

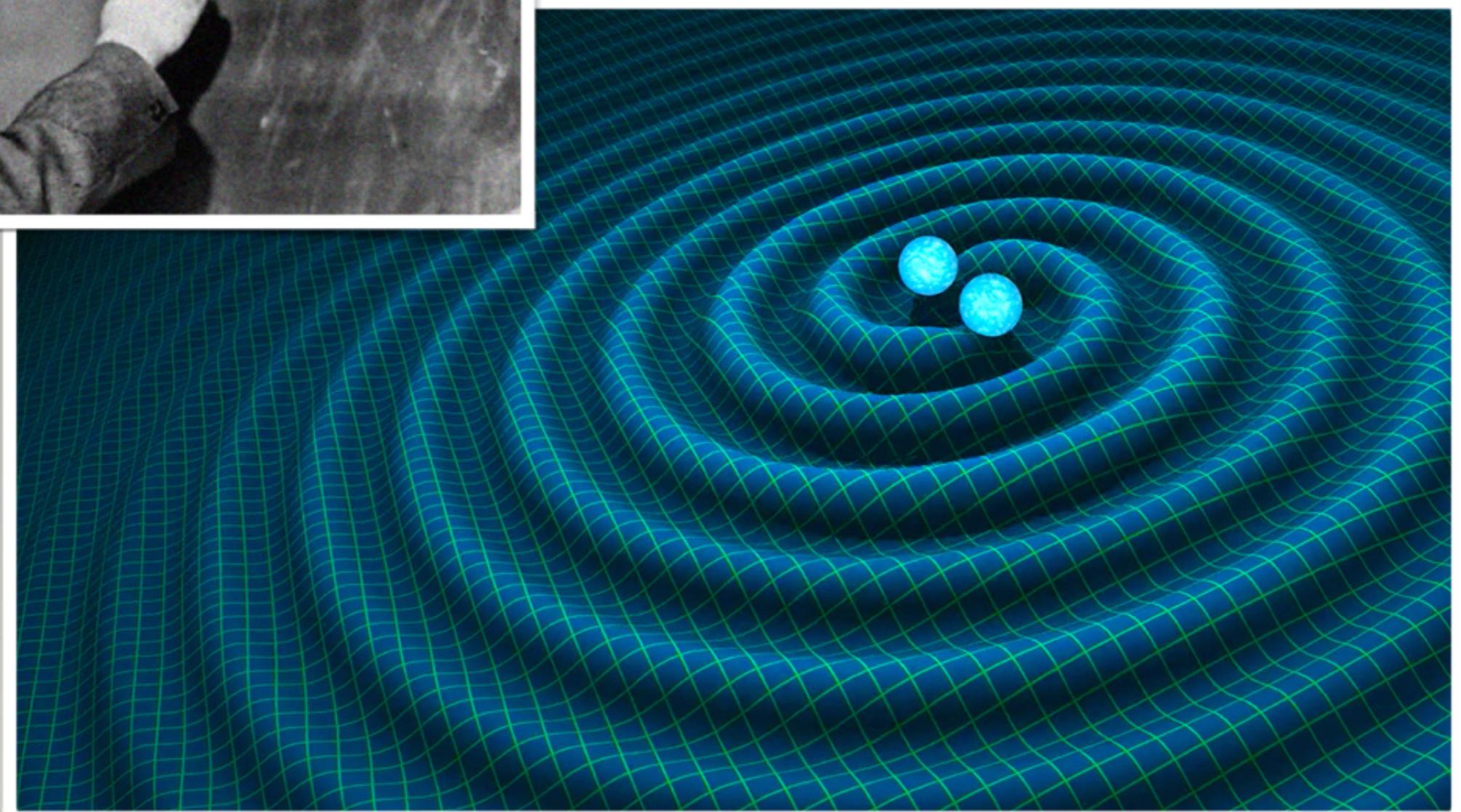
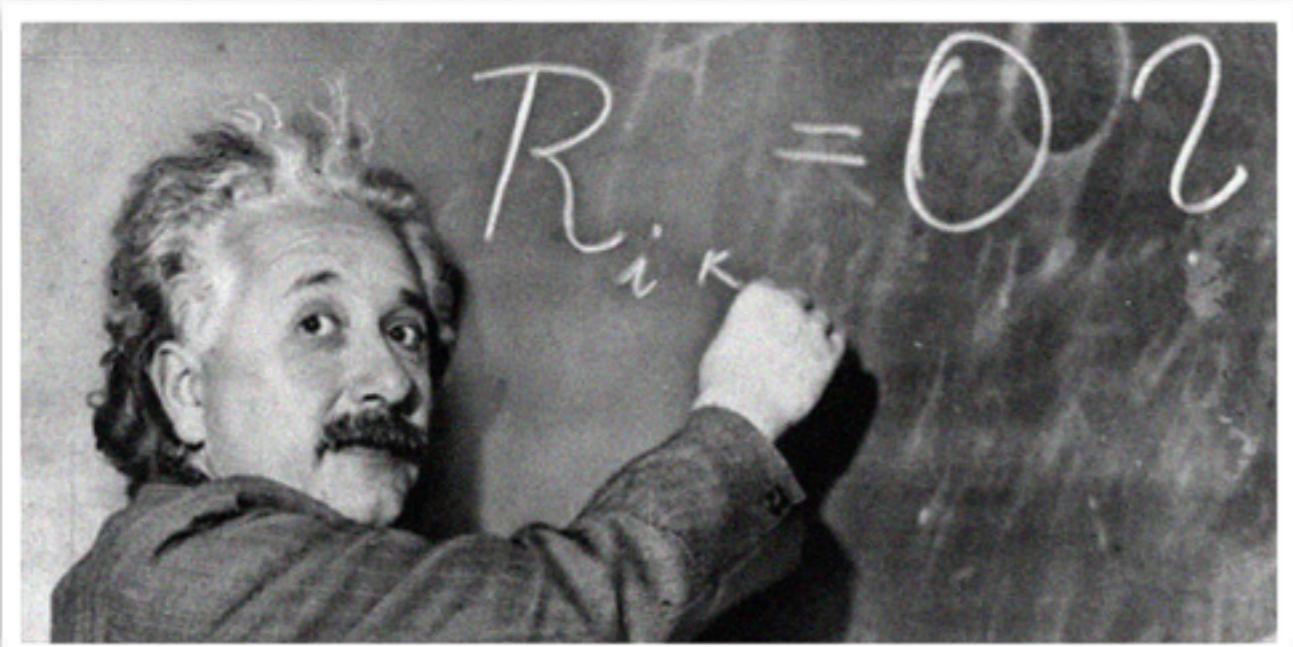
Andrew Osherooff's SciPy'16 talk:
<https://www.youtube.com/watch?v=OK6M4w7LYIc>

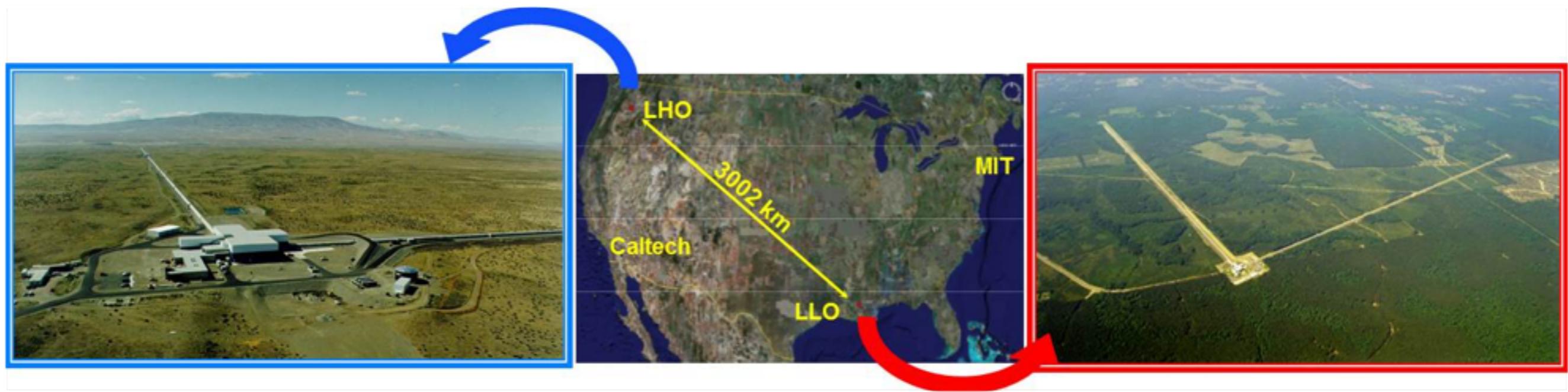
... to black holes and deep
learning

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:

- $\sim 1/1000$ proton over 4 km.
- Sensitivity $\sim 1\text{e-}21$
- Milky Way: $1\text{e+}21\text{m}$ across!

September 14, 2015

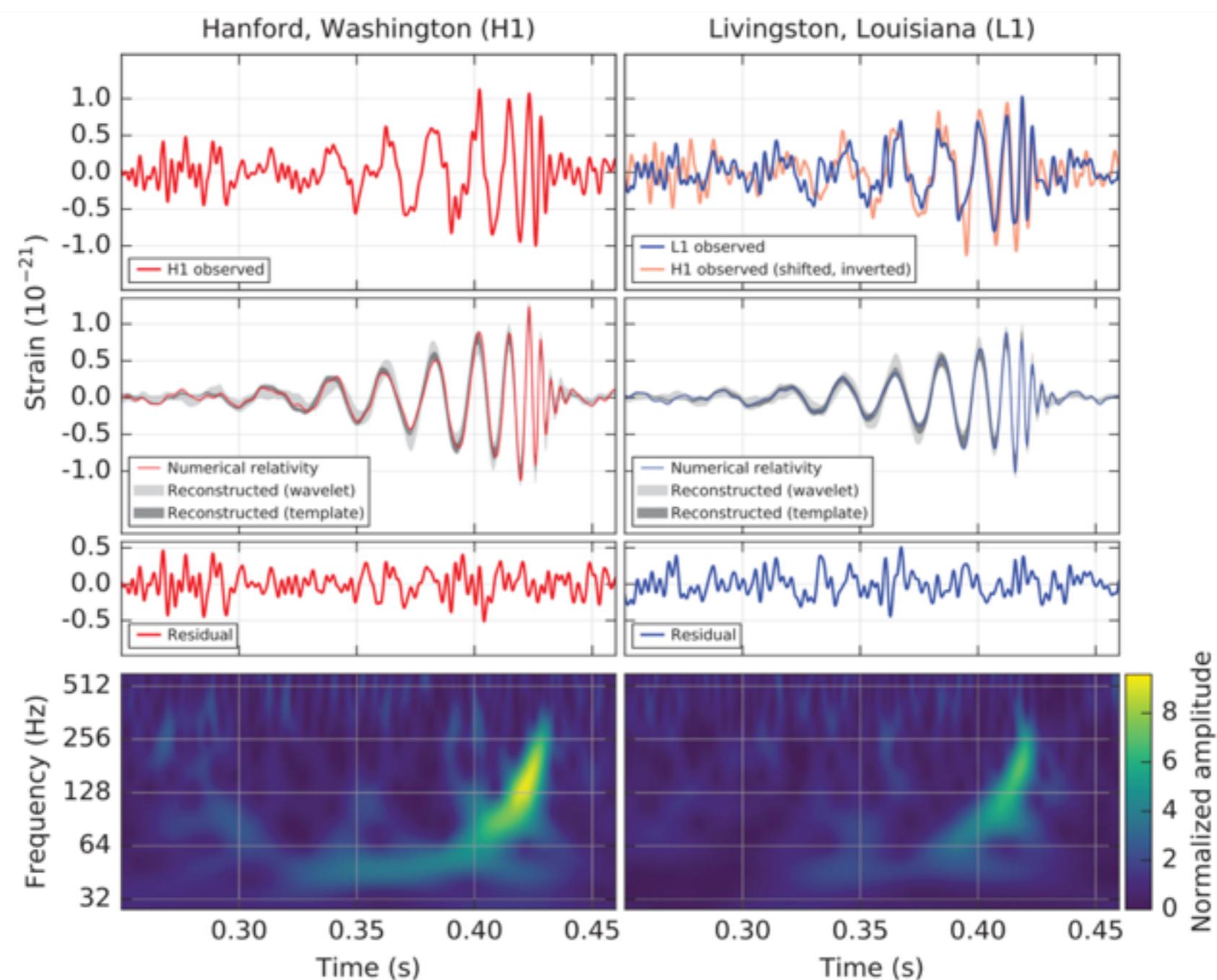
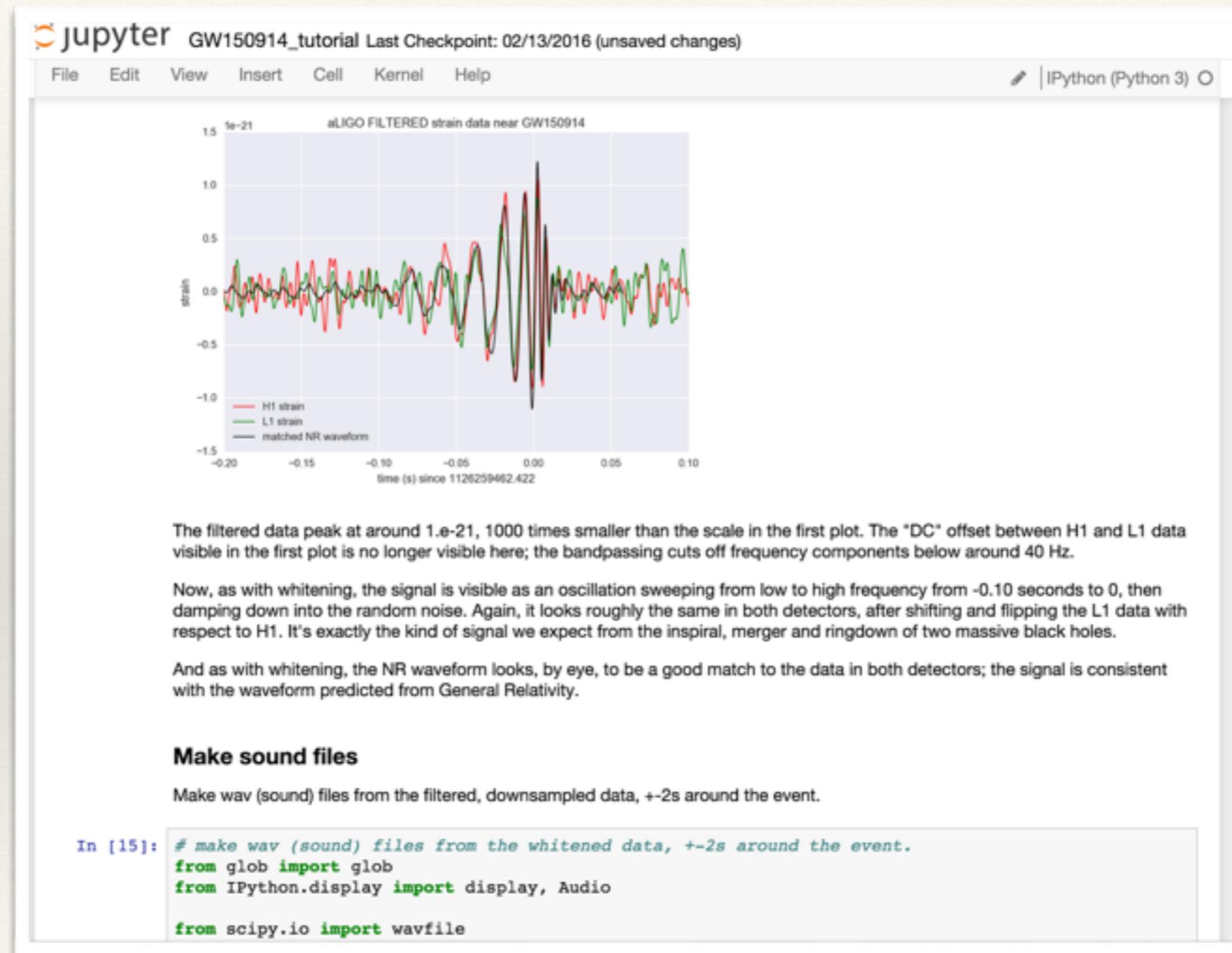
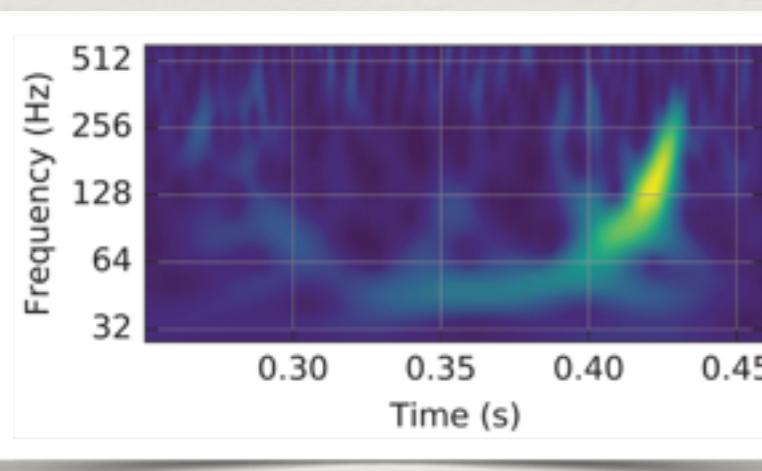
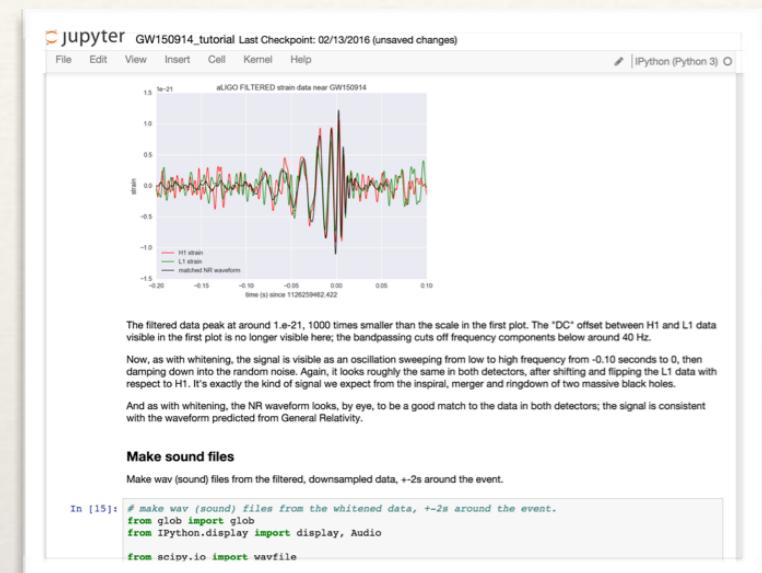


FIG. 1. The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress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

Gravitational waves detected on Jupyter!



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:
indx = np.where((time >= tevent-deltat) & (time < tevent+deltat))

# write the files:
write_wavfile("GW150914_H1_whitenbp.wav",int(fs), strain_H1_whitenbp[indx])
write_wavfile("GW150914_L1_whitenbp.wav",int(fs), strain_L1_whitenbp[indx])
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'
```



Using the IPython.display.Audio object

LIGO: Open Science with Jupyter

The diagram illustrates the integration of three platforms for open science:

- Microsoft Azure Notebooks** (left): A screenshot of the Microsoft Azure Notebooks interface, showing a "jupyter" logo and the text "Notebooks hosted on Microsoft Azure".
- LIGO Open Science Center** (center): A screenshot of the LIGO Open Science Center website. It features a "Tutorials" section with three examples: "Binary Black Hole Events", "Quickview Notebook", and "Signal Processing with GW150914". Each tutorial card includes "Run: Azure" and "mybinder" options, and download links for zip files, IPython notebooks, and Python scripts.
- binder** (right): The logo for the binder service, consisting of three interlocking rings in orange, pink, and blue.

Arrows indicate the integration and connection between these platforms:

- A blue arrow points from the "jupyter" logo on the Azure Notebooks page to the "Run: Azure" button on the LIGO tutorial cards.
- An orange arrow points from the "mybinder" button on the LIGO tutorial cards to the "binder" logo.

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?

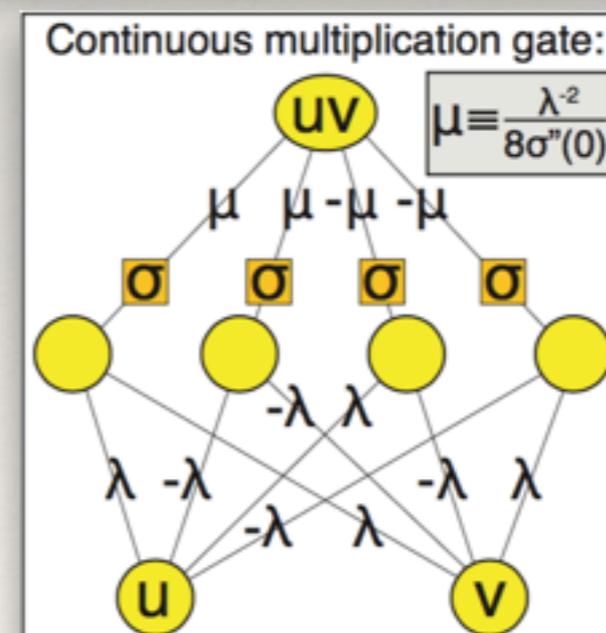
Henry W. Lin and Max Tegmark

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 f be a neural network of the form $f = \mathbf{A}_2\sigma\mathbf{A}_1$, where σ 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 f can approximate a multiplication gate arbitrarily well.



What I cannot create,
I do not understand.

I know how to solve every
problem that has been solved

Why const \times Sait. pg

TO LEARN:

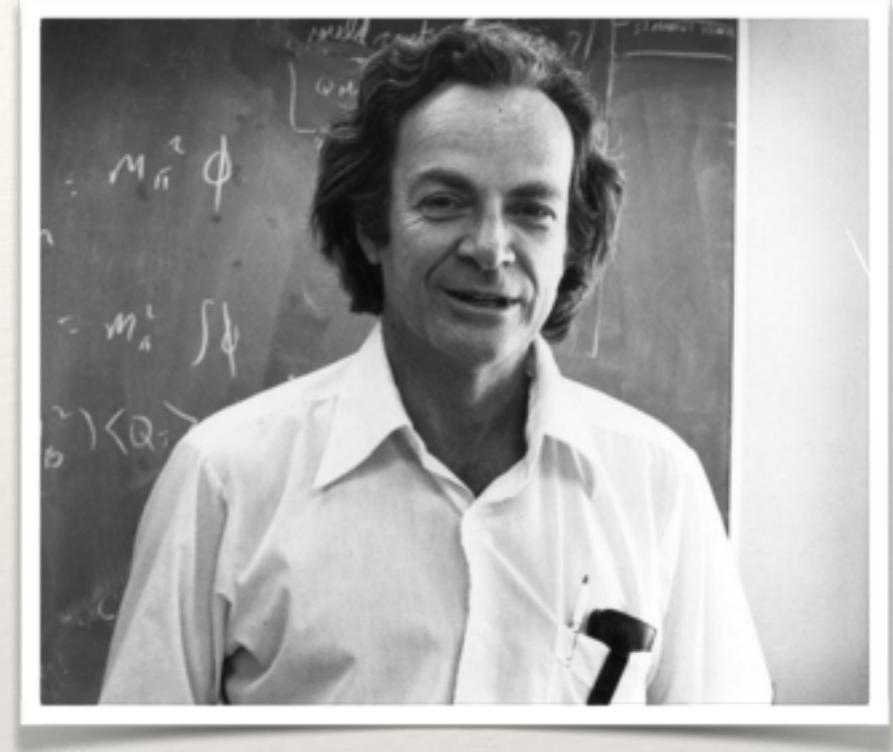
Bethe Ansatz Probs.
Kondo
2-D Hall
accel. Temp
Non Linear Classical Hydro

$$\textcircled{A} \quad f = u(r, \alpha)$$

$$g = 4(r-z) u(r, z)$$

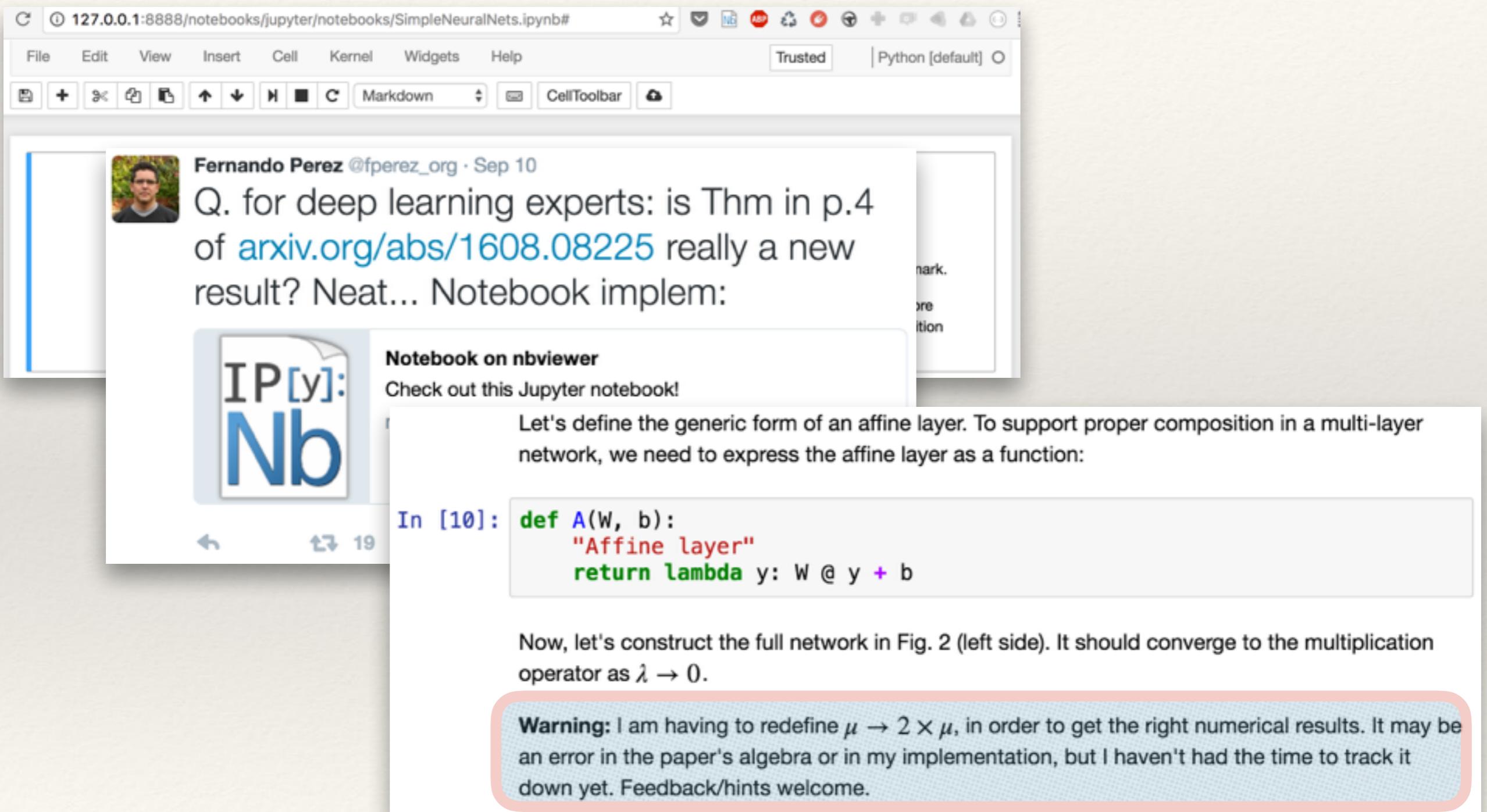
$$\textcircled{B} \quad f = 1/r \cdot \alpha / (u \cdot a)$$

© Copyright California Institute of Technology. All rights reserved.
Commercial use or modification of this material is prohibited.



What I can not create, I do not understand
Richard Feynman

Let's implement it!



Fernando Perez @fperez_org · Sep 10
Q. for deep learning experts: is Thm in p.4
of arxiv.org/abs/1608.08225 really a new
result? Neat... Notebook implem:

Notebook on nbviewer
Check out this Jupyter notebook!

In [10]:

```
def A(W, b):
    "Affine layer"
    return lambda y: W @ y + b
```

Now, let's construct the full network in Fig. 2 (left side). It should converge to the multiplication operator as $\lambda \rightarrow 0$.

Warning: I am having to redefine $\mu \rightarrow 2 \times \mu$, in order to get the right numerical results. It may be an error in the paper's algebra or in my implementation, but I haven't had the time to track it down yet. Feedback/hints welcome.

<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



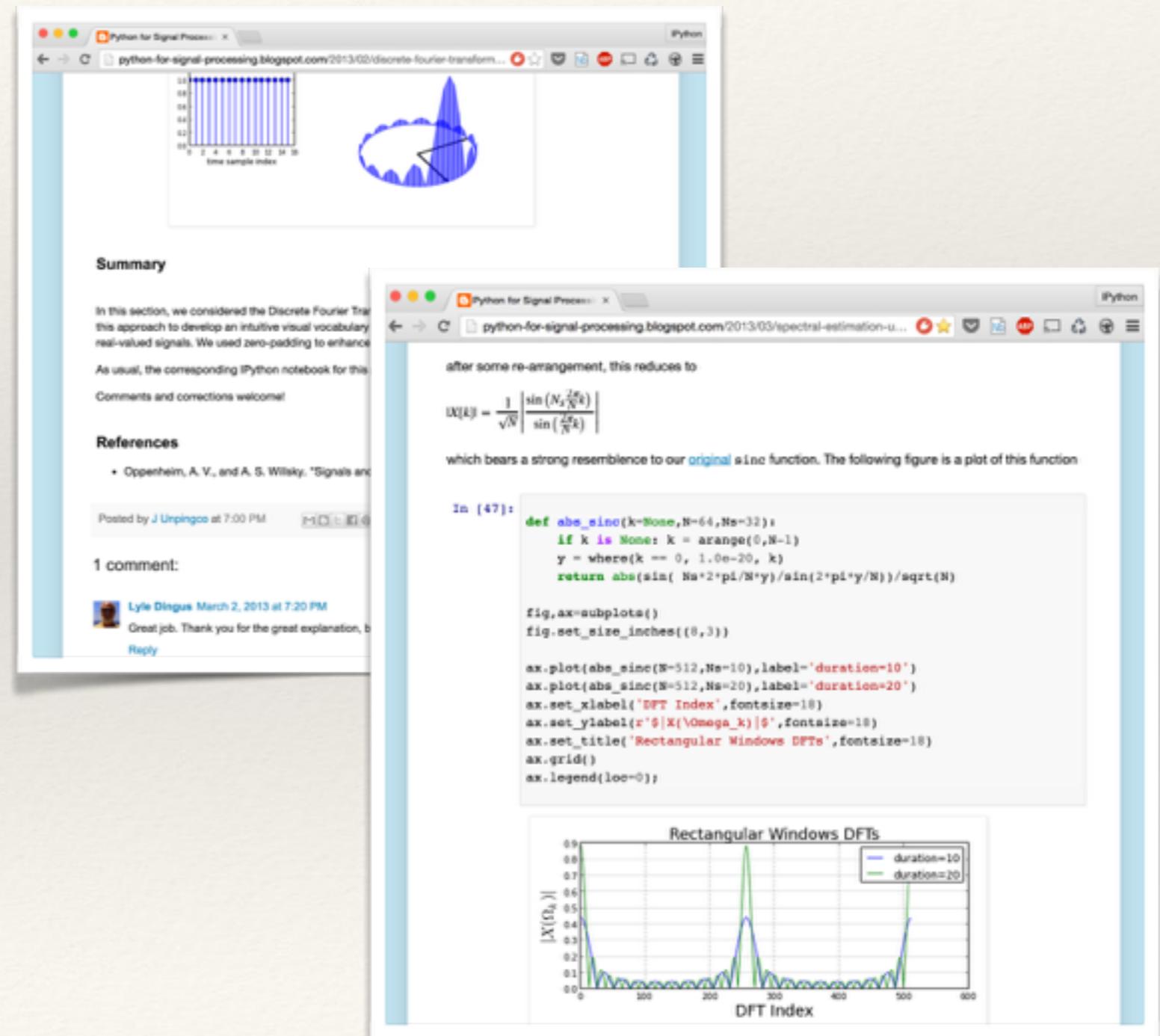
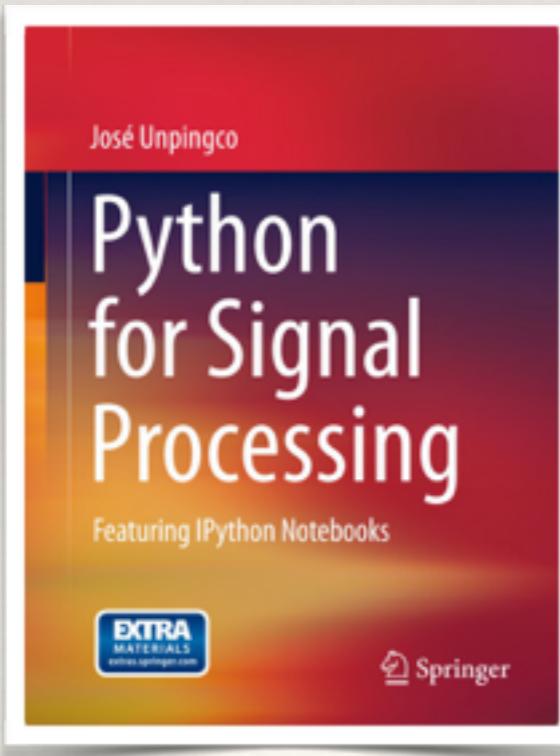
11

...

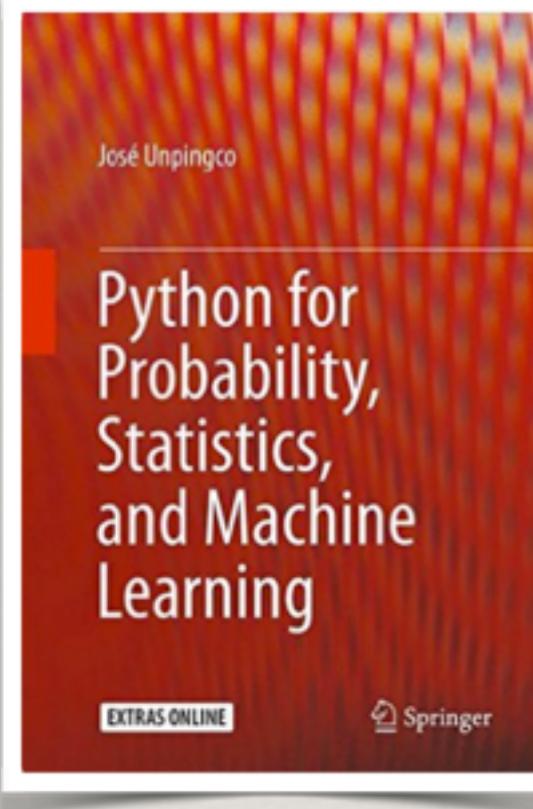
Executable books

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

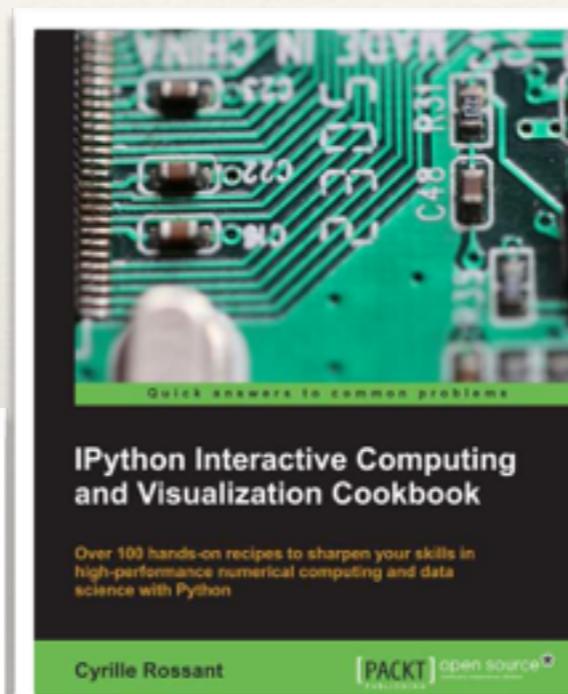


By Jose Unpingco

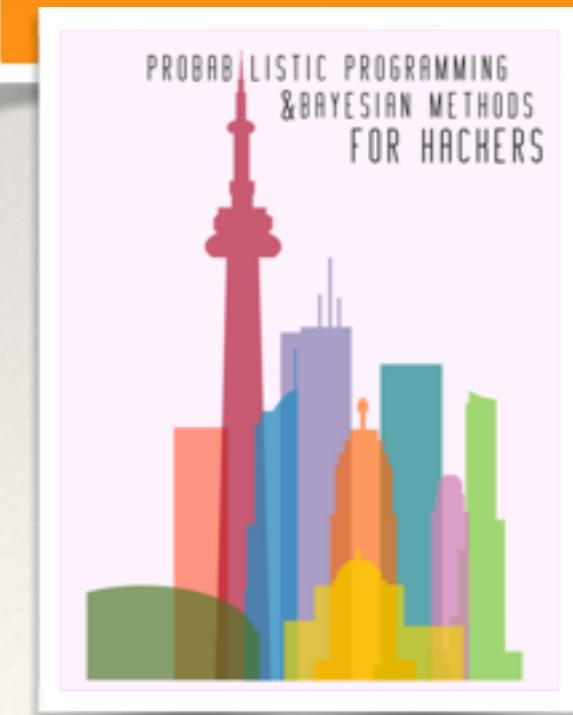
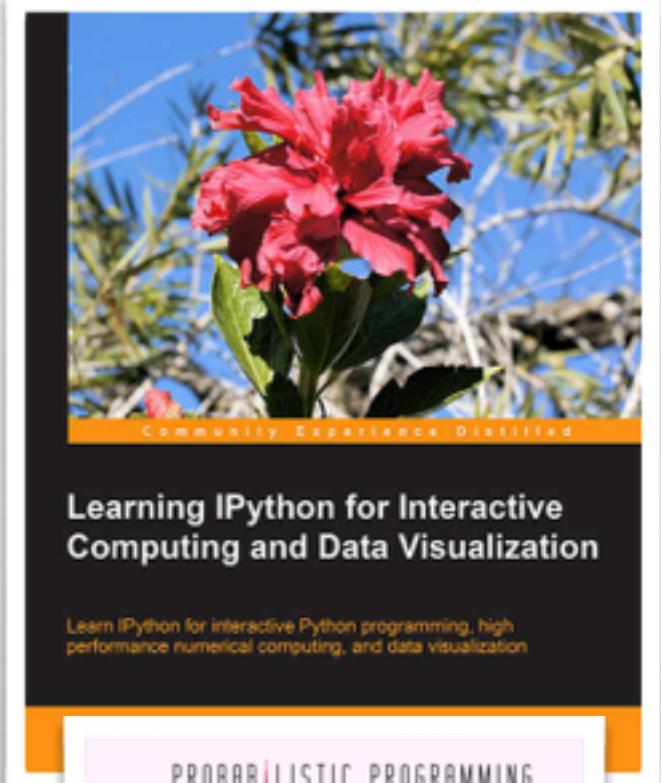


By Matthew Russell

By Cyrille Rossant



By Cameron Davidson-Pilon



Changing the scientific culture

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Authors

Archive > Volume 515 > Issue 7525 > Toolbox > Article

NATURE | TOOLBOX

Interactive notebooks: Sharing the code

The free IPython notebook makes data analysis easier to record, understand and reproduce.

Helen Shen

05 November 2014

PDF Rights & Permissions



Illustrations by The Project Twins

Search Go [Advanced search](#)

E- alert RSS Facebook Twitter

Top story



Beloved *Brontosaurus* makes a comeback

Jurassic giant's taxonomic status is restored.

Recent Read Comments Emailed

1. **History: Women at the edge of science**
Nature | 08 April 2015
2. **Scientific instrumentation: The aided eye**
Nature | 08 April 2015
3. **Books in brief**
Nature | 08 April 2015
4. **Antibody shows promise as**

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

Executable papers: the future?

nature.com · Sitemap · Log in · Close

IP[y]: Notebook · Nature (autosaved) · IPython (Python 3)

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

nature · rackspace

Introduction

Welcome! You have just launched a live example of an IPython Notebook. The notebook is an open-source, interactive computing environment that lets you combine live code, narrative text, mathematics, plots and rich media in one document. Notebook documents provide a complete reproducible record of a computation and its results and can be shared with colleagues (through, for example, email, web-hosting services such as GitHub, Dropbox, and nbviewer).

You can edit anything in this temporary demonstration notebook, including the text you are reading. To see it full-screen, click on the 'Expand' icon in the lower right corner of the frame around this notebook.

This notebook showcases some of IPython's capabilities for researchers.

This demonstration is hosted by [Rackspace](#) and is running on its bare metal offering, [OnMetal](#). Try out these cloud services yourself through [Rackspace's developer+ page](#).

Basic Python code and plotting

The box below (known as a code cell) contains the Python code to plot $y = x^2$ over the range $[0, 5]$. The blue comments preceded by `#` explain what the code does.

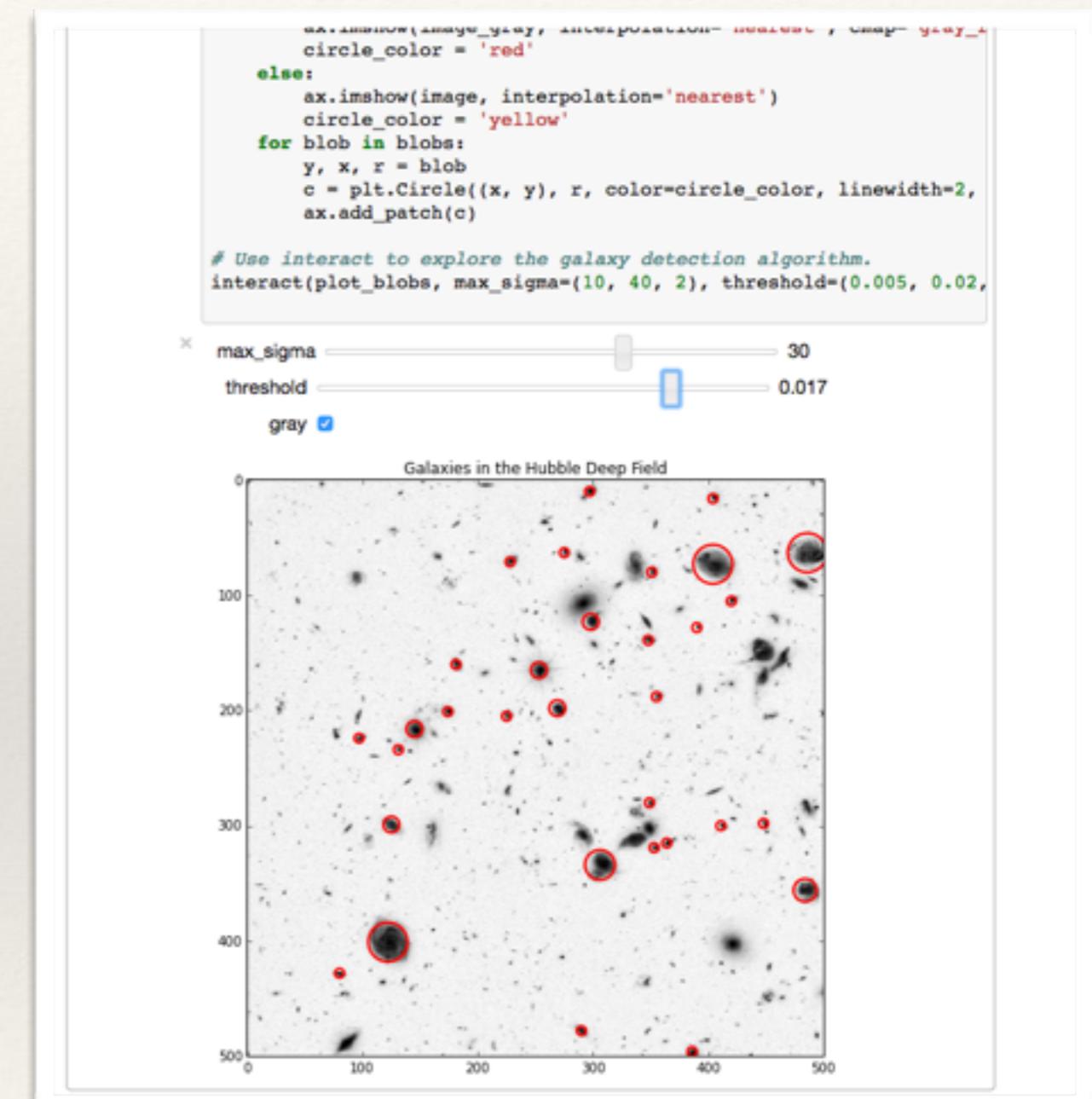
To run the code:

1. Click on the cell to select it.
2. Press SHIFT+ENTER on your keyboard or press the play button (\blacktriangleright) in the toolbar above.

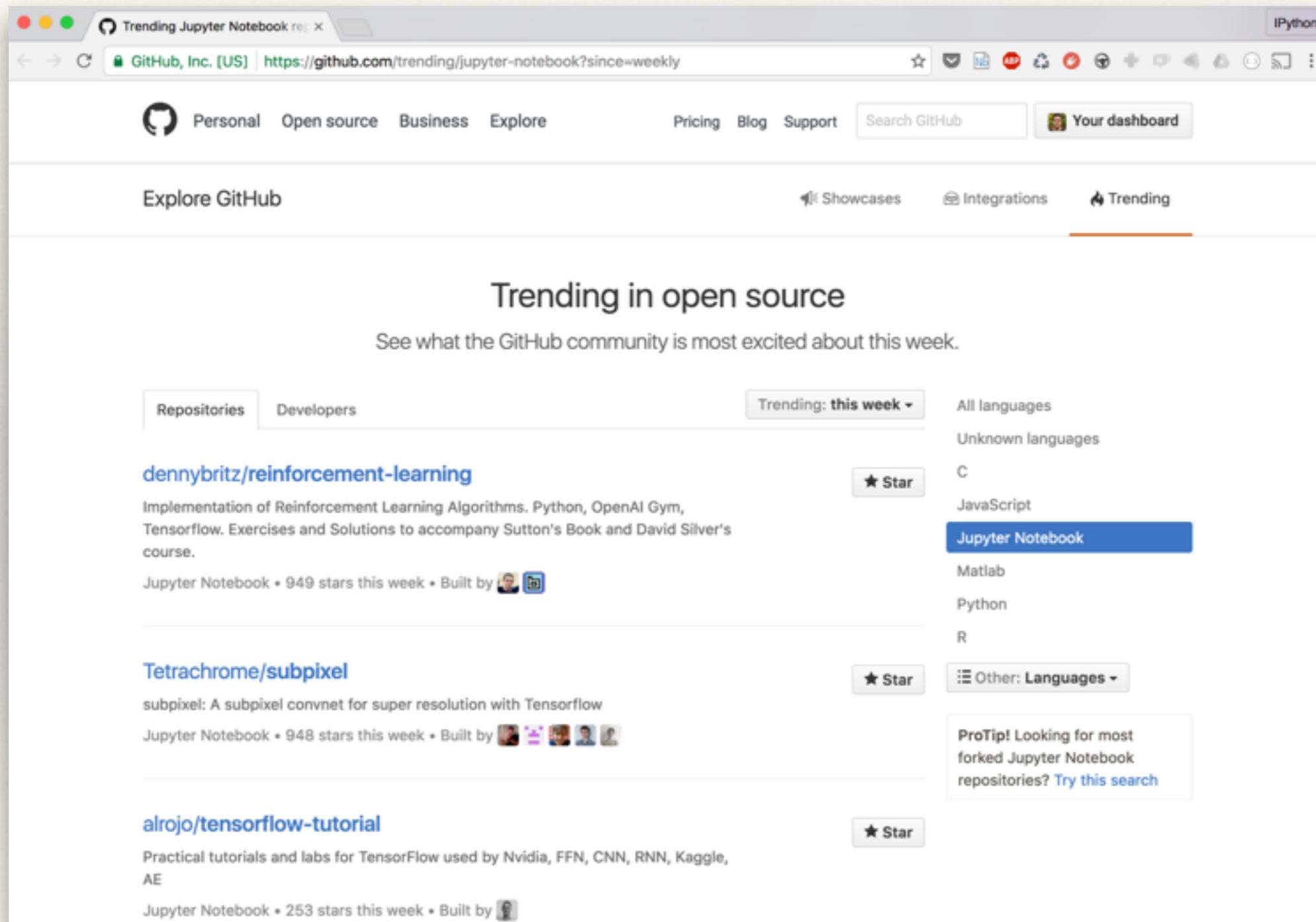
A full tutorial for using the notebook interface is available [here](#).

```
In [ ]: # Import matplotlib (plotting) and numpy (numerical arrays).
# This enables their use in the Notebook.
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

# Create an array of 30 values for x equally spaced from 0 to 5.
x = np.linspace(0, 5, 30)
```



Over 500,000 notebooks on Github



The screenshot shows a Mac OS X desktop with a browser window open to the GitHub trending Jupyter Notebook repository page. The URL in the address bar is <https://github.com/trending/jupyter-notebook?since=weekly>. The page title is "Trending Jupyter Notebook repositories". The GitHub header includes links for Personal, Open source, Business, Explore, Pricing, Blog, Support, and a search bar. The main content area is titled "Trending in open source" with the sub-instruction "See what the GitHub community is most excited about this week." It features three repository cards: 1. **dennybritz/reinforcement-learning**: A Jupyter Notebook repository with 949 stars this week, built by 2 people. It uses Python, OpenAI Gym, Tensorflow, and is related to Reinforcement Learning. 2. **Tetrachrome/subpixel**: A Jupyter Notebook repository with 948 stars this week, built by 6 people. It uses TensorFlow and is related to subpixel convolutional neural networks. 3. **alrojo/tensorflow-tutorial**: A Jupyter Notebook repository with 253 stars this week, built by 1 person. It uses TensorFlow and is related to practical tutorials and labs. A sidebar on the right lists trending languages: C, JavaScript, Jupyter Notebook (which is highlighted in blue), Matlab, Python, and R. A "ProTip!" box suggests trying a specific search for the most forked Jupyter Notebook repositories.

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

Notebook Workflows: The Big Picture

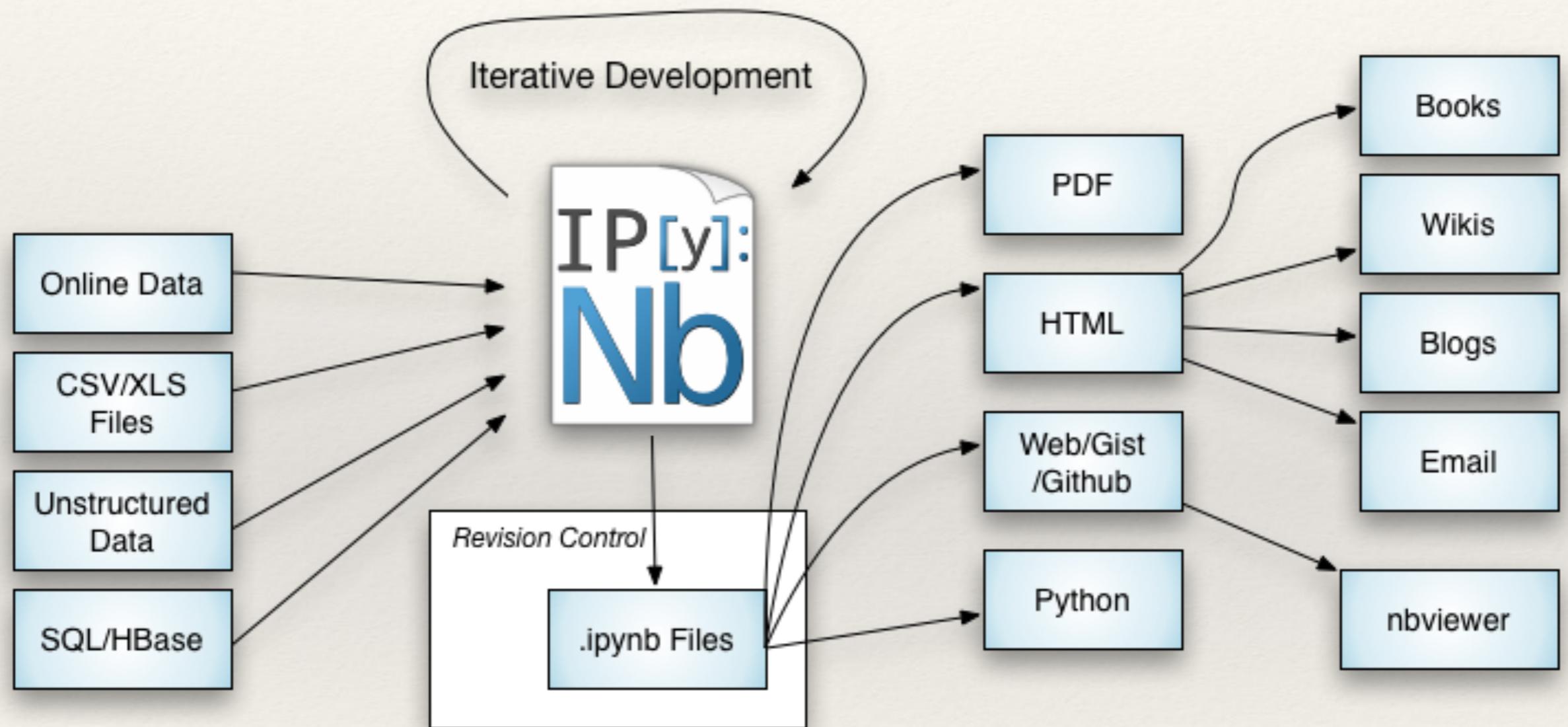


Image credit: [Joshua Barratt](#)

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.



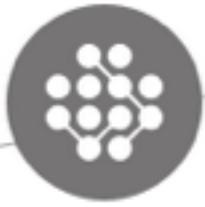
Centralized deployment

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

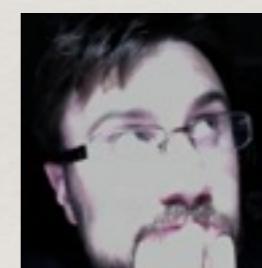
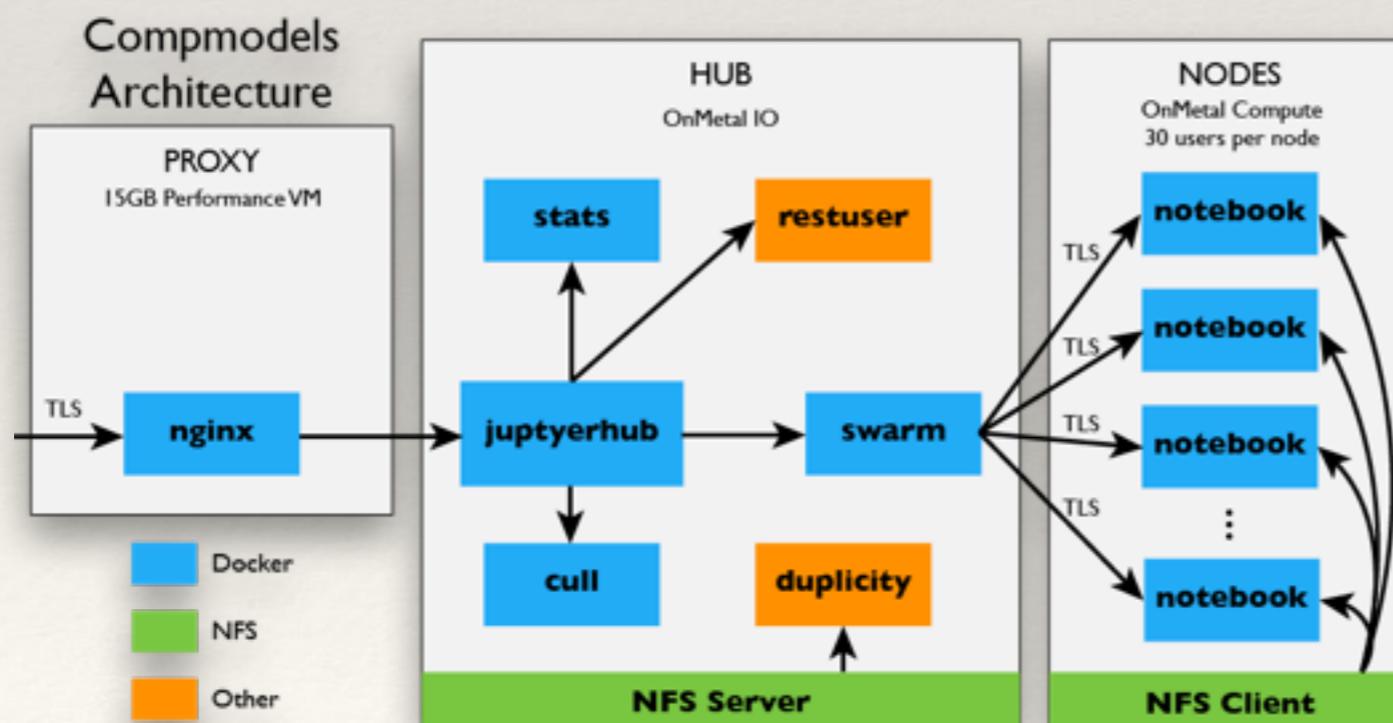
Deploy the Notebook next to your data to provide unified software management and data access within your organization.

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

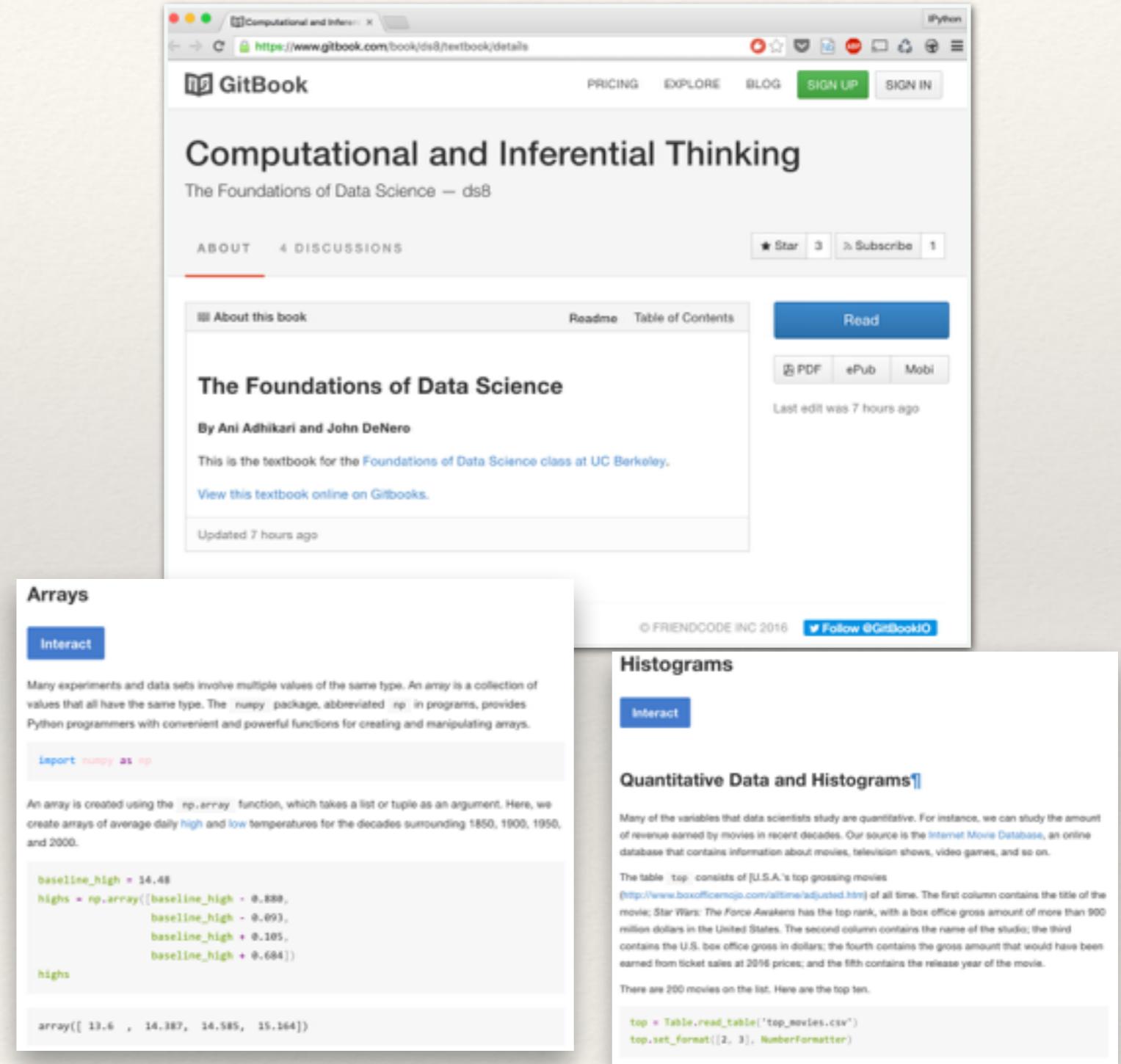


K. Kelley
Rackspace M. Ragan-Kelley
Cal B. Granger
Cal Poly



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



The screenshot shows the GitBook interface for the "Computational and Inferential Thinking" section of the "The Foundations of Data Science" textbook. The page title is "Computational and Inferential Thinking" and the subtitle is "The Foundations of Data Science – ds8". The page includes sections for "ABOUT" and "4 DISCUSSIONS". A "Read" button is prominent, along with links for "PDF", "ePub", and "Mobi". Below the main content, there are two Jupyter Notebook cells. The first cell, titled "Arrays", shows Python code for creating a numpy array of average daily high and low temperatures for the decades surrounding 1850, 1900, 1950, and 2000. The second cell, titled "Histograms", shows code for reading a table of top-grossing movies from a CSV file and displaying the top 10 movies. The GitBook footer includes copyright information and social media links.

Arrays

Interact

```
import numpy as np
```

An array is created using the `np.array` function, which takes a list or tuple as an argument. Here, we create arrays of average daily `high` and `low` temperatures for the decades surrounding 1850, 1900, 1950, and 2000.

```
baseline_high = 14.48
highs = np.array([baseline_high + 0.889,
                 baseline_high + 0.093,
                 baseline_high + 0.105,
                 baseline_high + 0.684])
highs
```

```
array([ 13.6 ,  14.387,  14.585,  15.164])
```

Histograms

Interact

Quantitative Data and Histograms

```
top = Table.read_table('top_movies.csv')
top.set_format([2, 3], NumberFormatter)
```

Many of the variables that data scientists study are quantitative. For instance, we can study the amount of revenue earned by movies in recent decades. Our source is the [Internet Movie Database](#), an online database that contains information about movies, television shows, video games, and so on. The table `top` consists of [U.S.A.'s top grossing movies (<http://www.boxofficemojo.com/alltime/adjusted.htm>) of all time. The first column contains the title of the movie; *Star Wars: The Force Awakens* has the top rank, with a box office gross amount of more than 900 million dollars in the United States. The second column contains the name of the studio; the third contains the U.S. box office gross in dollars; the fourth contains the gross amount that would have been earned from ticket sales at 2016 prices; and the fifth contains the release year of the movie. There are 200 movies on the list. Here are the top ten.

Data Science: *Connector Courses*



Data Science,
Demography, &
Immigration



Children in the
Developing World



Data Science
For Smart Cities



Data and Ethics



Social Networks



Computational
Structures in
Data Science

BERKELEY DATA SCIENCE EDUCATION PROGRAM

Fall 2016 Connector Course Offerings

Designed to Complement

Data 8: Foundations of Data Science

data.berkeley.edu



Social Data
Revolution



Making Sense of
Cultural Data



Genomics and Data
Science



Data Science for
Cognitive
Neuroscience



Data Science and
the Mind

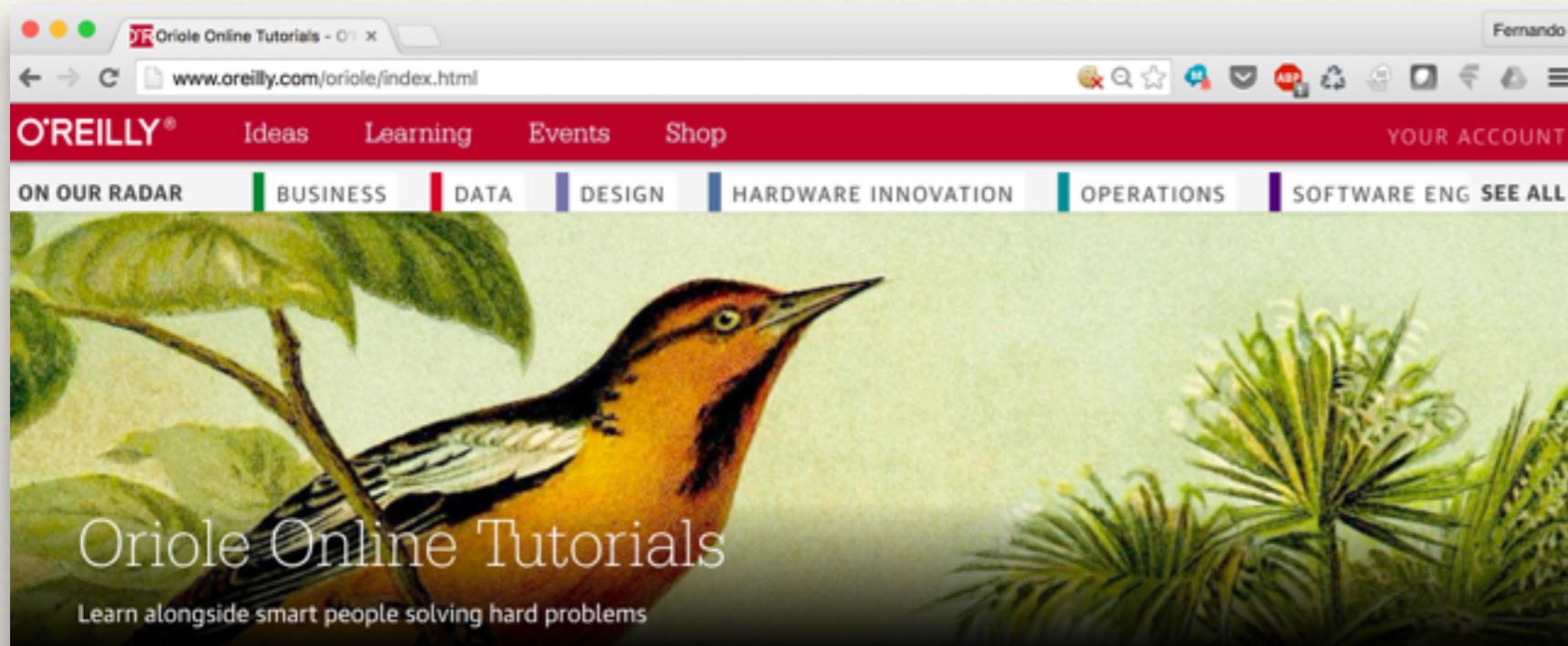


Probability and
Math Stats in
Data Science

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



Oriole Online Tutorials

Learn alongside smart people solving hard problems

Oriole is a unique new medium that blends code, data, text, and video into a narrated learning experience with executable content.

Led by some of the most brilliant minds in technology, each lesson is an easily digestible and engaging thought-by-thought tour of the instructor's approach to the problem in both narrative and executable code. No set-up or installation is necessary; Oriole Online Tutorials require nothing more than an internet connection and a laptop. You can write and run code within the environment. Make a mistake? Change it, and try again.

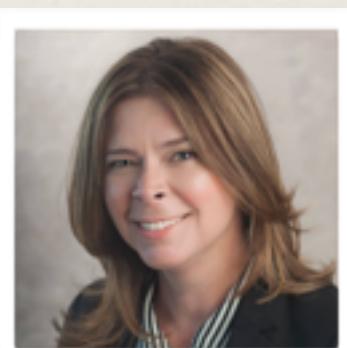
Oriole combines the expert insight and hands-on learning of in-person or online courses with the on-demand, at-your-own desk, back-up-and run-it-again convenience of video training. You learn by doing, on your own schedule, and at your own pace.

In Oriole, we get the complete integration of video synchronized with the flow of the text, as well as the ability to execute the code: **this is probably as close as we can get to learning side-by-side with Peter himself.**

Fernando Perez, creator of IPython, which evolved into Project Jupyter.



Paco Nathan



Taylor Martin



Andrew Odewahn

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

Oriole: from static notebook...



The screenshot shows a Jupyter Notebook interface with a comic strip and some code.

Comic Strip:

META-REGEX GOLF:
SO I WROTE A PROGRAM THAT PLAYS REGEX GOLF WITH ARBITRARY LISTS...

UH OH...

In [15]:

```
drugs = words('lipitor nexium plavix advair ablify seroquel singulair crestor actos epogen')
cities = words('paris trinidad capetown riga zurich shanghai vancouver chicago adelaide auckland')
report(drugs, cities)
```

Characters: 15, Parts: 6, Competitive ratio: 5.3, Winners: 10, Losers: 10

Out[15]: 'o.\$lxlrlqlfipo'

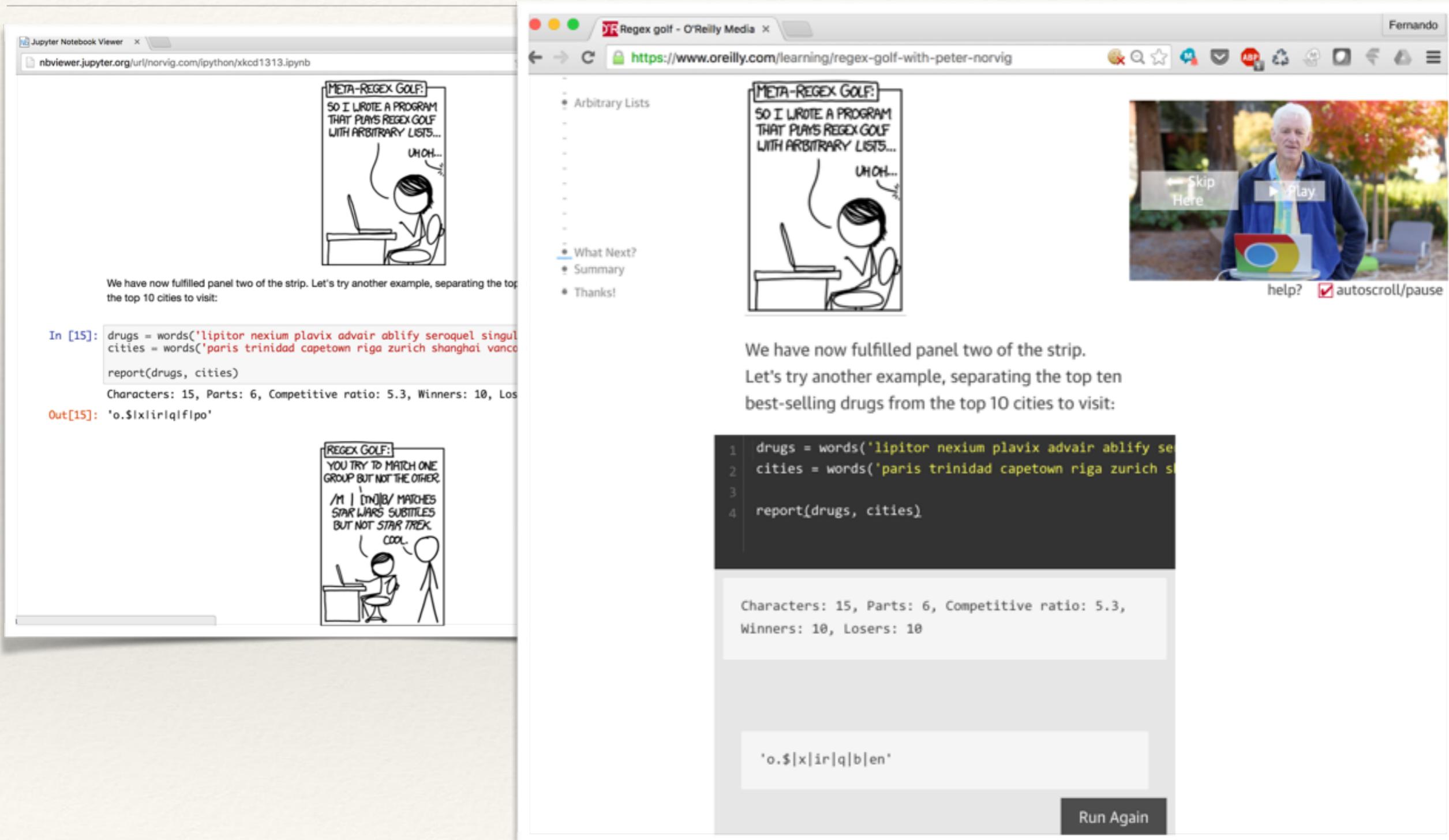
REGEX GOLF:
YOU TRY TO MATCH ONE GROUP BUT NOT THE OTHER.
/M | [tn]B/ MATCHES STAR WARS SUBTLES BUT NOT STAR TREK.
COOL.

Photo:



A portrait of a man with white hair, smiling, wearing a green patterned shirt.

... to executable, video-narrated tutorial:



The image shows two side-by-side screenshots of a tutorial on regular expressions. The left screenshot is from a Jupyter Notebook viewer, and the right is from a web-based O'Reilly Media tutorial.

Jupyter Notebook Viewer (Left):

- Header:** nbviewer.jupyter.org/url/norvig.com/ipython/xkcd1313.ipynb
- Content:**
 - Image:** A comic strip titled "META-REGEX GOLF" showing a person at a computer. The text in the box says: "SO I WROTE A PROGRAM THAT PLAYS REGEX GOLF WITH ARBITRARY LISTS..." with a "UH OH..." thought bubble.
 - Text:** "We have now fulfilled panel two of the strip. Let's try another example, separating the top ten best-selling drugs from the top 10 cities to visit:
 - Code:**

```
In [15]: drugs = words('lipitor nexium plavix advair abilify seroquel singul
cities = words('paris trinidad capetown riga zurich shanghai vanc
report(drugs, cities)
Characters: 15, Parts: 6, Competitive ratio: 5.3, Winners: 10, Los
Out[15]: 'o.$!x!r!q!f!po'
```
 - Image:** A comic strip titled "REGEX GOLF" showing two people. The text in the box says: "YOU TRY TO MATCH ONE GROUP BUT NOT THE OTHER. /M | [n]B/ MATCHES STAR WARS SUBTITLES BUT NOT STAR TREK. COOL."

O'Reilly Media (Right):

- Header:** Regex golf - O'Reilly Media
- Content:**
 - Image:** The same "META-REGEX GOLF" comic strip as in the Jupyter viewer.
 - Navigation:** Arbitrary Lists, What Next?, Summary, Thanks!
 - Text:** "We have now fulfilled panel two of the strip. Let's try another example, separating the top ten best-selling drugs from the top 10 cities to visit:
 - Code:**

```
1 drugs = words('lipitor nexium plavix advair abilify se
2 cities = words('paris trinidad capetown riga zurich s
3
4 report(drugs, cities)
```
 - Text:** "Characters: 15, Parts: 6, Competitive ratio: 5.3, Winners: 10, Losers: 10
 - Output:** 'o.\$!x!r!q!b!en'
 - Video Player:** A video player interface with a thumbnail of a man, a "Skip Here" button, a "Play" button, and a "help?" checkbox.

Microsoft, IBM, Google, Continuum...

Microsoft

Machine Learning Blog
Introducing Jupyter Notebooks in Azure ML Studio

Microsoft Azure Notebooks PREVIEW

WHAT IS JUPYTER?

- Interactive Notebooks for Data Science and Technical Computing
- Browser-based REPL with Markdown and inline interactive graphics
- Support for Python 2, Python 3 and R

ABOUT THIS SERVICE

- This notebook service is provided by the Azure Data Group
- Your notebooks are stored in Azure and linked to your Microsoft account
- Enjoy some free cycles on us

IBM

Data Scientist Workbench
Prepare data. Analyze data. Get answers.

Prepare data effortlessly.

Explore Data.
Find and explore large data sets with ease.

Clean and Transform Data
Easily clean messy data and transform formats.

Reconcile and Match Data
Link and extend your datasets with web services.

Analyze data interactively.

Powerful Notebook Environment
Use Python/Jupyter notebooks to combine code execution, text, plots and rich media.

Cloud DataLab

Google Cloud Platform

Cloud DataLab™
An easy-to-use interactive tool for large-scale data exploration, analysis, and visualization.

TRY IT FREE

Powerful Data Exploration

Cloud DataLab is a powerful interactive tool created to explore, analyze and visualize data with a single click on Google Cloud Platform. It runs on Google App Engine and orchestrates multiple services automatically so you can focus on exploring your data.

Google

CONTINUUM ANALYTICS

CONTINUUM ANALYTICS

Gallery About Pricing Anaconda Help Download Anaconda Sign In

Interactive Stock Prices Downsampling

Hover Over Points

My Gist Activity

Data Visualization in Python

Scientific Programming in Python

Texas Unemployment Choropleth

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>



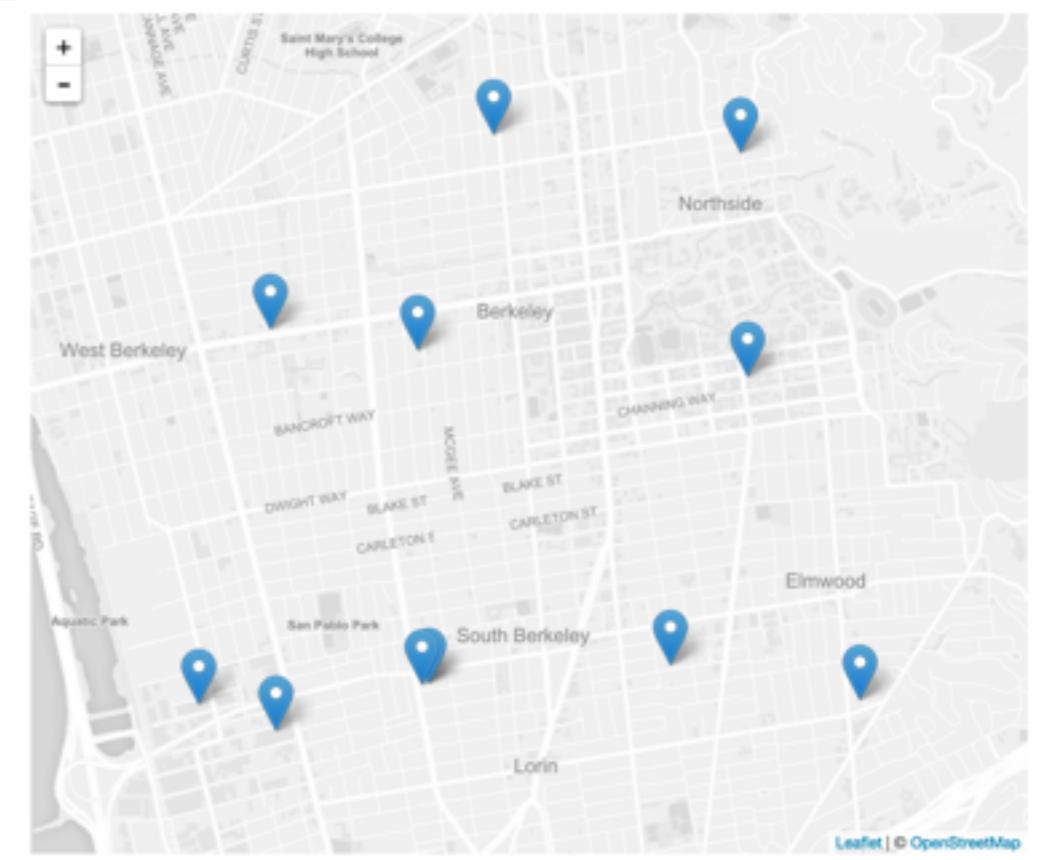
Bamboo

Converting from Pandas DataFrame to GeoJSON to Leaflet Map

```
[12] import IPython
cols = ['street_address', 'issue_description', 'issue_type', 'ticket_status']
geojson = df_to_geojson(df_geo, cols)

def plot_geo(geojson):
    IPython.display.display({'application/vnd.geo+json': geojson}, raw=True)
```

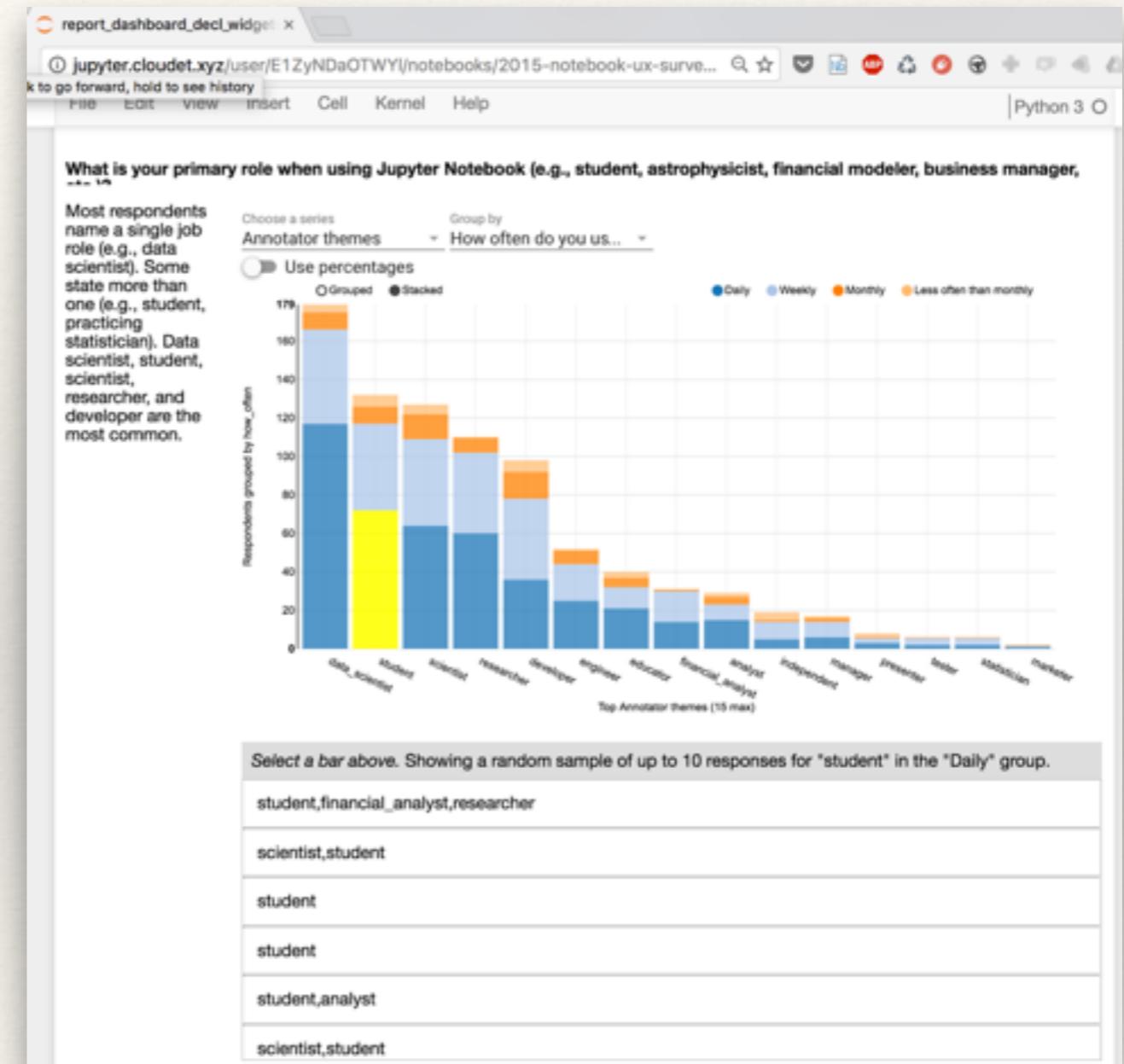
[22] plot_geo(geojson)



Leaflet | © OpenStreetMap

JupyterLab: the notebook,
evolved...

Estimated user base: at least ~3M

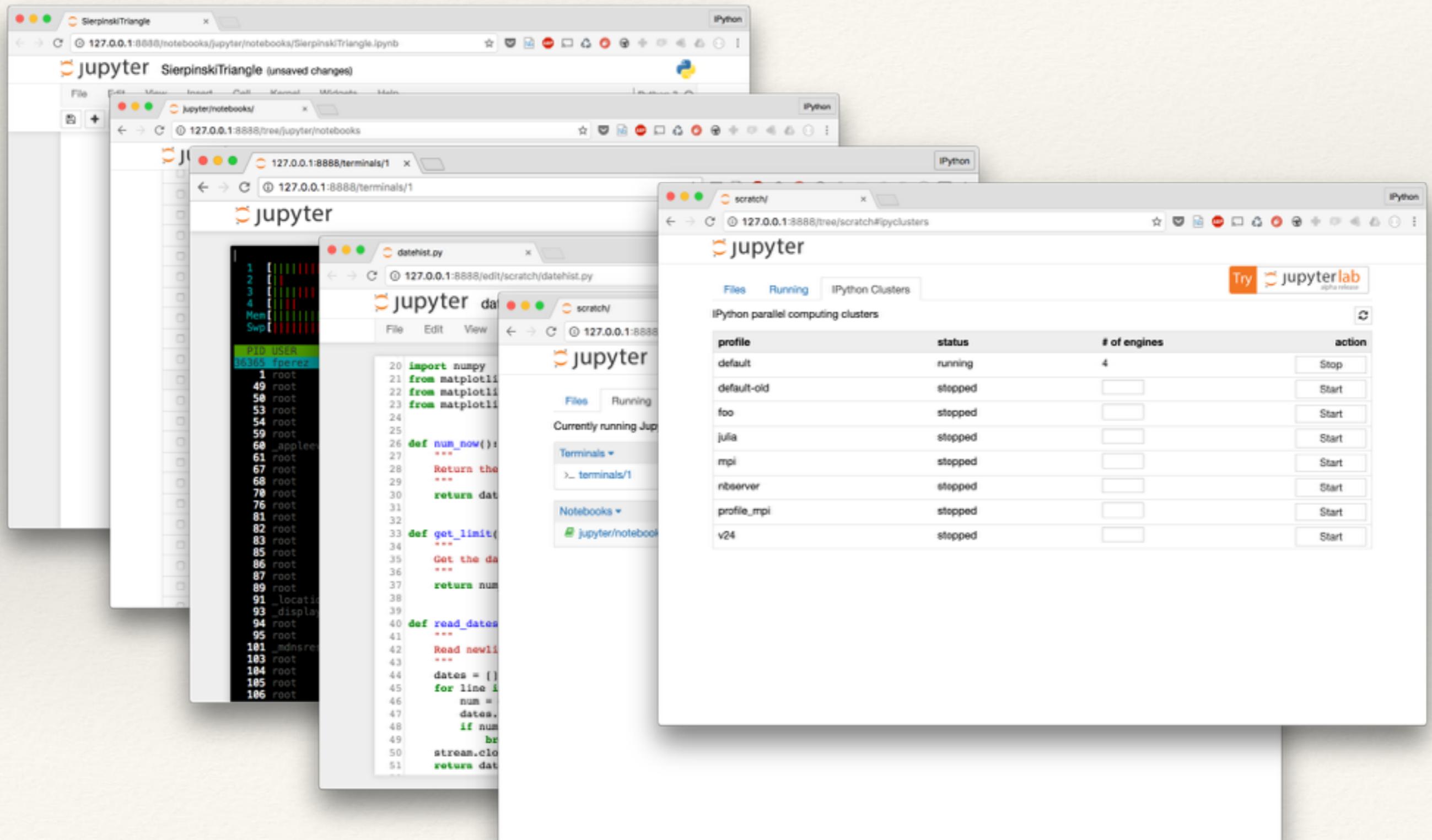


IBM UX User Survey: a live, interactive dashboard

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

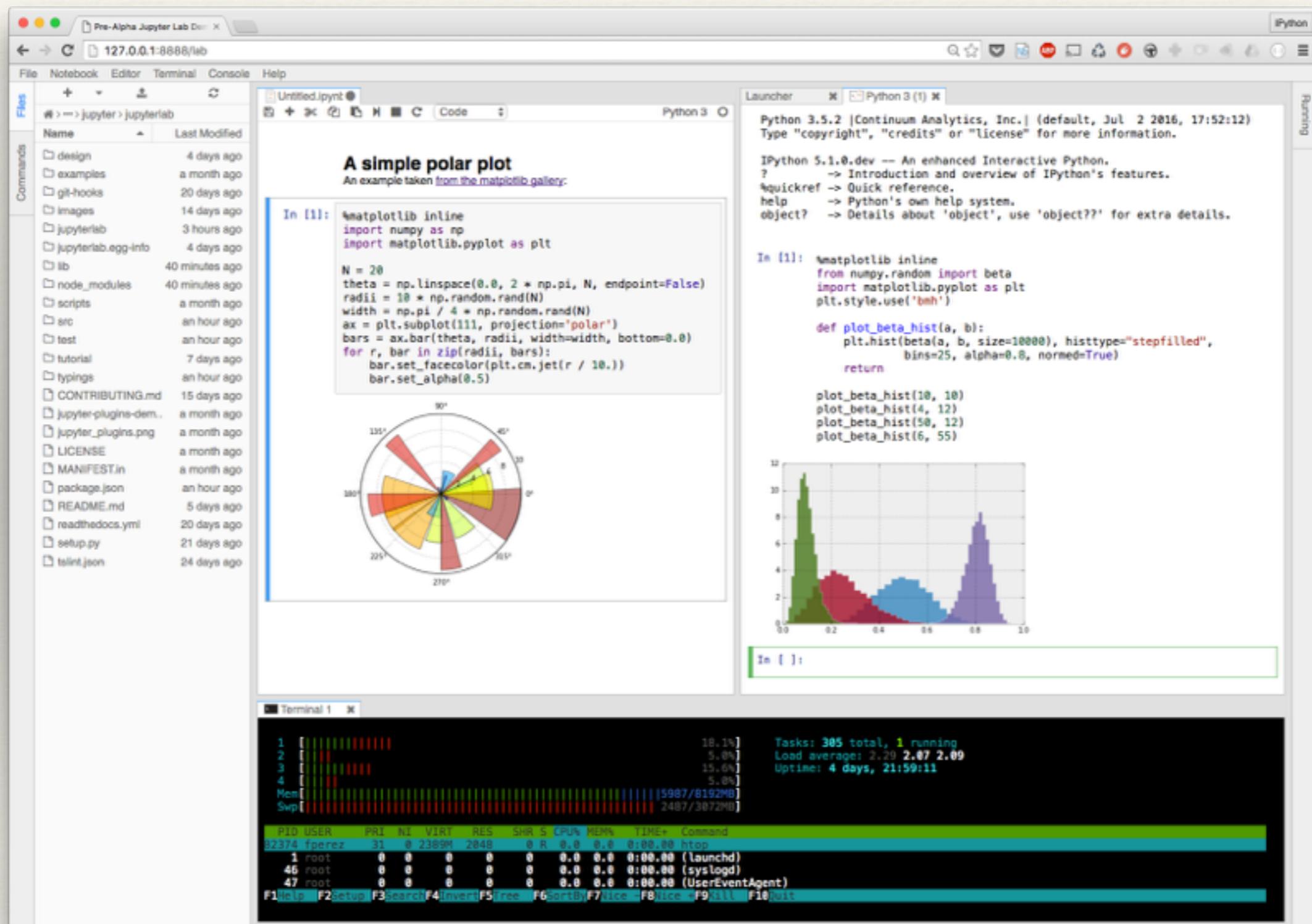
Terminal

Text
Editor

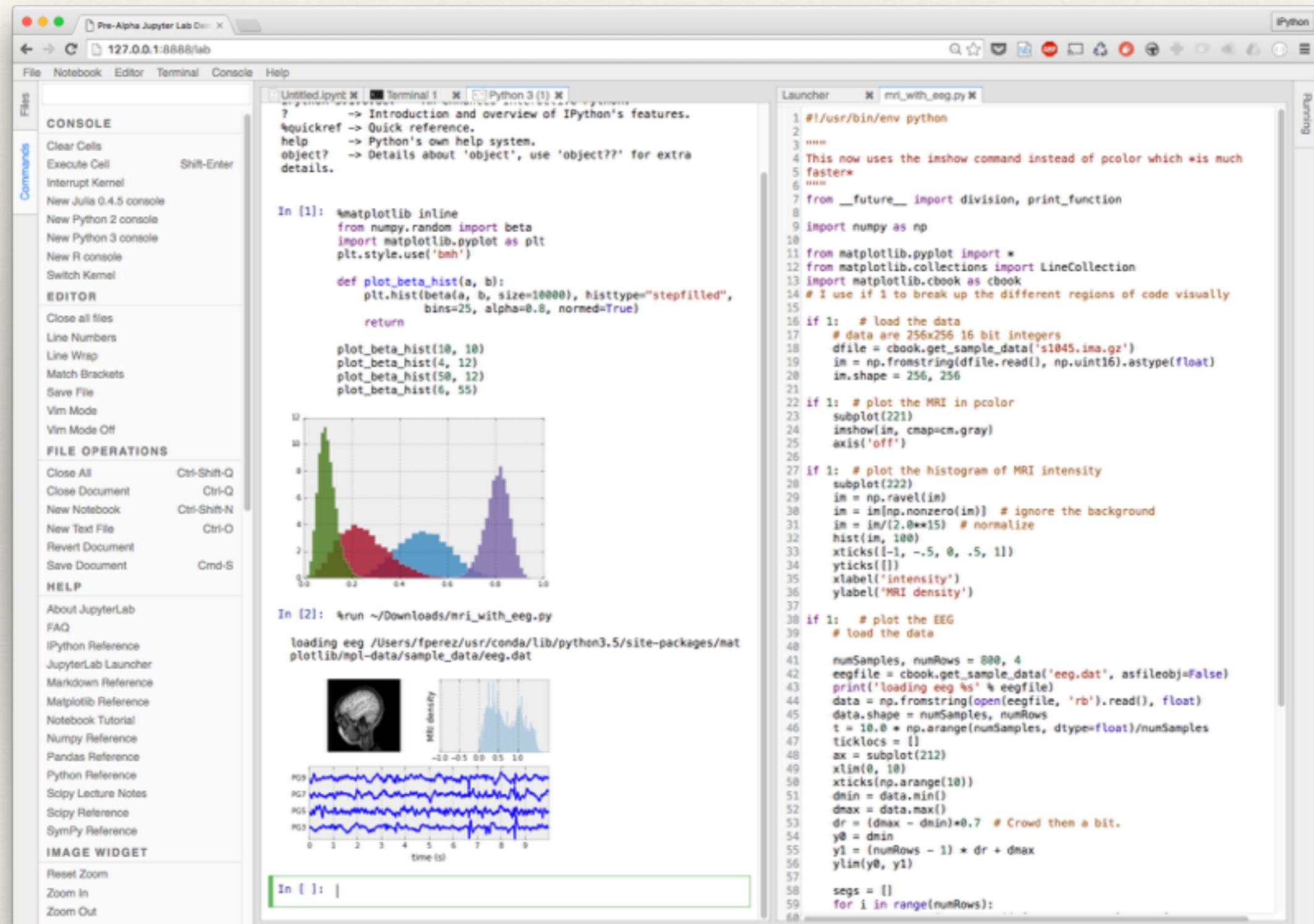
Kernels

Output

JupyterLab: unifying these ideas



A modular, plugin-based architecture



A collaboration with Bloomberg and Continuum



blog.jupyter.org/2016/07/14/jupyter-lab-alpha/

JupyterLab: the next generation of the Jupyter Notebook

14 JULY 2016

Learning the lessons of the Jupyter Notebook

It's been a long time in the making, but today we want to start engaging our community with an early (pre-alpha) release of the next generation of Jupyter Notebook application, which we are calling [JupyterLab](#).

https://www.techatbloomberg.com/blog/inside-the-collaboration-that-built-the-open-source-jupyterlab-project

Inside the Collaboration That Built the Open Source JupyterLab Project

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