Jupyter - An Interactive Platform for Scientific Computing

Shreyas Cholia

Group Leader, Usable Software Systems
Lawrence Berkeley National Laboratory

Why?

Why?

- **Ethical:** openness as fairness
- **Human/social:** openness fosters collaboration.
- **Epistemological:** proprietary science is an oxymoron.
- **Technical:** Python was cool :)

Designed for Learning

In reality, programming languages are how programmers express and communicate *ideas* — and the audience for those ideas is other programmers, not computers.

– Guido van Rossum

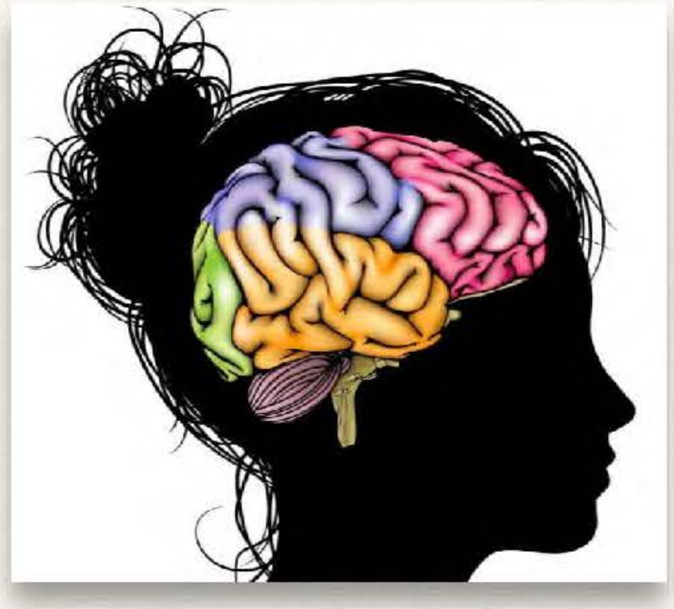
http://neopythonic.blogspot.com/2016/04/kings-day-speech.html

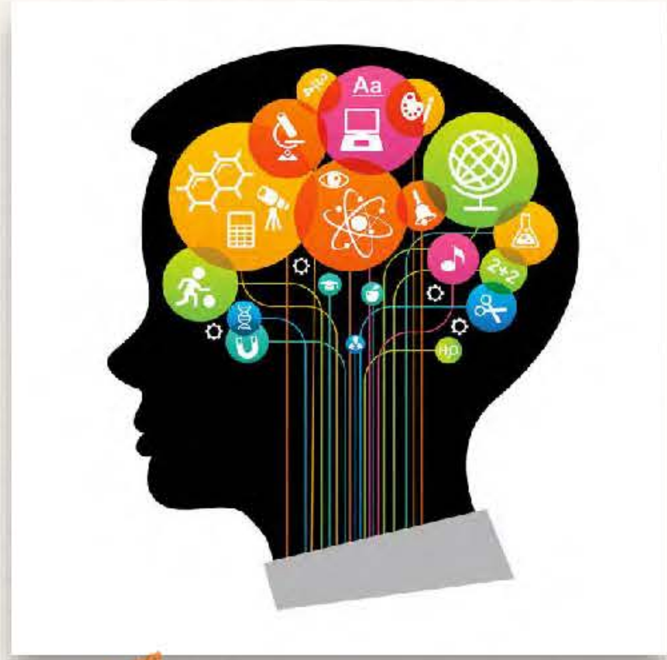
Slide credit: C. Willing

a *community* of people and an *ecosystem* of open tools and standards for interactive computing create things that are **language-agnostic** and **modular**.

Empower people to use **other open tools**.

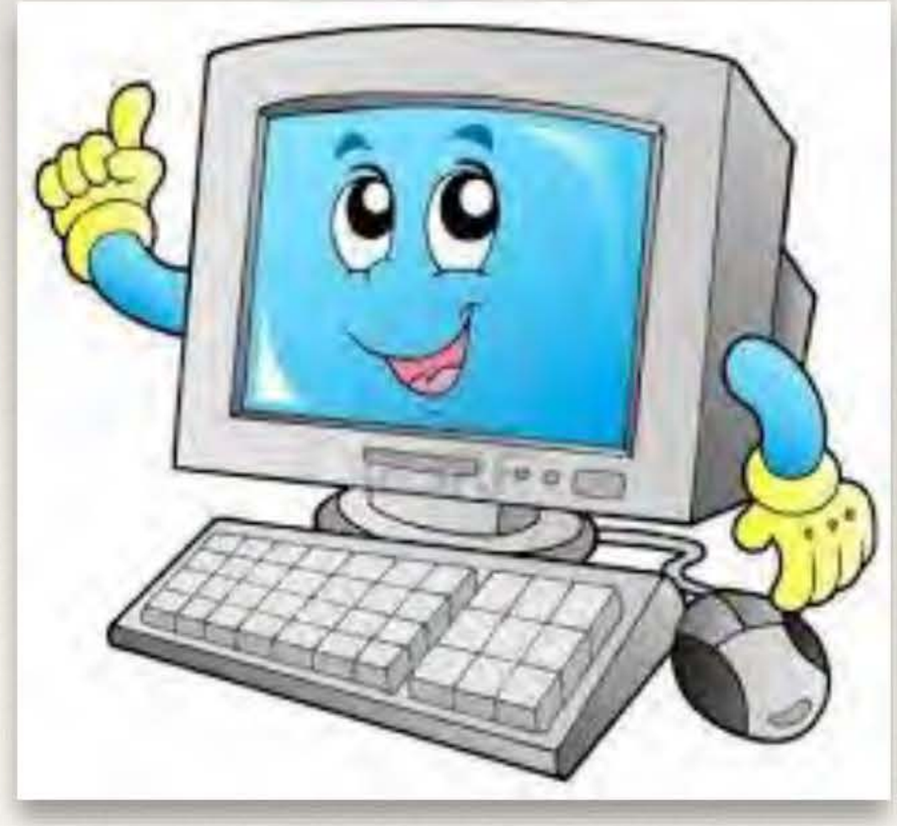
What?

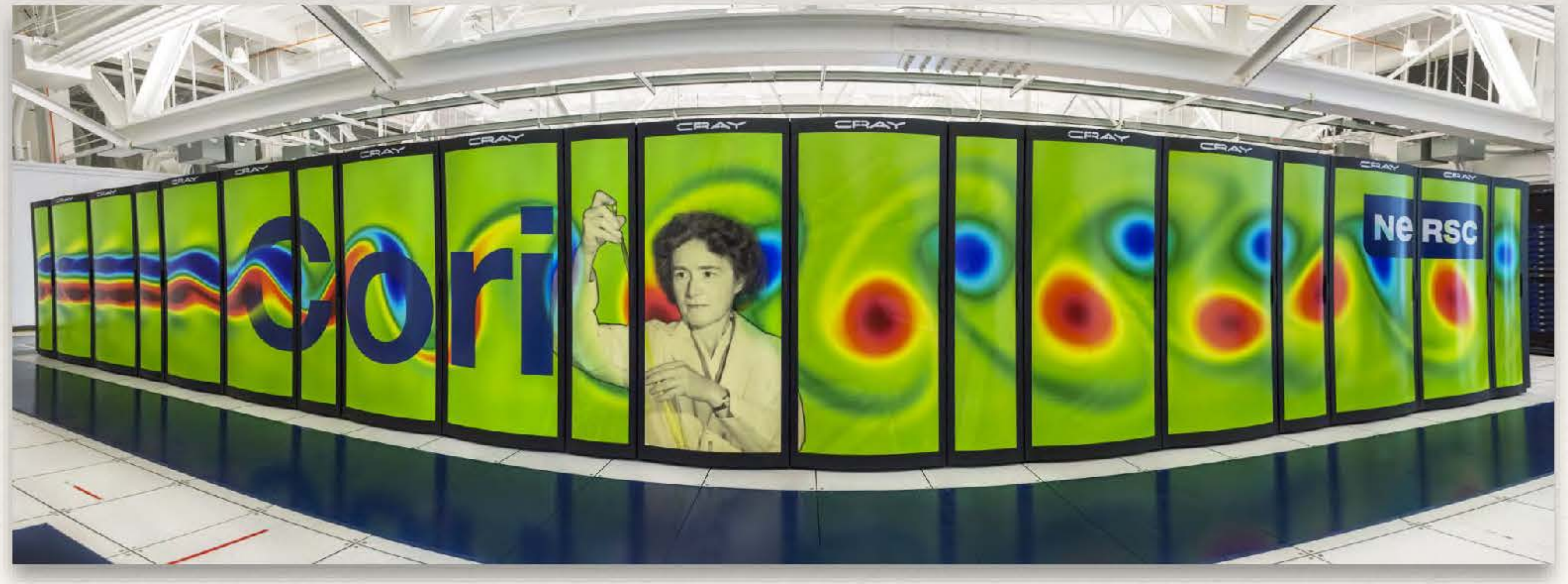




IP[y]:
IPython

jupyter



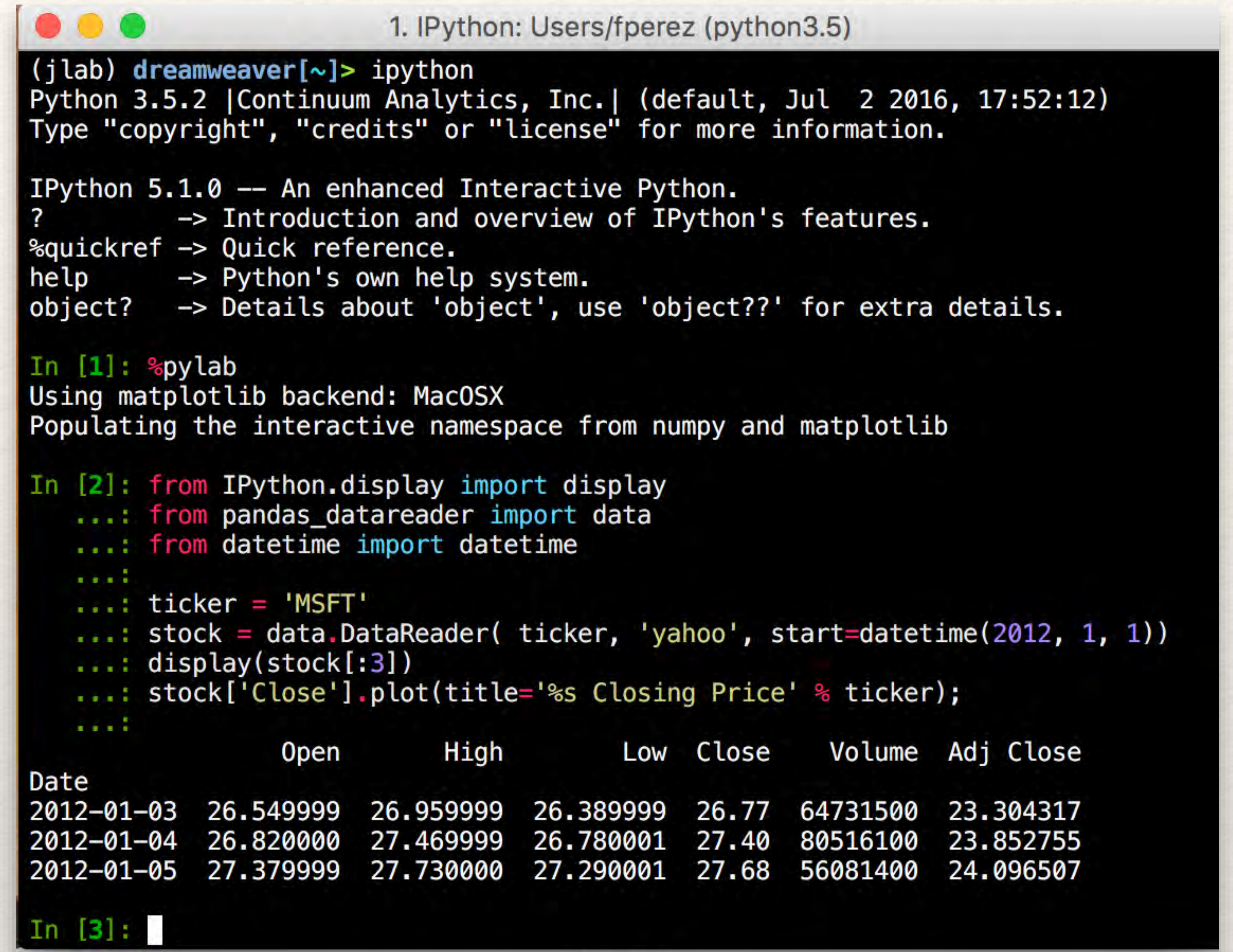


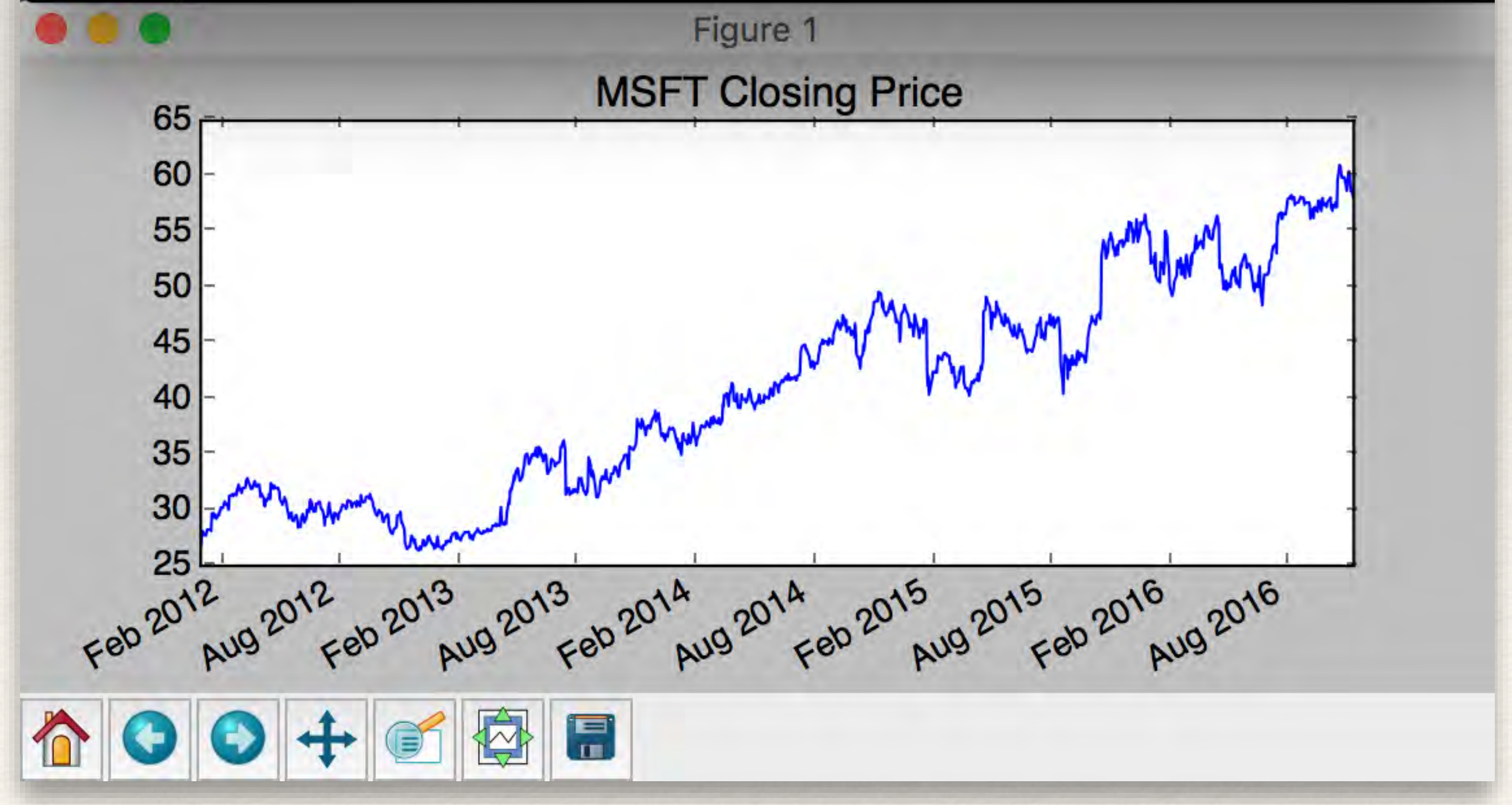
IPython: Interactive Python, 2001

A humble start:

IPython 0.0.1, 259 LOC

“Just an afternoon hack”





https://gist.github.com/fperez/1579699

Team today: where *all the credit* goes

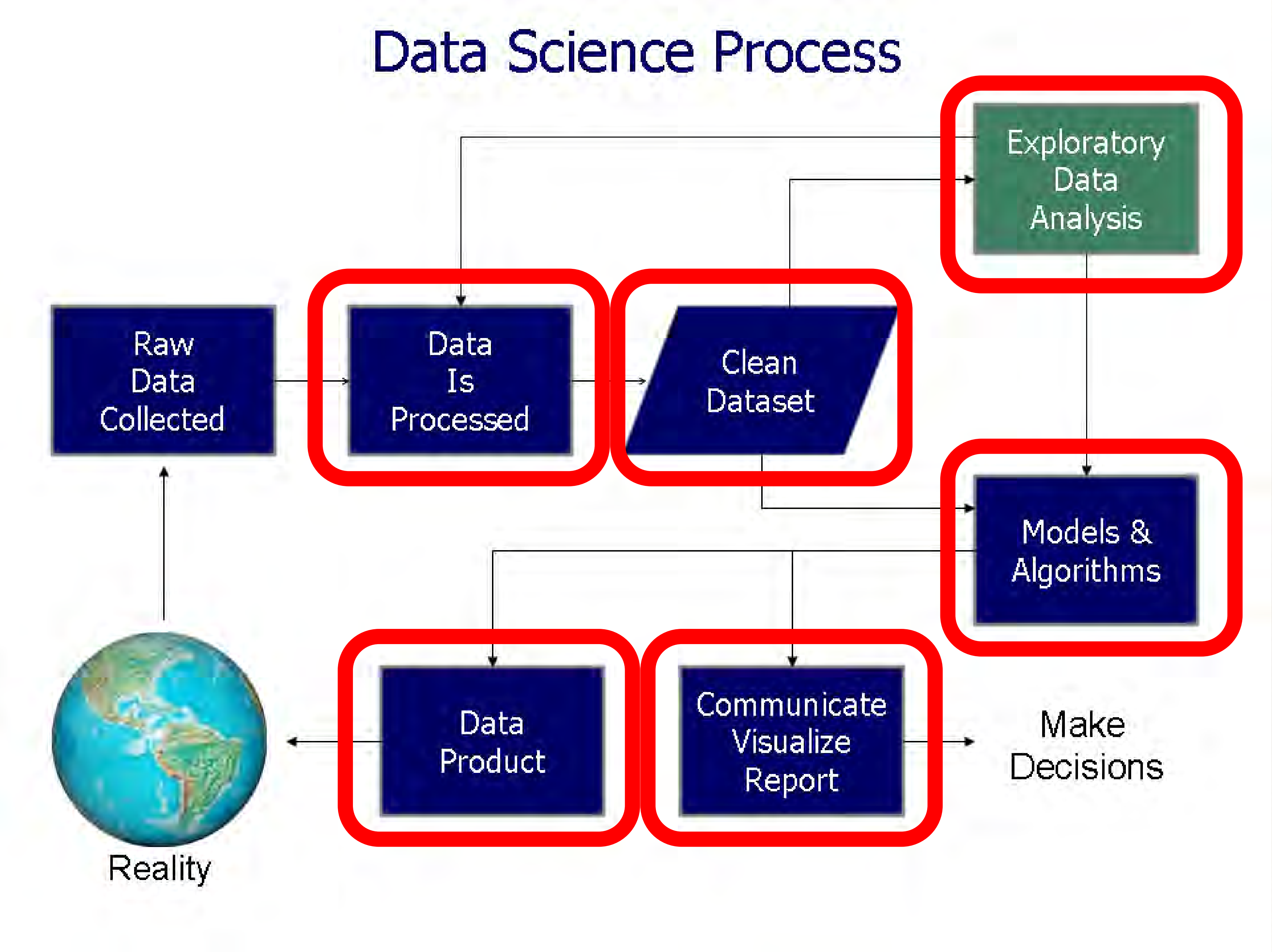
Plus ~ 1500 more Open source contributors!

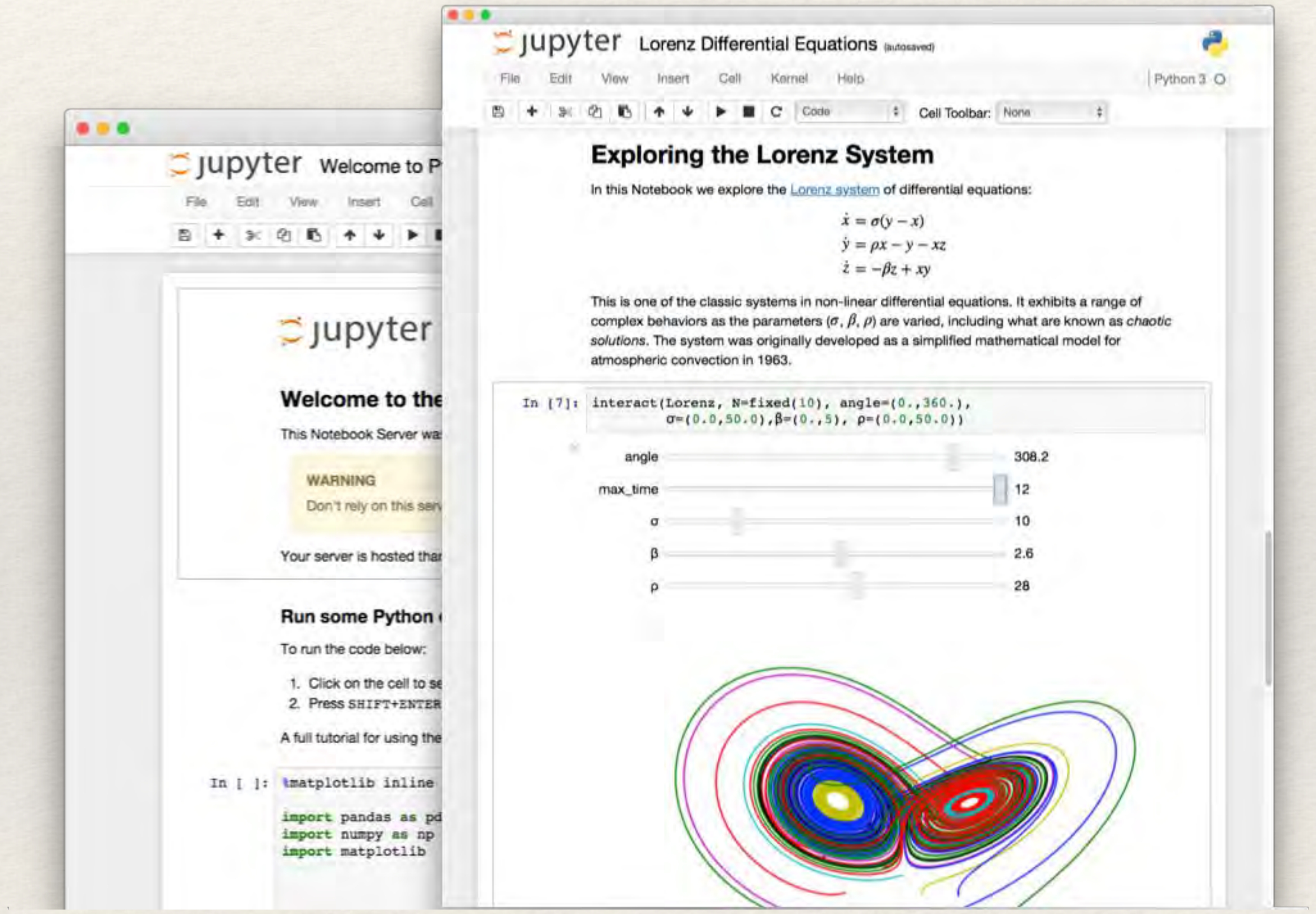
What is Jupyter?

Tool for reproducible, shareable narratives, literate computing:

***Notebook*: Document containing code, comments, outputs.**

Rich text, interactive plots, equations, widgets, etc.

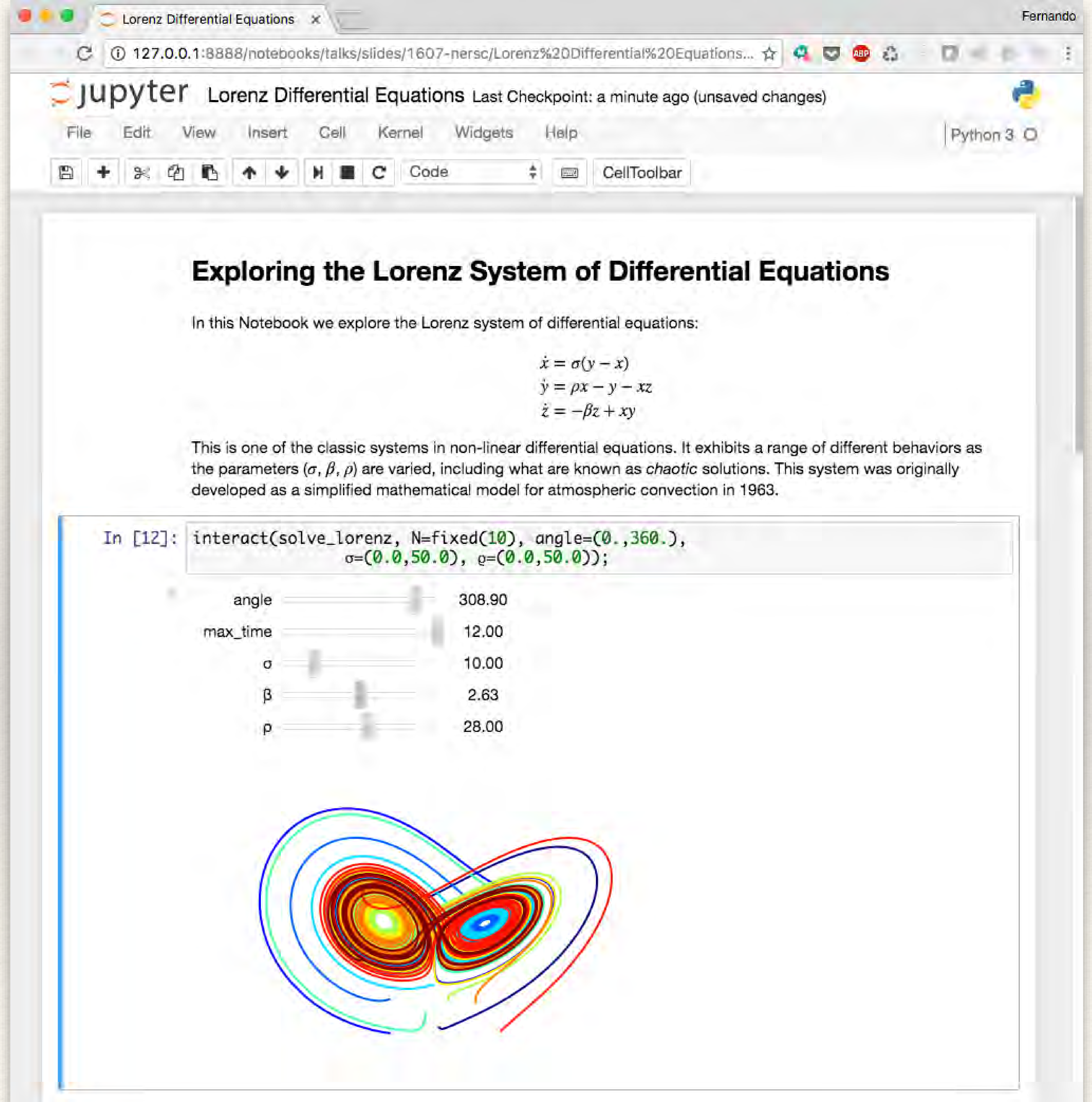




Goal: Enable exploratory data analytics, deep learning, workflows, and more through Jupyter on NERSC systems.

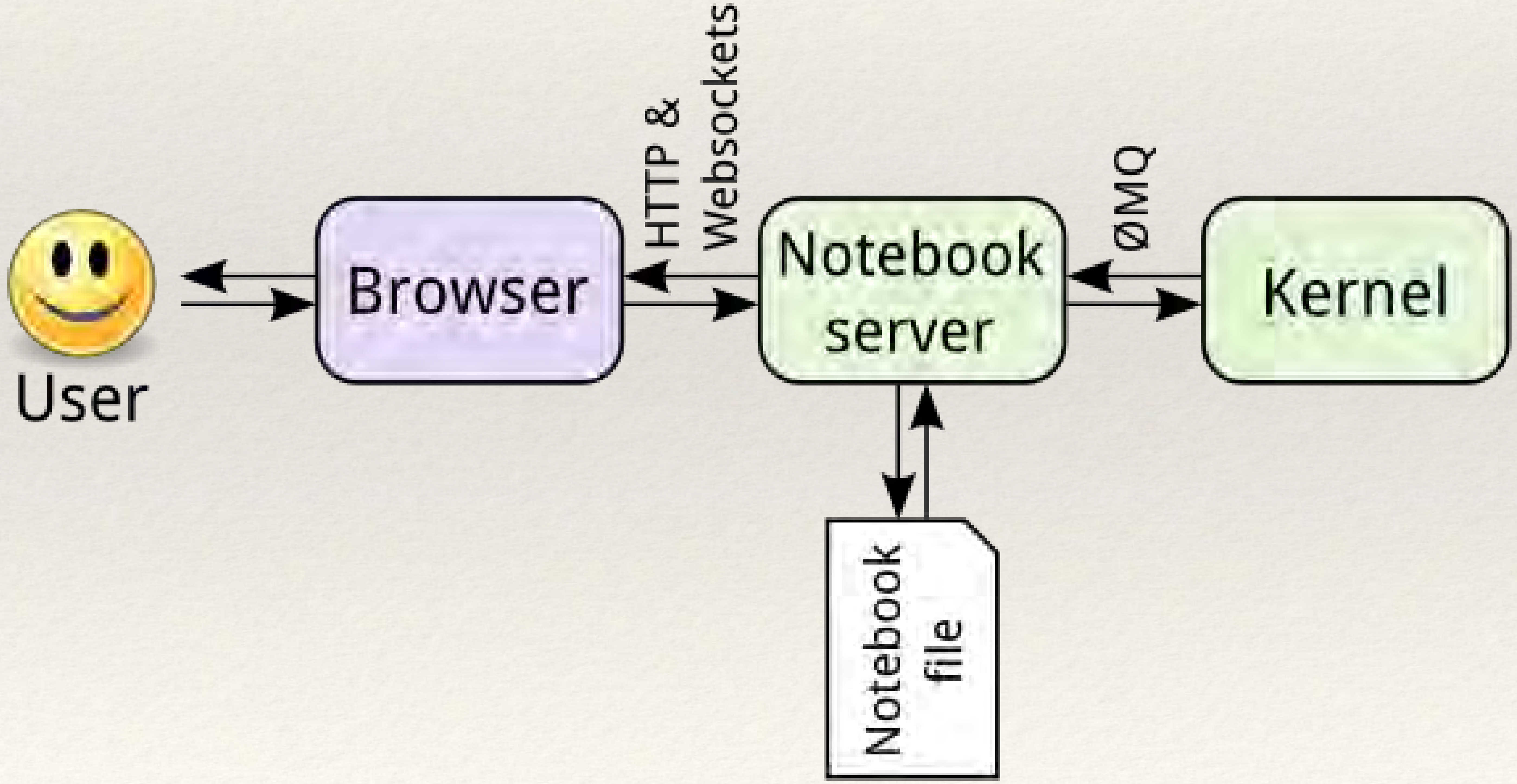
The IPython/Jupyter Notebook

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

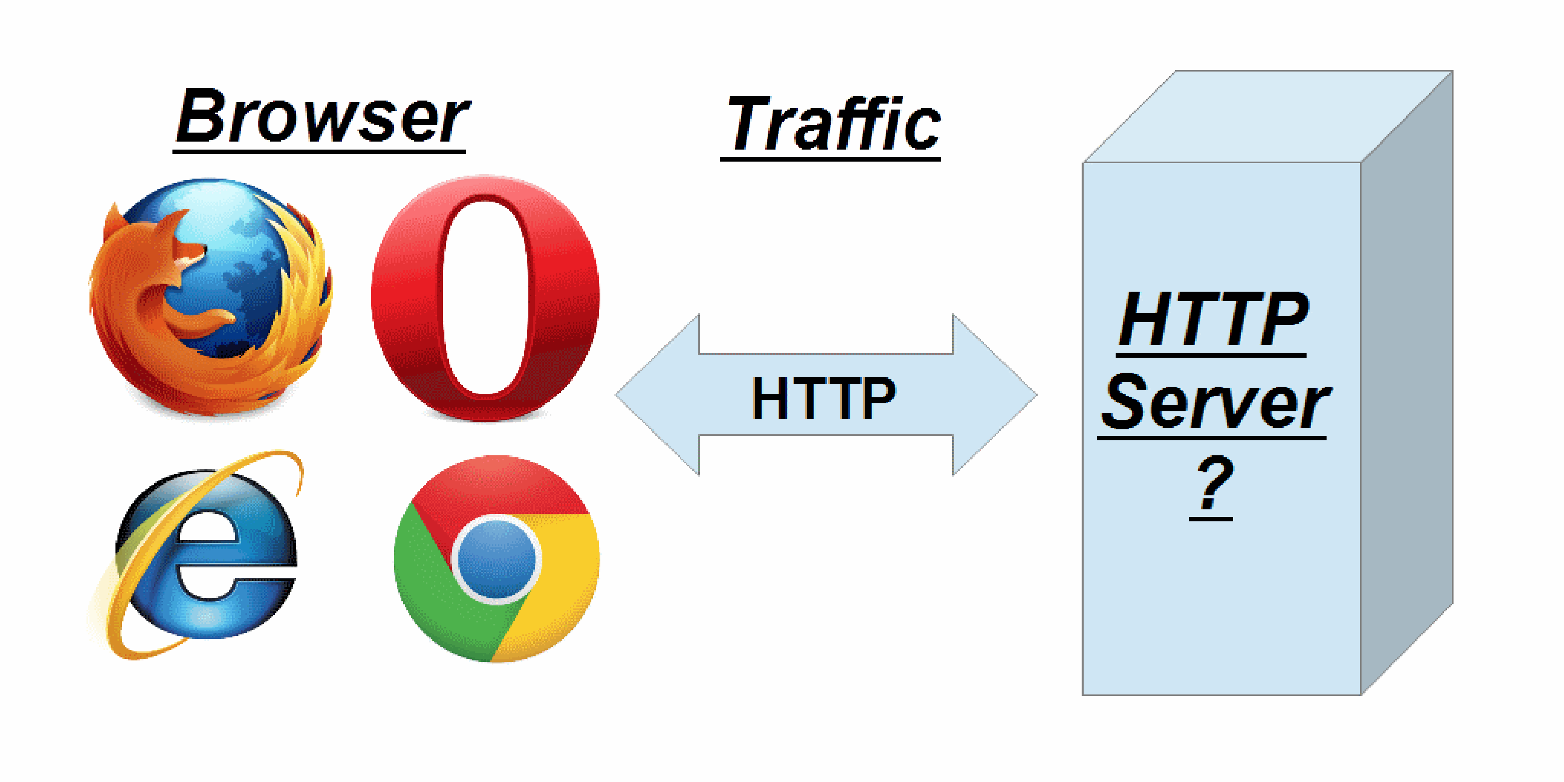


How It Works

The Notebook Server sends code (via 0mq) to a language “kernel” that executes this code. In addition to running your code, it stores code and output, together with markdown notes, in an editable document called a notebook, saved as a JSON file with a .ipynb extension.



Core ideas of the web: HTTP & HTML



HTTP: protocol to connect clients and servers

HyperText Transport Protocol

Image credit: eviltester.com

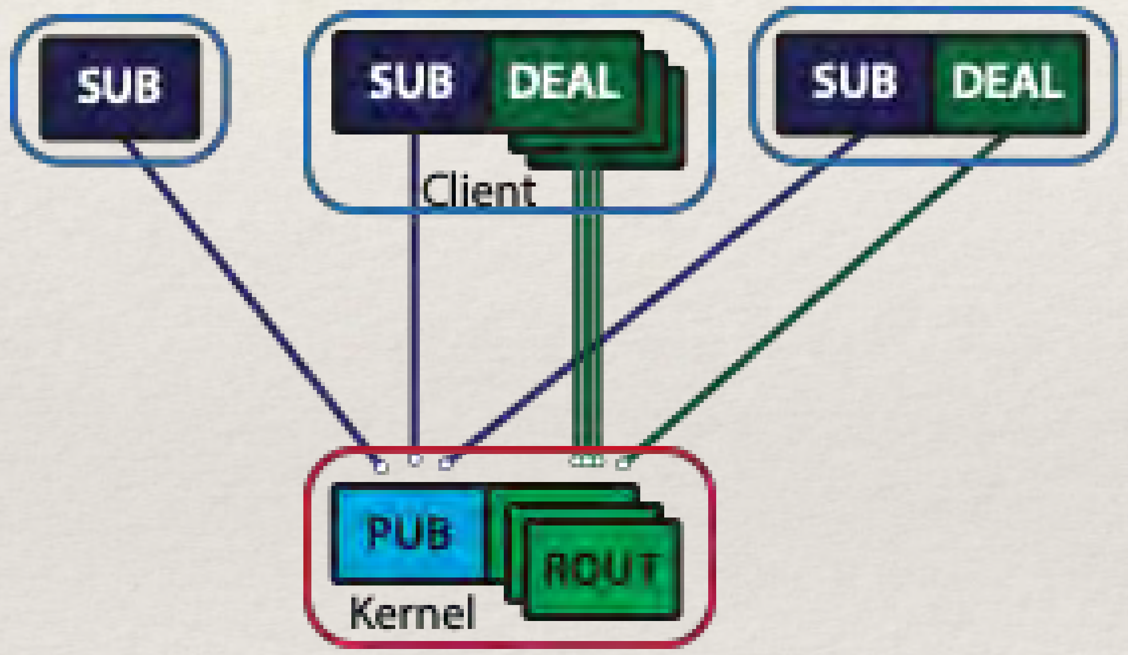
```
<div id="page-wrapper">
  <div id="main" class="clearfix">
    <div id="content" class="column" role="main"><div id="content-inner" class="clearfix">
                  <a id="main-content"></a>
                    <h1 class="title" id="page-title">Statistics at UC Berkeley</h1>
```

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

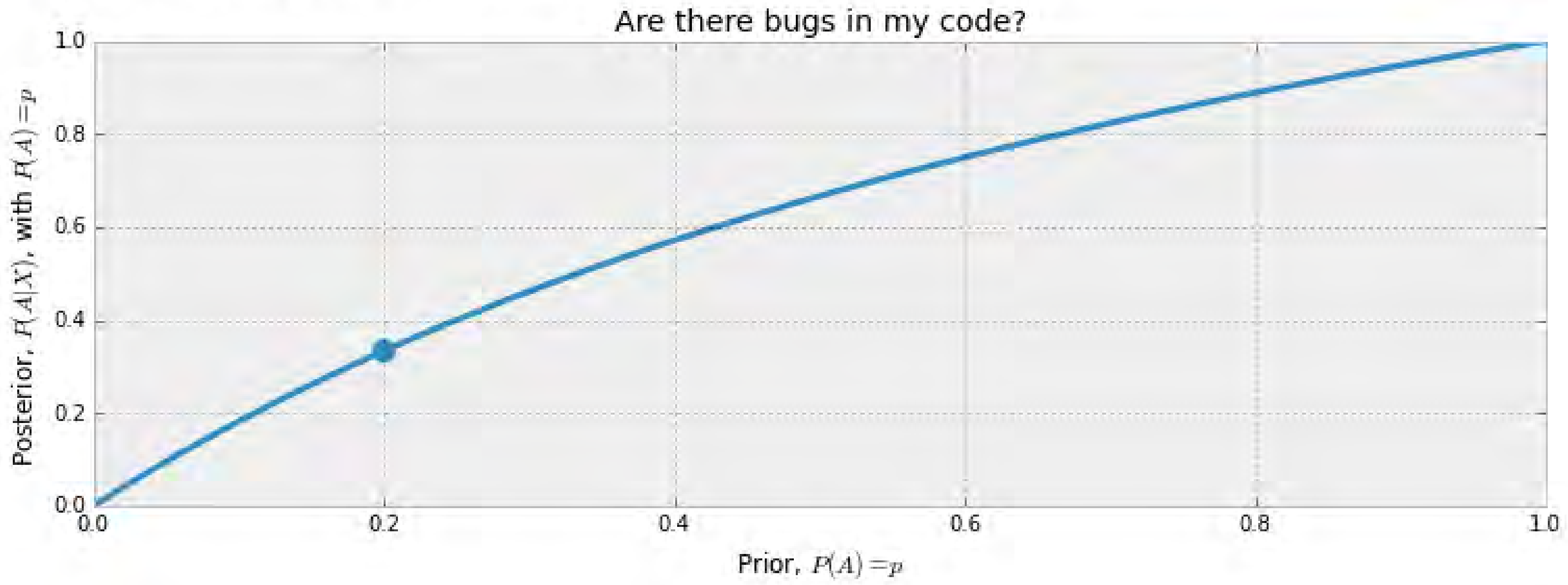
$$P(A|X) = \frac{1 \cdot p}{1 \cdot p + 0.5(1-p)}$$

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

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>
```



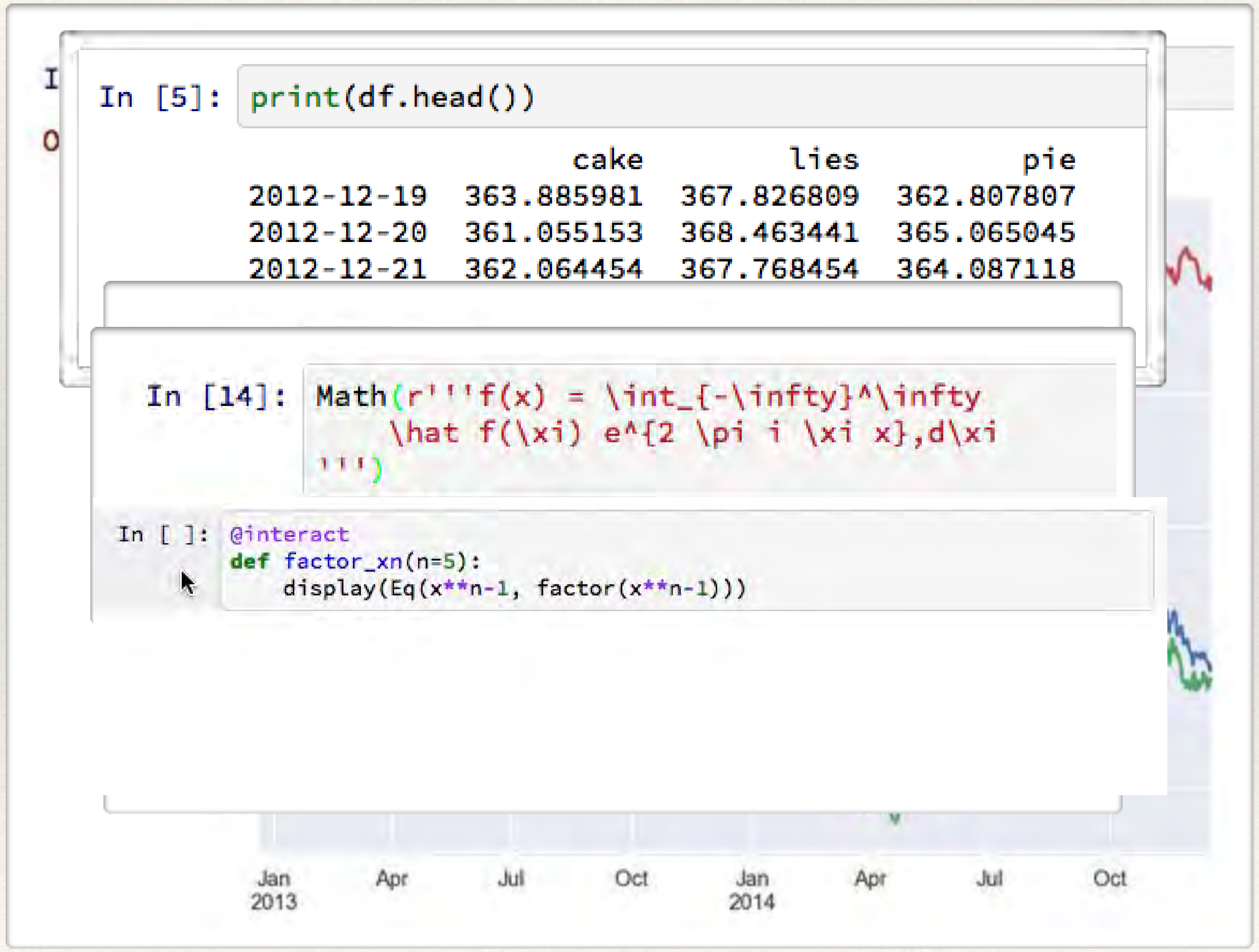
https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers

Jupyter Protocol

web-age capture of the process of interactive computing

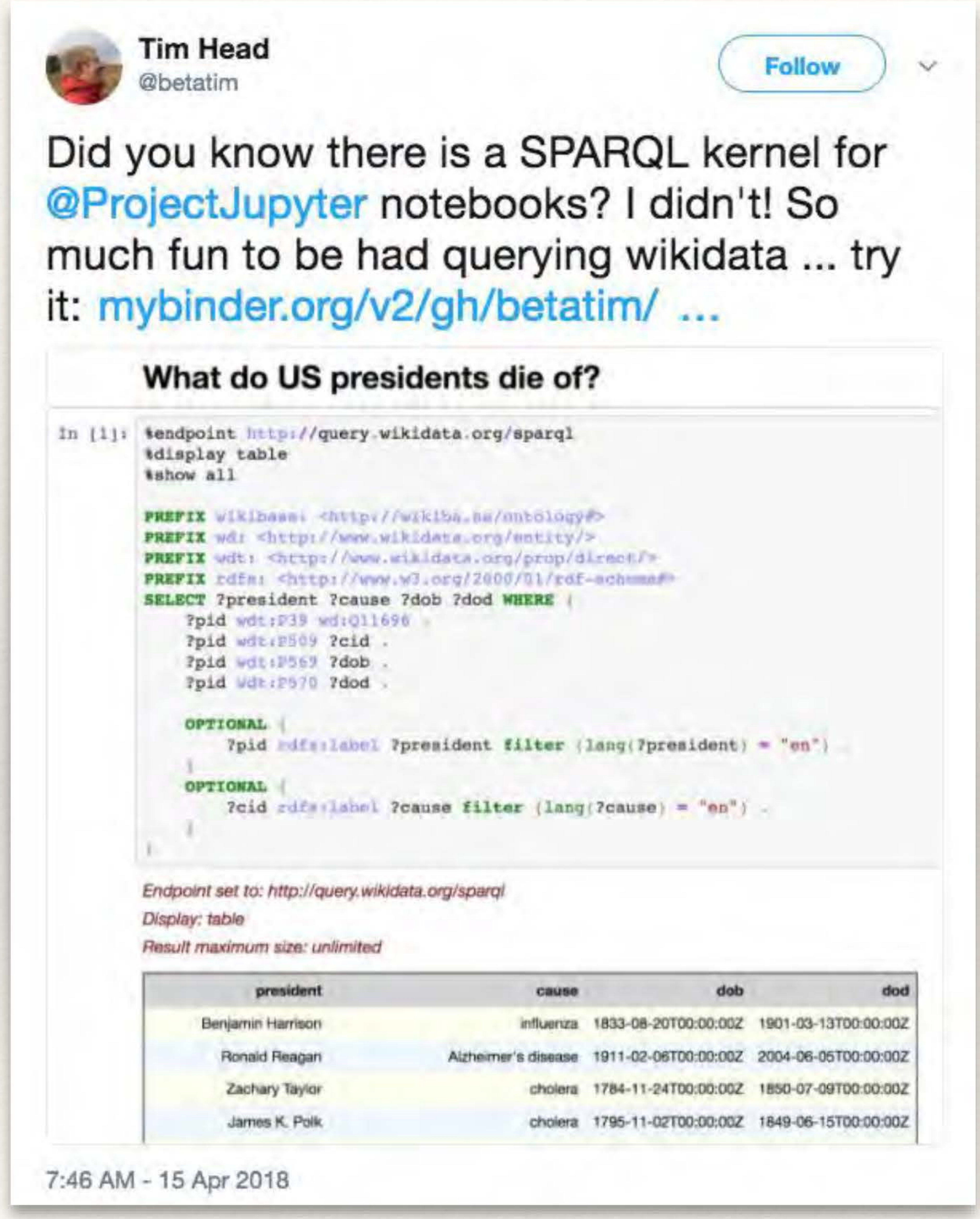
any mime-type output

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



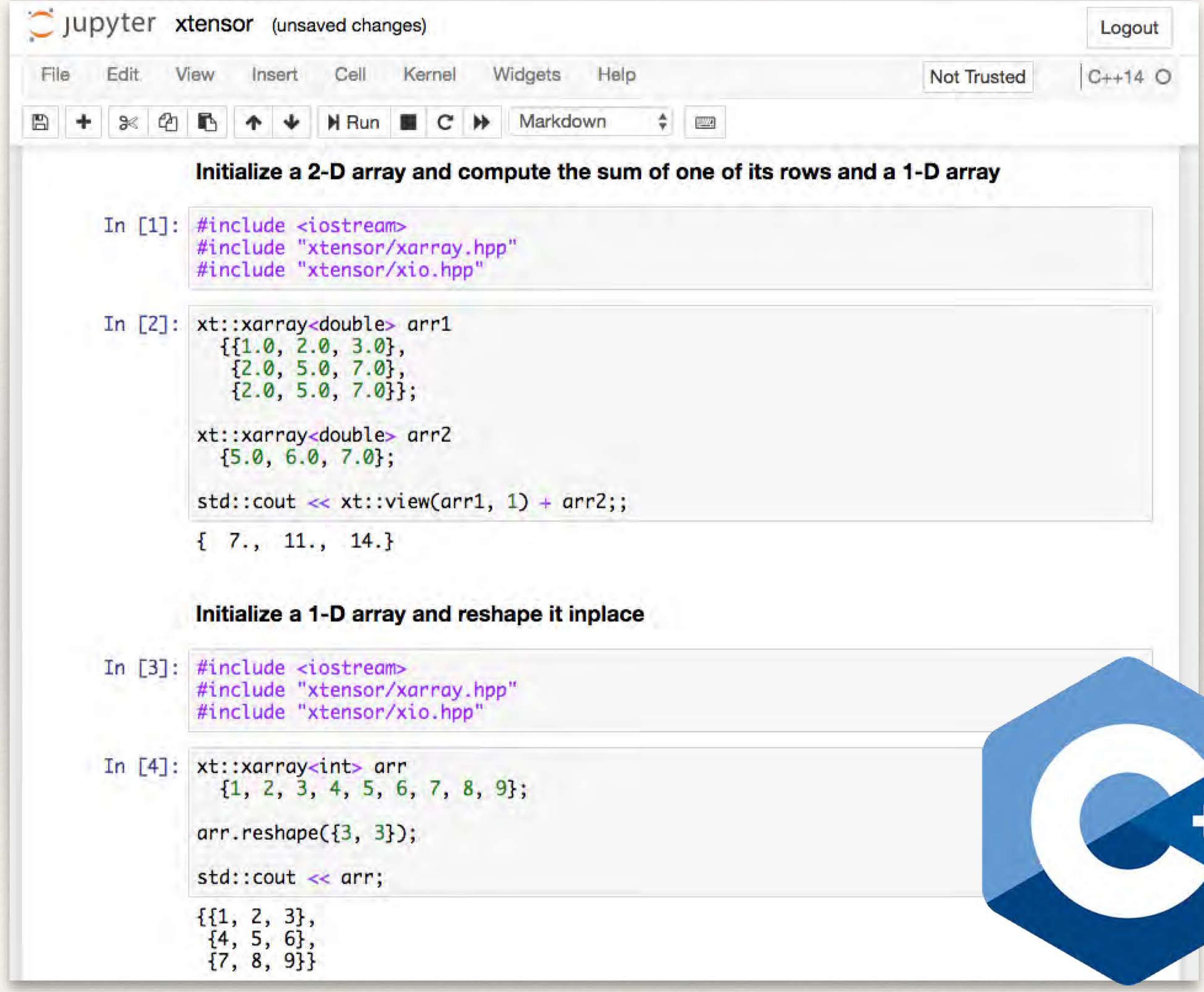
Jupyter Protocol
is language agnostic

~100 different kernels: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

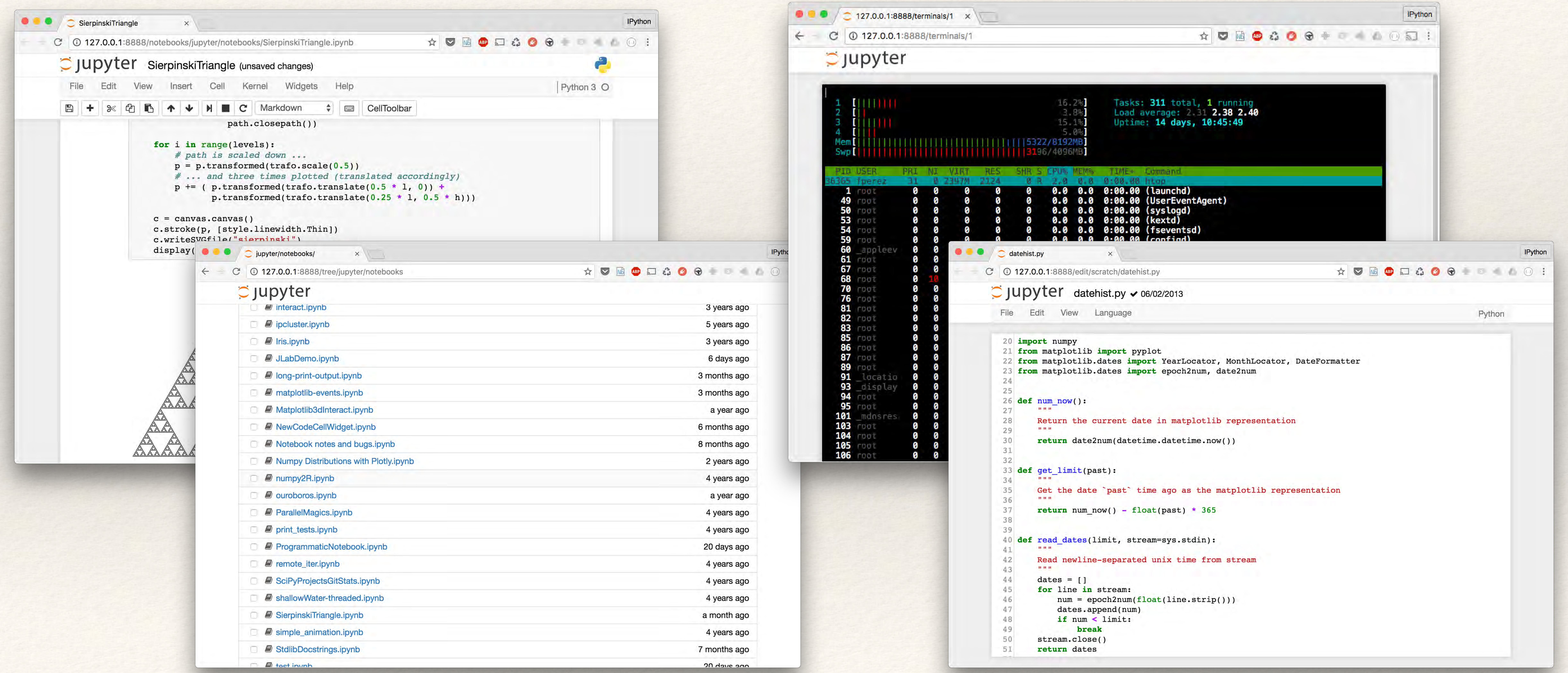


xtensor

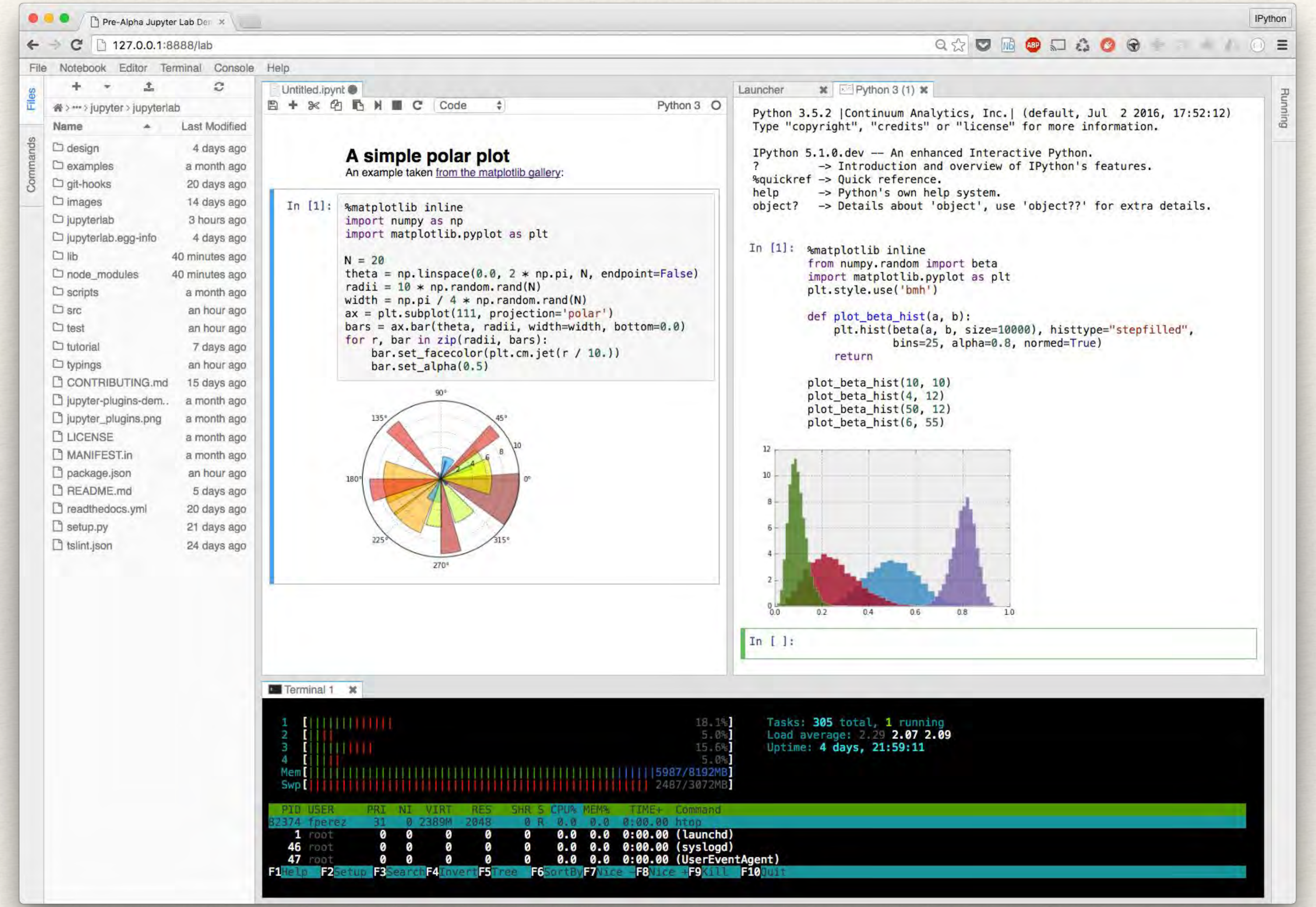
Cling

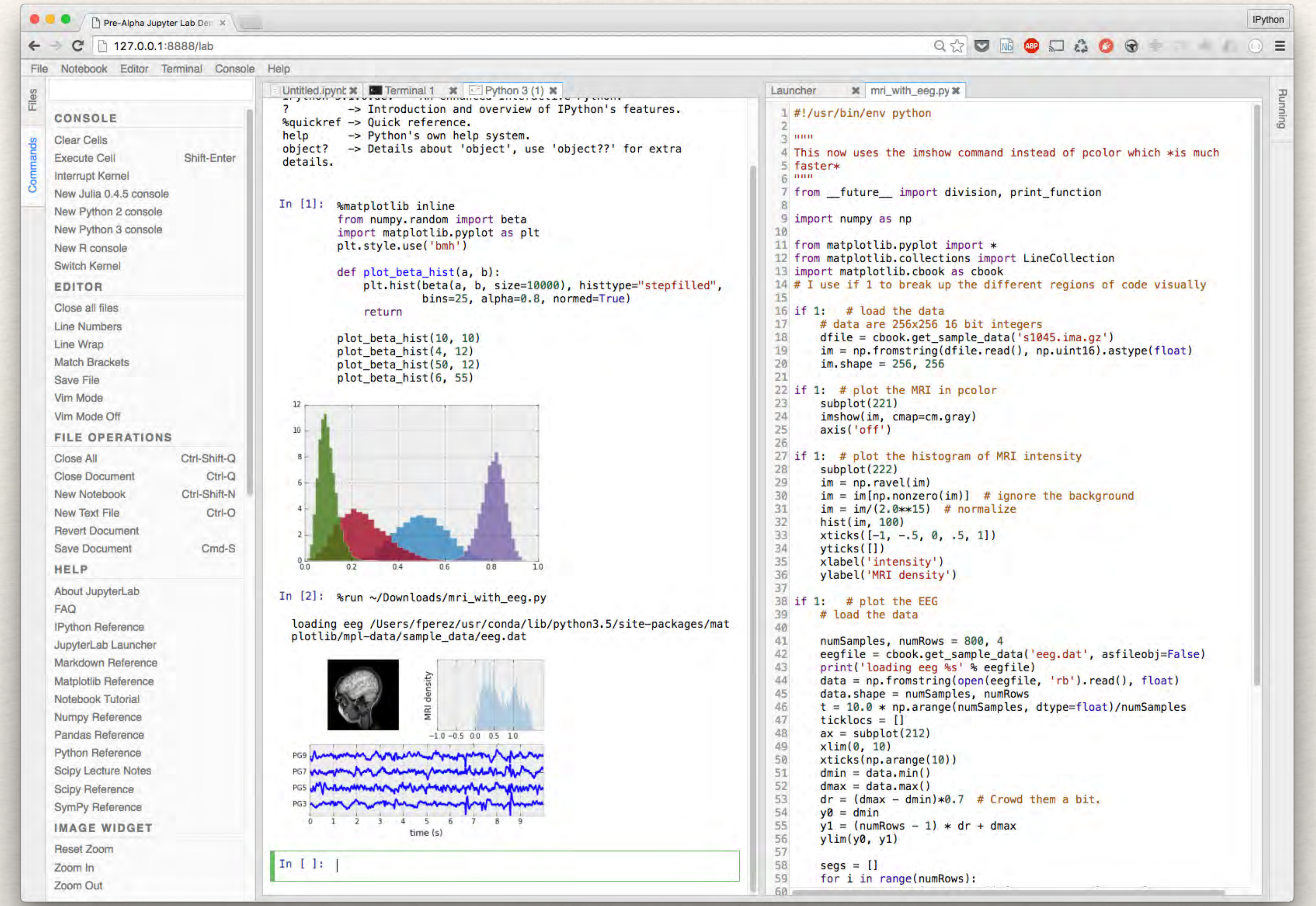


Classic 'Notebook'...



JupyterLab: a grand unified theory of Jupyter





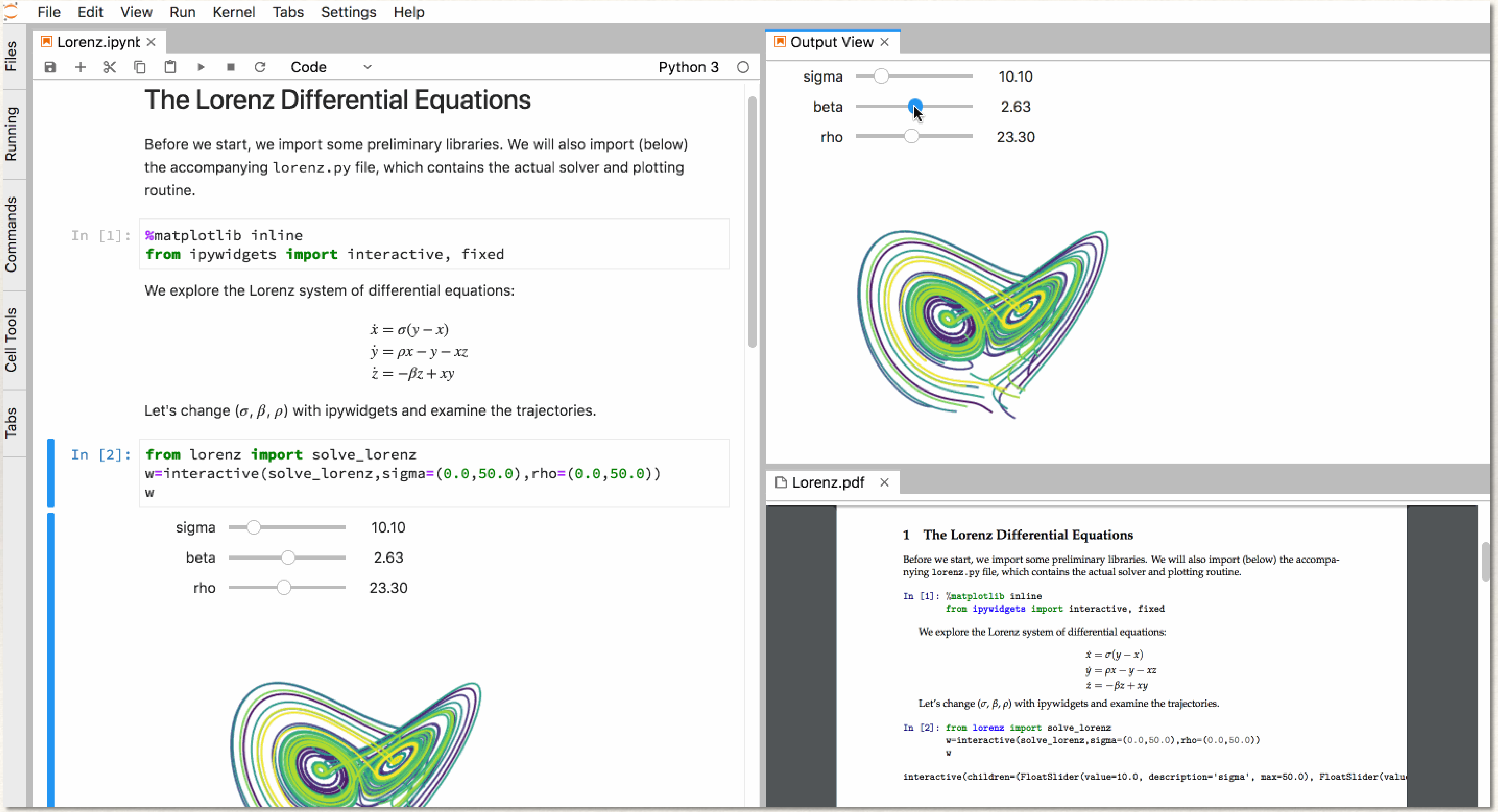
Huge Team Effort!

C. Colbert, S. Corlay, A. Darian, B. Granger, J. Grout, P. Ivanov, I. Rose, S. Silvester, C. Willing, J. Zosa-Forde …

ANACONDA

Bloomberg

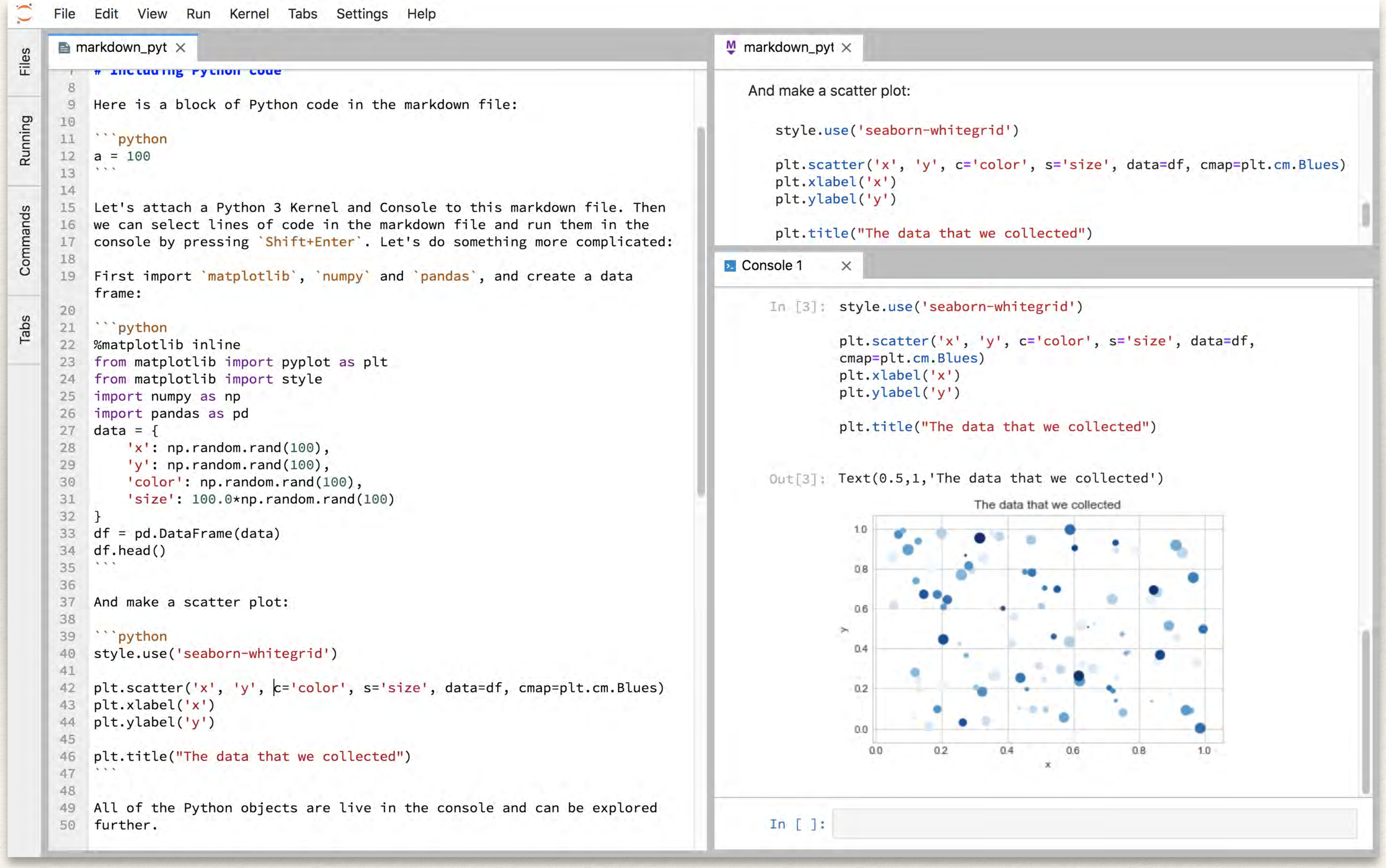
Notebooks++



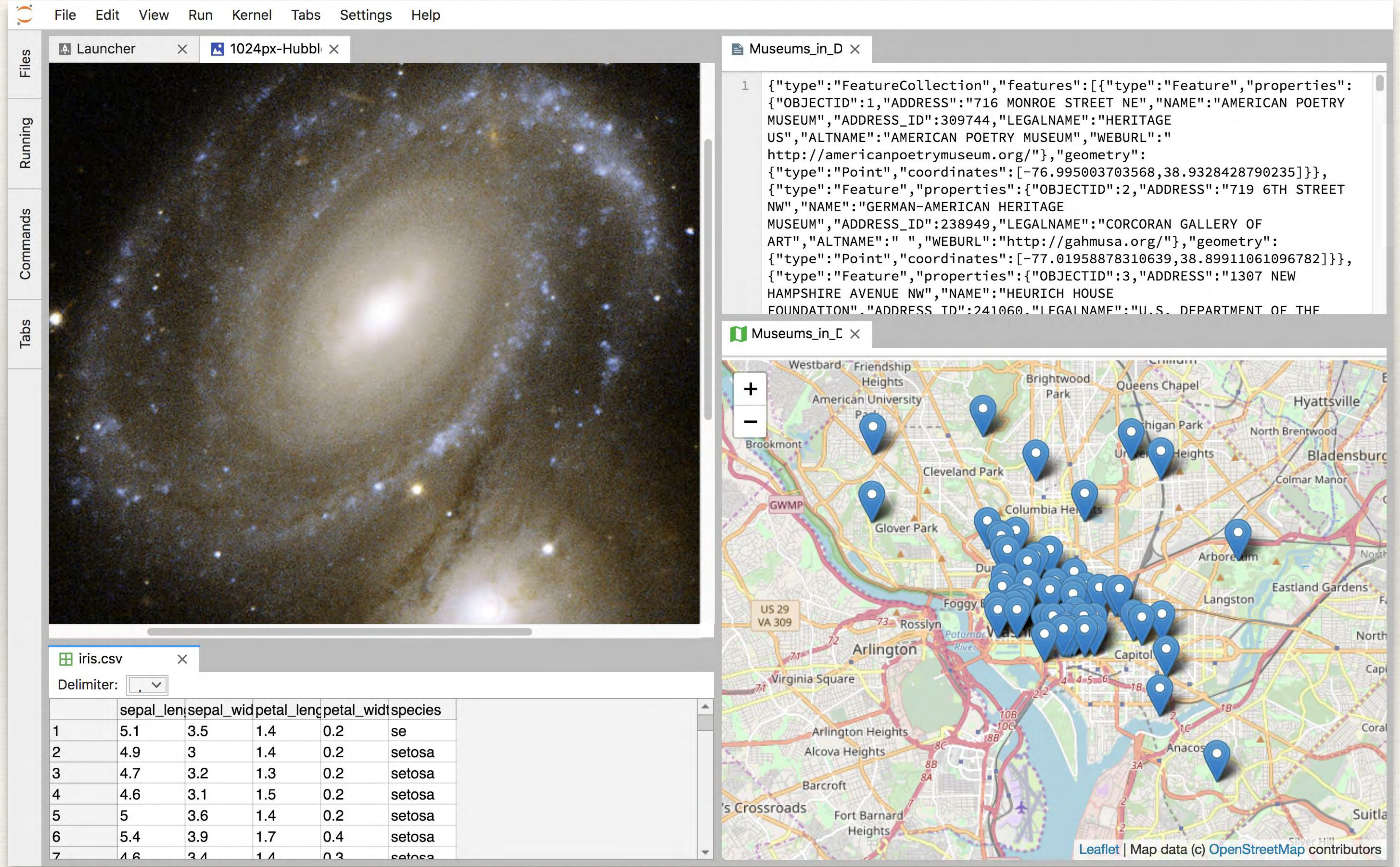
File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
Lorenz.ipynb
Code
Python 3
The Lorenz Differential Equations
Before we start, we import some preliminary libraries. We will also import (below) the accompanying lorenz.py file, which contains the actual solver and plotting routine.
In [1]:
%matplotlib inline
from ipywidgets import interactive, fixed
We explore the Lorenz system of differential equations:
$\dot{x} = \sigma(y - x)$
$\dot{y} = \rho x - y - xz$
$\dot{z} = -\beta z + xy$
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.
In [2]:
from lorenz import solve_lorenz
w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
sigma 10.10
beta 2.63
rho 23.30
Output View
sigma 10.10
beta 2.63
rho 23.30
Lorenz.pdf
1 The Lorenz Differential Equations
Before we start, we import some preliminary libraries. We will also import (below) the accompanying lorenz.py file, which contains the actual solver and plotting routine.
In [1]: %matplotlib inline
from ipywidgets import interactive, fixed
We explore the Lorenz system of differential equations:
$\dot{x} = \sigma(y - x)$
$\dot{y} = \rho x - y - xz$
$\dot{z} = -\beta z + xy$
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.
In [2]: from lorenz import solve_lorenz
w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
interactive(children=(FloatSlider(value=10.0, description='sigma', max=50.0), FloatSlider(valu

Live Demo!

Beyond notebooks

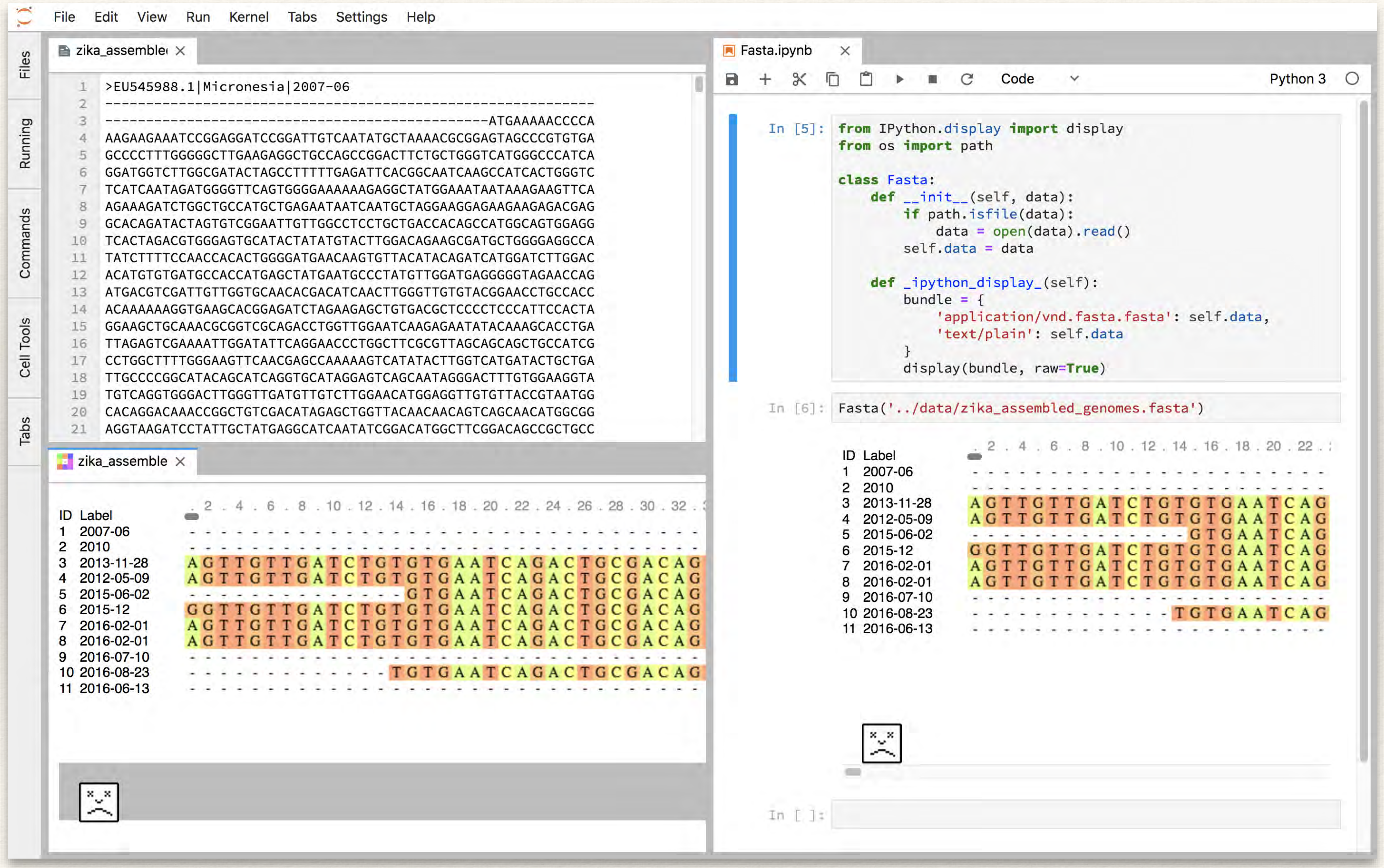


Data

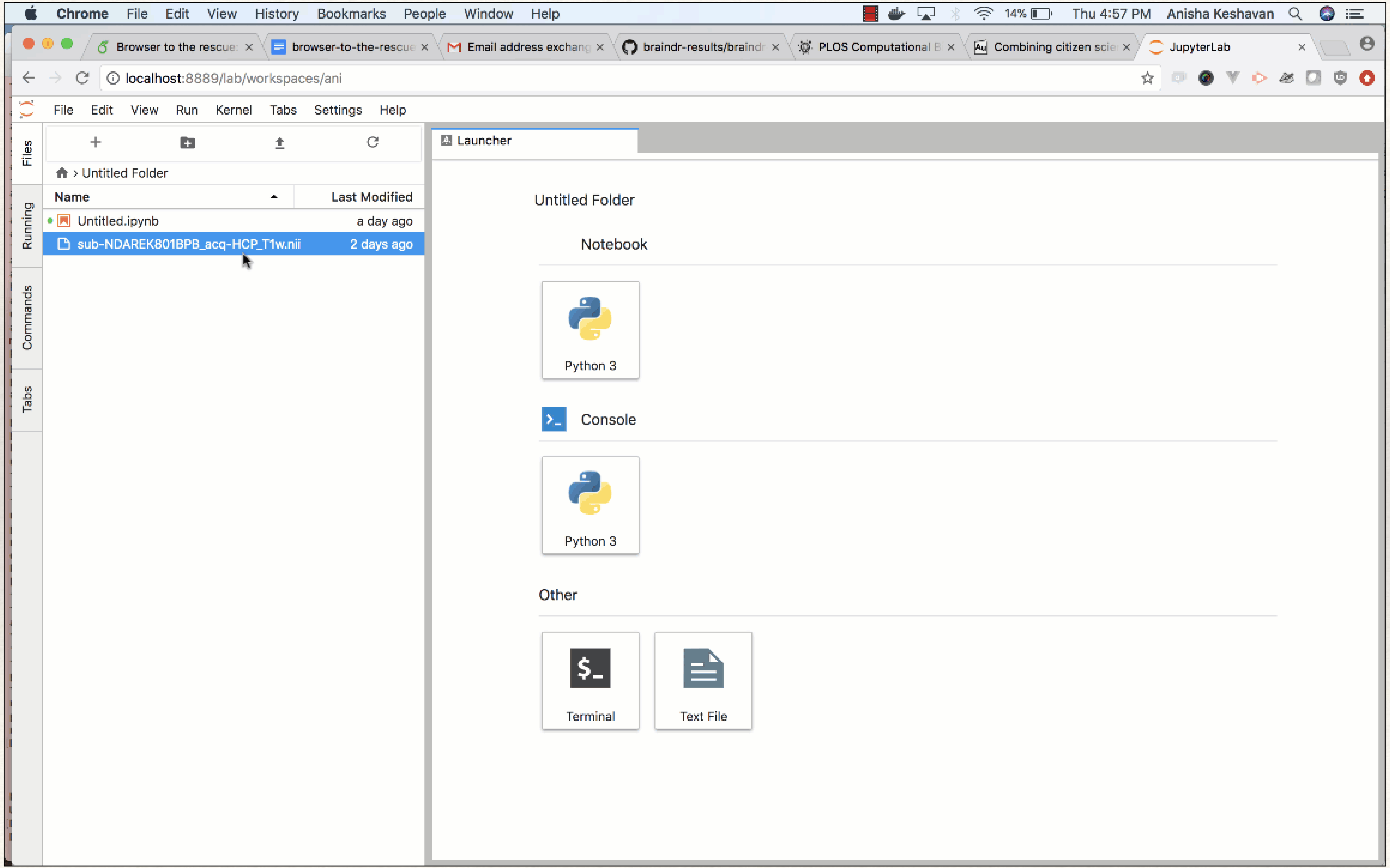


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Tabs
Launcher
1024px-Hubbl
Museums_in_D
{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"OBJECTID":1,"ADDRESS":"716 MONROE STREET NE","NAME":"AMERICAN POETRY MUSEUM","ADDRESS_ID":309744,"LEGALNAME":"HERITAGE US","ALTNAME":"AMERICAN POETRY MUSEUM","WEBURL":"http://americanpoetrymuseum.org/"},"geometry":{"type":"Point","coordinates":[-76.995003703568,38.9328428790235]}},{"type":"Feature","properties":{"OBJECTID":2,"ADDRESS":"719 6TH STREET NW","NAME":"GERMAN-AMERICAN HERITAGE MUSEUM","ADDRESS_ID":238949,"LEGALNAME":"CORCORAN GALLERY OF ART","ALTNAME":" ","WEBURL":"http://gahmusa.org/"},"geometry":{"type":"Point","coordinates":[-77.01958878310639,38.89911061096782]}},{"type":"Feature","properties":{"OBJECTID":3,"ADDRESS":"1307 NEW HAMPSHIRE AVENUE NW","NAME":"HEURICH HOUSE
Museums_in_D
iris.csv
Delimiter:
sepal_len sepal_wid petal_leng petal_widt species
1 5.1 3.5 1.4 0.2 se
2 4.9 3 1.4 0.2 setosa
3 4.7 3.2 1.3 0.2 setosa
4 4.6 3.1 1.5 0.2 setosa
5 5 3.6 1.4 0.2 setosa
6 5.4 3.9 1.7 0.4 setosa
Westbard
Friendship Heights
American University
Brookmont
Brightwood Park
Queens Chapel
Hyattsville
North Brentwood
Bladensburg
Colmar Manor
Cleveland Park
Columbia Heights
Glover Park
GWMP
Arboretum
Eastland Gardens
Langston
Foggy
Rosslyn
US 29
VA 309
Arlington
Capitol
Virginia Square
Arlington Heights
Alcova Heights
Barcroft
Fort Barnard Heights
Anacos
Leaflet | Map data (c) OpenStreetMap contributors

Data: reusable within code



Community-extensible: Neurohackademy 2018



JupyterHub: multiuser support

jupyterhub

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

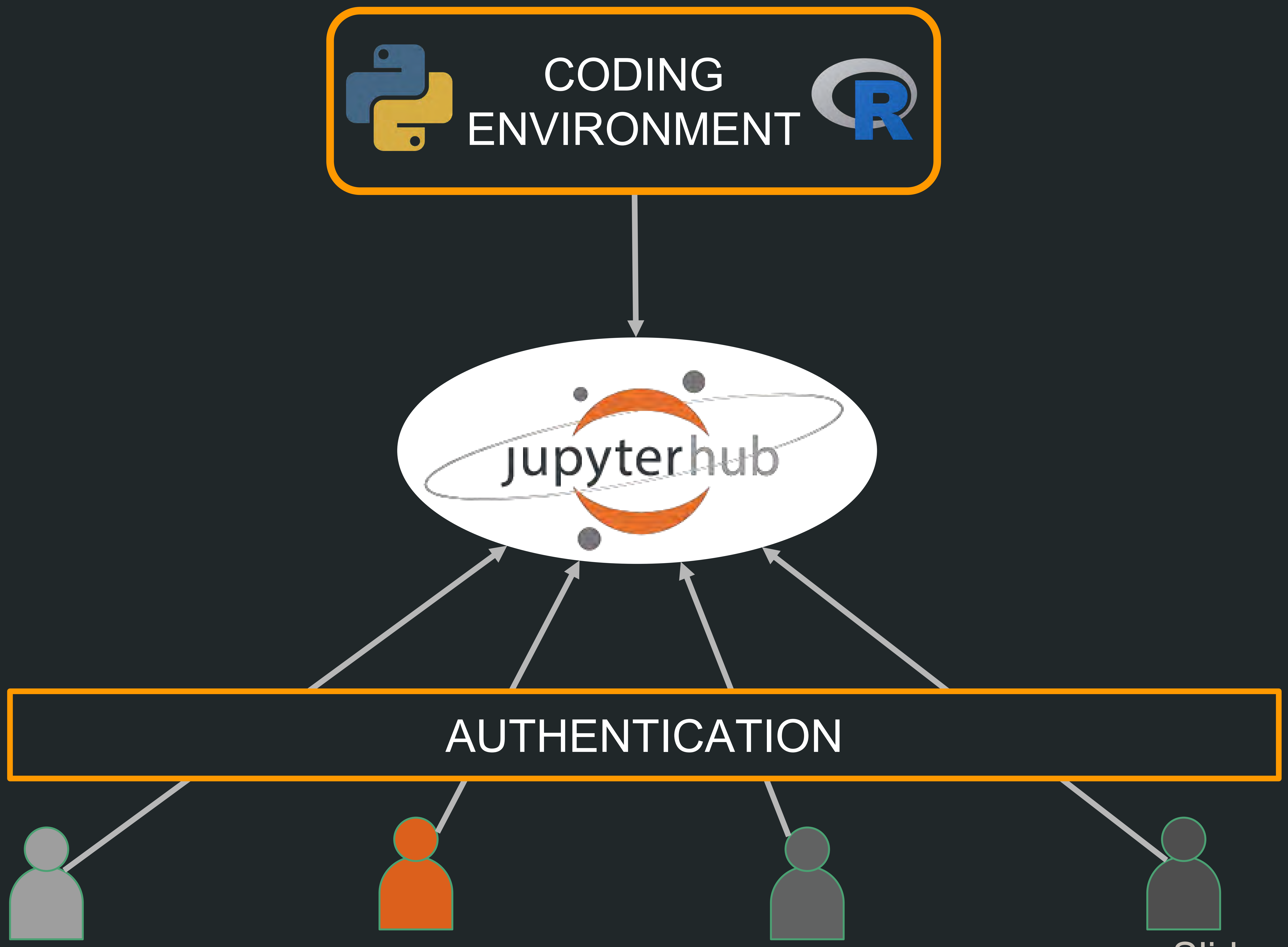
Deploy the Jupyter Notebook to all users in your organization on centralized servers on- or off-site.

Container friendly

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

Code meets data

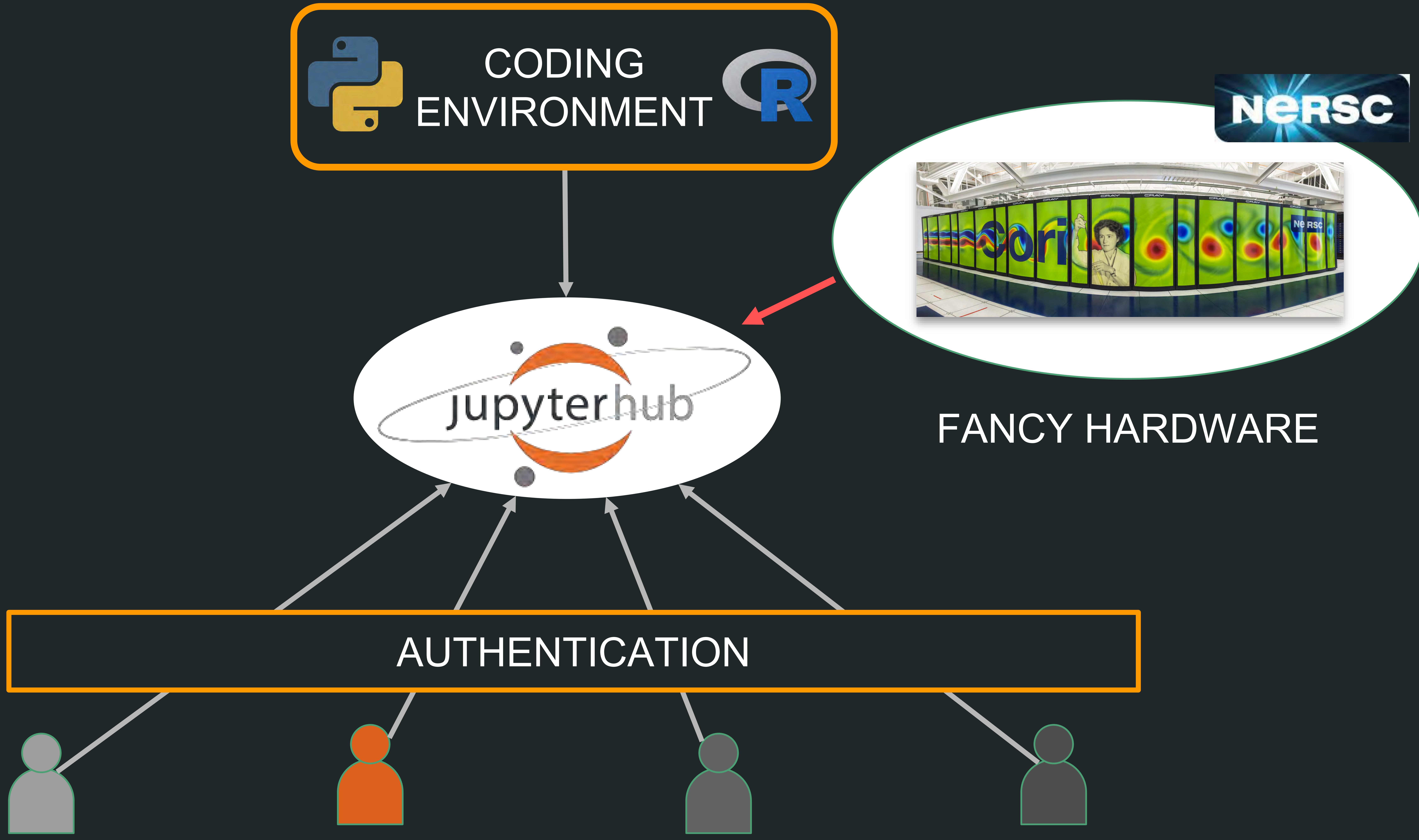
Deploy the Notebook next to your data to provide unified software management and data access within your organization.



Slides credit: C. Holdgraf

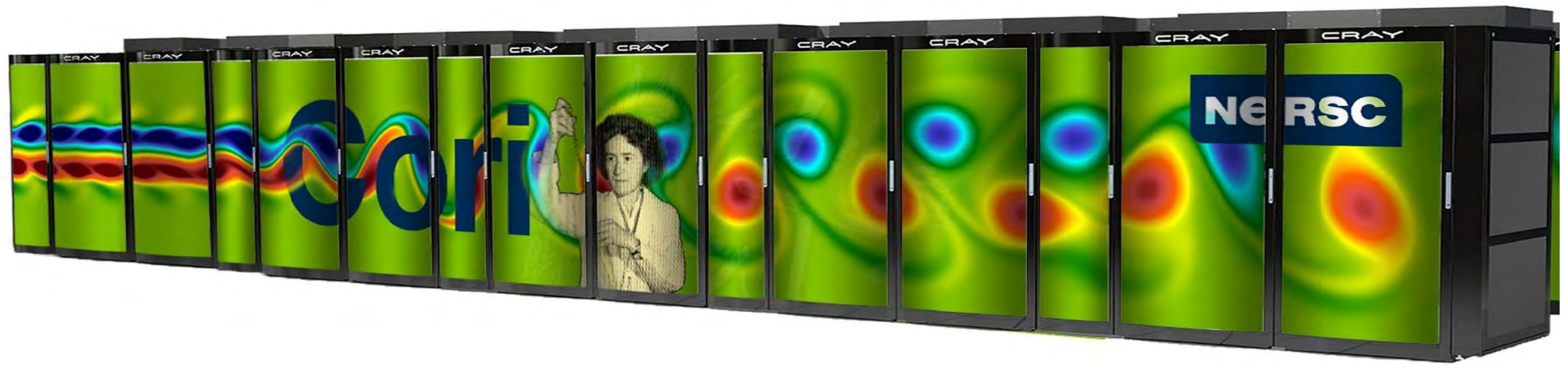
What does this mean for science + education?

- Can utilize…
  - ...shared hardware/compute for running code
  - ...shared data storage for big datasets
  - ...shared environments for doing work
  - ...shared workflows, ideas, and results

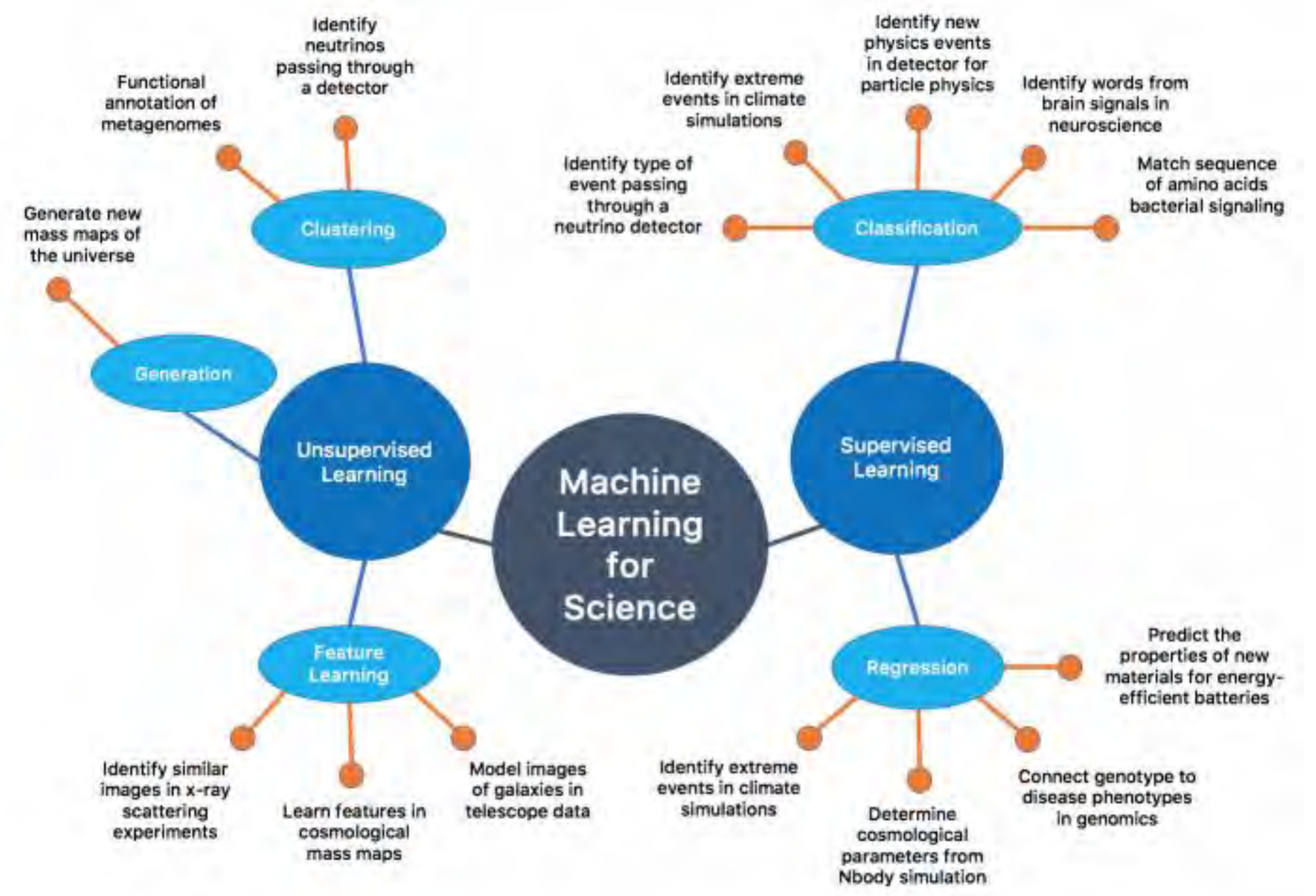

CODING
ENVIRONMENT
R
NERSC
Cori
jupyterhub
FANCY HARDWARE
AUTHENTICATION

NERSC

- **Mission HPC center for US Dept. of Energy**
  - 7000+ diverse users across science (e.g. cosmology, climate, biosciences)
- **Cori – Cray XC40 (31.4 PF Peak)**
  - 9668 Intel Knights Landing (KNL), 2388 Haswell nodes

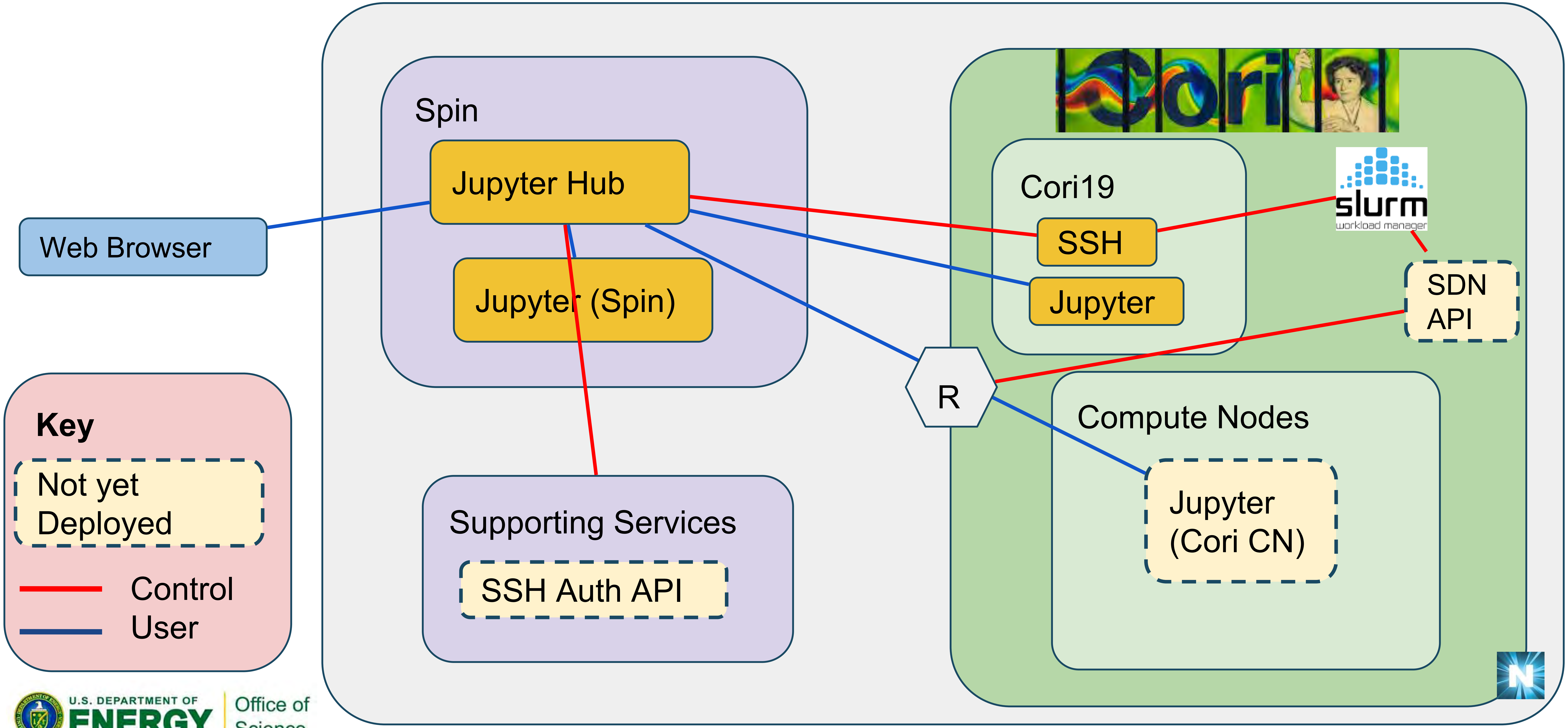


- **Deep learning: Data and analytics (DAS) group:**
  - Tools for machine learning; optimized for scale
  - Cutting-edge methods/Collaborations/Training
- **Interactive Computing at NERSC:**
  - Modifications to SLURM including real-time and interactive queues with dedicated resource
  - Also other interactive features not described here: (visualization; science gateways etc.)

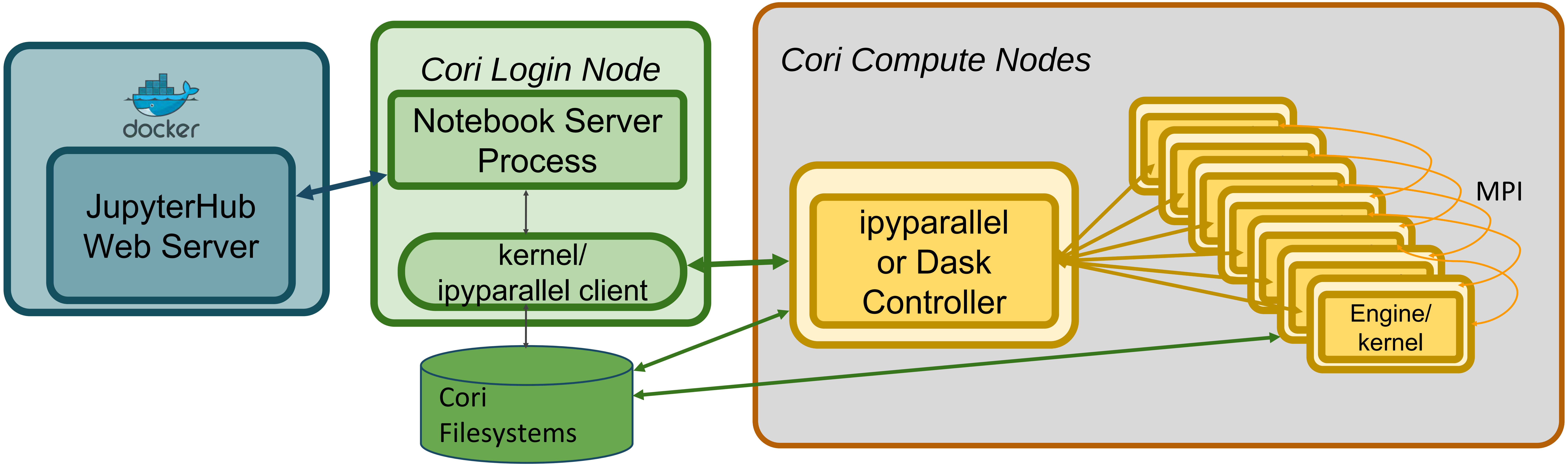


Slides credit: S. Farrell et al, LBNL

Jupyter at NERSC Architecture

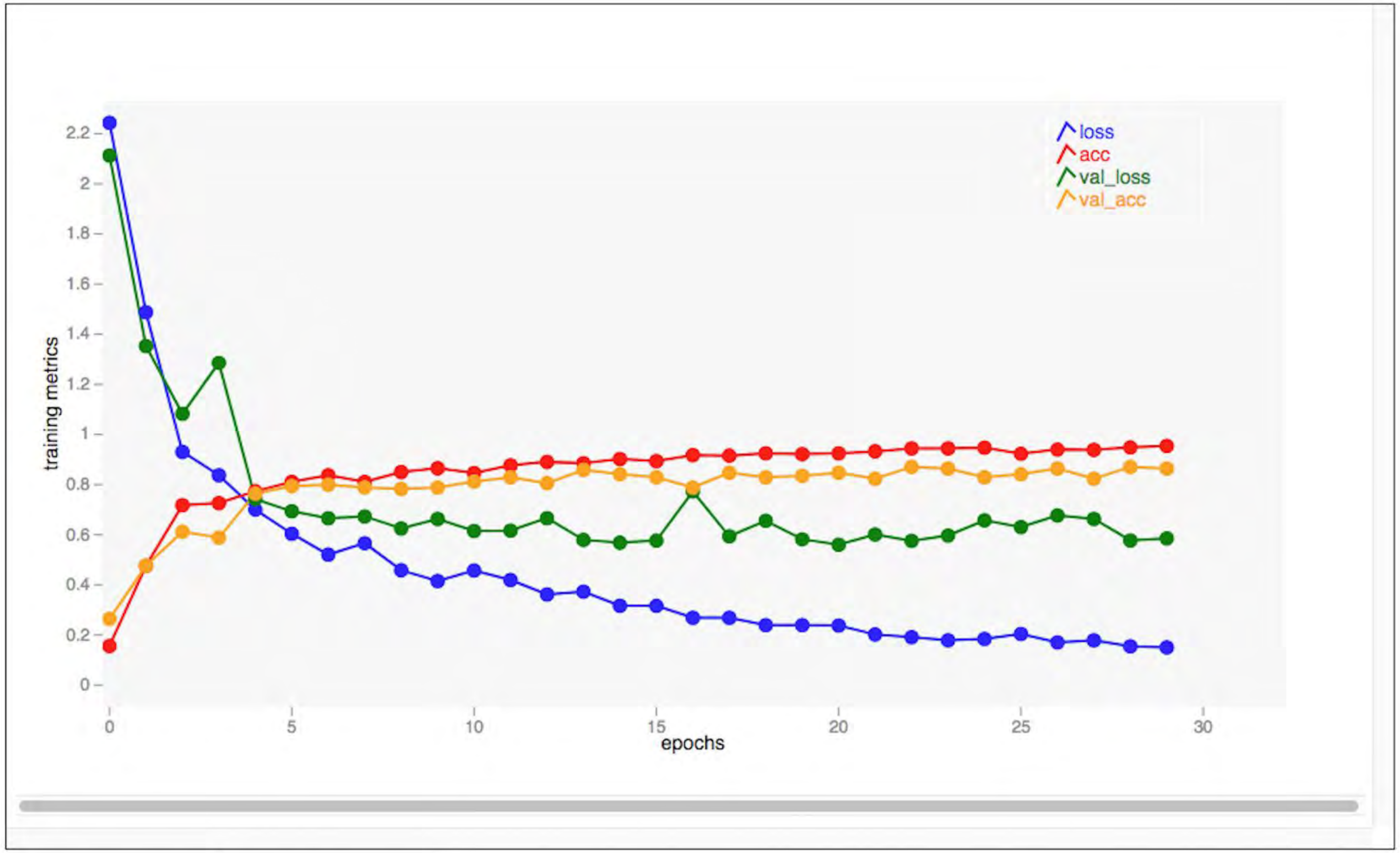


Parallel Jobs via Jupyter



Interactive Distributed Deep Learning

https://github.com/sparticlesteve/cori-intml-examples/



Stop selected Restart selected

index	status	epoch	h1	h2	h3	dropout	optimizer	loss	val_loss	acc	val_acc
3	Ended Tra...	31	16	64	16	0.88467	Adam	2.003565...	1.689878...	0.213253...	0.682352...
2	Ended Tra...	31	16	8	8	0.19765	Adam	0.852827...	0.829521...	0.763855...	0.800000...
0	Ended Tra...	31	64	8	8	0.04836	Adadelta	0.157987...	0.579903...	0.944578...	0.870588...
1	Ended Tra...	31	4	8	16	0.03825	Adadelta	0.151153...	0.585079...	0.954216...	0.864705...

DistWidgetHPO_rpv

https://jupyter-dev.nersc.gov/user/mhenders/notebooks/global/homes/m/mhend...

jupyter DistWidgetHPO_rpv Last Checkpoint: Last Thursday at 3:44 PM (autosaved)

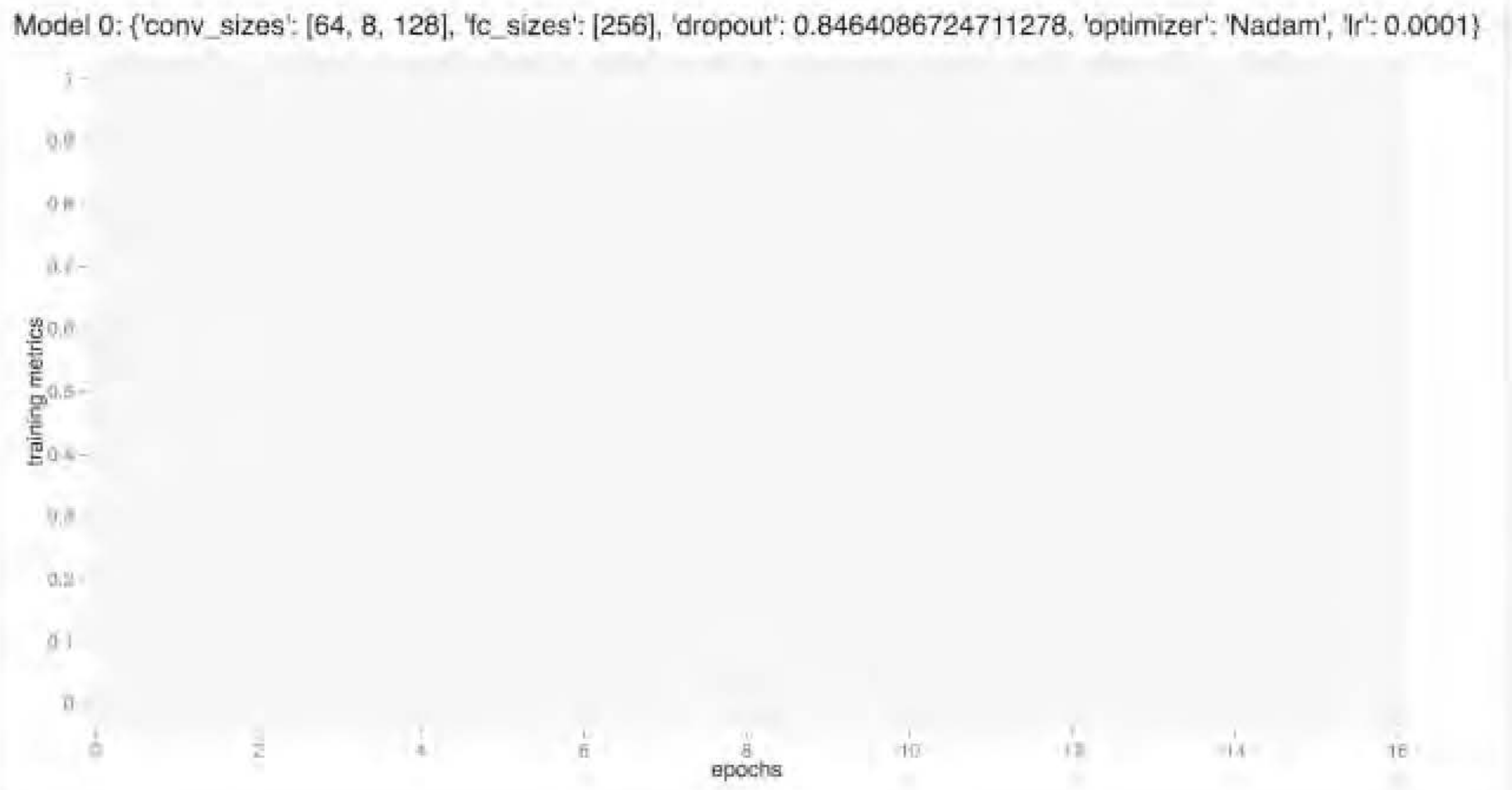
Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | lsc-hpo

Run Code Edit Presentation Show Presentation

```
    conv_sizes=conv_sizes,
    fc_sizes=fc_sizes,
    dropout=dropout,
    optimizer=optimizer,
    lr=lr,
)

psw = ParamSpanWidget(
    compute_func=train_func,
    vis_func=plot_func,
    params=hpo_params,
    ipp_cluster_id=cluster_id
)

psw.submit_computations()

psw
```



Stop selected Restart selected

index	status	epoch	conv_sizes	fc_sizes	dropout	optimizer	lr	loss	val_loss	acc	val_acc
0	Not Started	-1	[64, 8, 128]	[256]	0.84641	Nadam	0.0001	nan	nan	nan	nan
1	Not Started	-1	[4, 8, 16]	[32]	0.69948	Nadam	0.01	nan	nan	nan	nan
2	Not Started	-1	[32, 8, 32]	[128]	0.29744	Adam	0.0001	nan	nan	nan	nan
3	Not Started	-1	[32, 4, 128]	[128]	0.8138	Adam	0.0001	nan	nan	nan	nan
4	Not Started	-1	[32, 16, 64]	[256]	0.39651	Adadelta	0.01	nan	nan	nan	nan
5	Not Started	-1	[8, 64, 128]	[128]	0.8811	Adadelta	0.0001	nan	nan	nan	nan
6	Not Started	-1	[32, 32, 1...	[256]	0.58127	Adadelta	0.01	nan	nan	nan	nan
7	Not Started	-1	[16, 32, 1...	[256]	0.88174	Nadam	0.01	nan	nan	nan	nan
8	Not Started	-1	[64, 16, 64]	[256]	0.69253	Nadam	0.01	nan	nan	nan	nan
9	Not Started	-1	[4, 64, 128]	[128]	0.72525	Adam	0.0001	nan	nan	nan	nan
10	Not Started	-1	[4, 16, 128]	[64]	0.50132	Adam	0.0001	nan	nan	nan	nan
11	Not Started	-1	[64, 4, 128]	[128]	0.95608	Adadelta	0.0001	nan	nan	nan	nan
12	Not Started	-1	[16, 4, 8]	[128]	0.64399	Adadelta	0.001	nan	nan	nan	nan
13	Not Started	-1	[8, 64, 128]	[256]	0.42386	Nadam	0.01	nan	nan	nan	nan
14	Not Started	-1	[4, 4, 64]	[128]	0.60639	Adam	0.0001	nan	nan	nan	nan
15	Not Started	-1	[8, 64, 32]	[256]	0.01919	Adam	0.001	nan	nan	nan	nan

Jupyter Matters to NERSC Users

Users appreciate Jupyter @ NERSC...

[Venkitesh: "… jupyter notebooks are very important for me: ***The 3 most important things in life: food, shelter and jupyter… everything else is optional.*"]

"I really like the jupyter interface."

"New jupyter notebooks are awesome!"

"Great interactive workflow (e.g. for postprocessing) via JupyterHub"

"As mentioned, the ability to access data from the scratch directories through the Jupyter hub is very important to my workflow."

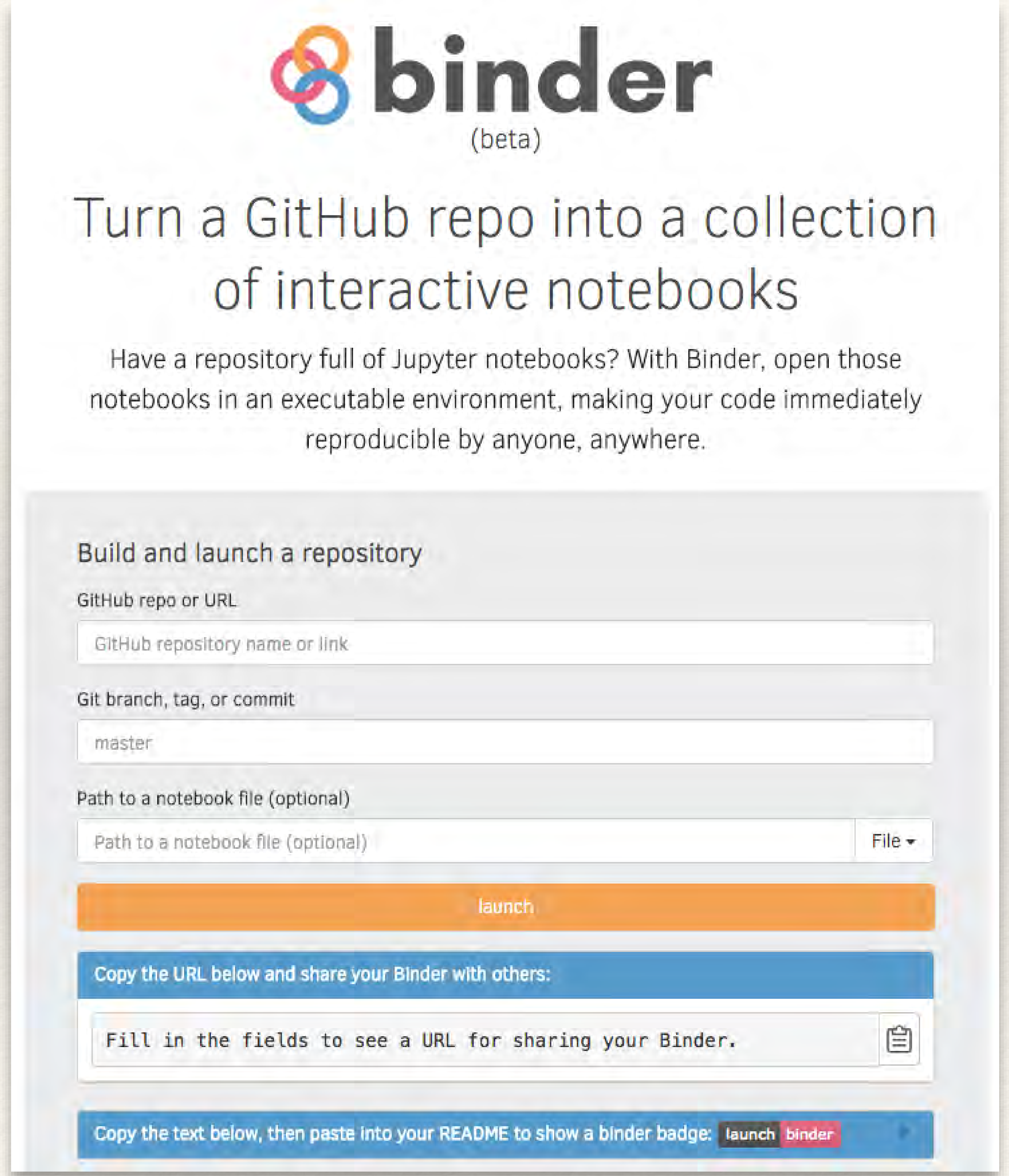
"I absolutely love the fact that I can use the Jupyter hub to access the Cori scratch directory. This allows me to analyze data through the browser ... or to quickly check that simulation runs are going as expected without having to transfer data to a different location. ***I actually also have access to other supercomputer clusters, but this is one of the biggest reasons I mainly use Cori and Edison for debugging and production runs.***"

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.

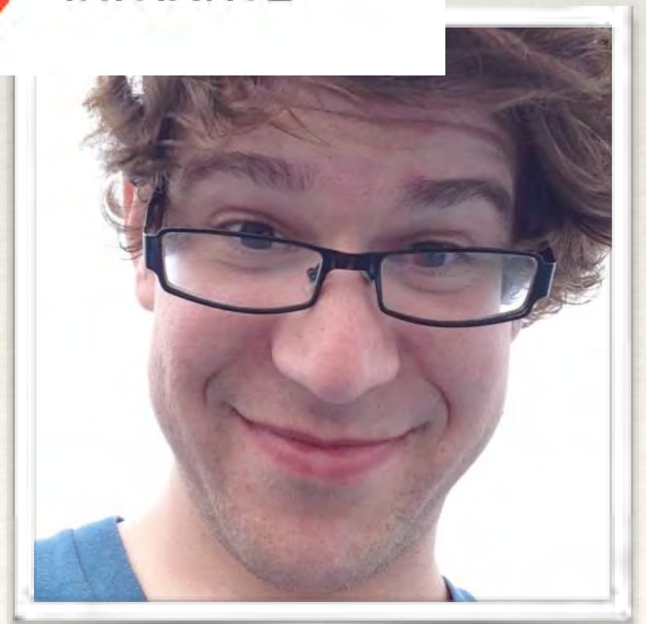
Buckheit and Donoho, WaveLab and Reproducible Research, ***1995***

mybinder.org: shareable reproducibility



Origins

CHAN ZUCKERBERG INITIATIVE



github.com/freeman-lab

jupyter GitHub

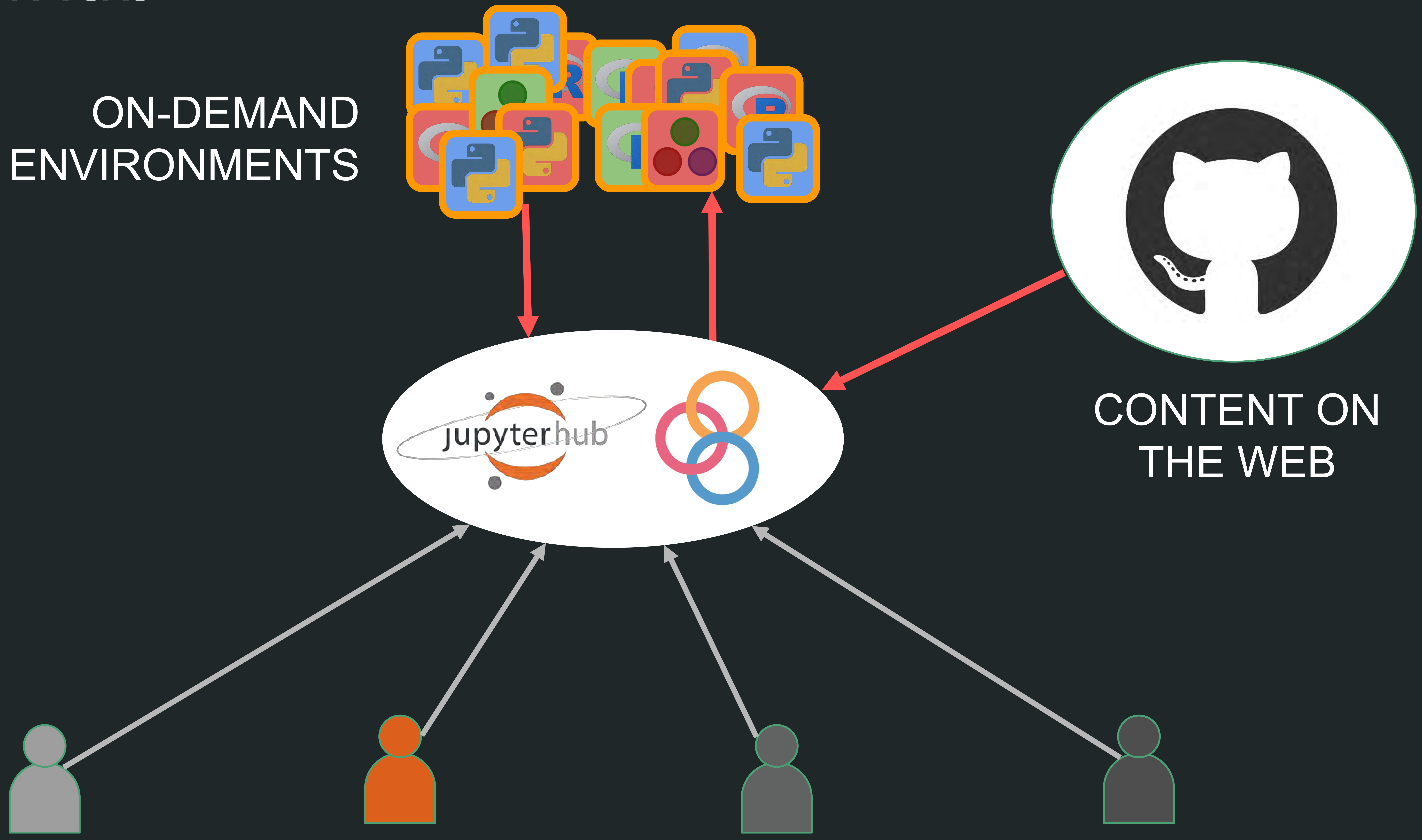
+

Explicit Dependencies

+

docker kubernetes

BinderHub



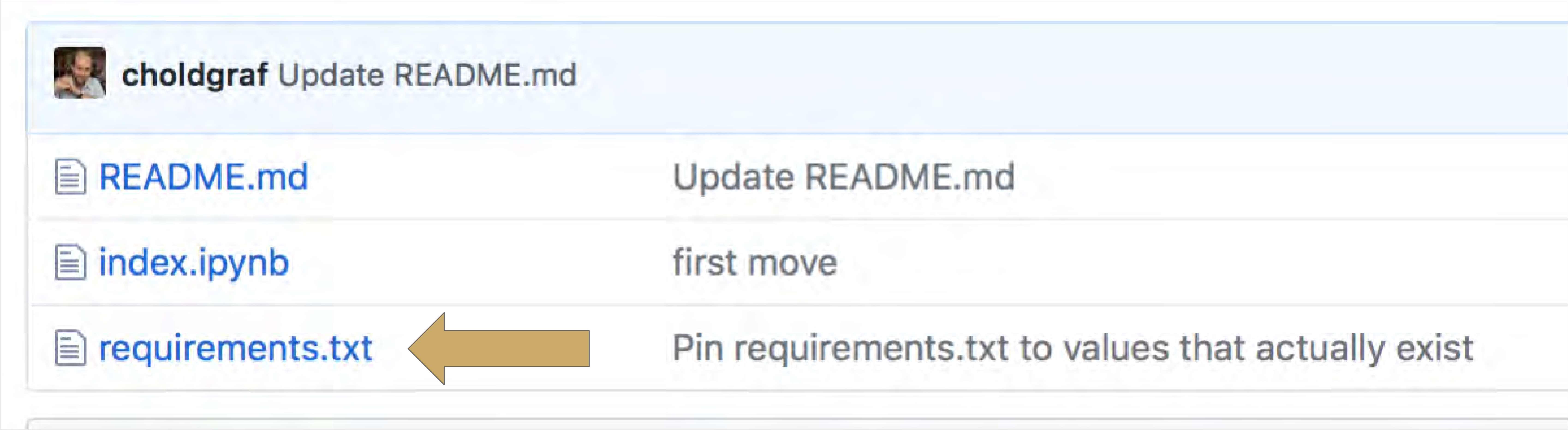
repo2docker

Convert a repository into a Docker image that runs the code inside.

repo2docker.readthedocs.io

repo2docker
what does it do?

Identify **requirements**



repo2docker
what does it do?

generate **Dockerfile**

assemble the environment for the repo

```
...
# Copy and chown stuff.
COPY src/ ${HOME}
RUN chown -R ${NB_USER}:${NB_USER} ${HOME}

# Run assemble scripts! These will actually build the sp
# in the repository into the image.
USER ${NB_USER}
RUN ${KERNEL_PYTHON_PREFIX}/bin/pip install --no-cache-d
    -r "requirements.txt"
...
```

repo2docker
what does it do?

build (& **push**) image

```
docker build -t myimage
docker push myimage
```

repo2docker

what does it do?

```
$ jupyter repo2docker \
>       https://github.com/minrk/ligo-binder

Cloning into
'/var/folders/.../T/repo2dockermu6z66sd'...
Using CondaBuildPack builder
Step 1/31 : FROM buildpack-deps:bionic
 ---> 29f4eef41002
Step 2/31 : ENV DEBIAN_FRONTEND=noninteractive
 ---> Using cache
 ---> ee1ba7c4f5f4
Step 3/31 : RUN apt-get update &&     apt-get
install --yes --no-install-recommends locales &&
apt-get purge &&     apt-get clean &&     rm -rf
/var/lib/apt/lists/*
```

BinderHub

One-click sharable, interactive, reproducible environments from your public git repository

mybinder.org

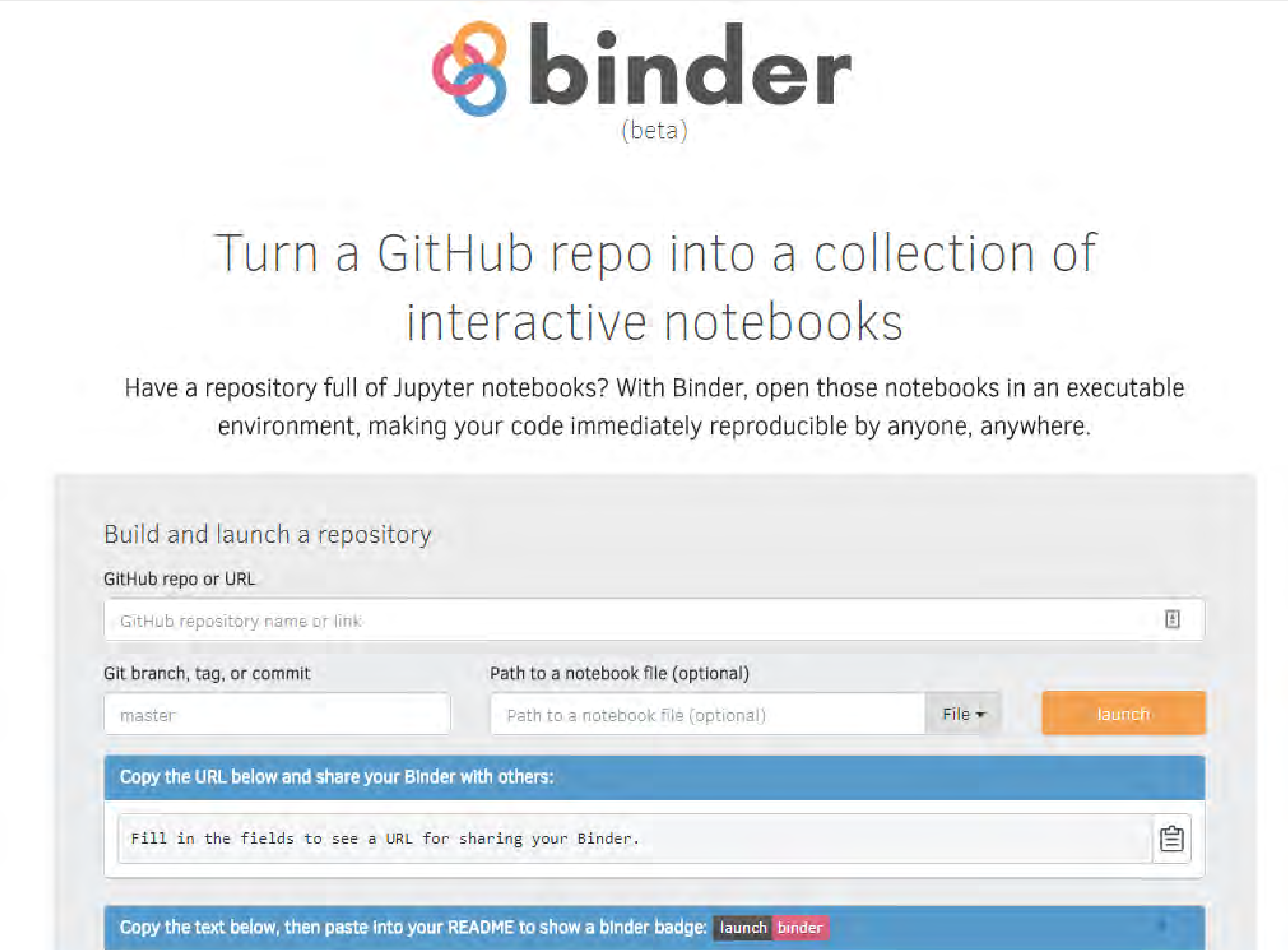
binderhub.readthedocs.io

binderhub = repo2docker + jupyterhub

BinderHub is open tech...

- Built on Kubernetes
- Cloud-agnostic
- Scalable
- **Community driven and deployable by anyone**

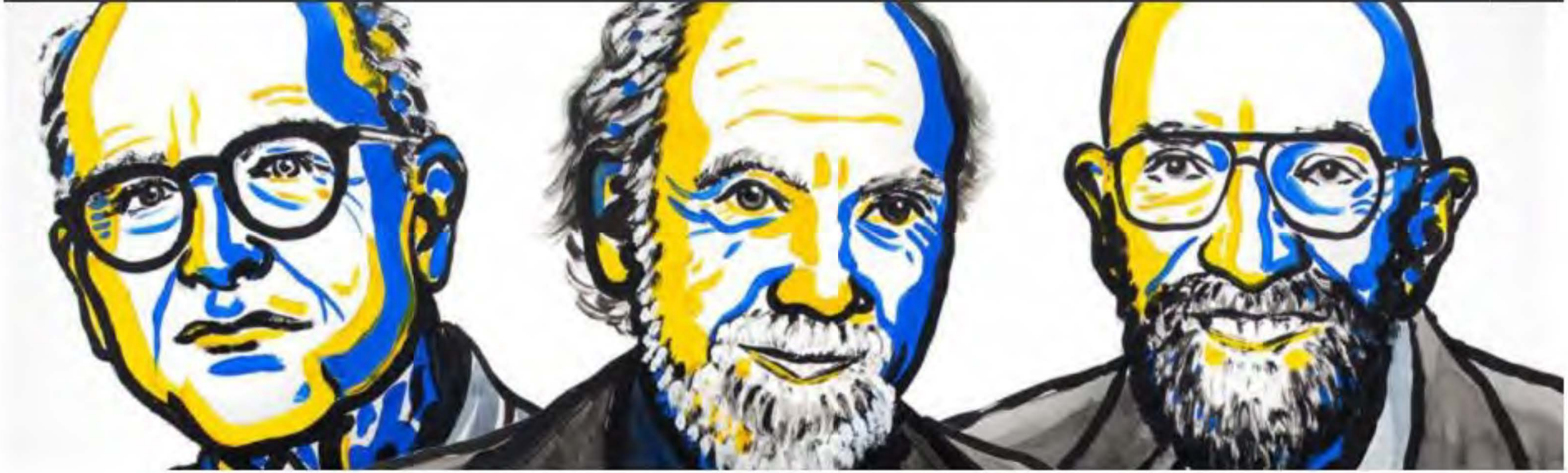
mybinder.org



Physics @BERKELEY

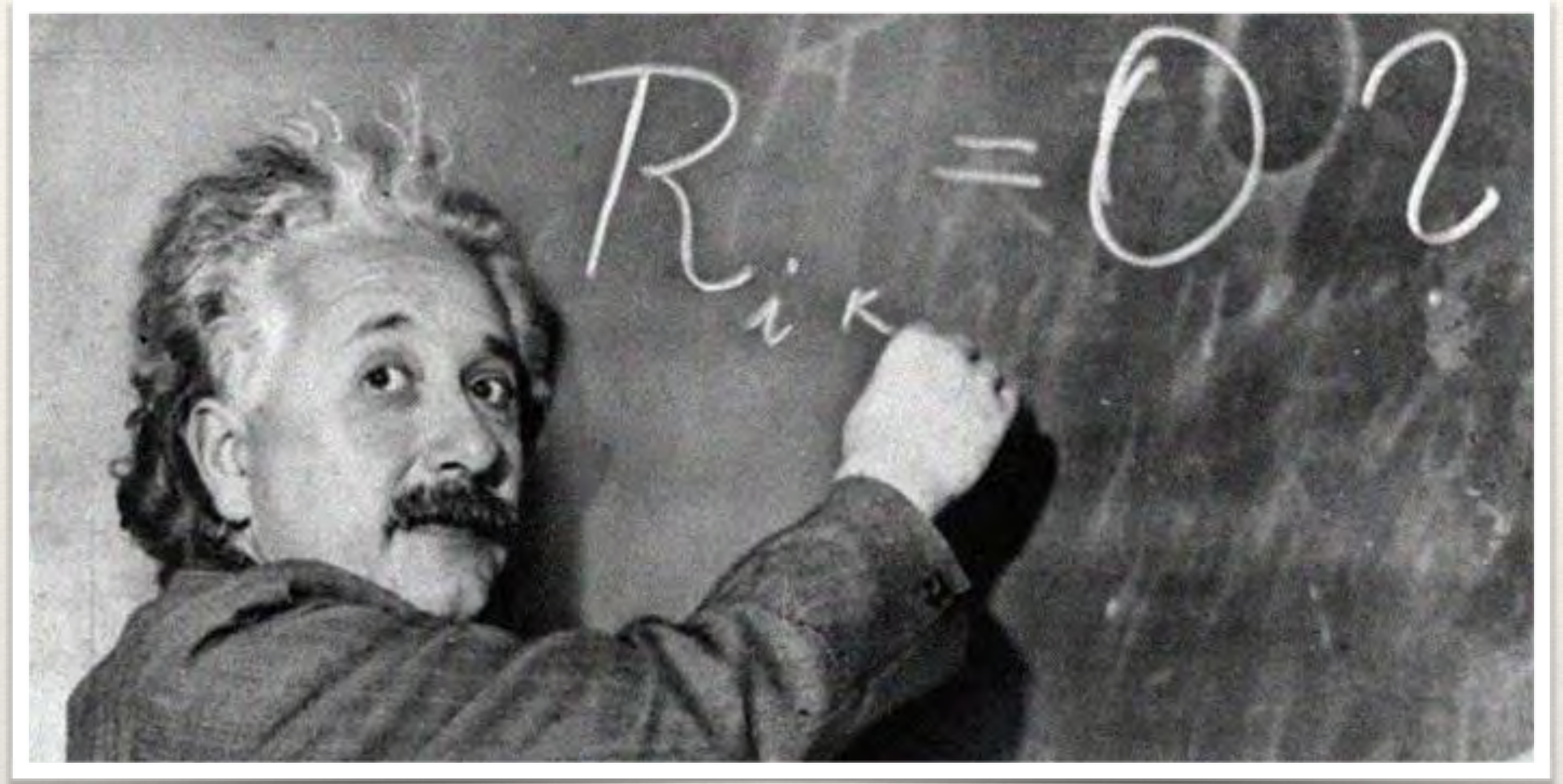
ABOUT US | DIRECTORY | EQUITY & INCLUSION | INDUSTRY | GIVE

START HERE | NEWS : EVENTS : ACADEMICS : RESEARCH & FACULTY : ADMINISTRATION



2017 NOBEL PRIZE IN PHYSICS AWARDED FOR DISCOVERY OF GRAVITATIONAL WAVES

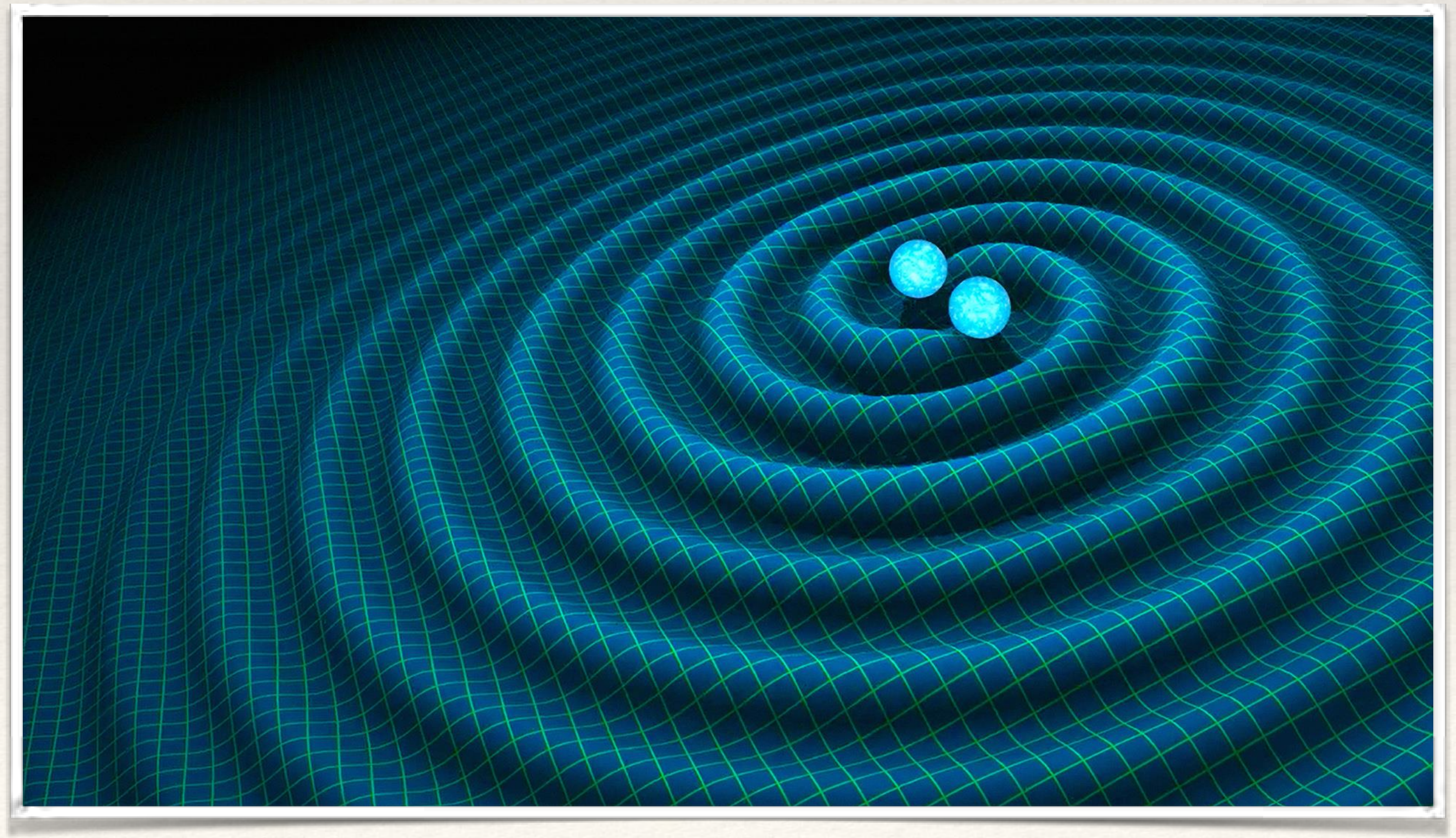
A long time ago in a galaxy far, far away…



$$R_{\mu\nu} - \frac{1}{2}R\,g_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$$

Einstein's Field Equations of General Relativity

Annalen der Physik, 1916



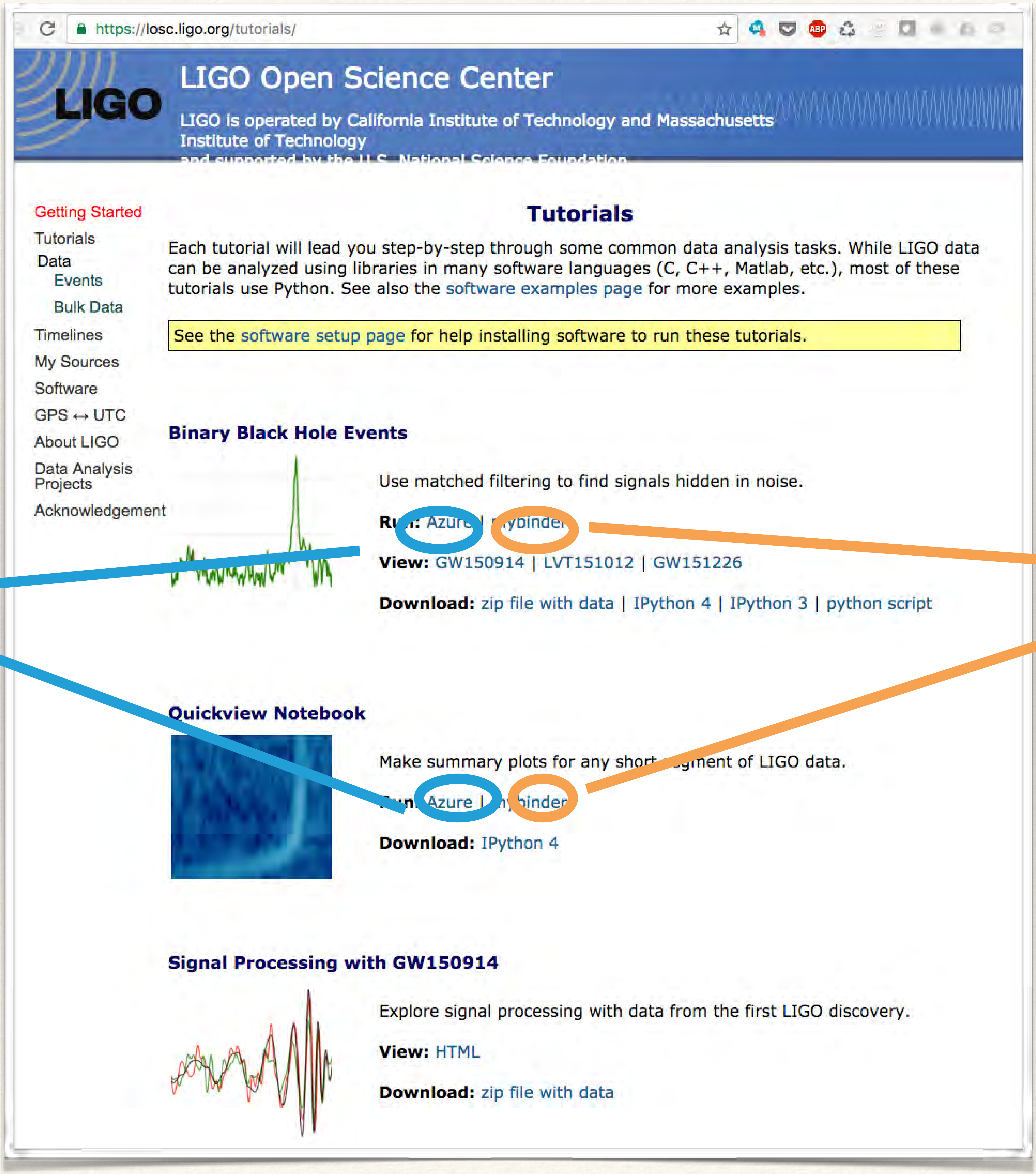
Two identical detectors: Hanford, WA and Livingston, LA

LIGO: a feat of science & engineering

Detection problem:

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

LIGO: Open Science with Jupyter



binder

September 14, 2015

NumPy

SciPy

matplotlib

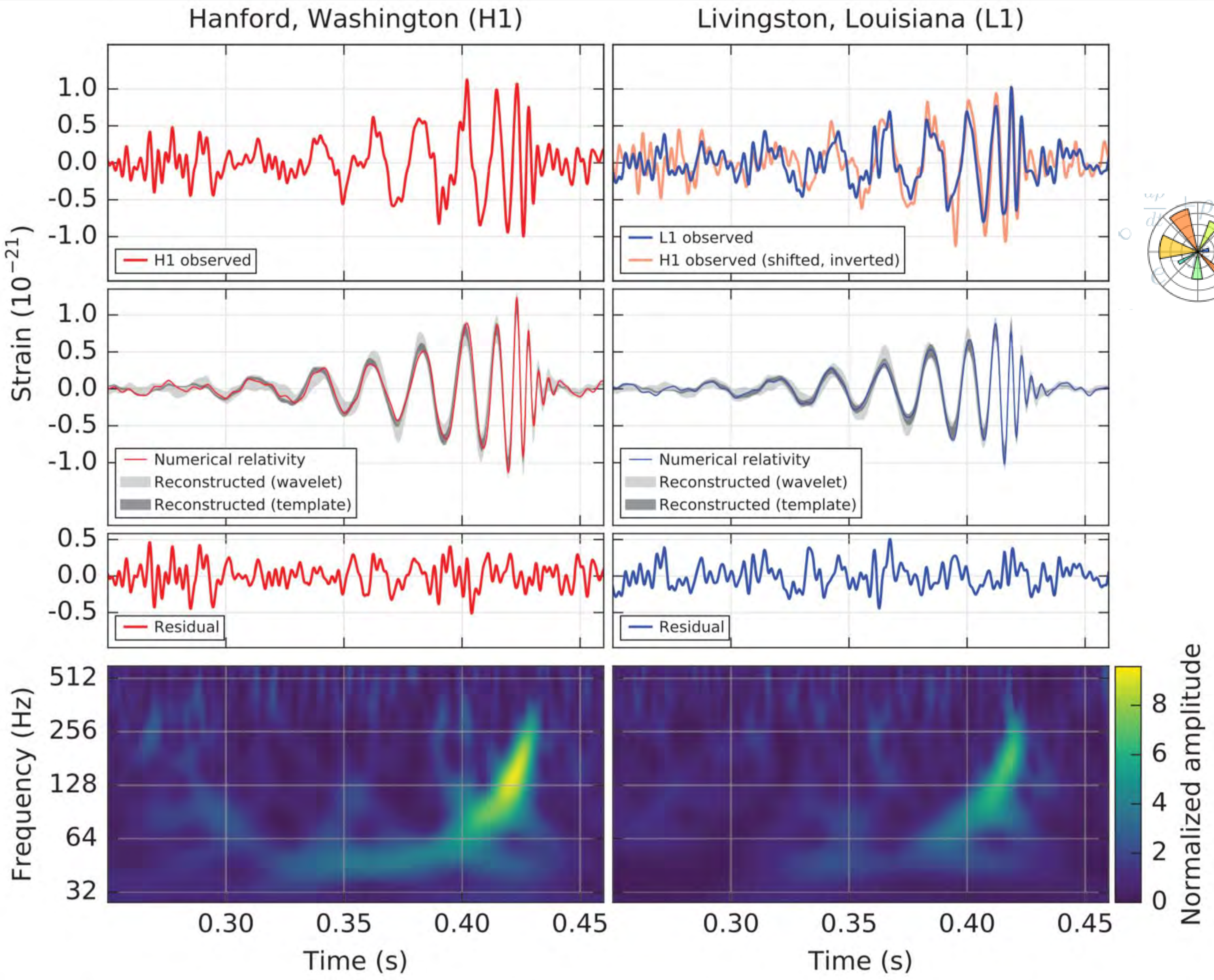
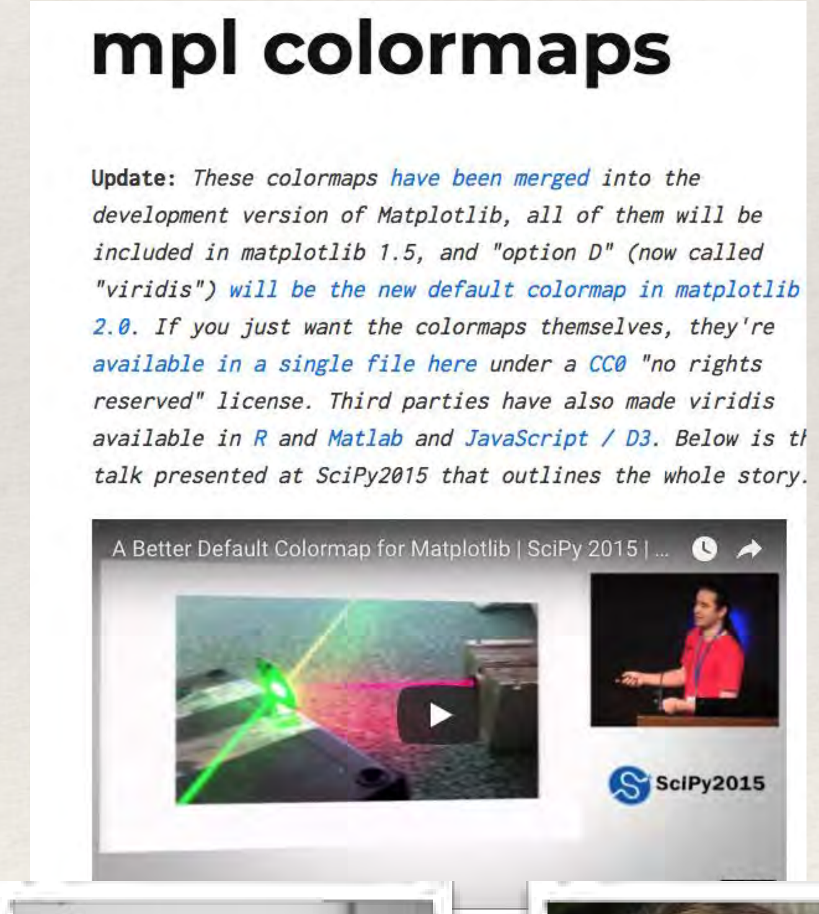


FIG. 1. The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress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



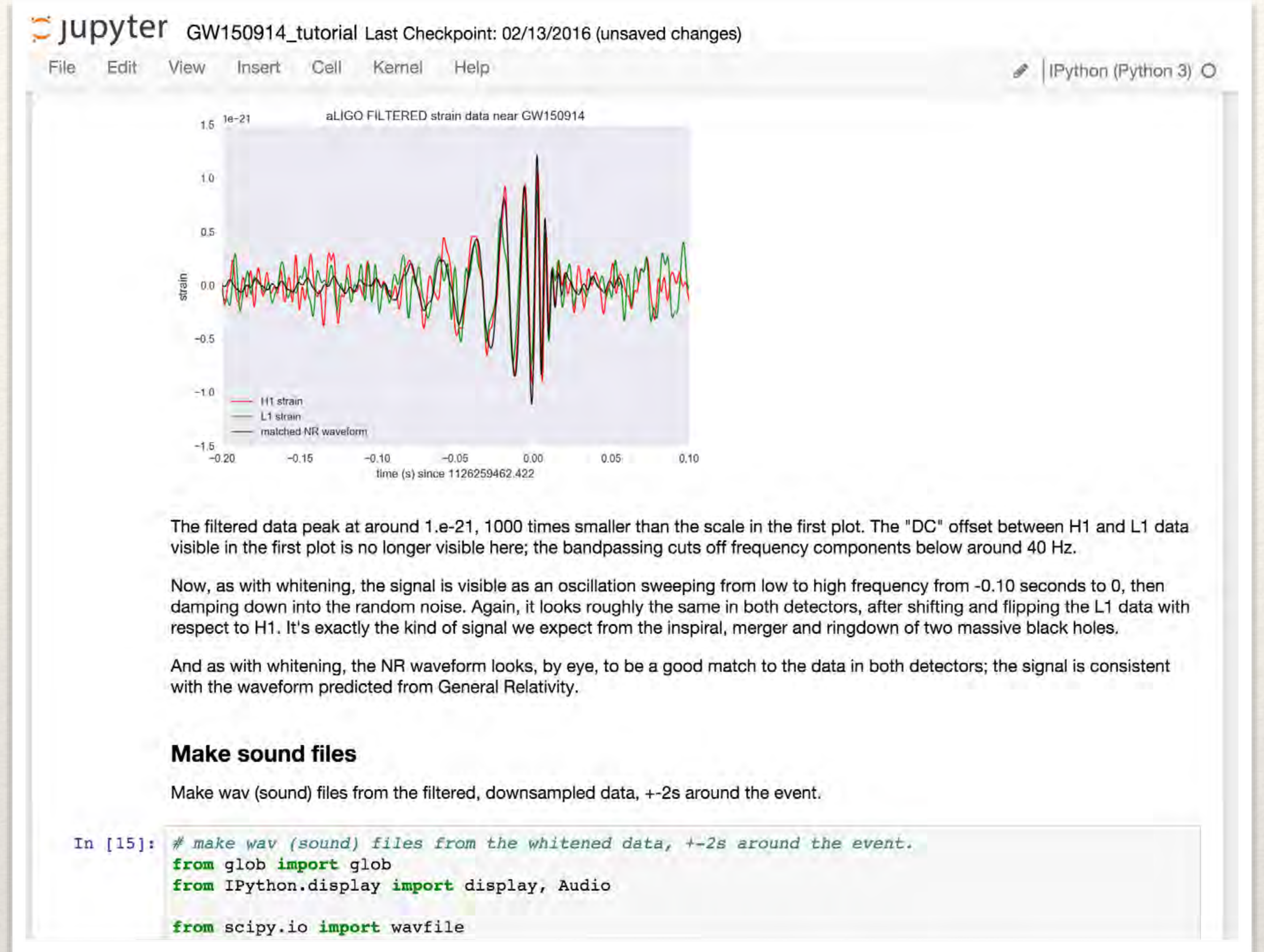
mpl colormaps

Update: *These colormaps have been merged into the development version of Matplotlib, all of them will be included in matplotlib 1.5, and "option D" (now called "viridis") will be the new default colormap in matplotlib 2.0. If you just want the colormaps themselves, they're available in a single file here under a CC0 "no rights reserved" license. Third parties have also made viridis available in R and Matlab and JavaScript / D3. Below is th talk presented at SciPy2015 that outlines the whole story.*

A Better Default Colormap for Matplotlib | SciPy 2015 | ...

SciPy2015

The song of the universe



The filtered data peak at around 1.e-21, 1000 times smaller than the scale in the first plot. The "DC" offset between H1 and L1 data visible in the first plot is no longer visible here; the bandpassing cuts off frequency components below around 40 Hz.

Now, as with whitening, the signal is visible as an oscillation sweeping from low to high frequency from -0.10 seconds to 0, then damping down into the random noise. Again, it looks roughly the same in both detectors, after shifting and flipping the L1 data with respect to H1. It's exactly the kind of signal we expect from the inspiral, merger and ringdown of two massive black holes.

And as with whitening, the NR waveform looks, by eye, to be a good match to the data in both detectors; the signal is consistent with the waveform predicted from General Relativity.

Make sound files

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

```
In [15]: # 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
```

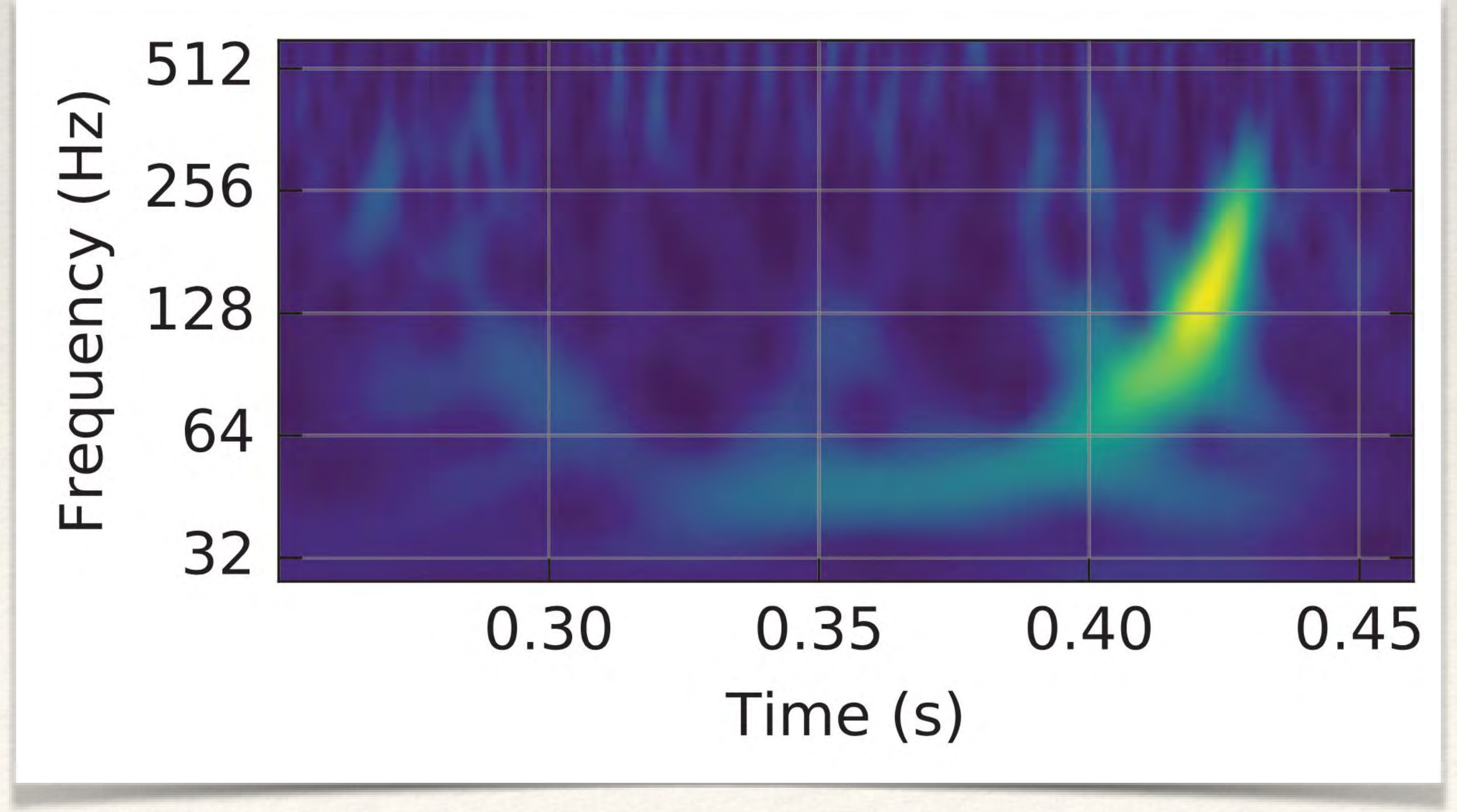
Make sound files

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'

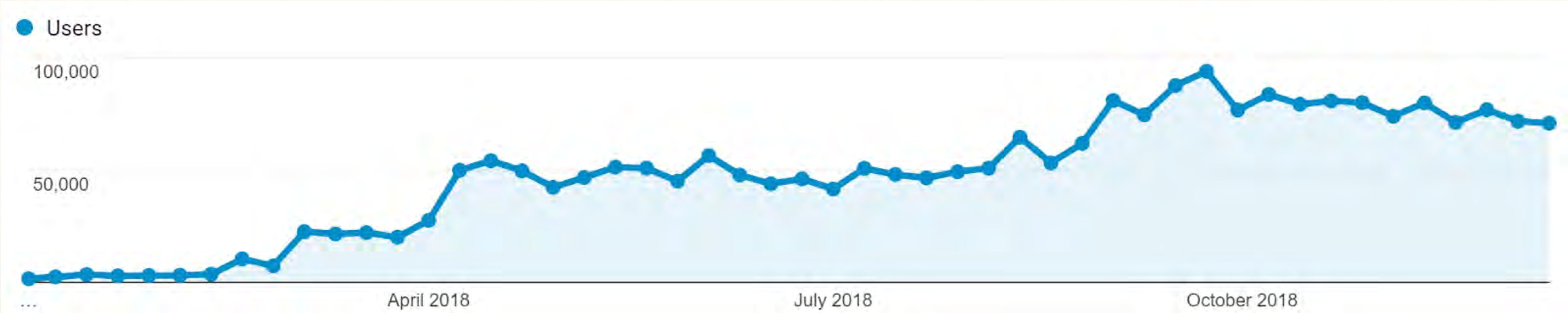
0:00



Using the IPython.display.Audio object

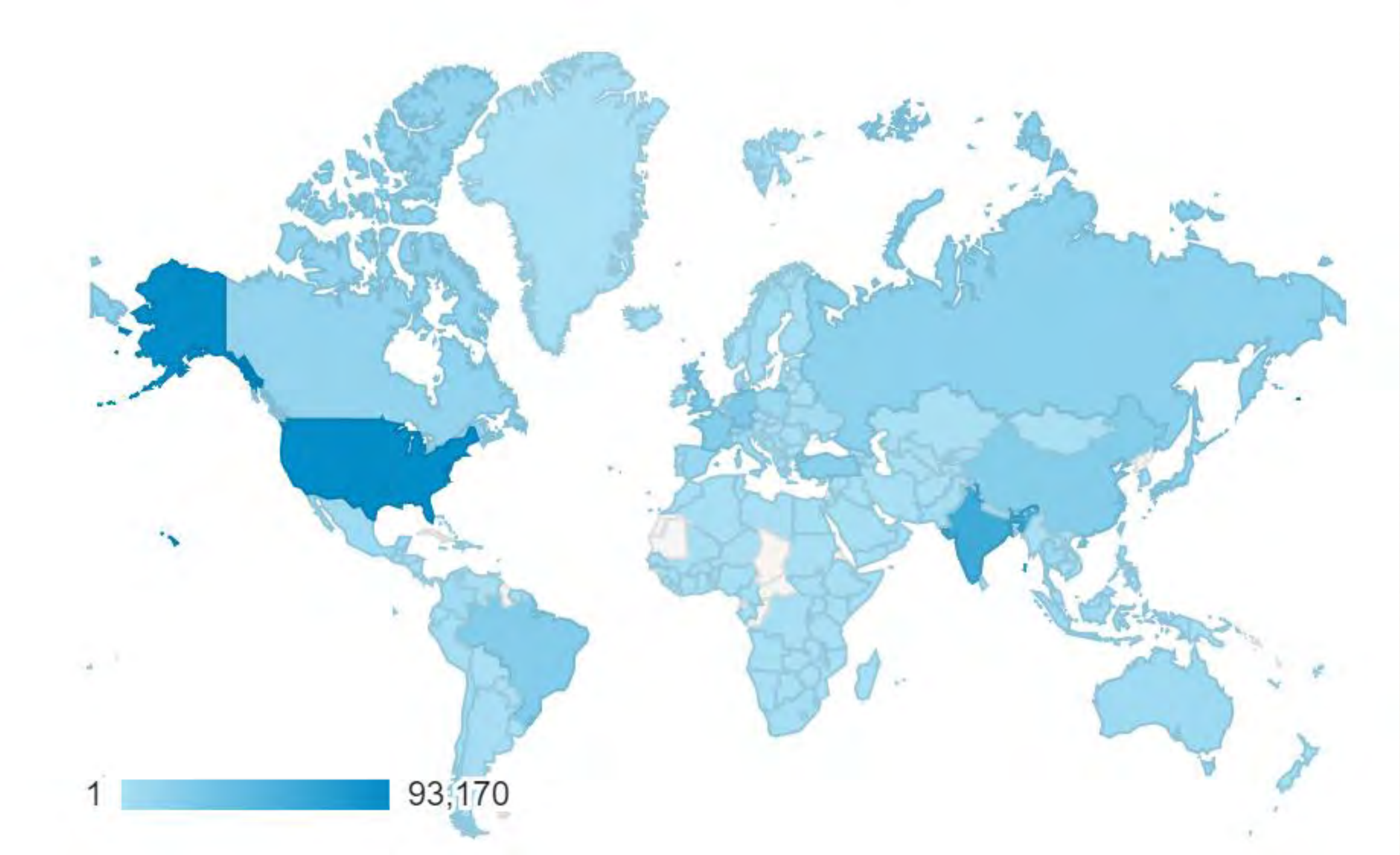
mybinder.org: usage

Weekly Binder users, Jan 1 2018 to present (~350k/month)



GORDON AND BETTY MOORE FOUNDATION

Google



Binder users, Nov 1 2018 to present

Berkeley: Yuvi Panda, Chris Holdgraf

Cal Poly: Carol Willing

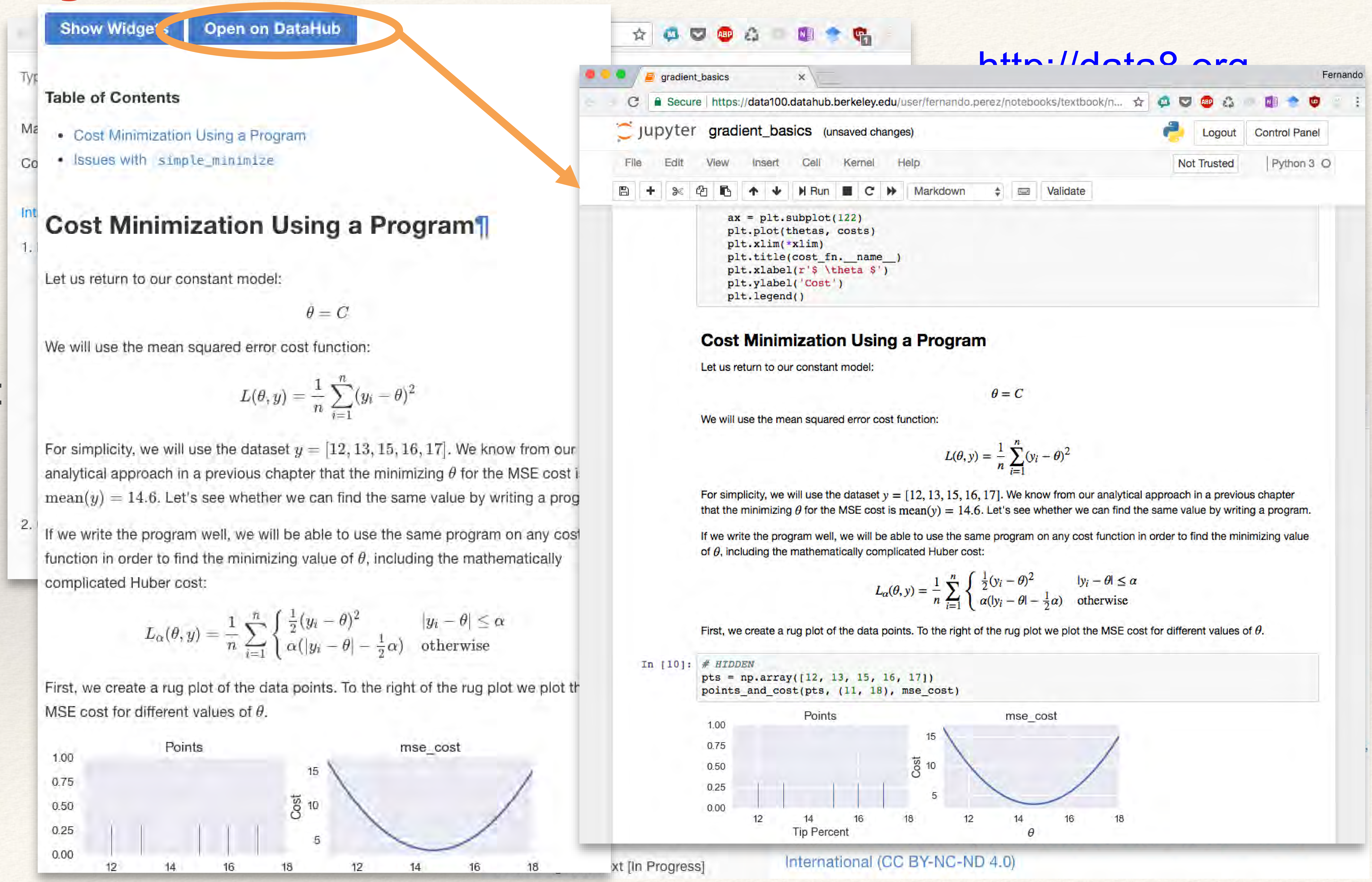
Simula: Min Ragan-Kelley

Jessica Zosa-Forde, Tim Head

Education

Berkeley's Data Science Courses

- Freshmen & upper division
- Interactive textbooks: Jupyter Notebooks
- Course deployment: JupyterHub



DataHub

datahub.berkeley.edu

Supporting 2,500+ users

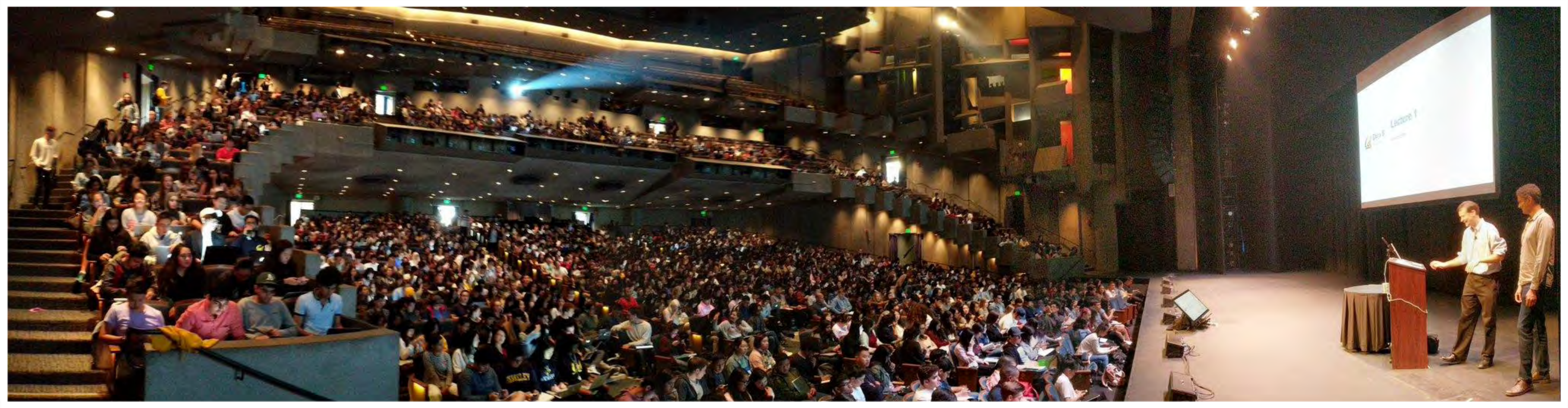
Being used for Data 8, as well as several other courses

Requires @berkeley.edu to access

Running on Azure with almost zero maintenance

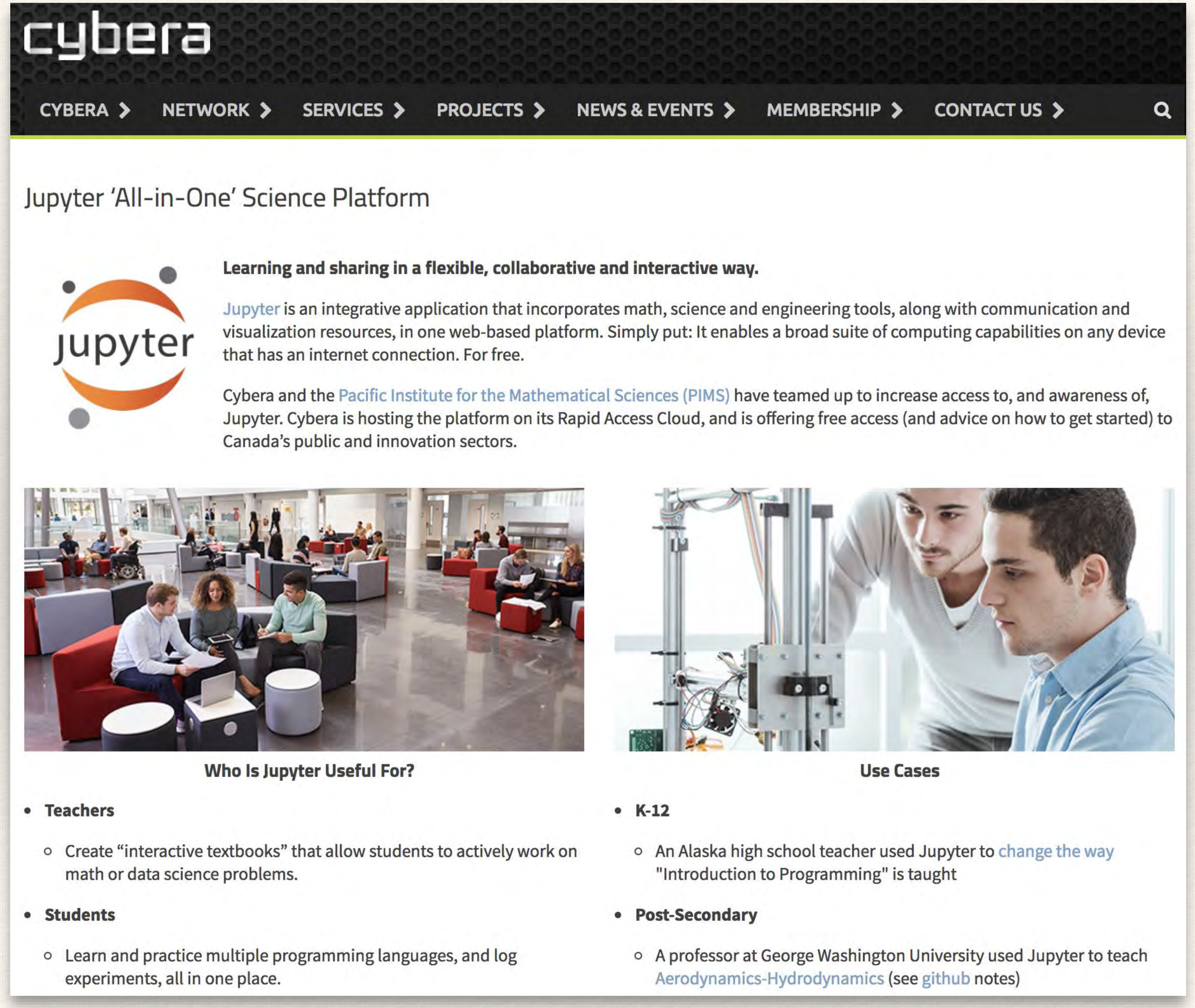
Slide: C. Holdgraf

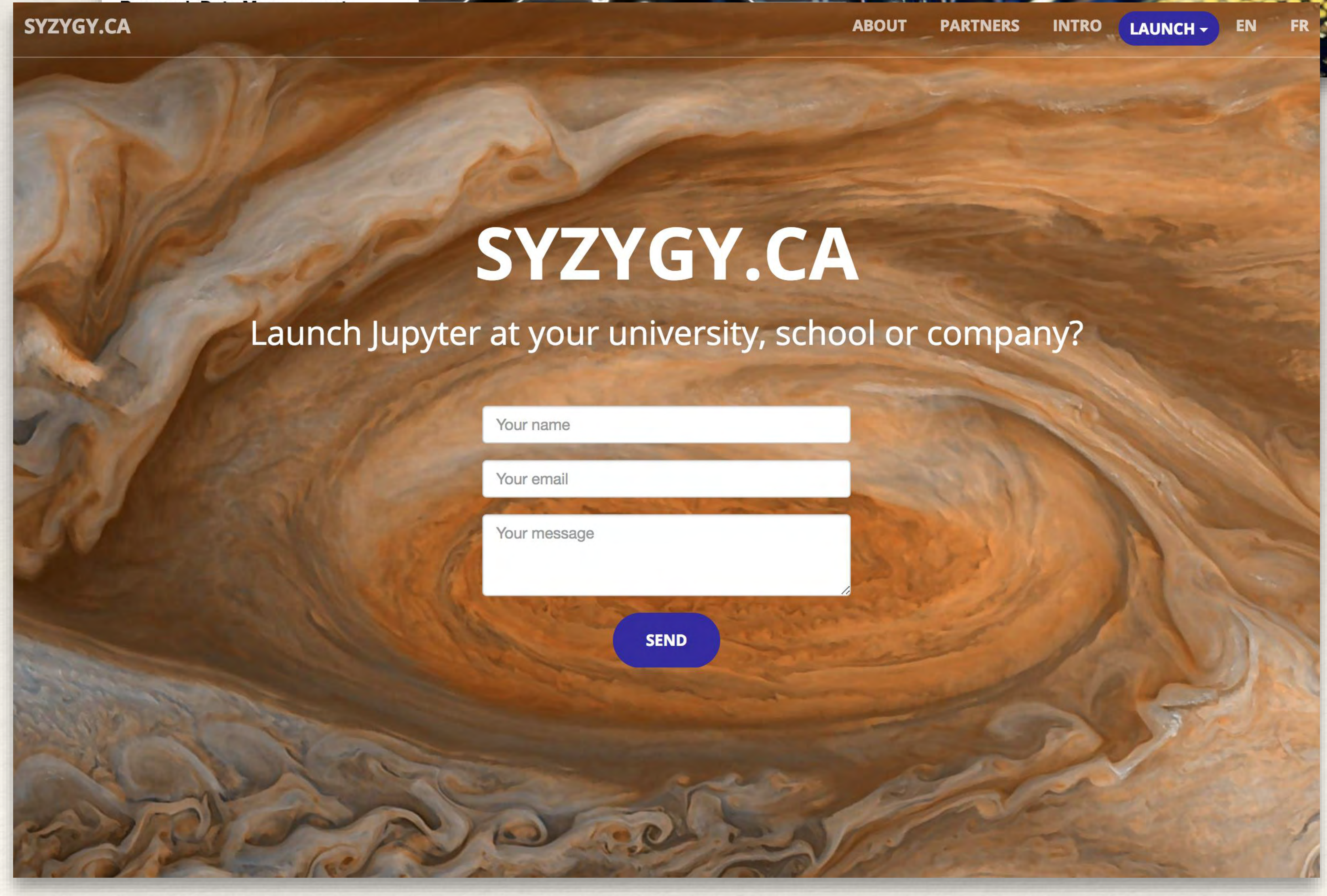
Fall 2018, Berkeley: Data 8, ~1,300 students



Data 100, ~820

CA From K-12 to HPC





Jupyter @ Berkeley and LBNL



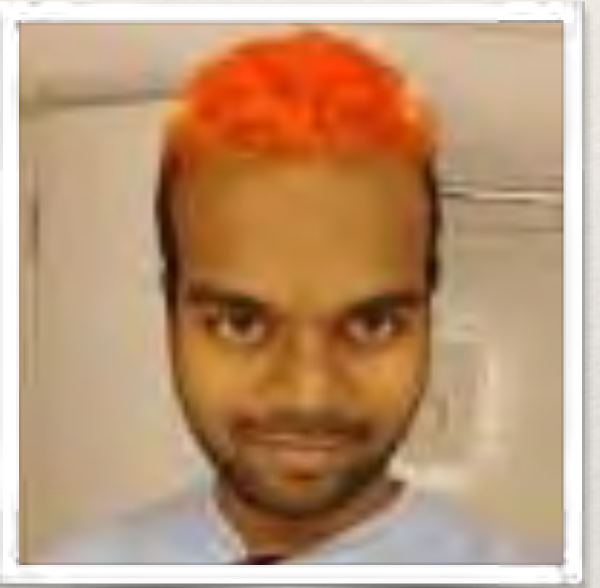


University of California, Berkeley
DEPARTMENT OF STATISTICS

BIDS
BERKELEY INSTITUTE FOR DATA SCIENCE








BERKELEY LAB






NERSC

Bloomberg

Wide industrial adoption

Funding and resources

What’s Next for Jupyter in HPC?

Jupyter R&D at LBL

- **Superfacility** - Shreyas Cholia
    - Exploring how Jupyter can act as the primary *Superfacility* interface through interactive widgets and tools for experimental and observational data. Follow up on LDRD work integrating Jupyter with HPC.
- **Usable Data Abstractions** - Lavanya Ramakrishnan
    - Enabling access to very large remote datasets through Jupyter and Jupyterlab
- **Jupyterhub** - Rollin Thomas
    - Software development to enable integration of Jupyterhub with NERSC systems in a scalable, secure and intuitive manner

Where Do We Go From Here?

Scientists and researchers are already using Jupyter. Now we need to enable this at scale.

- Jupyter is designed to be modular and pluggable
- increased emphasis on integration with big data and compute

- How do we enable them to use this on the big systems and large datasets?
- How do we connect and capture distributed facilities and workflows in Jupyter?
- How do we enable reproducible science at scale? Binder for HPC?
- What would YOU like to see?

Jupyter Community Workshop

Jupyter for Scientific User Facilities and High-Performance Computing Workshop to be held in Berkeley, California, from **Tuesday June 11 to Thursday June 13, 2019.**

http://bit.ly/jup-sfhpc

NERSC

BIDS
BERKELEY INSTITUTE
FOR DATA SCIENCE

Thank You!

Contact: scholia@lbl.gov

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov