

IP[y]:
IPython



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>





a *community* of people and an *ecosystem* of open tools and
standards for interactive computing



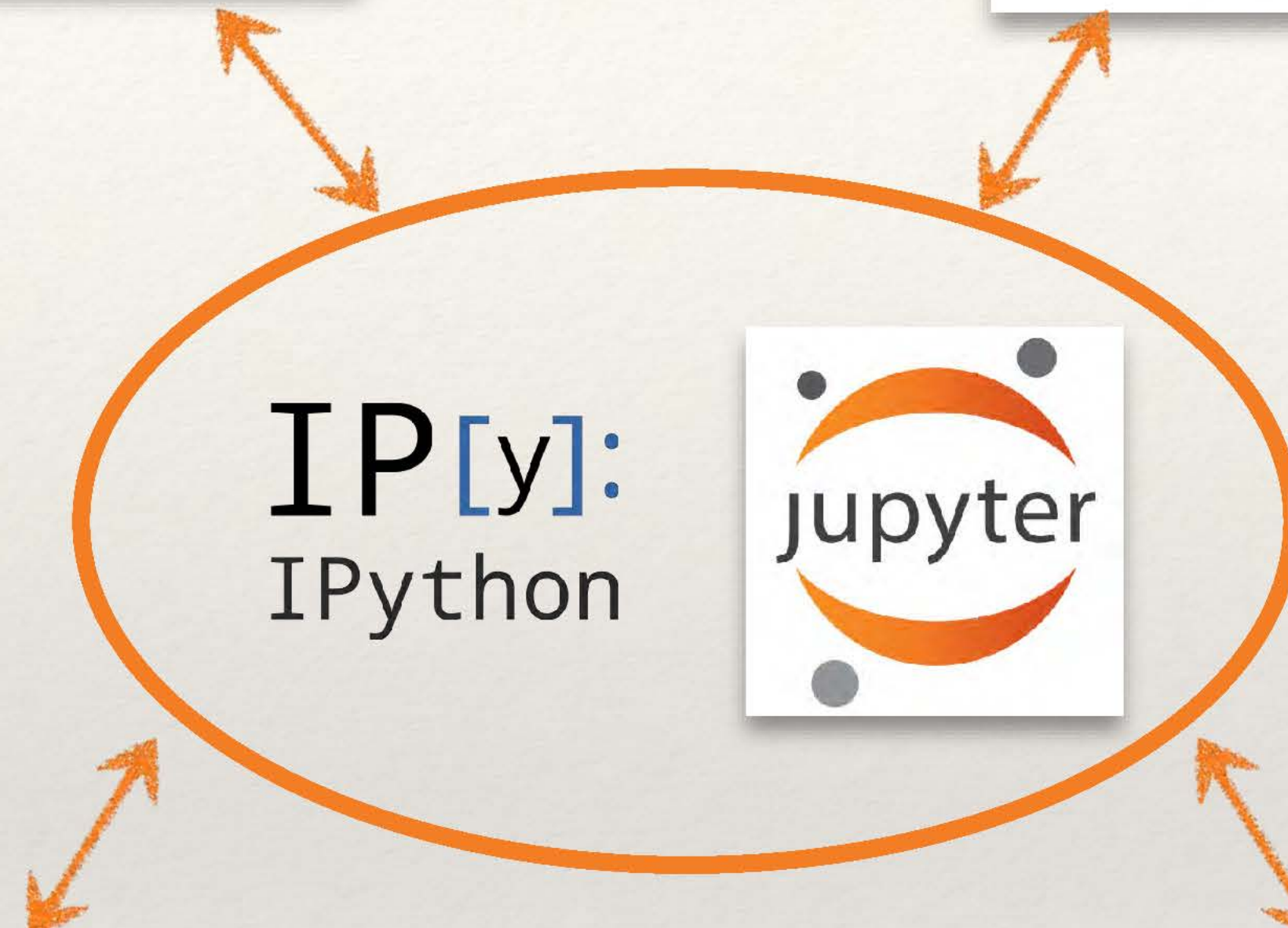
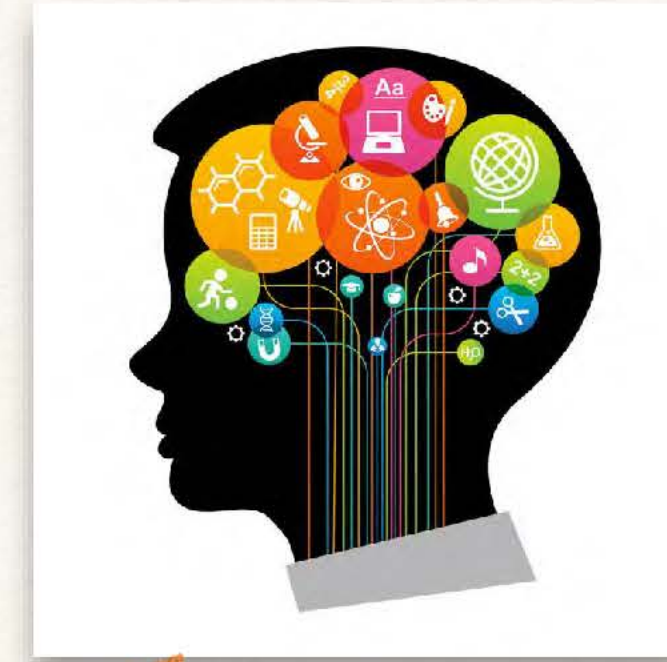
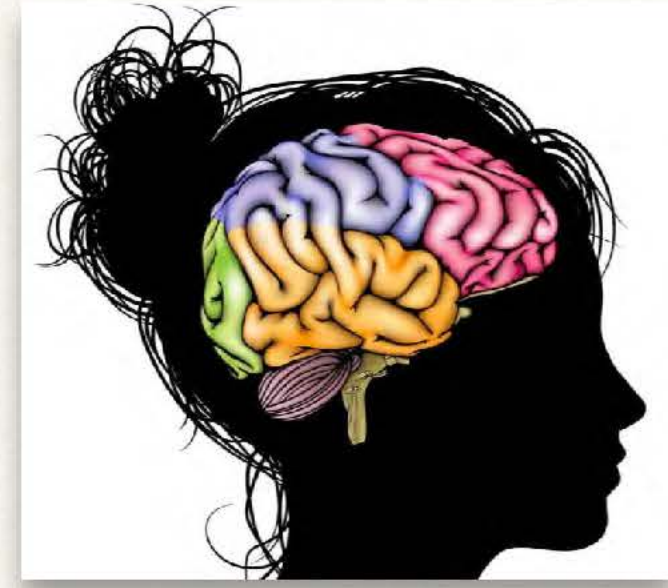


create things that are **language-agnostic** and **modular**.

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



What?



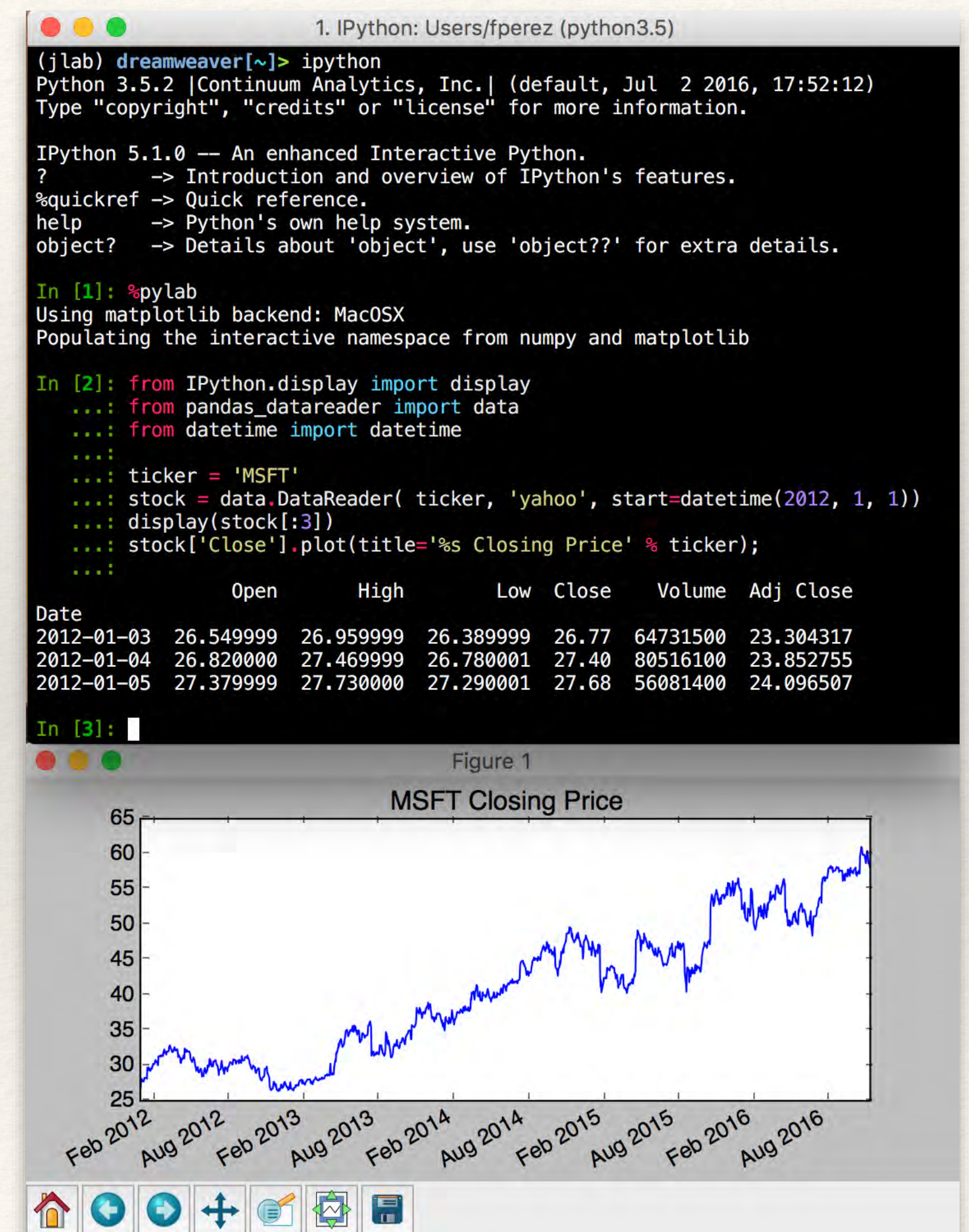
IPython: Interactive Python, 2001

A humble start:

IPython 0.0.1, 259 LOC

“Just an afternoon hack”

```
ipython-0.0.1.py x
32 Globals for SI units (including g=9.8) : _load_units = %(_load_units)s
33 Starting number for prompt counter : _prompt_ini = %(_prompt_ini)s
34 Number of history items to store in cache : _cache_size = %(_cache_size)s
35 """
36 #*****
37 # Configure here
38 _load_Numeric = 1
39 _load_Gnuplot = 1
40 _load_gracePlot = 1
41 _load_units = 1
42 _cache_size = 1000
43 _prompt_ini = 1
44 #** Don't modify below unless you know what you're doing. **
45 #
46 # Crude first version, with minimal object structure. This could be done much
47 # better, by defining a Cache class (probably using weak references or
48 # generators). But it seems to work ok. Haven't checked for memory circularity
49 # problems, though.
50 #
51 #*****
52 # Copyright (C) 2001 Fernando P0rez. <fperez@pizero.colorado.edu>
53 #
54 # Distributed under the terms of the GNU General Public License.
55 #
56 # The full text of the GPL is available at:
57 #
58 # http://www.gnu.org/copyleft/gpl.html
59 #*****
60 _author_ = 'Fernando P0rez. <fperez@pizero.colorado.edu>'
61 _version_ = '0.1'
62 #*****
63 # Class definitions
64 #
65 class HistPrompt1:
66     """Simple interactive prompt like Mathematica's."""
67     def __str__(self):
68         return '\nIn[ %s _prompt_count `+` ]:= '
69 #
70 class HistPrompt2:
71     """Simple interactive continuation prompt."""
72     def __str__(self):
73         return '... '+ '*(len('In[ %s _prompt_count `+` ]:= ')-3)
74 #*****
75 # Function definitions
76 #
77 def _history_print(arg):
78     """Printing with history cache management.
79     This is invoked everytime the interpreter needs to print, and is activated
80     by setting the variable sys.displayhook to it."""
81 #
82 global _p, _pp, _ppp, _cache, _prompt_count
83 #
```



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

Team today: where *all the credit* goes



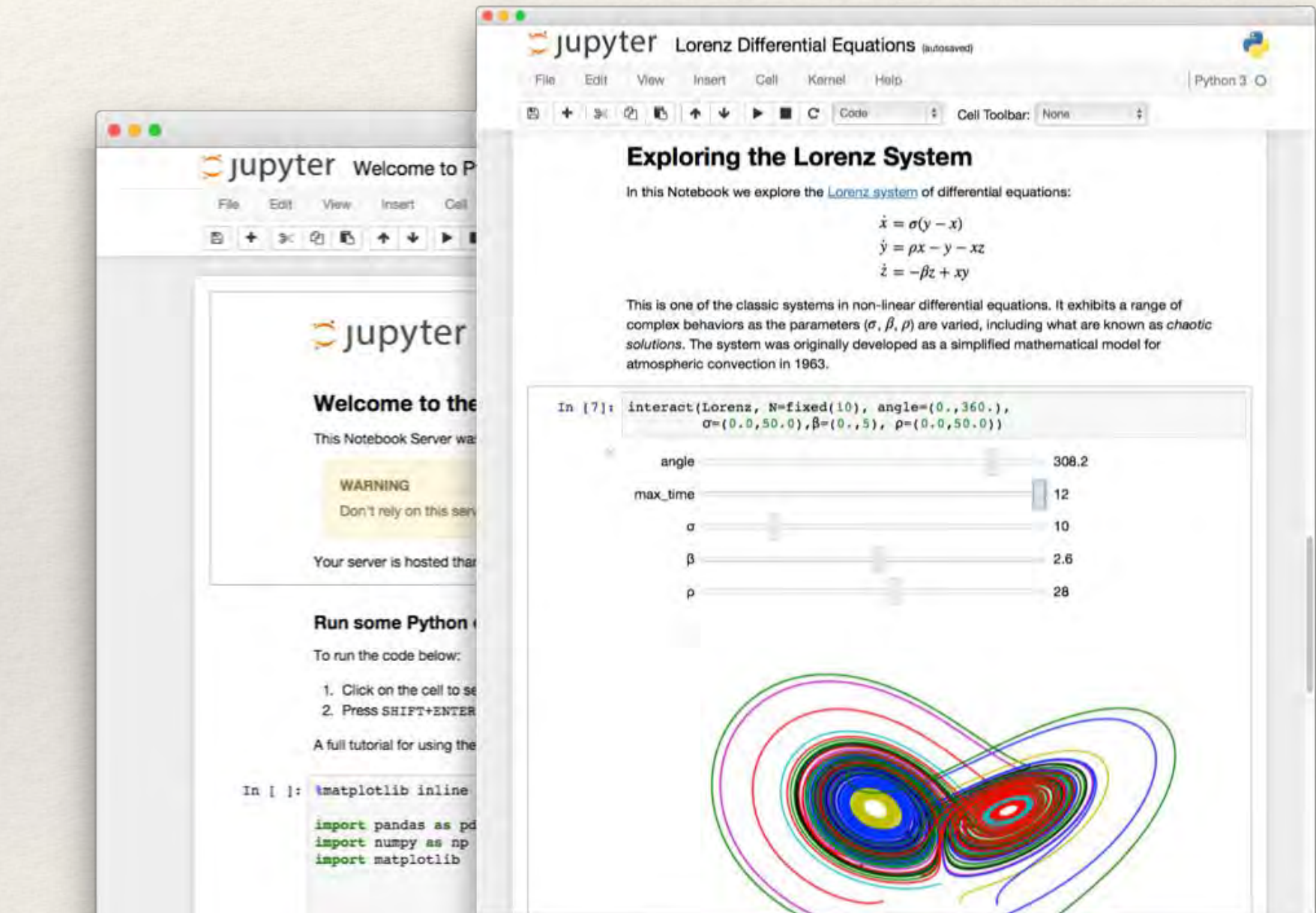
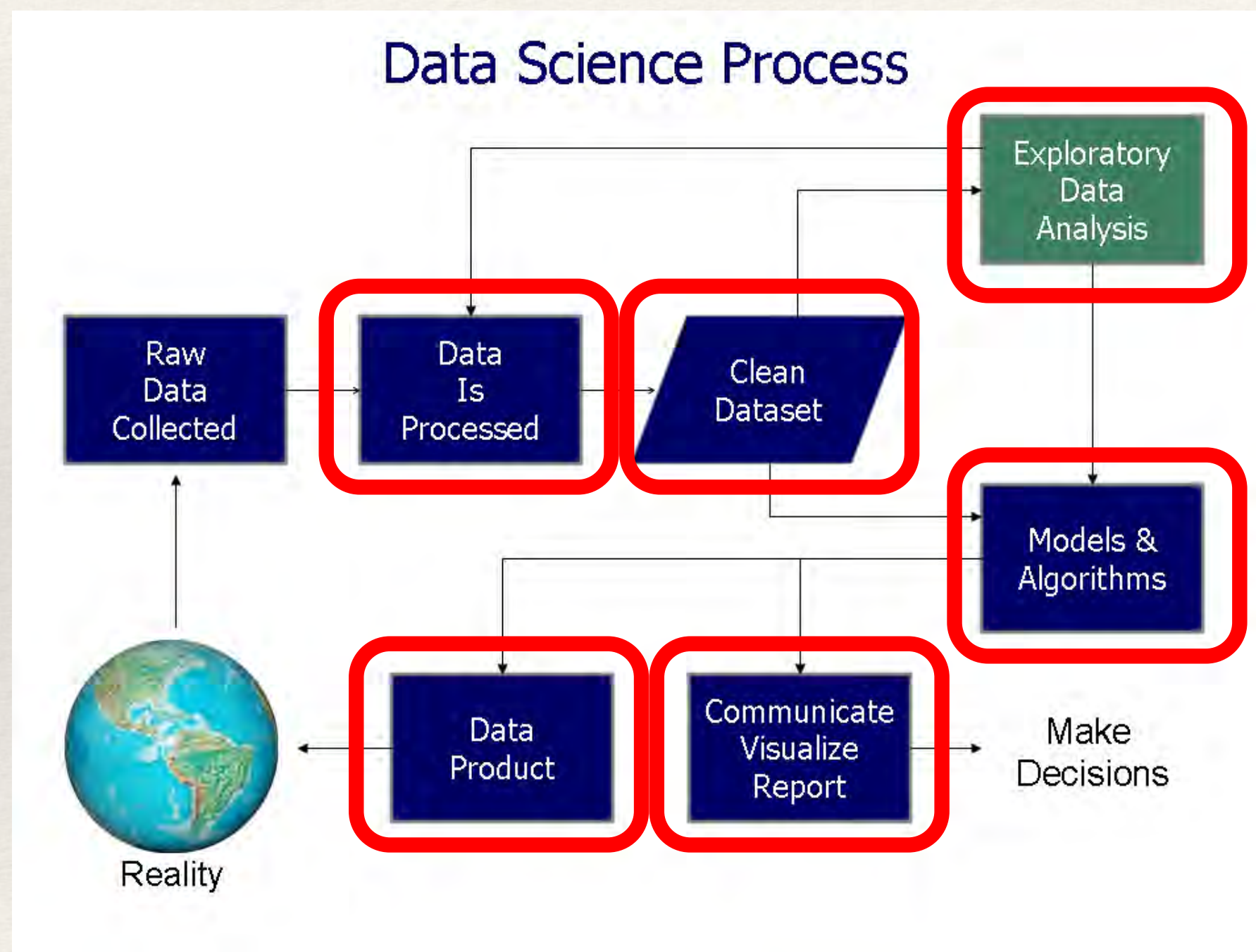
Plus ~ 1500 more Open source contributors!

What is Jupyter?

Tool for reproducible, shareable narratives, literate computing:

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

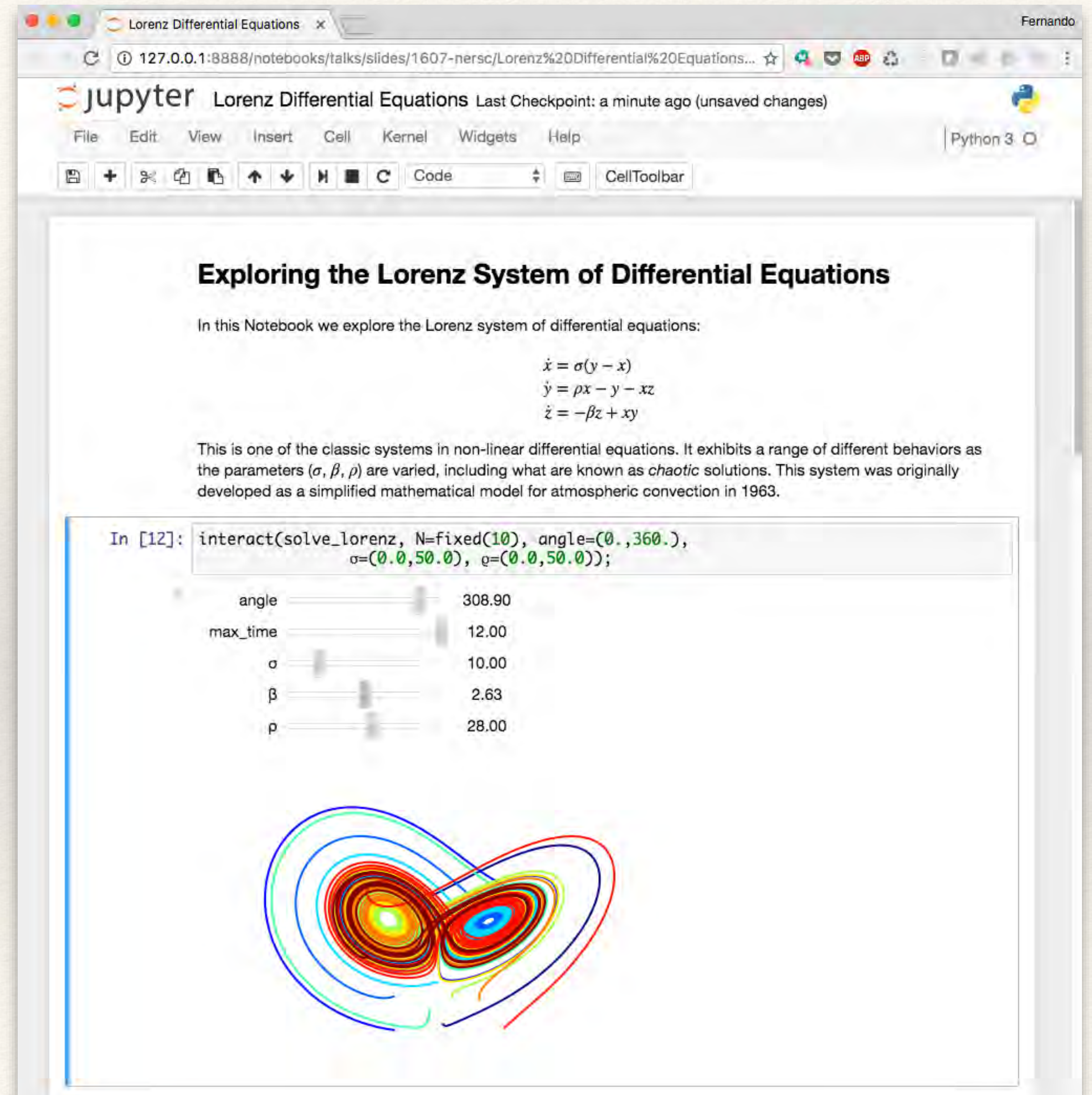
Rich text, interactive plots, equations, widgets, etc.



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

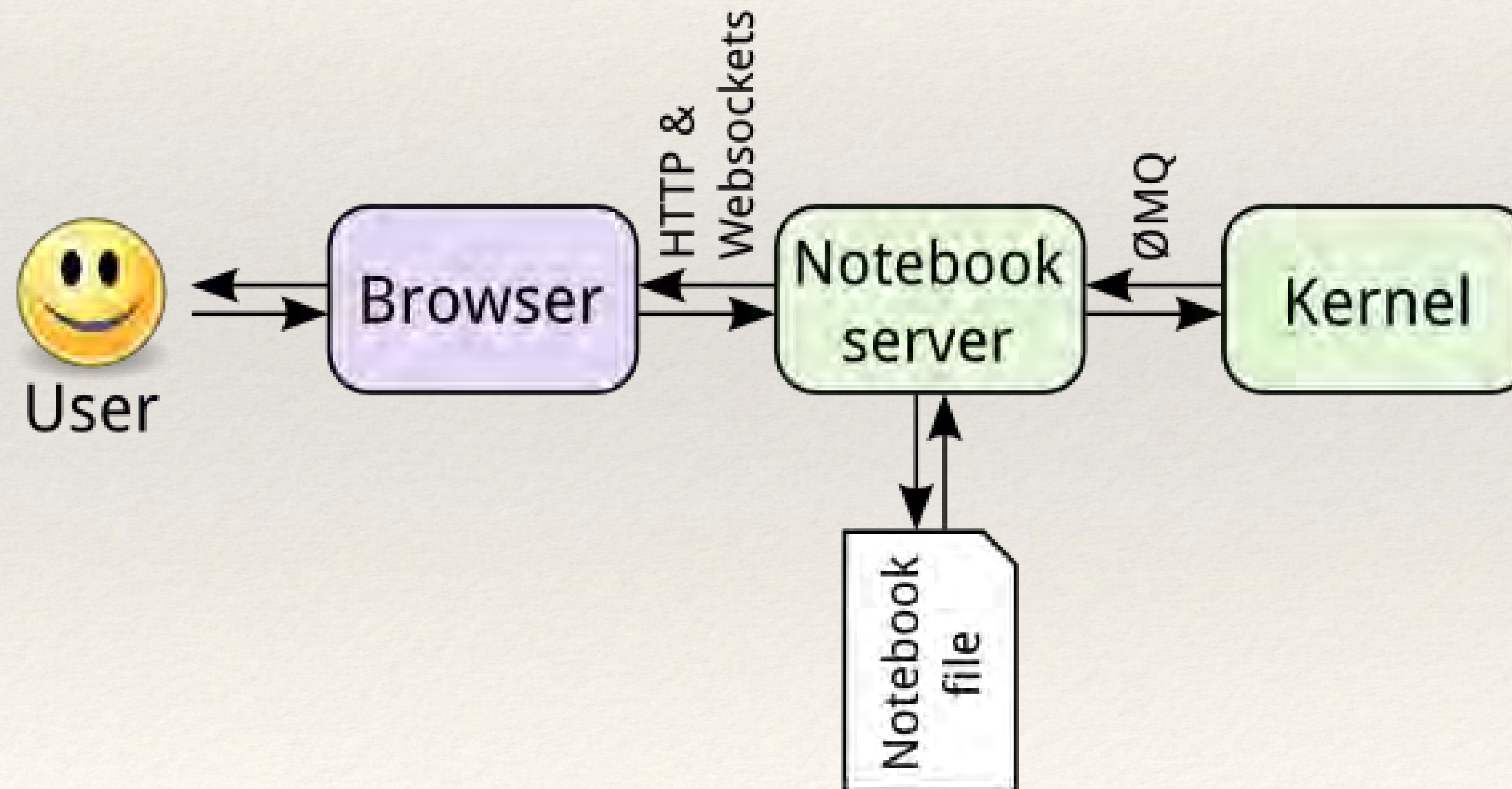
The IPython/Jupyter Notebook

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

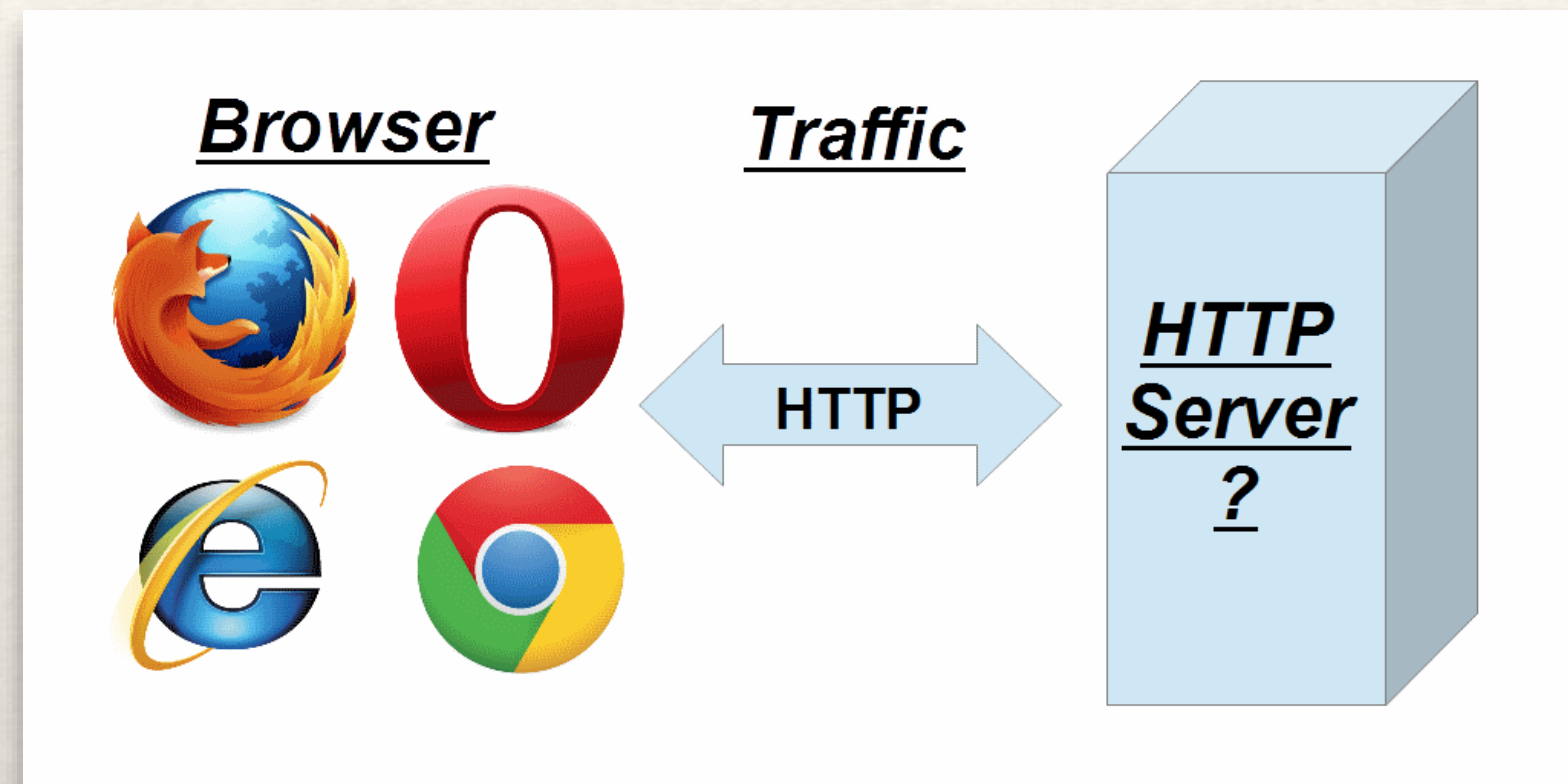


How It Works

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

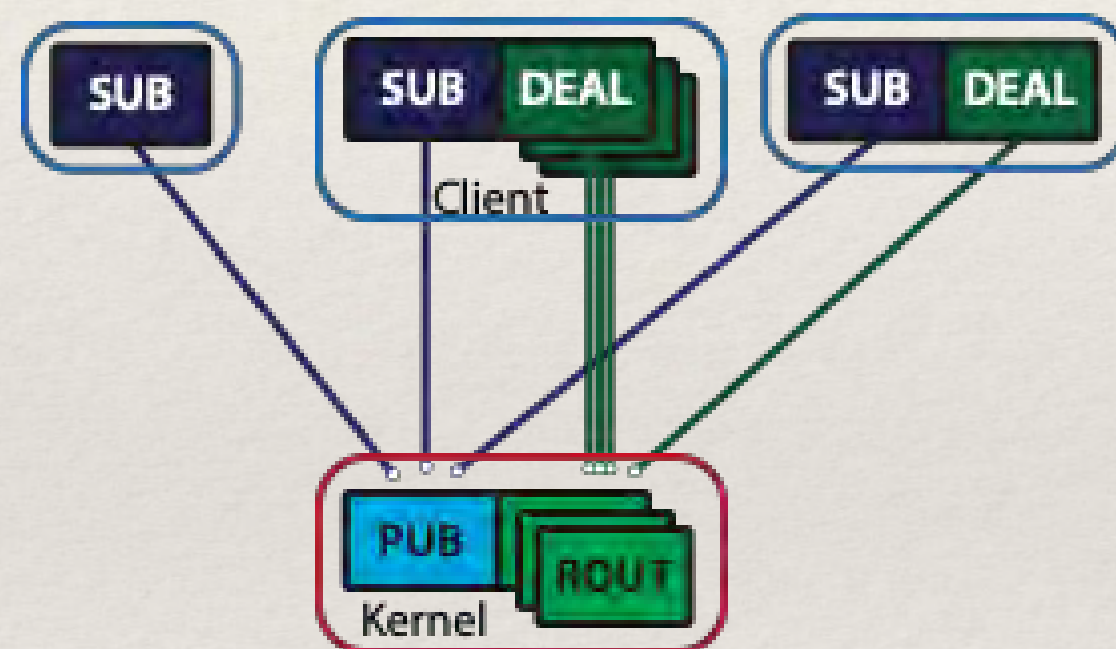
```
<div id="page-wrapper">
  <div id="main" class="clearfix">
    <div id="content" class="column" role="main"><div id="content-inner" class="clearfix">
      <a id="main-content"></a>
      <h1 class="title" id="page-title">Statistics at UC Berkeley</h1>
```



HTML: format to represent content
HyperText Markup Language

Core ideas of Jupyter

Interactive Computing Protocol



ØMQ + JSON

Document Format

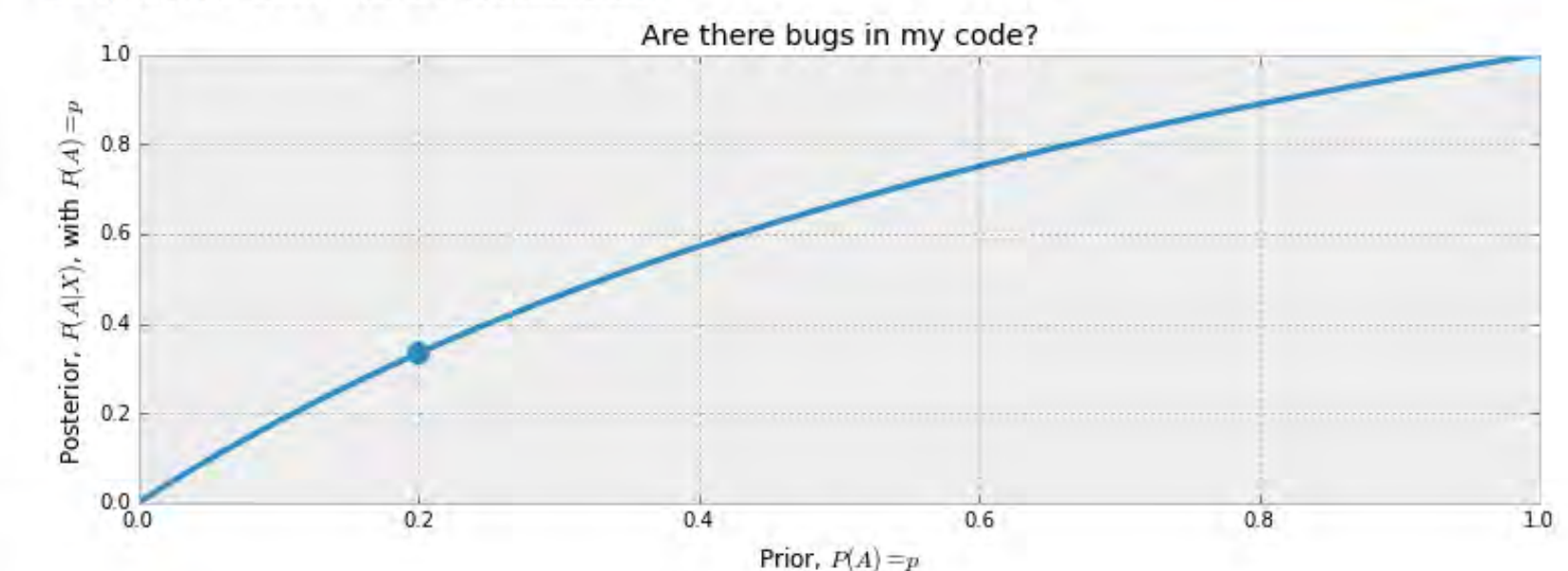
We have already computed $P(X|A)$ above. On the other hand, $P(X| \sim A)$ is subjective: our code can pass tests but still have a bug in it, though the probability there is a bug present is reduced. Note this is dependent on the number of tests performed, the degree of complication in the tests, etc.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

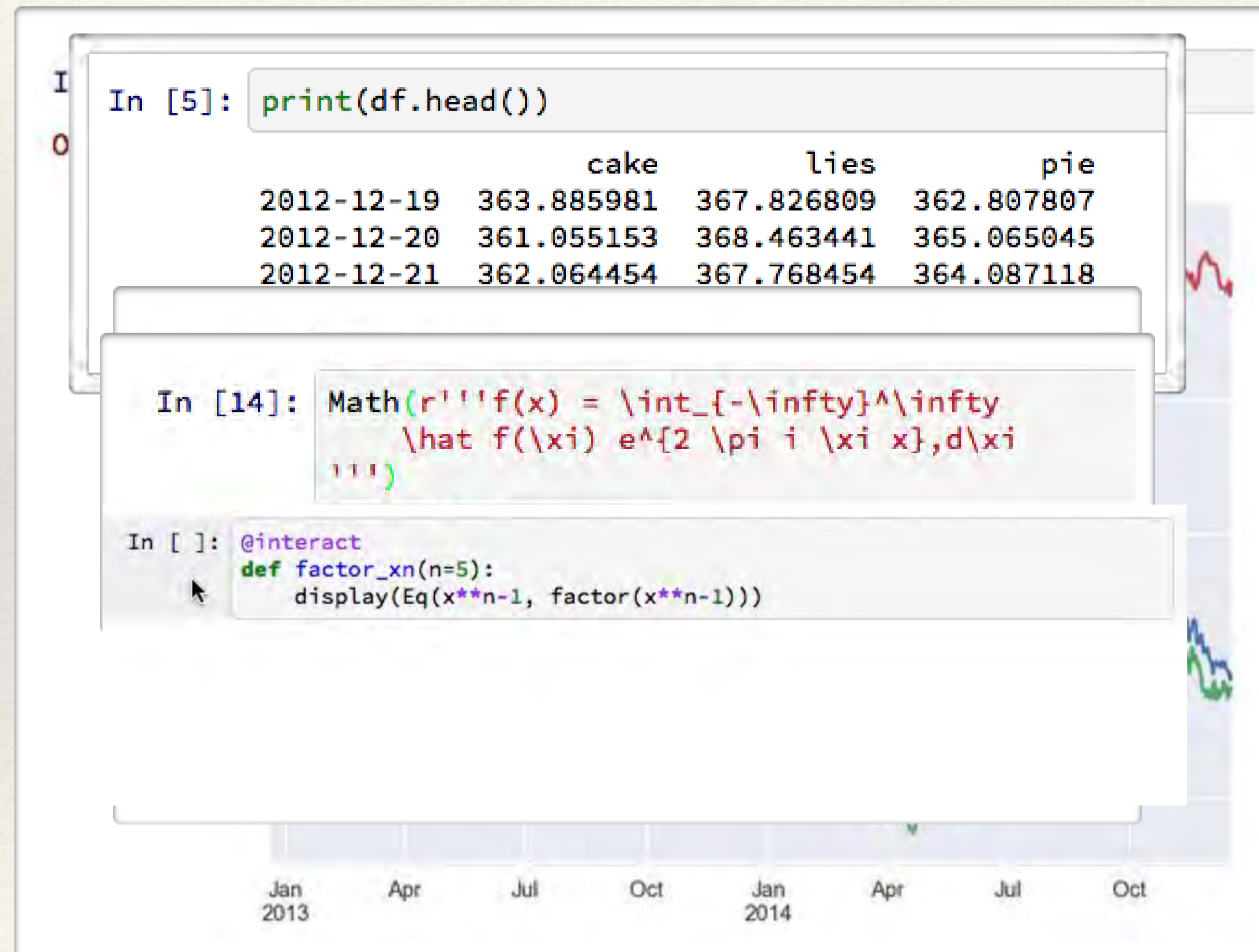


Jupyter Protocol

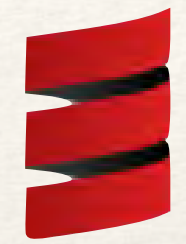
web-age capture of the process of interactive computing

any mime-type output

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



Jupyter Protocol is language agnostic

 **Scala**



 **julia**



Spark 



pythonTM IP[y]:
IPython



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



Tim Head

@betatim

Follow

Did you know there is a SPARQL kernel for @ProjectJupyter notebooks? I didn't! So much fun to be had querying wikidata ... try it: mybinder.org/v2/gh/betatim/ ...

What do US presidents die of?

```
In [1]: %endpoint http://query.wikidata.org/sparql
%display table
%show all

PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX wdt: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?president ?cause ?dob ?dod WHERE {
  ?pid wdt:P39 wdt:Q11696 .
  ?pid wdt:P509 ?cid .
  ?pid wdt:P569 ?dob .
  ?pid wdt:P570 ?dod .

  OPTIONAL {
    ?pid rdfs:label ?president filter (lang(?president) = "en")
  }
  OPTIONAL {
    ?cid rdfs:label ?cause filter (lang(?cause) = "en")
  }
}
```

Endpoint set to: <http://query.wikidata.org/sparql>

Display: table

Result maximum size: unlimited

president	cause	dob	dod
Benjamin Harrison	influenza	1833-08-20T00:00:00Z	1901-03-13T00:00:00Z
Ronald Reagan	Alzheimer's disease	1911-02-06T00:00:00Z	2004-06-06T00:00:00Z
Zachary Taylor	cholera	1784-11-24T00:00:00Z	1850-07-09T00:00:00Z
James K. Polk	cholera	1795-11-02T00:00:00Z	1849-06-15T00:00:00Z

7:46 AM - 15 Apr 2018



jupyter xtensor (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

C++14

Run

Markdown

Initialize a 2-D array and compute the sum of one of its rows and a 1-D array

```
In [1]: #include <iostream>
#include "xtensor/xarray.hpp"
#include "xtensor/xio.hpp"
```

```
In [2]: xt::xarray<double> arr1
        {{1.0, 2.0, 3.0},
         {2.0, 5.0, 7.0},
         {2.0, 5.0, 7.0}};

xt::xarray<double> arr2
        {5.0, 6.0, 7.0};

std::cout << xt::view(arr1, 1) + arr2;;
{ 7., 11., 14.}
```

Initialize a 1-D array and reshape it inplace

```
In [3]: #include <iostream>
#include "xtensor/xarray.hpp"
#include "xtensor/xio.hpp"
```

```
In [4]: xt::xarray<int> arr
        {1, 2, 3, 4, 5, 6, 7, 8, 9};

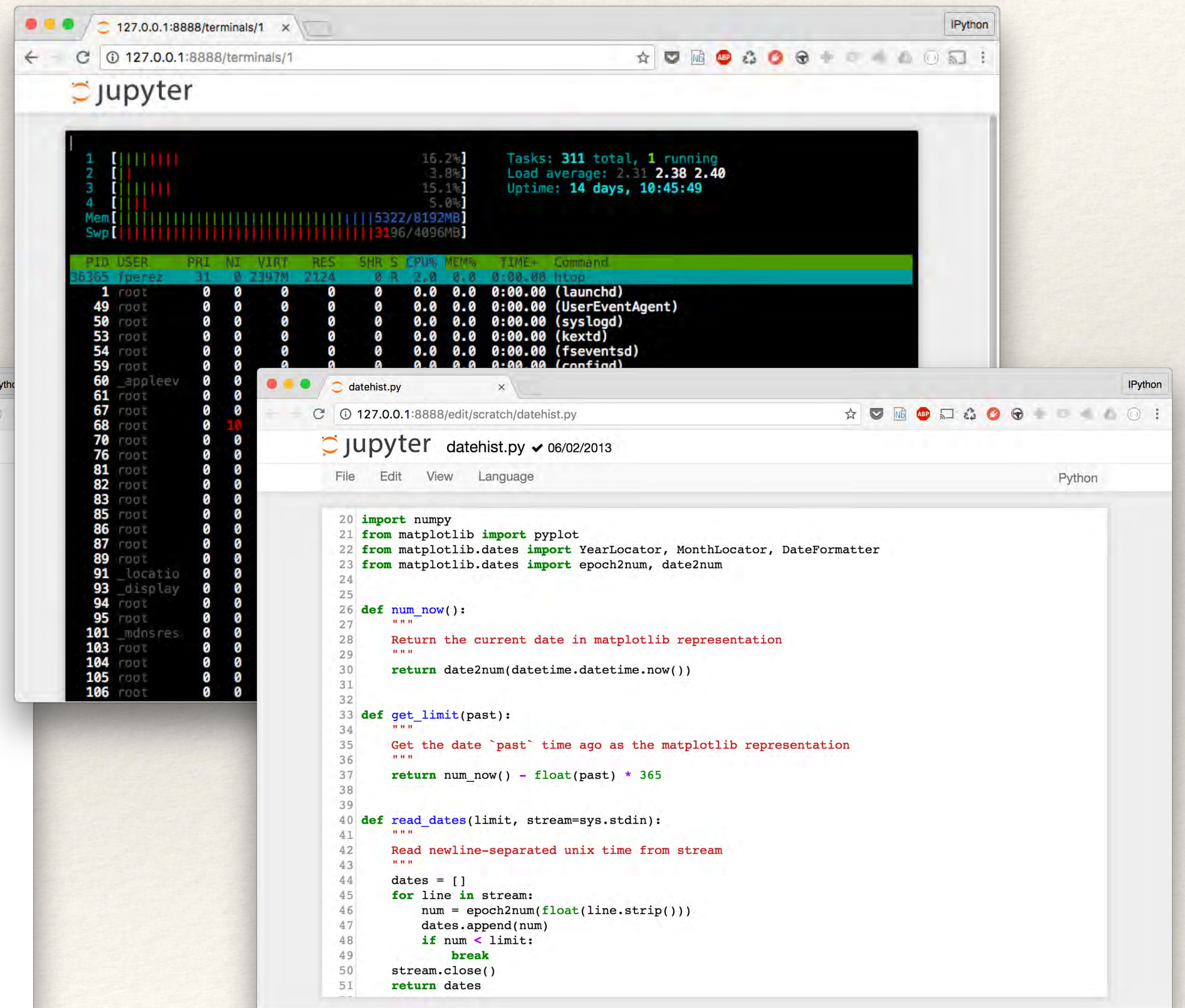
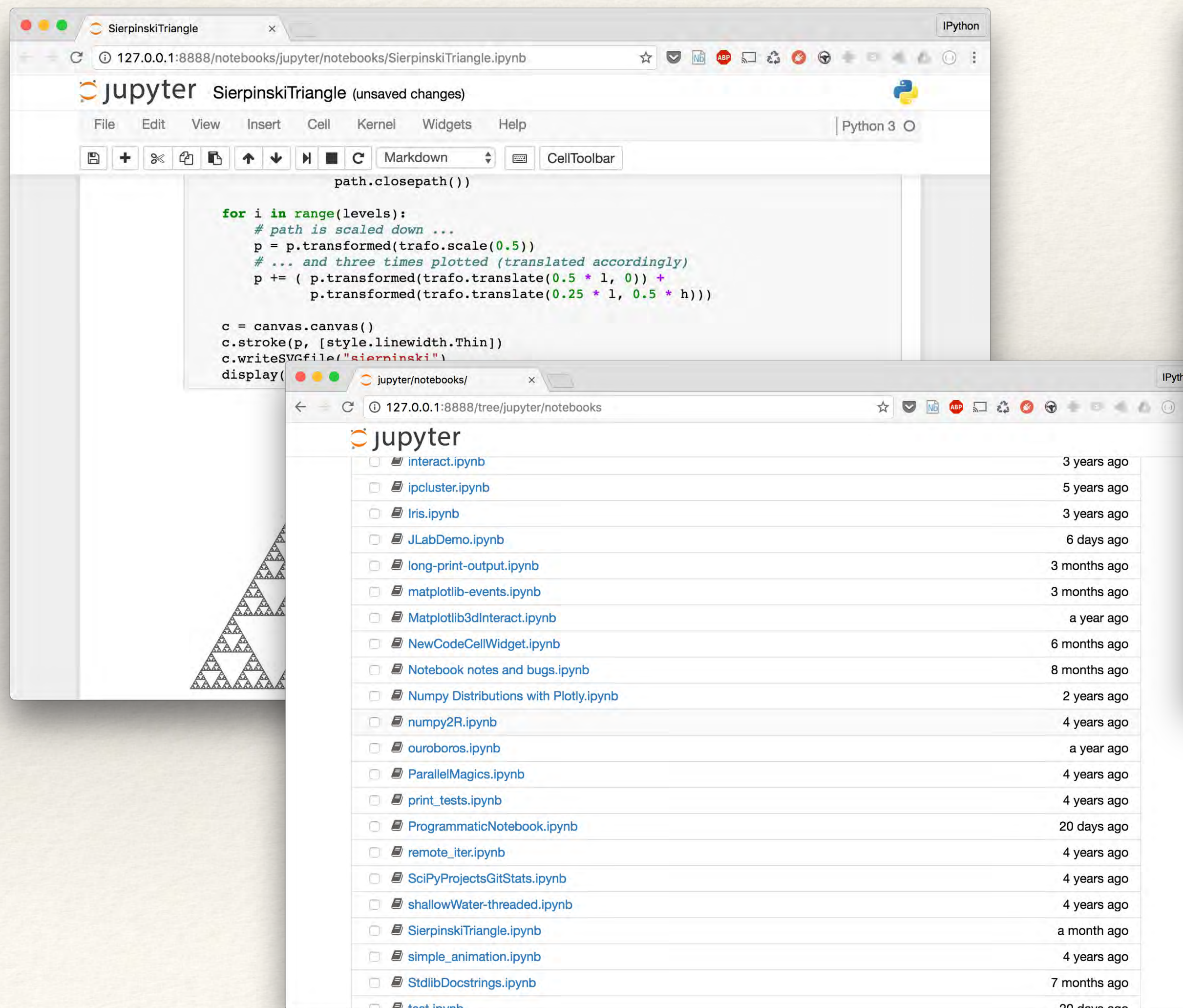
arr.reshape({3, 3});

std::cout << arr;

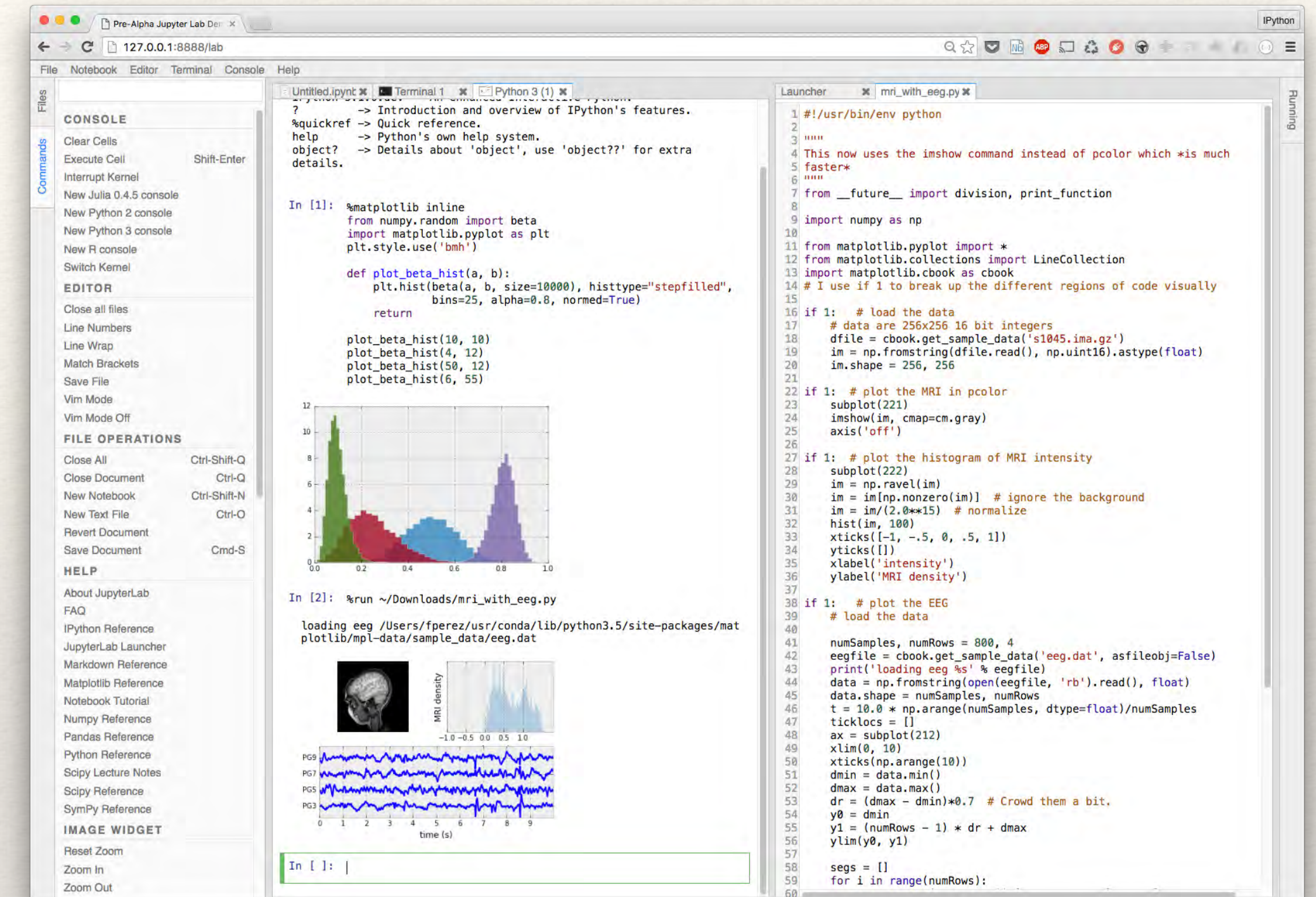
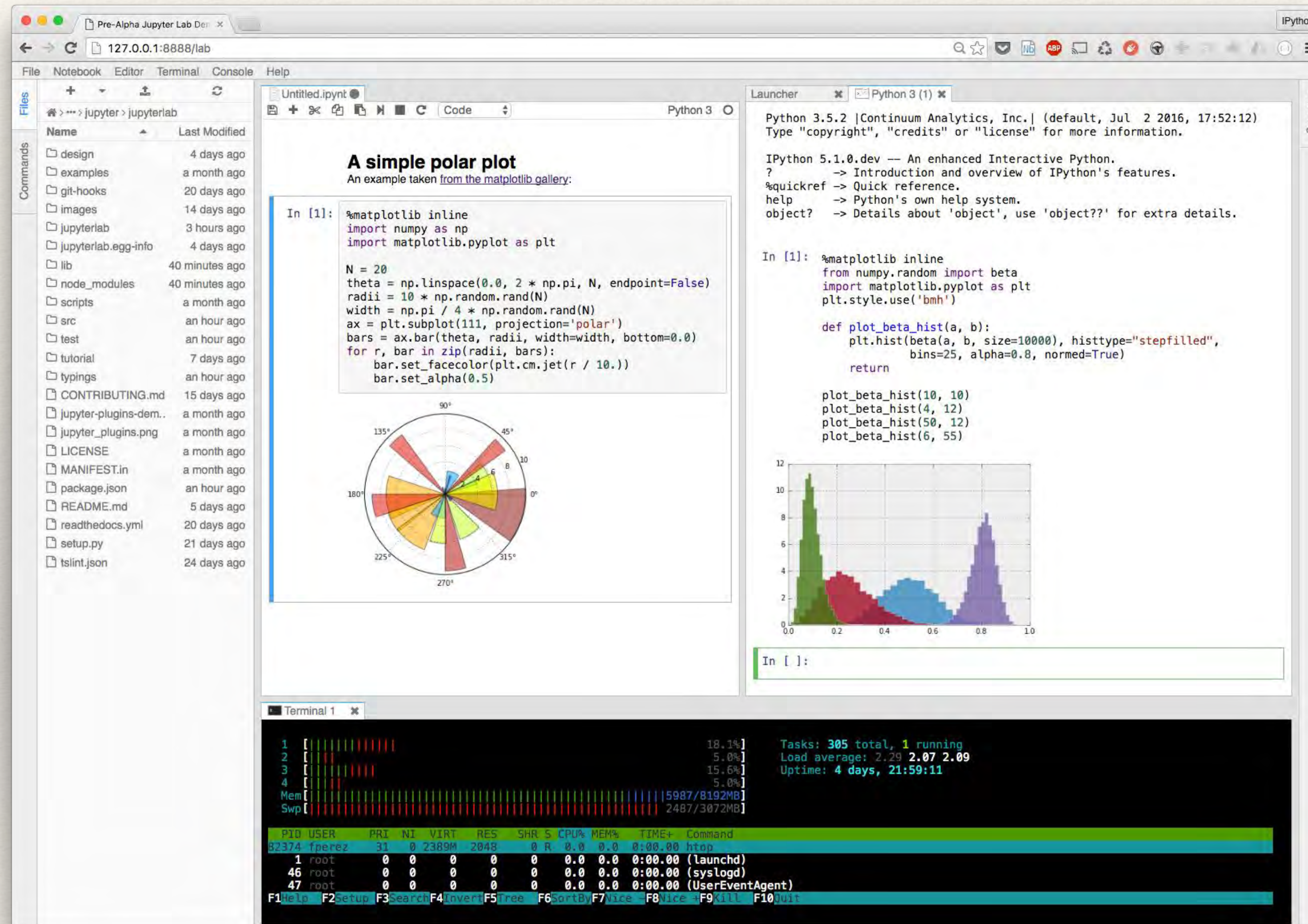
{{1, 2, 3},
 {4, 5, 6},
 {7, 8, 9}}
```



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 ...



Notebooks++

Files

Running

Commands

Cell Tools

Tabs

File Edit View Run Kernel Tabs Settings Help

Lorenz.ipynk ×

+

✂

📄

📄

▶

■

🔄

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:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

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

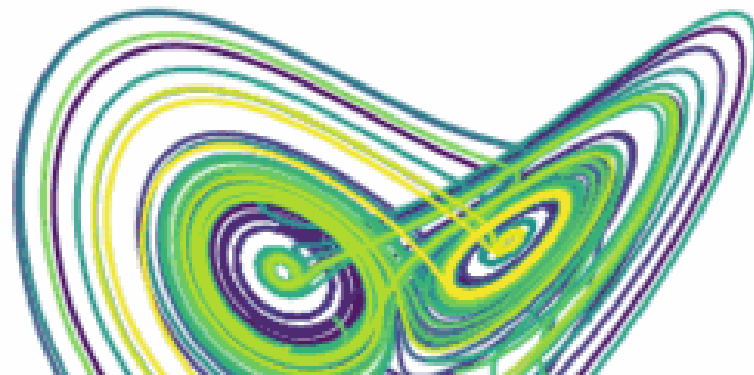
10.10

beta

2.63

rho

23.30



Output View ×

sigma

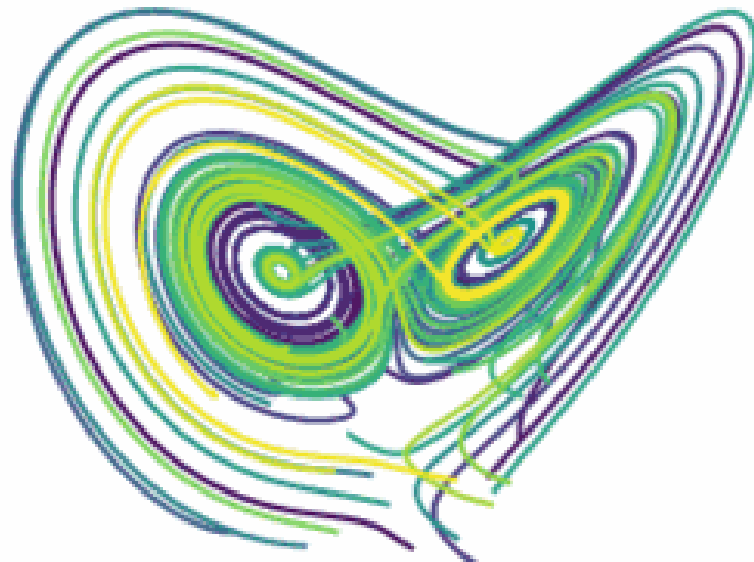
10.10

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:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

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(value=
```




Live Demo!

Beyond notebooks

File Edit View Run Kernel Tabs Settings Help

markdown_pyt x

```
7 # Including Python code
8
9 Here is a block of Python code in the markdown file:
10
11 ```python
12 a = 100
13 ```
14
15 Let's attach a Python 3 Kernel and Console to this markdown file. Then
16 we can select lines of code in the markdown file and run them in the
17 console by pressing `Shift+Enter`. Let's do something more complicated:
18
19 First import `matplotlib`, `numpy` and `pandas`, and create a data
20 frame:
21 ```python
22 %matplotlib inline
23 from matplotlib import pyplot as plt
24 from matplotlib import style
25 import numpy as np
26 import pandas as pd
27 data = {
28     'x': np.random.rand(100),
29     'y': np.random.rand(100),
30     'color': np.random.rand(100),
31     'size': 100.0*np.random.rand(100)
32 }
33 df = pd.DataFrame(data)
34 df.head()
35 ```
36
37 And make a scatter plot:
38
39 ```python
40 style.use('seaborn-whitegrid')
41
42 plt.scatter('x', 'y', c='color', s='size', data=df, cmap=plt.cm.Blues)
43 plt.xlabel('x')
44 plt.ylabel('y')
45
46 plt.title("The data that we collected")
47 ```
48
49 All of the Python objects are live in the console and can be explored
50 further.
```

Files

Running

Commands

Tab

markdown_pyt x

And make a scatter plot:

```
style.use('seaborn-whitegrid')

plt.scatter('x', 'y', c='color', s='size', data=df, cmap=plt.cm.Blues)
plt.xlabel('x')
plt.ylabel('y')

plt.title("The data that we collected")
```

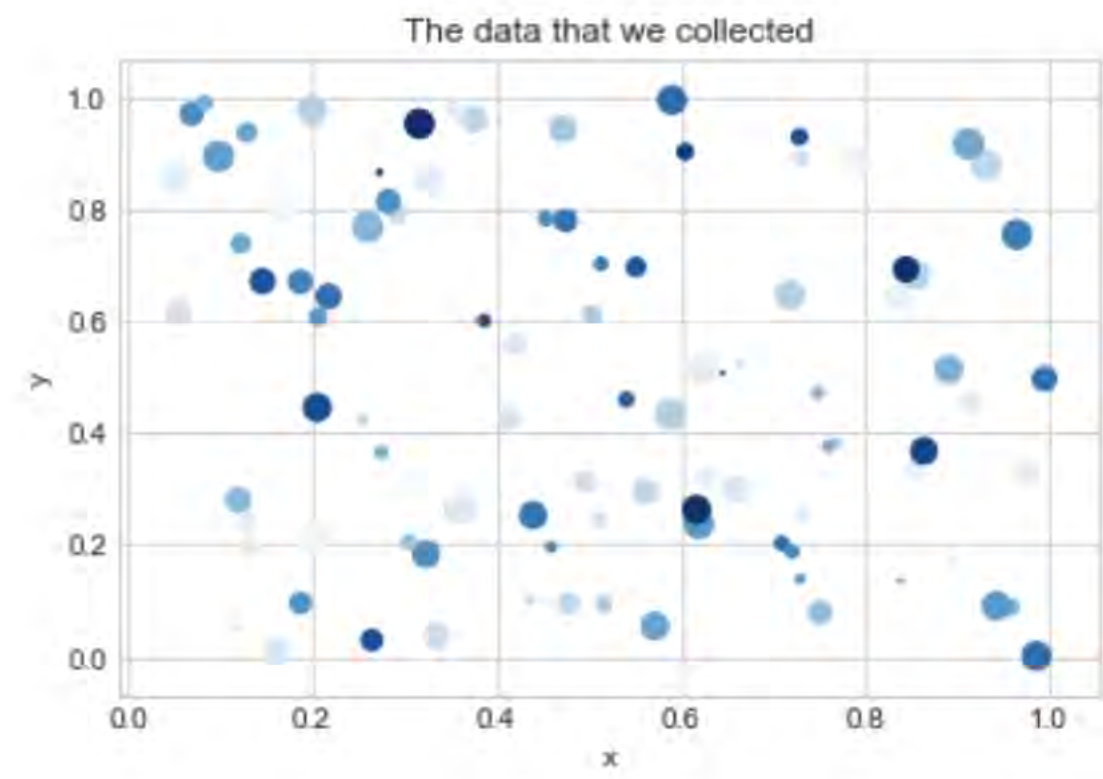
Console 1 x

```
In [3]: style.use('seaborn-whitegrid')

plt.scatter('x', 'y', c='color', s='size', data=df,
cmap=plt.cm.Blues)
plt.xlabel('x')
plt.ylabel('y')

plt.title("The data that we collected")

Out[3]: Text(0.5,1,'The data that we collected')
```



In []:

Data

File Edit View Run Kernel Tabs Settings Help

Launcher

1024px-Hubble

Files

Running

Commands

Tabs



iris.csv

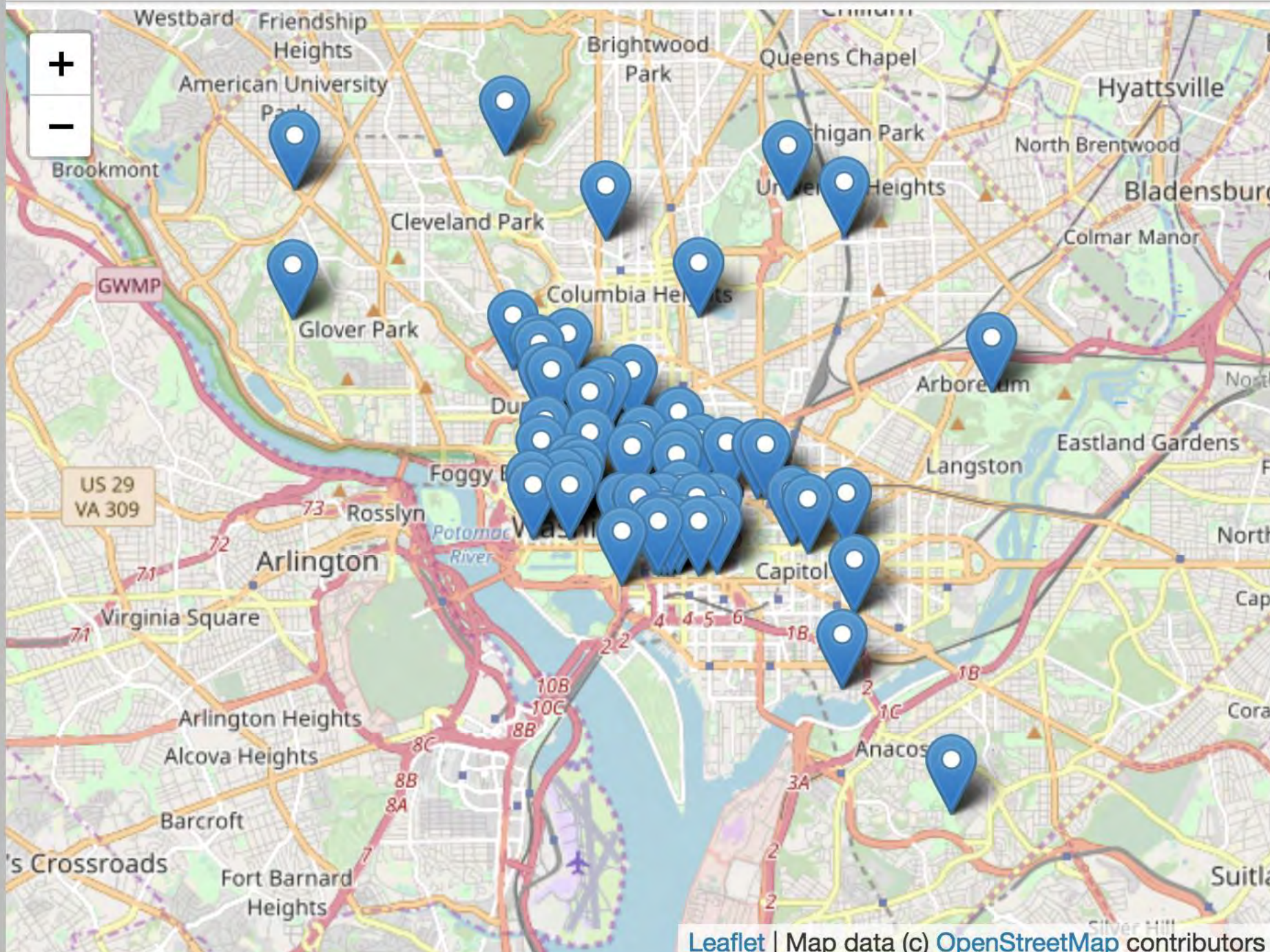
Delimiter: ,

	sepal_len	sepal_wid	petal_len	petal_wid	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
7	4.6	3.4	1.4	0.3	setosa

Museums_in_D

```
1 [{"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 FOUNDATION", "ADDRESS_ID": 241060, "LEGALNAME": "U.S. DEPARTMENT OF THE"}]
```

Museums_in_C

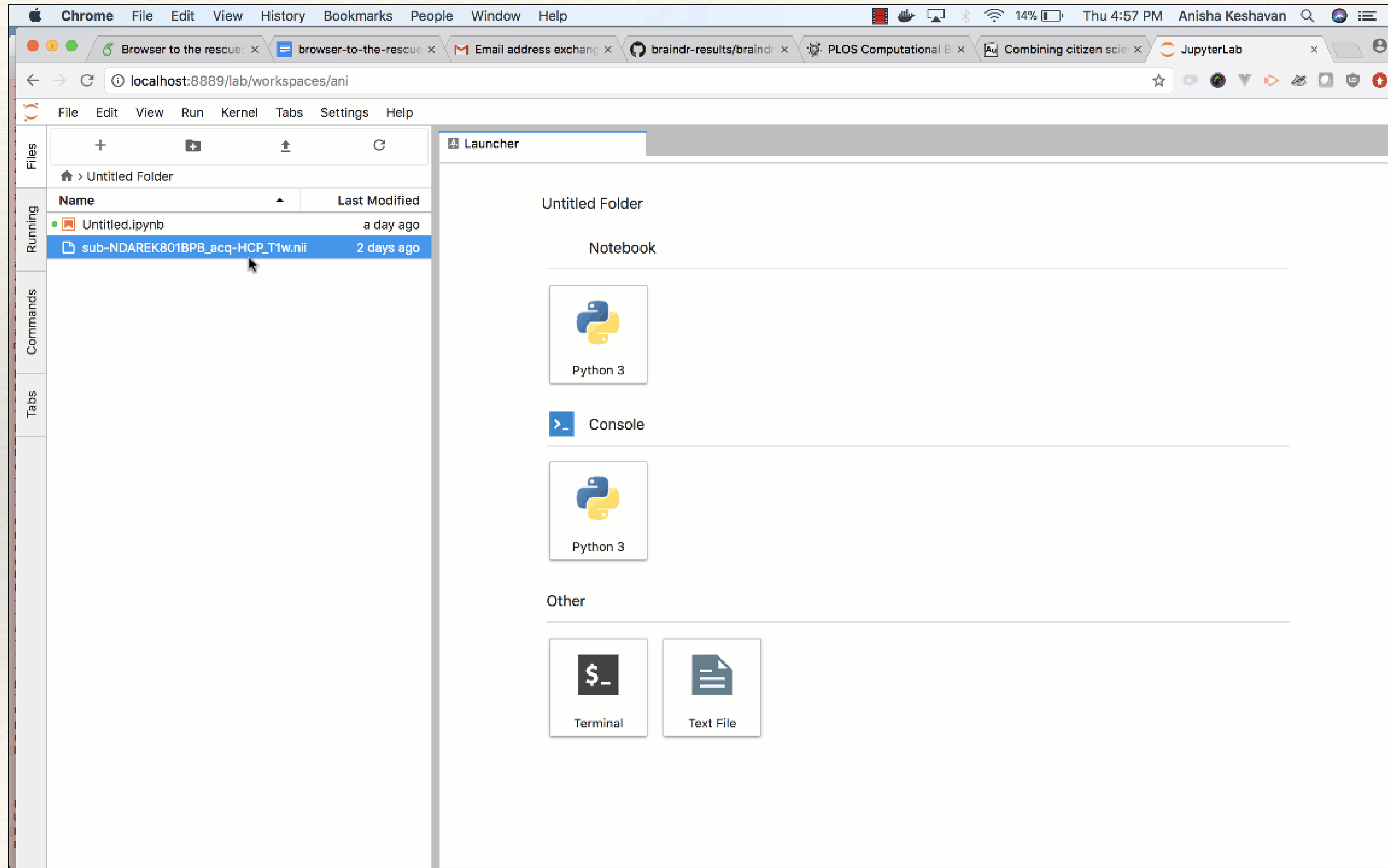


Leaflet | Map data (c) OpenStreetMap contributors

Data: reusable within code

[illegible]

Community-extensible: Neurohackademy 2018



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.



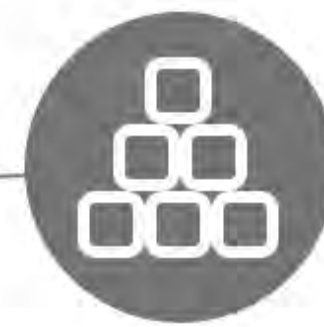
Pluggable authentication

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



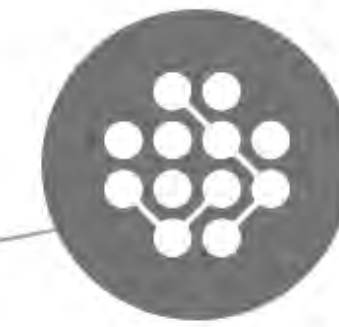
Centralized deployment

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



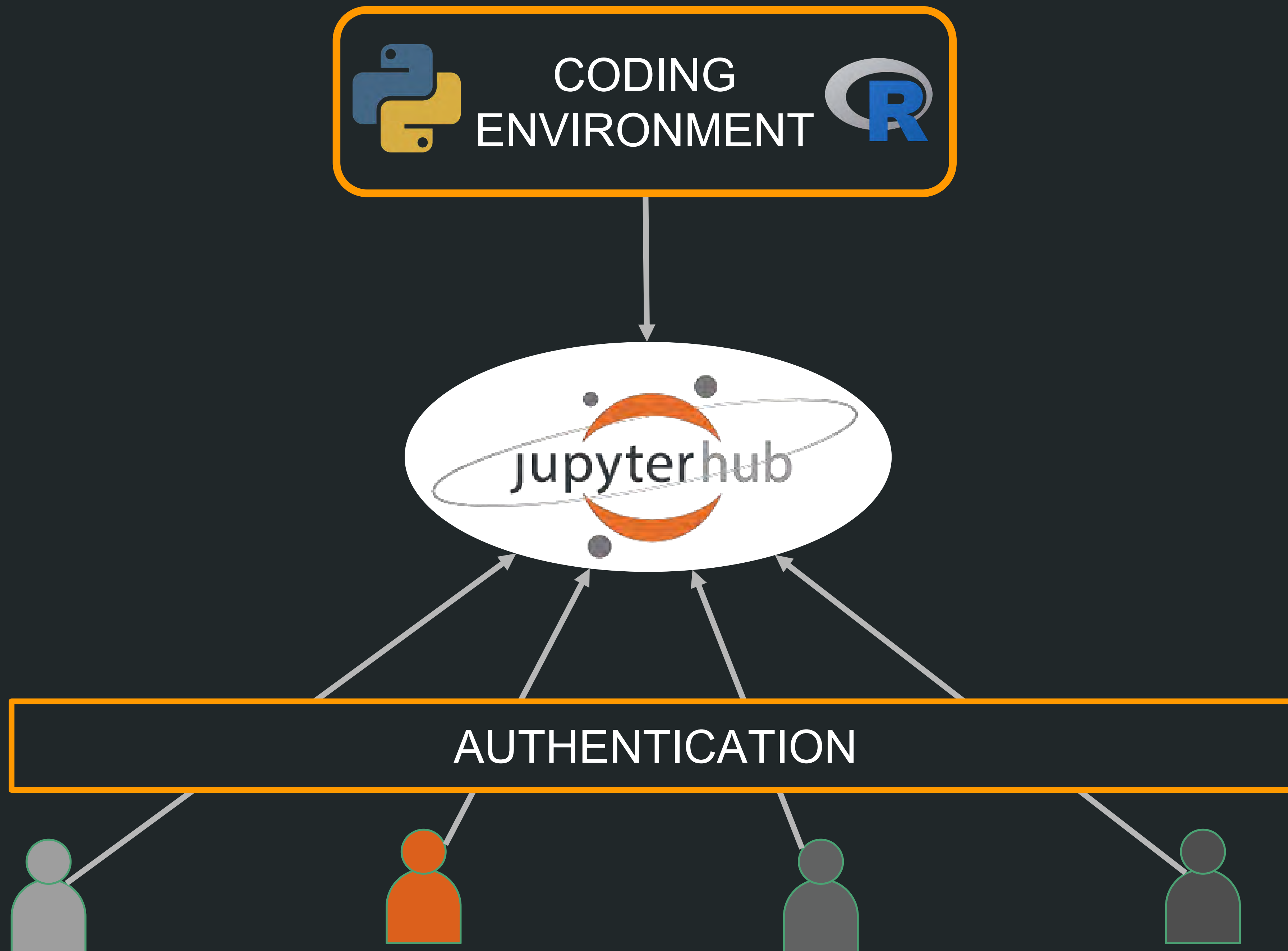
Container friendly

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



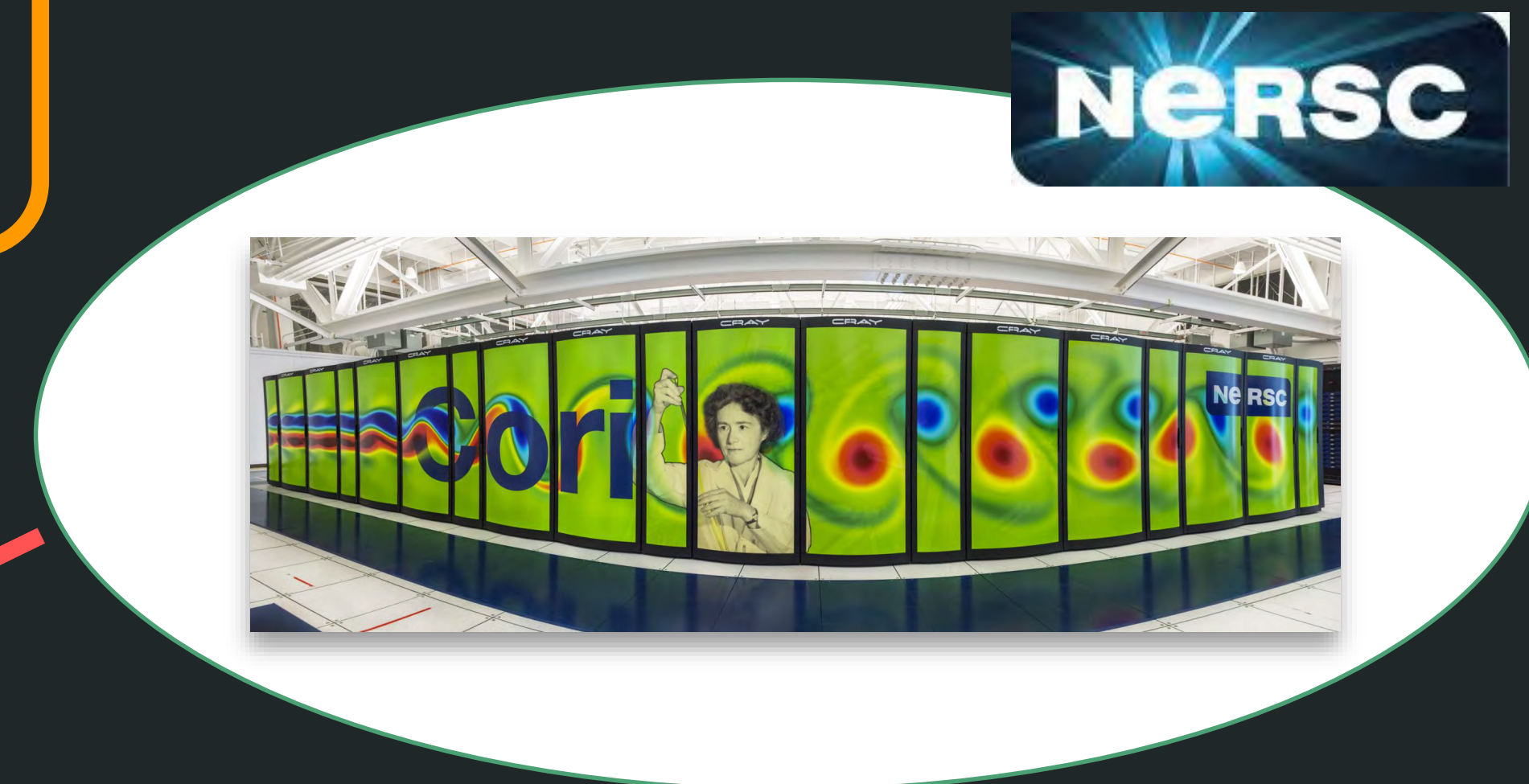
Code meets data

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



What does this mean for science + education?

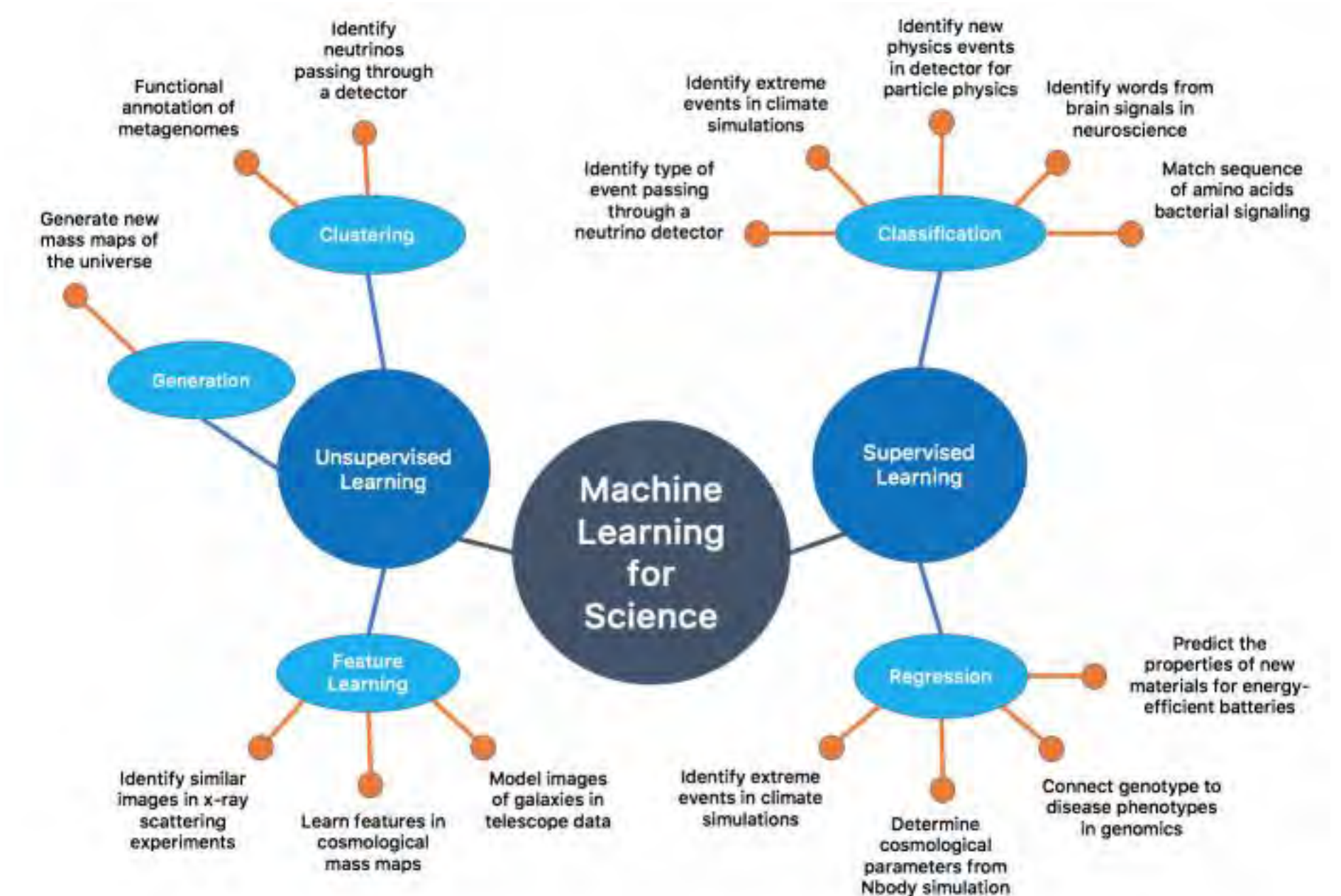
- ❖ Can utilize...
 - ❖ ...shared **hardware/compute** for running code
 - ❖ ...shared **data storage** for big datasets
 - ❖ ...shared **environments** for doing work
 - ❖ ...shared **workflows**, ideas, and results



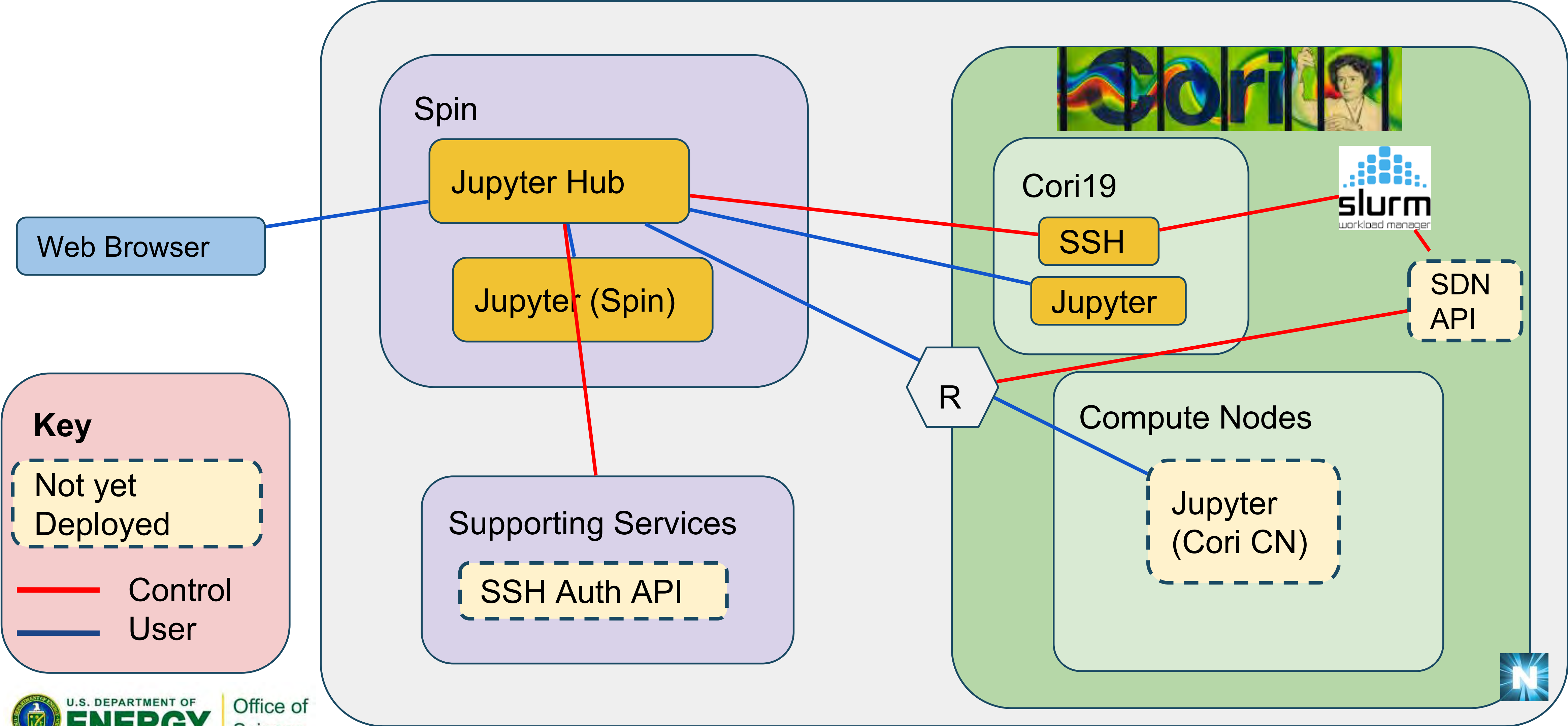
FANCY HARDWARE



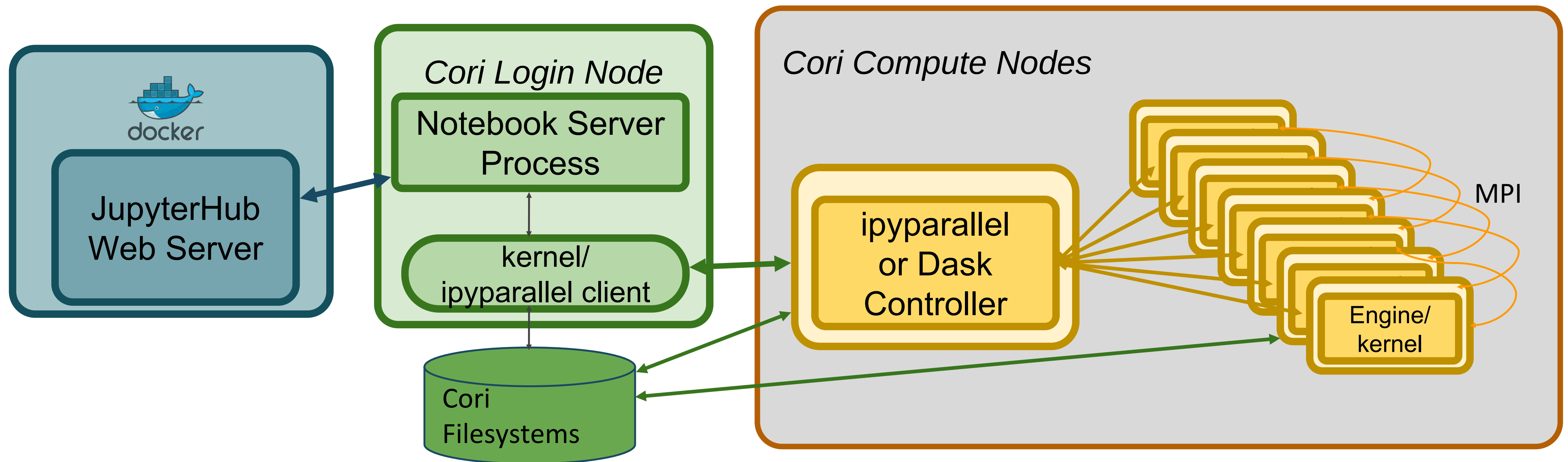
- **Mission HPC center for US Dept. of Energy**
 - 7000+ diverse users across science (e.g. cosmology, climate, biosciences)
- **Cori – Cray XC40 (31.4 PF Peak)**
 - 9668 Intel Knights Landing (KNL), 2388 Haswell nodes
- **Deep learning: Data and analytics (DAS) group:**
 - [Tools for machine learning](#); 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.)



Jupyter at NERSC Architecture



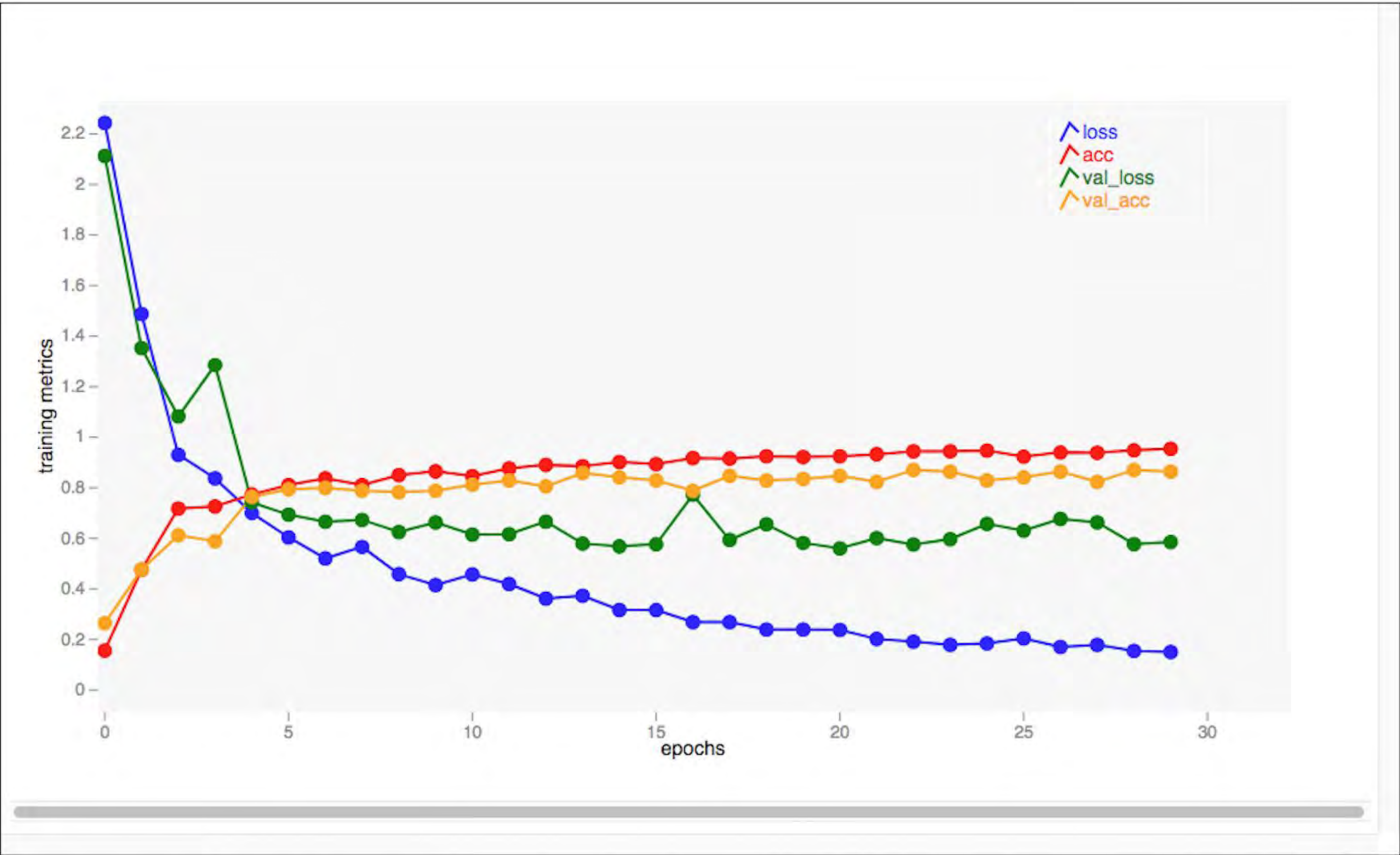
Parallel Jobs via Jupyter



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	Adadelata	0.157987...	0.579903...	0.944578...	0.870588...
1	Ended Tra...	31	4	8	16	0.03825	Adadelata	0.151153...	0.585079...	0.954216...	0.864705...

DistWidgetHPO.rpv

https://jupyter-dev.nersc.gov/user/nhanders/notebooks/global/nomes/m/hand...

Search

jupyter DistWidgetHPO.rpv Last Checkpoint: Last Thursday at 3:44 PM (autosaved)

Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | Set type

Run Code Edit Presentation Show Presentation

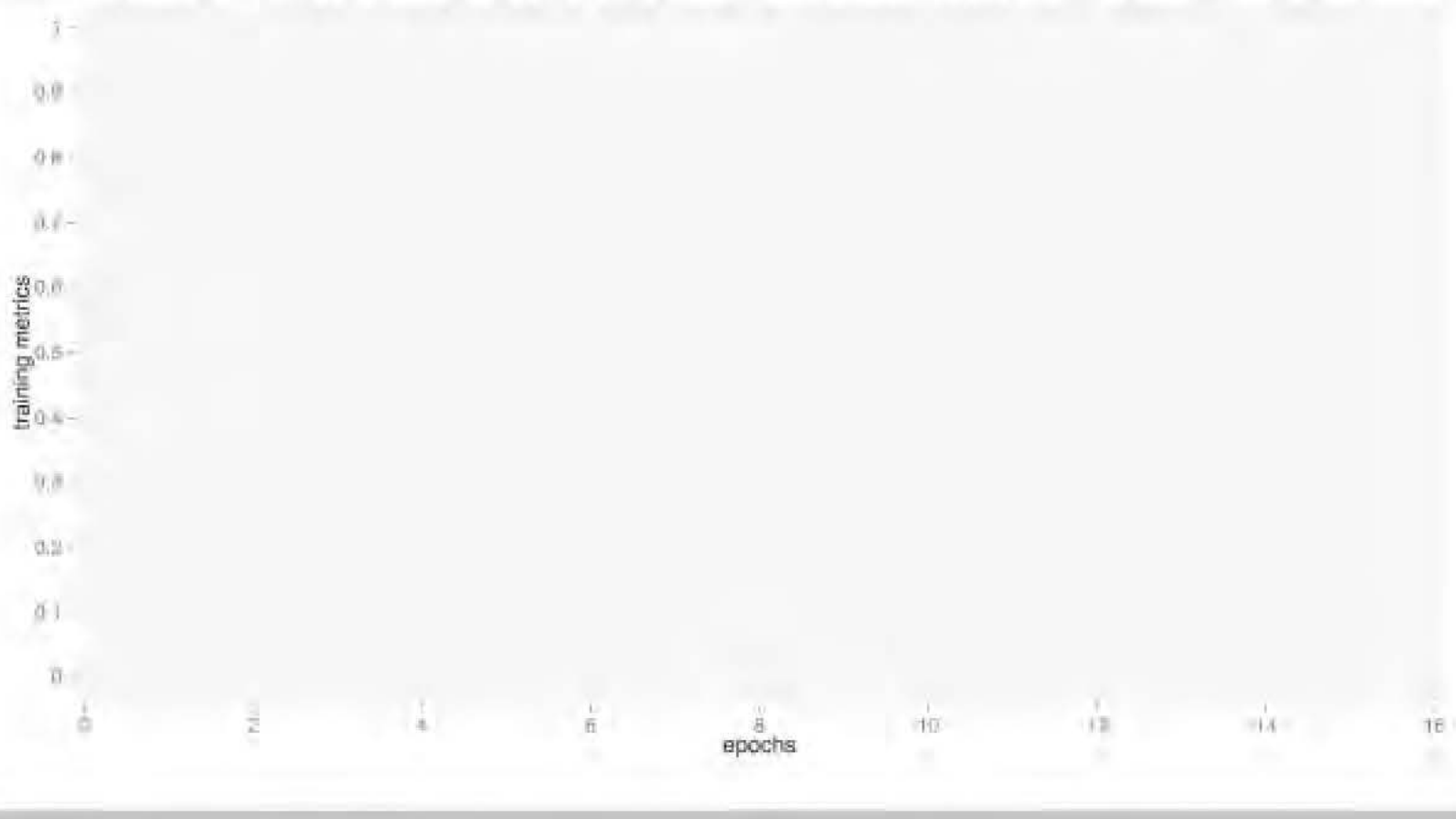
```
conv_sizes=conv_sizes,
fc_sizes=fc_sizes,
dropout=dropout,
optimizer=optimizer,
lr=lr,
)

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

paw.submit_computations()

paw
```

Model 0: {'conv_sizes': [64, 8, 128], 'fc_sizes': [256], 'dropout': 0.8464086724711278, 'optimizer': 'Nadam', 'lr': 0.0001}



Stop selected Restart selected

index	status	epoch	conv_size	fc_size	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.*"

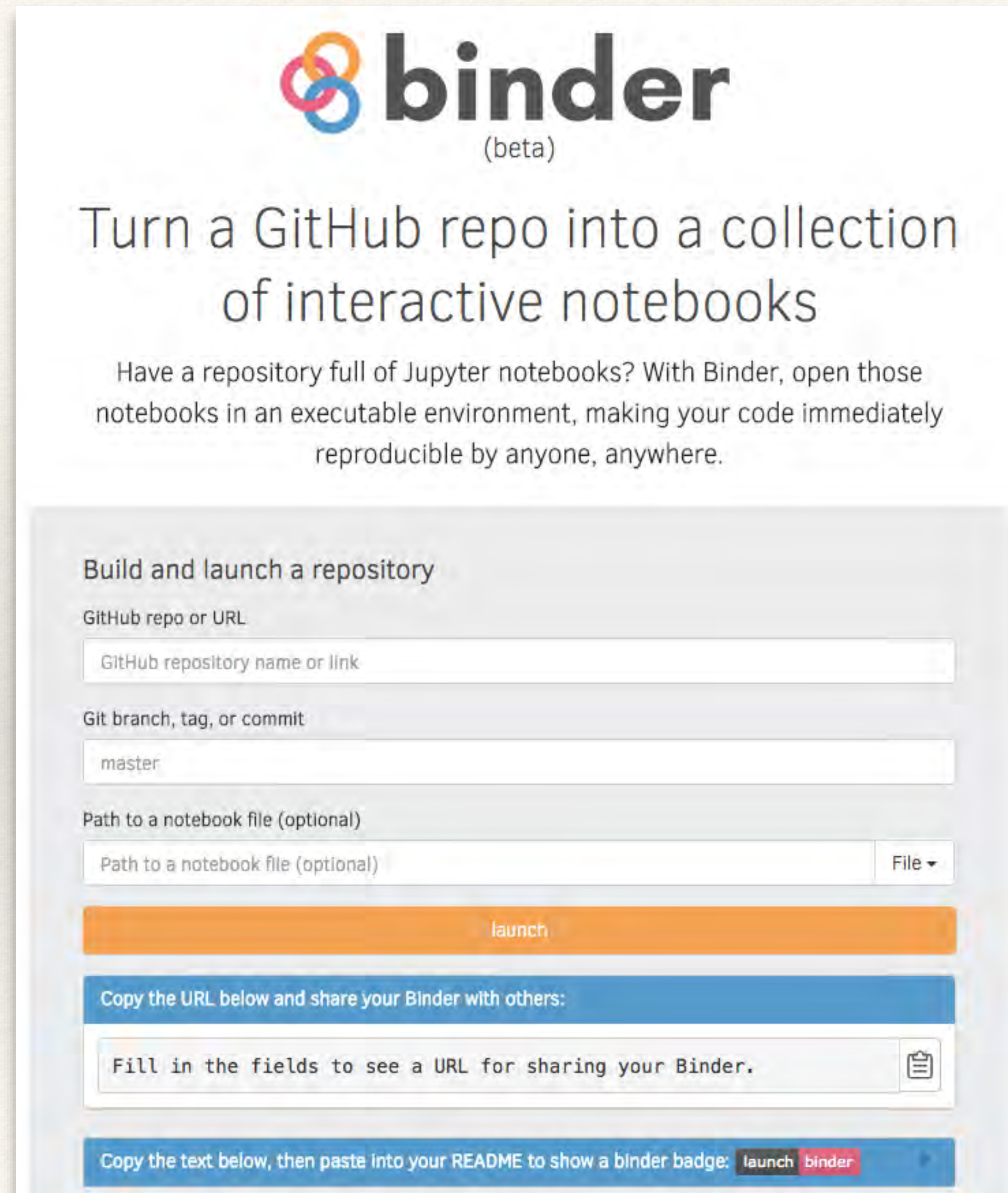
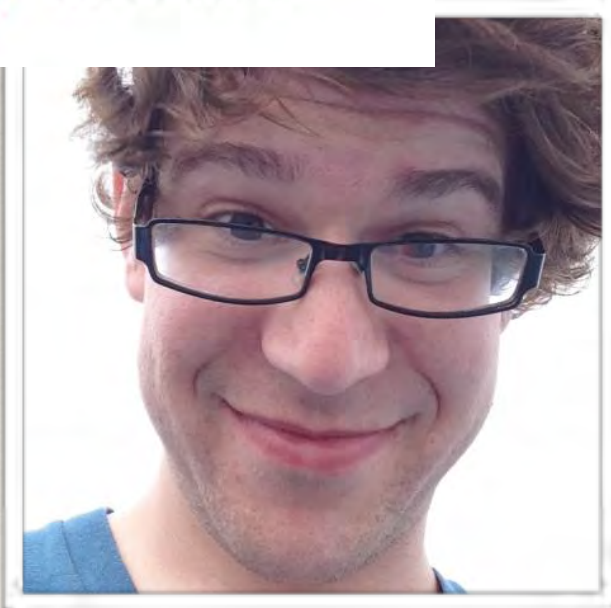
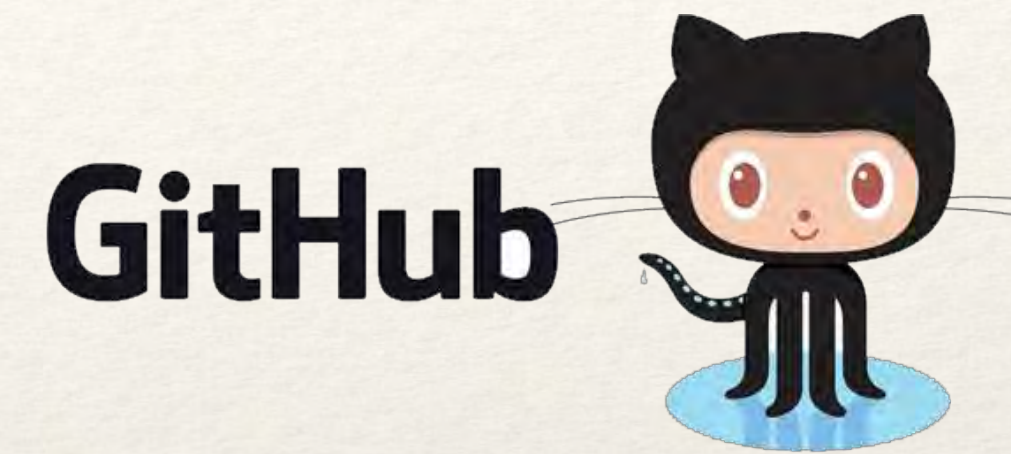
Reproducible Research

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.

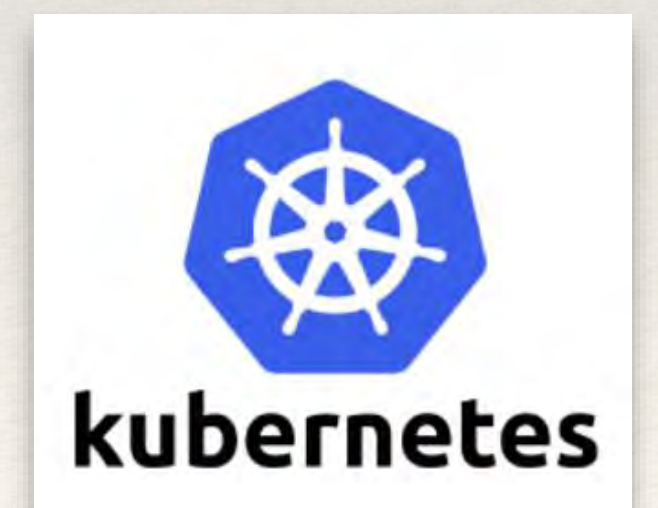
*Buckheit and Donoho, WaveLab and Reproducible Research, **1995***

mybinder.org: shareable reproducibility

Origins

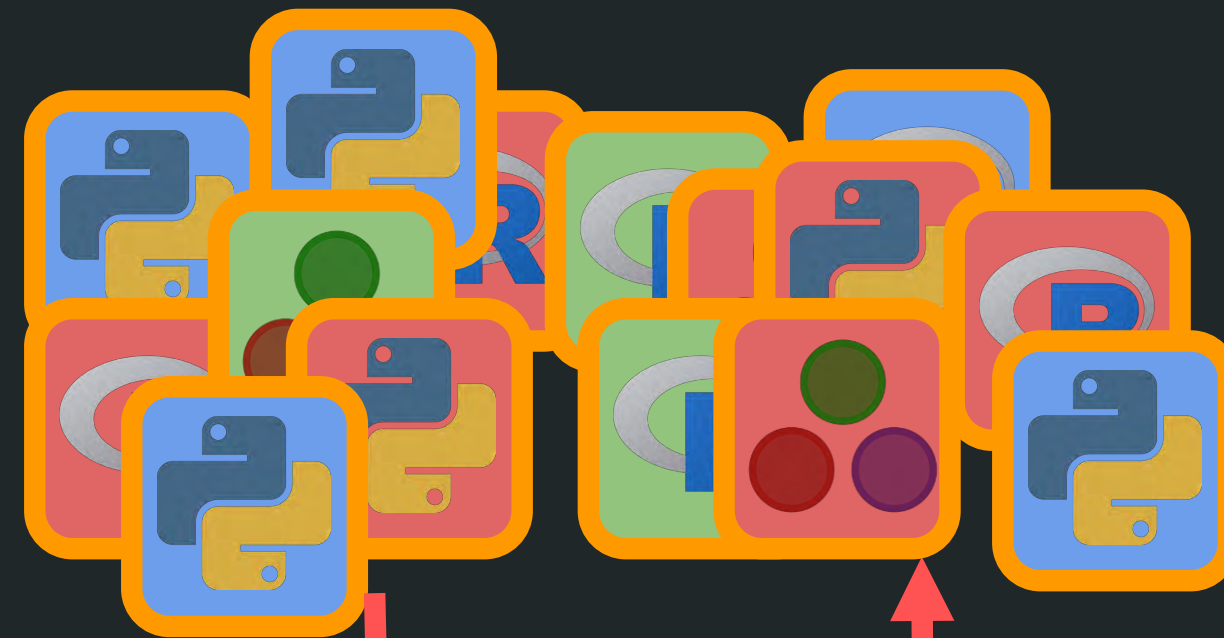
A screenshot of the Binder (beta) website. The header shows the Binder logo and the text 'Turn a GitHub repo into a collection of interactive notebooks'. Below this is a paragraph explaining that Binder allows users to open Jupyter notebooks in an executable environment. The main section is titled 'Build and launch a repository' and contains three input fields: 'GitHub repo or URL' (with a placeholder 'GitHub repository name or link'), 'Git branch, tag, or commit' (with a placeholder 'master'), and 'Path to a notebook file (optional)' (with a placeholder 'Path to a notebook file (optional)' and a 'File' dropdown menu). Below these fields is an orange 'launch' button. Underneath the button is a blue box with the text 'Copy the URL below and share your Binder with others:' and a text input field with the placeholder 'Fill in the fields to see a URL for sharing your Binder.' and a clipboard icon. At the bottom is another blue box with the text 'Copy the text below, then paste into your README to show a binder badge:' followed by 'launch' and 'binder' buttons.

Explicit Dependencies

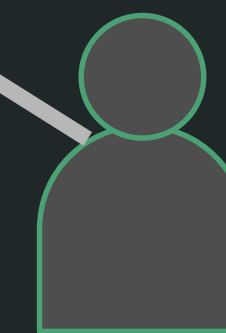
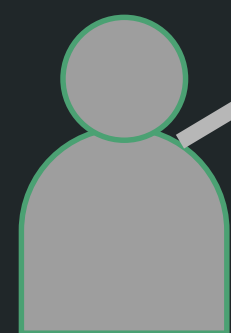


BinderHub

ON-DEMAND
ENVIRONMENTS



CONTENT ON
THE WEB



repo2docker

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



repo2docker.readthedocs.io



Identify requirements

repo2docker
*what does it
do?*

 choldgraf Update README.md		
 README.md		Update README.md
 index.ipynb		first move
 requirements.txt		Pin requirements.txt to values that actually exist



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-c
-r "requirements.txt"

...



build (& push) image

repo2docker
*what does it
do?*

```
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 &&
```

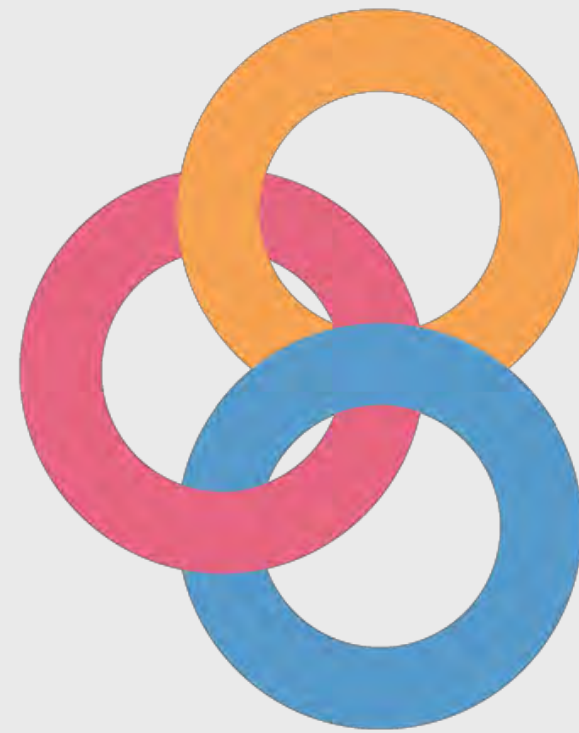
```
apt-get purge &&          apt-get clean &&          rm -rf
```

```
/var/lib/apt/lists/*
```



BinderHub

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



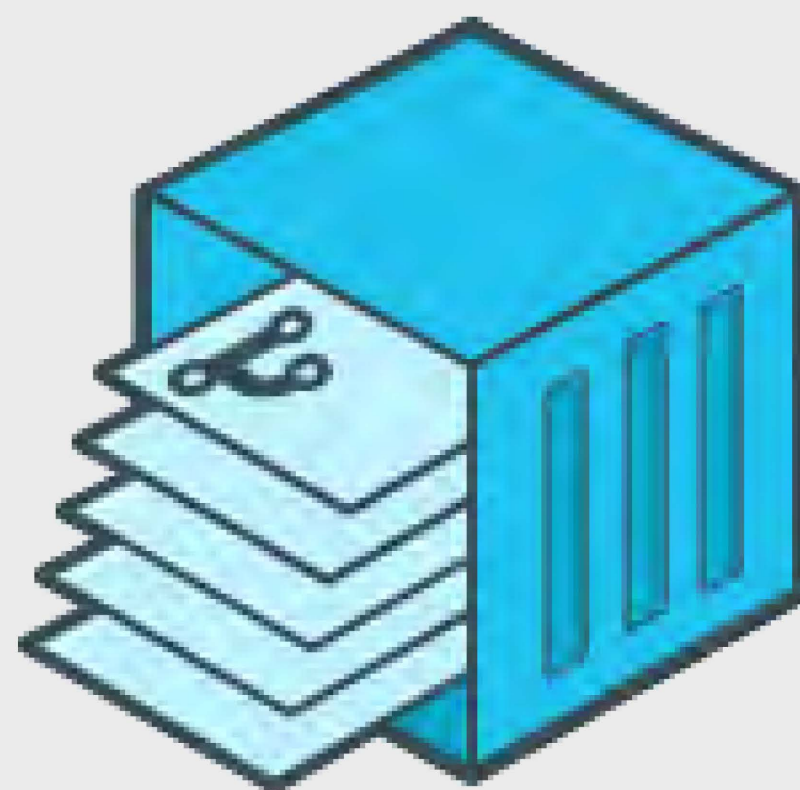
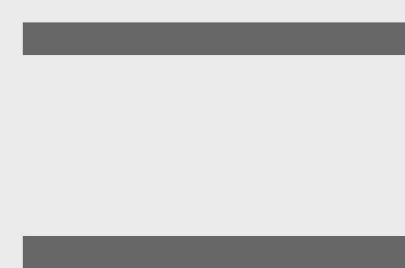
mybinder.org

binderhub.readthedocs.io

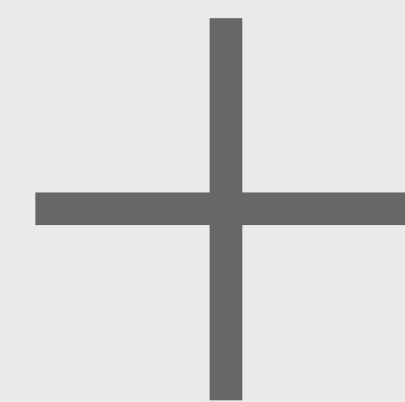




binderhub



repo2docker



jupyterhub



BinderHub is open tech...

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



mybinder.org



Turn a GitHub repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

Build and launch a repository

GitHub repo or URL

Git branch, tag, or commit

Path to a notebook file (optional)

File

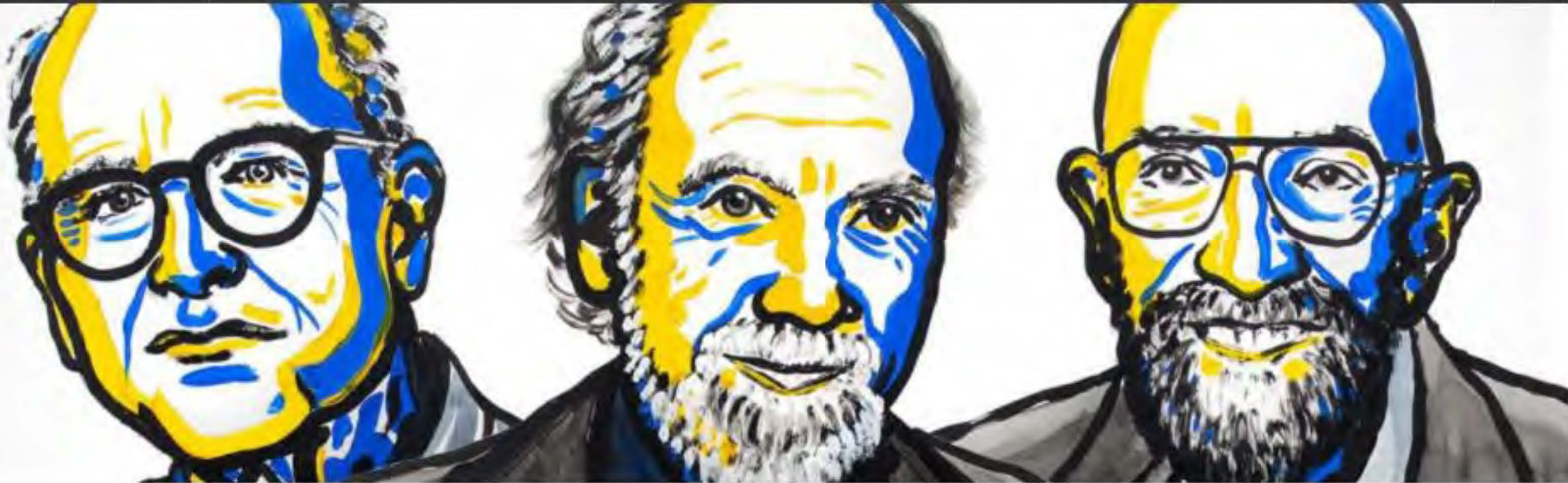
launch

Copy the URL below and share your Binder with others:

Copy the text below, then paste into your README to show a binder badge:

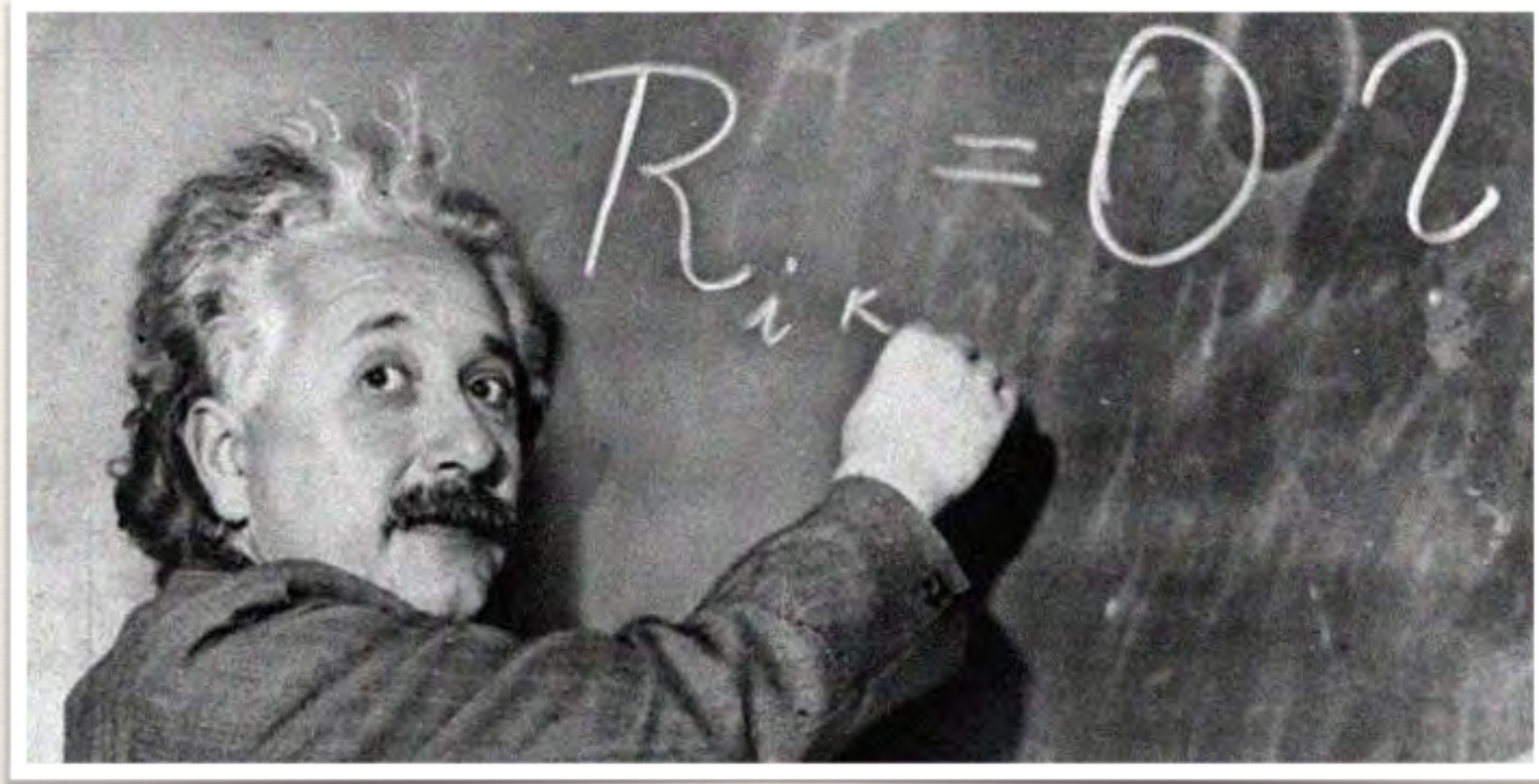
launch binder





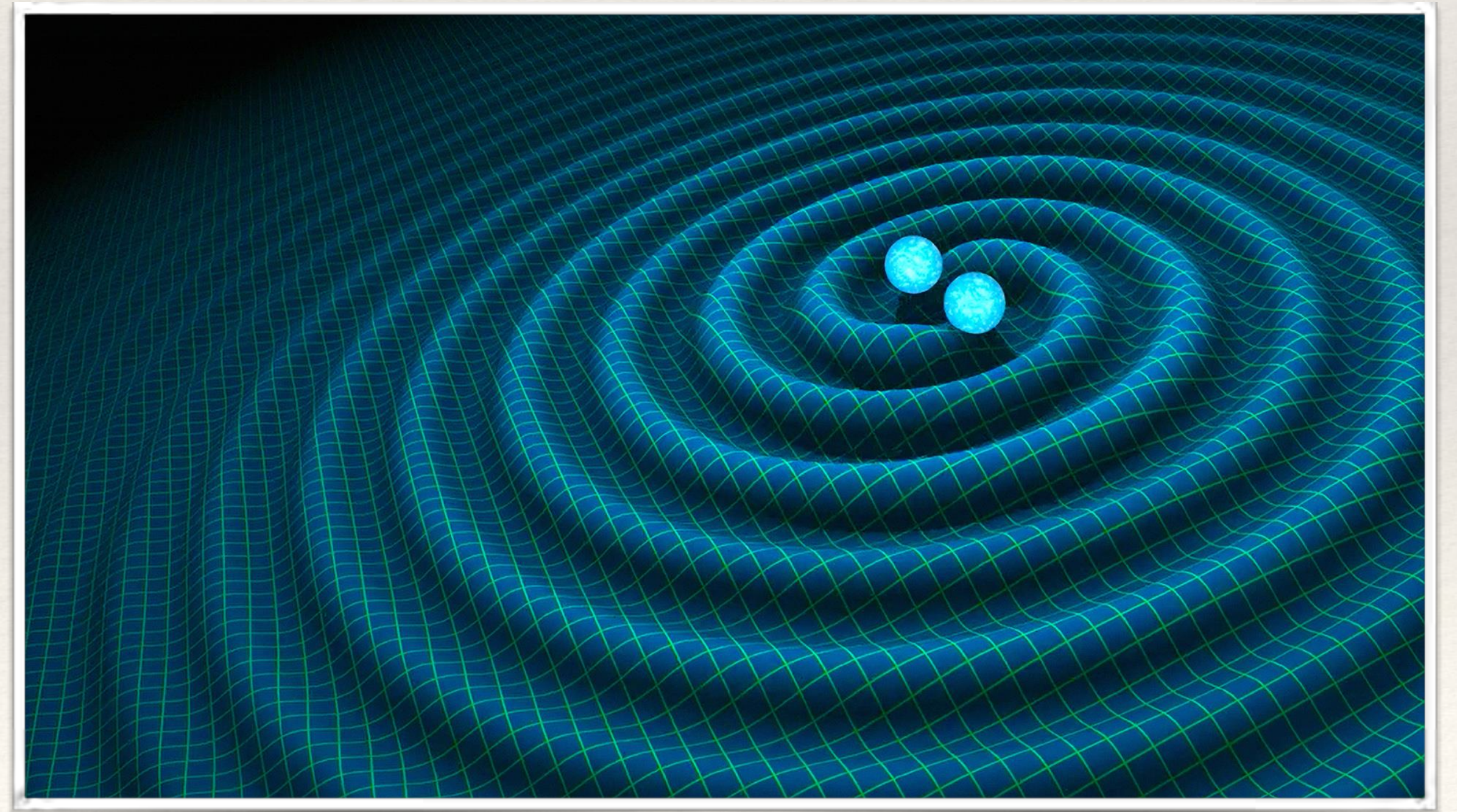
2017 NOBEL PRIZE IN PHYSICS AWARDED FOR DISCOVERY
OF GRAVITATIONAL WAVES

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





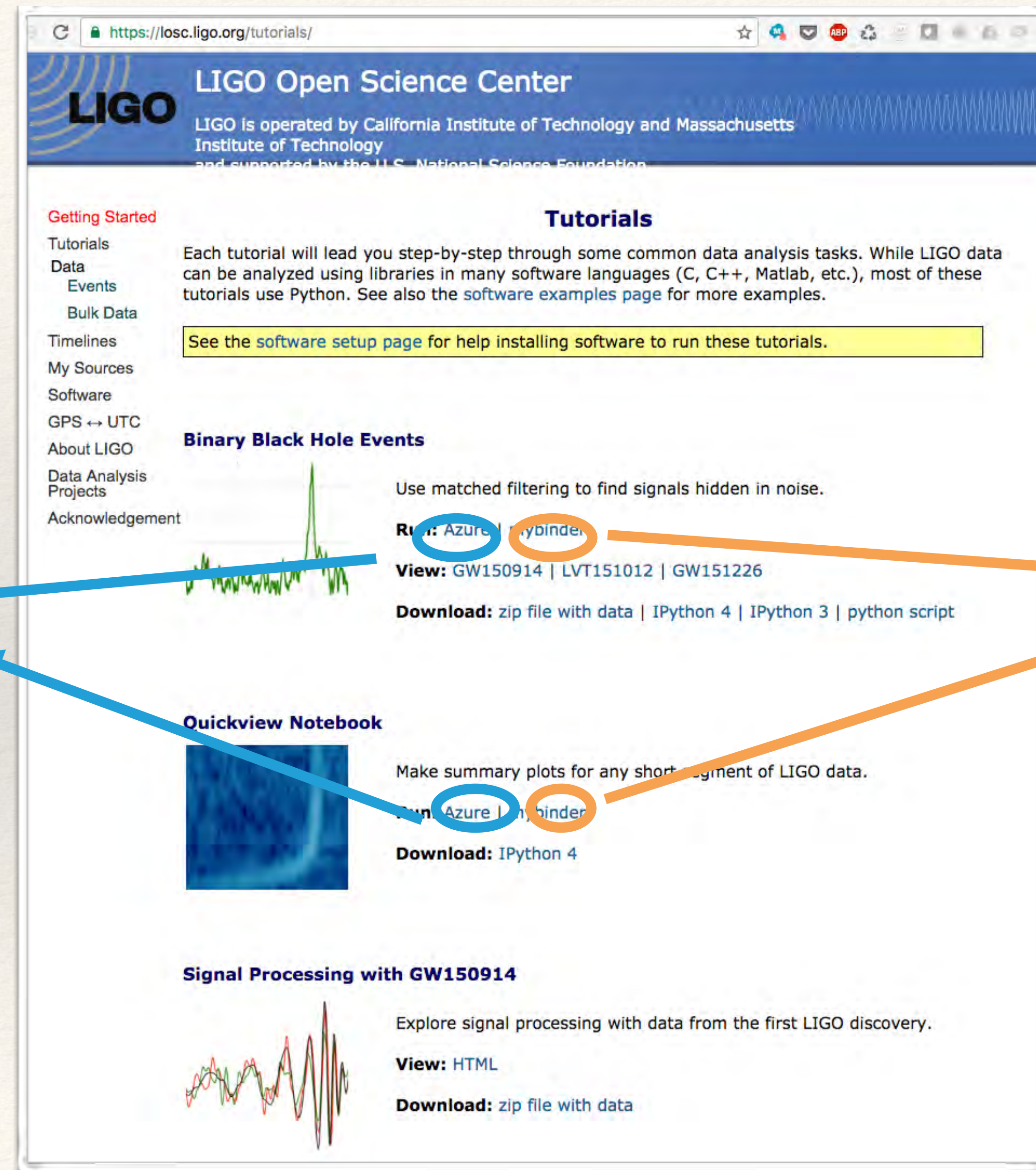
Two identical detectors: Hanford, WA and Livingston, LA

LIGO: a feat of science & engineering

Detection problem:

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

LIGO: Open Science with Jupyter



September 14, 2015

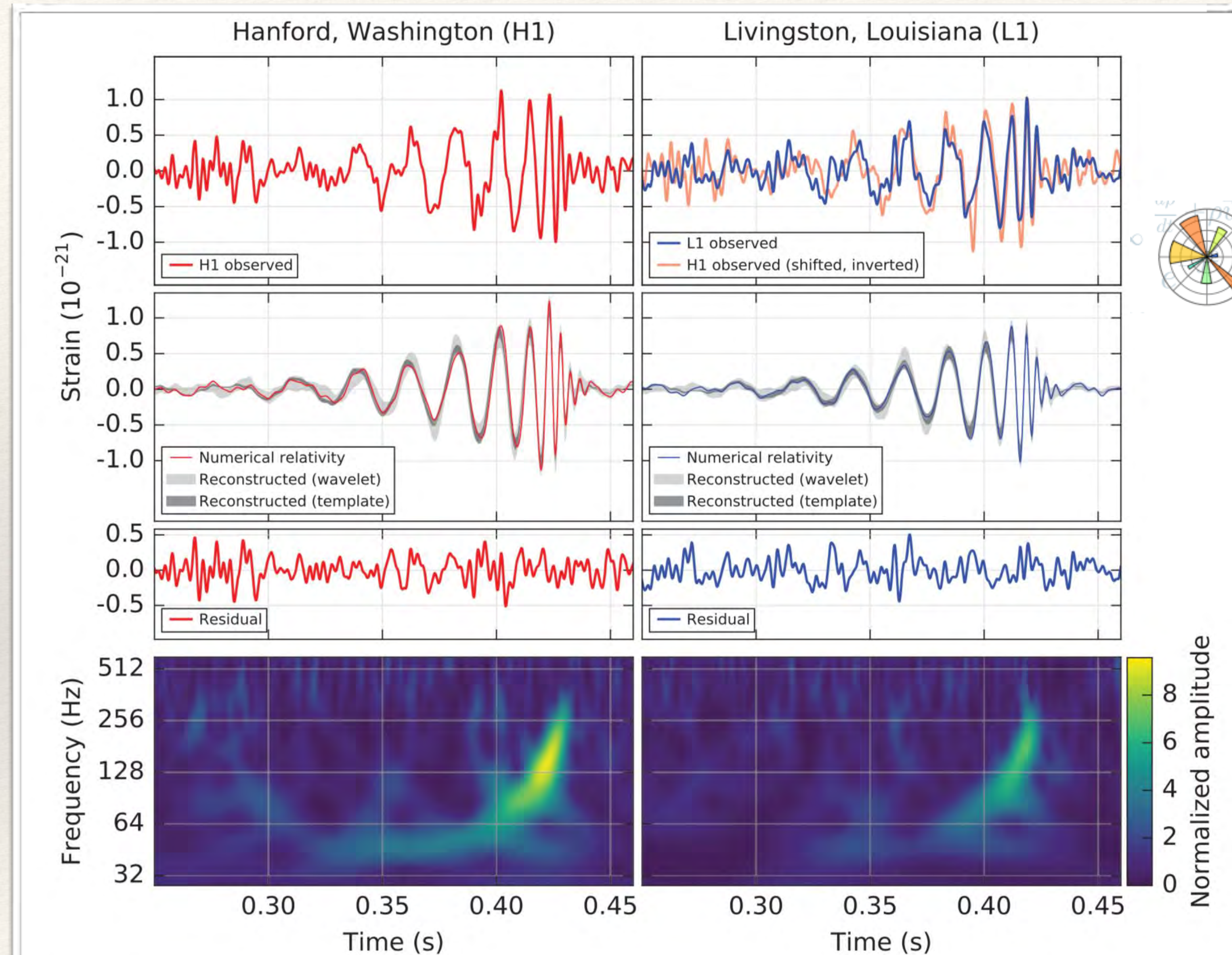
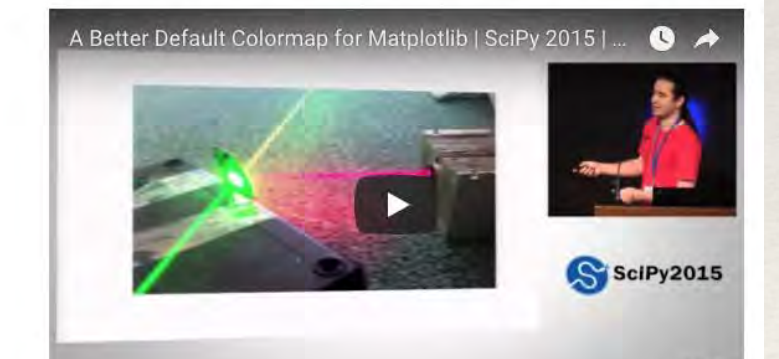


FIG. 1. The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress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

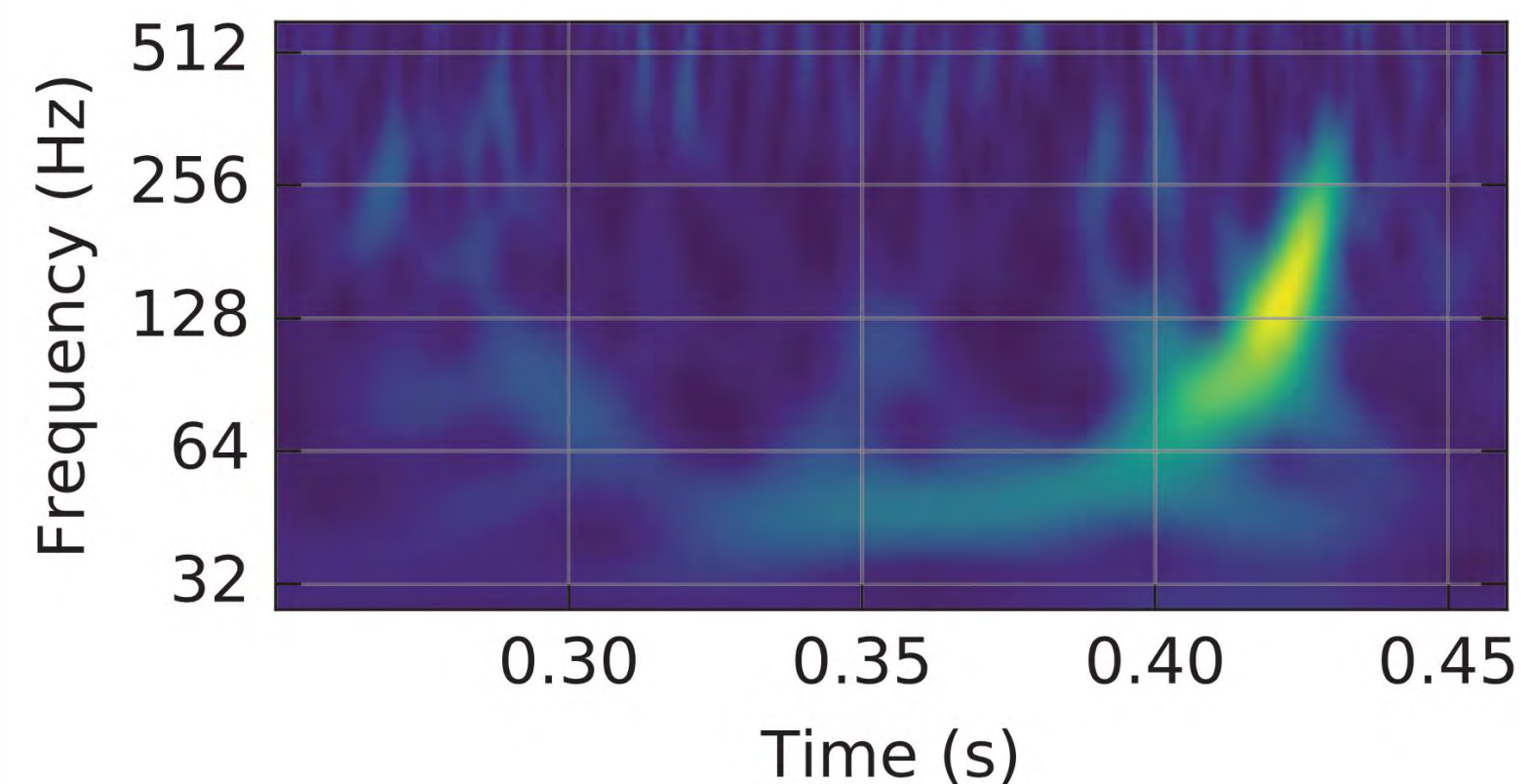
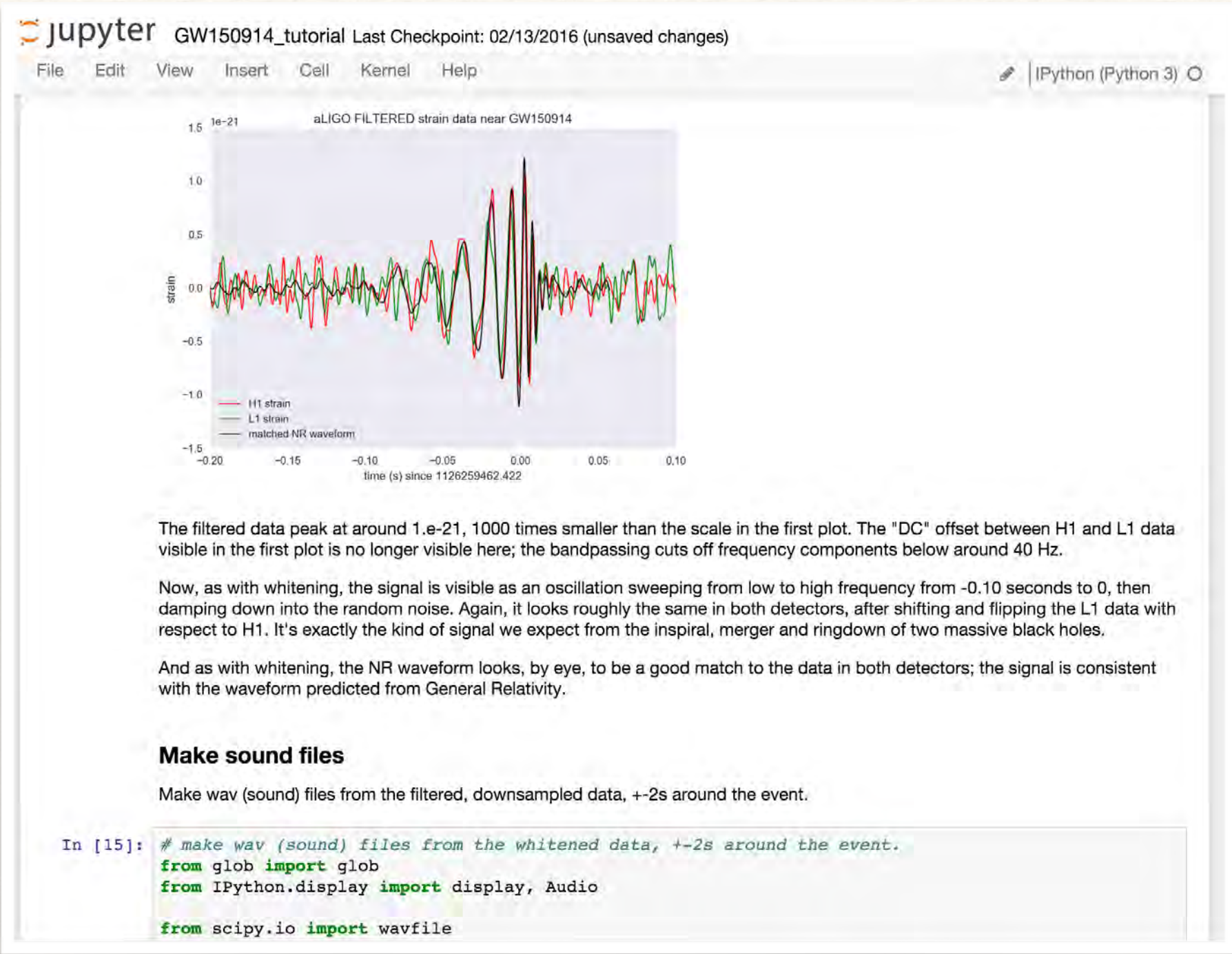


mpl colormaps

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 the talk presented at SciPy2015 that outlines the whole story.



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))

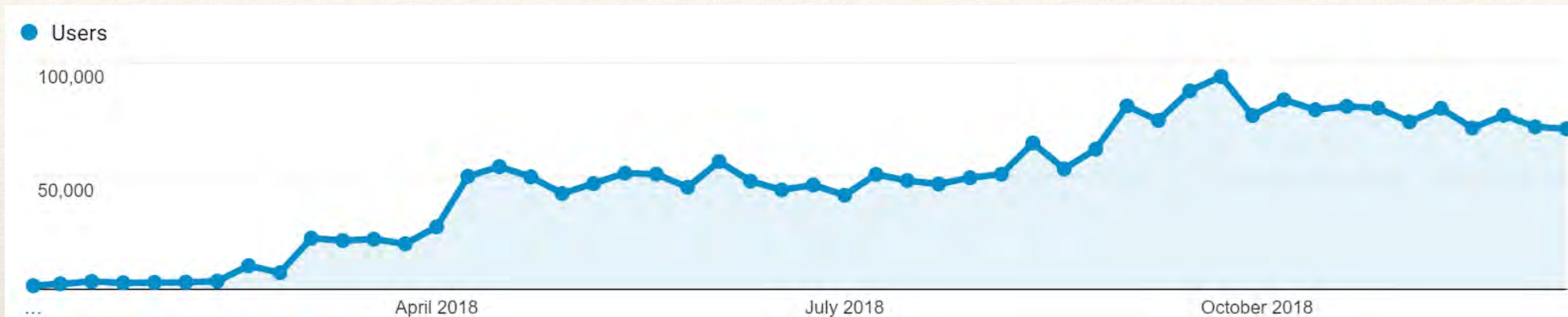
'GW150914_H1_whitenbp.wav'
```



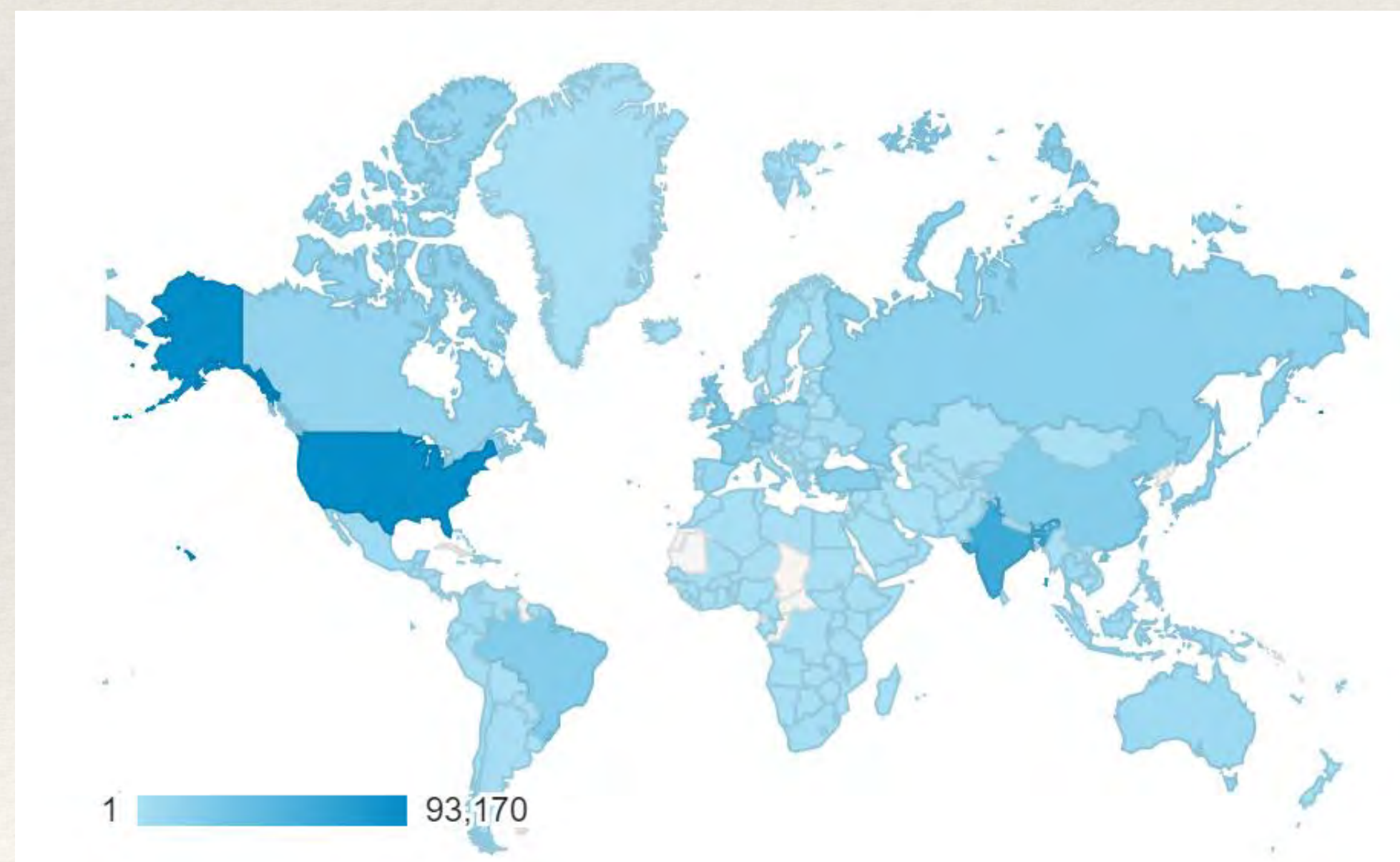
Using the IPython.display.Audio object

mybinder.org: usage

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



GORDON AND BETTY
MOORE
FOUNDATION



Berkeley: Yuvi Panda, Chris Holdgraf

Cal Poly: Carol Willing

Simula: Min Ragan-Kelley

Jessica Zosa-Forde, Tim Head

Binder users, Nov 1 2018 to present

Education

Berkeley's Data Science Courses

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

Table of Contents

- Cost Minimization Using a Program
- Issues with simple_minimize

Cost Minimization Using a Program

Let us return to our constant model:

$$\theta = C$$

We will use the mean squared error cost function:

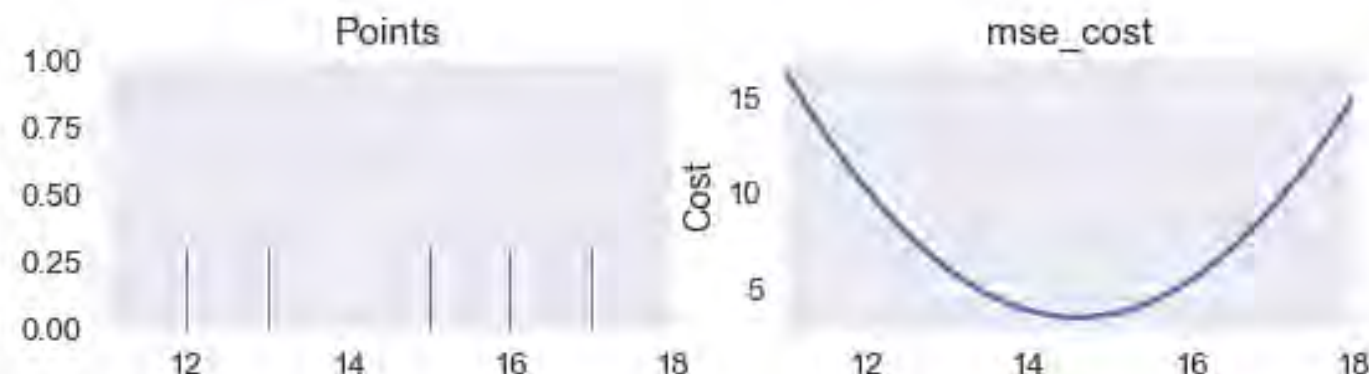
$$L(\theta, y) = \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

For simplicity, we will use the dataset $y = [12, 13, 15, 16, 17]$. We know from our analytical approach in a previous chapter that the minimizing θ for the MSE cost is $\text{mean}(y) = 14.6$. Let's see whether we can find the same value by writing a program.

If we write the program well, we will be able to use the same program on any cost function in order to find the minimizing value of θ , including the mathematically complicated Huber cost:

$$L_{\alpha}(\theta, y) = \frac{1}{n} \sum_{i=1}^n \begin{cases} \frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \leq \alpha \\ \alpha(|y_i - \theta| - \frac{1}{2}\alpha) & \text{otherwise} \end{cases}$$

First, we create a rug plot of the data points. To the right of the rug plot we plot the MSE cost for different values of θ .



Open on DataHub

gradient_basics

jupyter gradient_basics (unsaved changes)

File Edit View Insert Cell Kernel Help

Not Trusted Python 3

```
ax = plt.subplot(122)
plt.plot(thetas, costs)
plt.xlim(*xlim)
plt.title(cost_fn.__name__)
plt.xlabel(r'$\theta$')
plt.ylabel('Cost')
plt.legend()
```

Cost Minimization Using a Program

Let us return to our constant model:

$$\theta = C$$

We will use the mean squared error cost function:

$$L(\theta, y) = \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

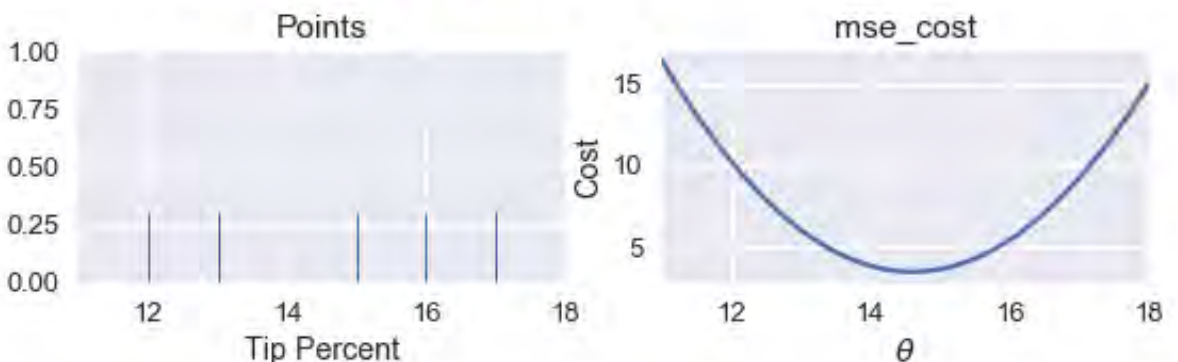
For simplicity, we will use the dataset $y = [12, 13, 15, 16, 17]$. We know from our analytical approach in a previous chapter that the minimizing θ for the MSE cost is $\text{mean}(y) = 14.6$. Let's see whether we can find the same value by writing a program.

If we write the program well, we will be able to use the same program on any cost function in order to find the minimizing value of θ , including the mathematically complicated Huber cost:

$$L_{\alpha}(\theta, y) = \frac{1}{n} \sum_{i=1}^n \begin{cases} \frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \leq \alpha \\ \alpha(|y_i - \theta| - \frac{1}{2}\alpha) & \text{otherwise} \end{cases}$$

First, we create a rug plot of the data points. To the right of the rug plot we plot the MSE cost for different values of θ .

```
In [10]: # HIDDEN
pts = np.array([12, 13, 15, 16, 17])
points_and_cost(pts, (11, 18), mse_cost)
```



International (CC BY-NC-ND 4.0)



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

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



Data 100, ~820

CA From K-12 to HPC

cybera

CYBERA > NETWORK > SERVICES > PROJECTS > NEWS & EVENTS > MEMBERSHIP > CONTACT US >

Jupyter 'All-in-One' Science Platform



Learning and sharing in a flexible, collaborative and interactive way.

Jupyter is an integrative application that incorporates math, science and engineering tools, along with communication and visualization resources, in one web-based platform. Simply put: It enables a broad suite of computing capabilities on any device that has an internet connection. For free.

Cybera and the [Pacific Institute for the Mathematical Sciences \(PIMS\)](#) have teamed up to increase access to, and awareness of, Jupyter. Cybera is hosting the platform on its Rapid Access Cloud, and is offering free access (and advice on how to get started) to Canada's public and innovation sectors.



Who Is Jupyter Useful For?

- **Teachers**
 - Create "interactive textbooks" that allow students to actively work on math or data science problems.
- **Students**
 - Learn and practice multiple programming languages, and log experiments, all in one place.



Use Cases

- **K-12**
 - An Alaska high school teacher used Jupyter to [change the way](#) "Introduction to Programming" is taught
- **Post-Secondary**
 - A professor at George Washington University used Jupyter to teach [Aerodynamics-Hydrodynamics](#) (see [github](#) notes)

compute
canada

calcul
canada

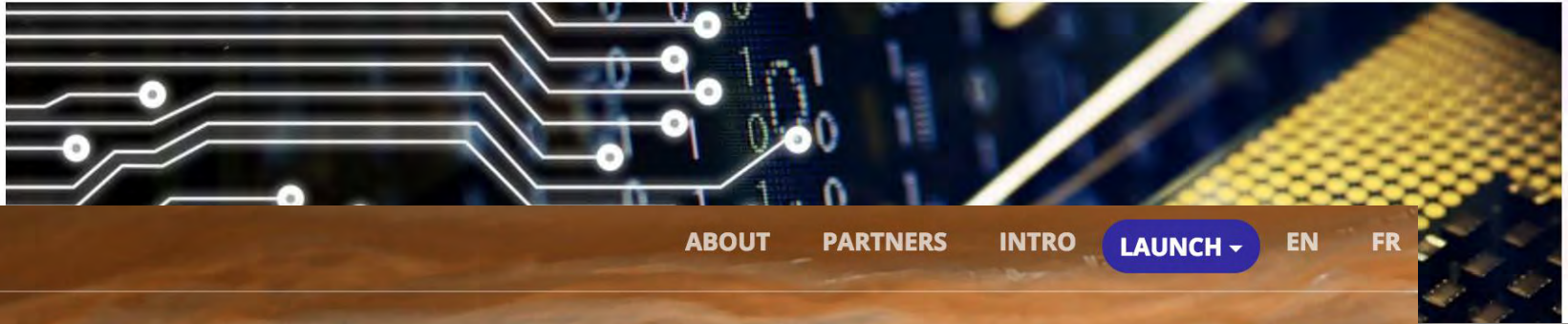
15/03/2017 :: Featured, News



compute canada and pims launch jupyter service for researchers

[←](#) [→](#)

[Home](#)
[About](#)
[Renewing Canada's Advanced
Research Computing Platform](#)



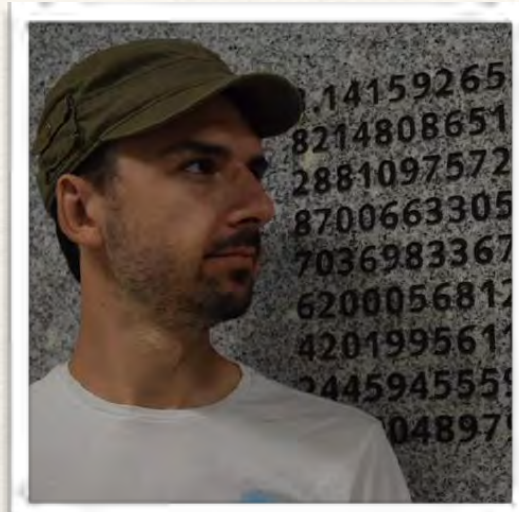
SYZYG.CA [ABOUT](#) [PARTNERS](#) [INTRO](#) [LAUNCH](#) [EN](#) [FR](#)

SYZYG.CA

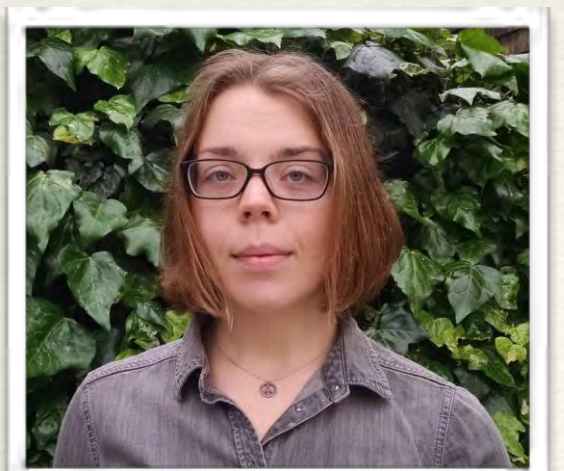
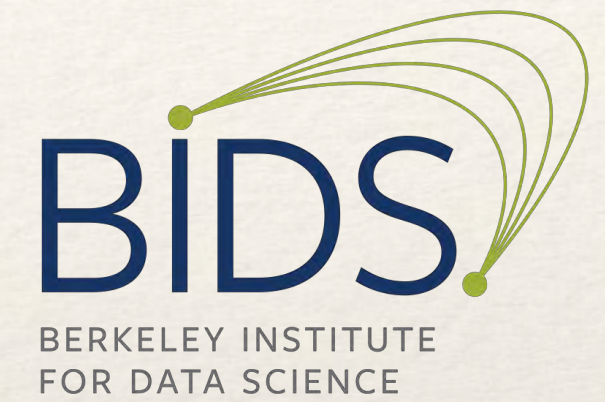
Launch Jupyter at your university, school or company?

[SEND](#)

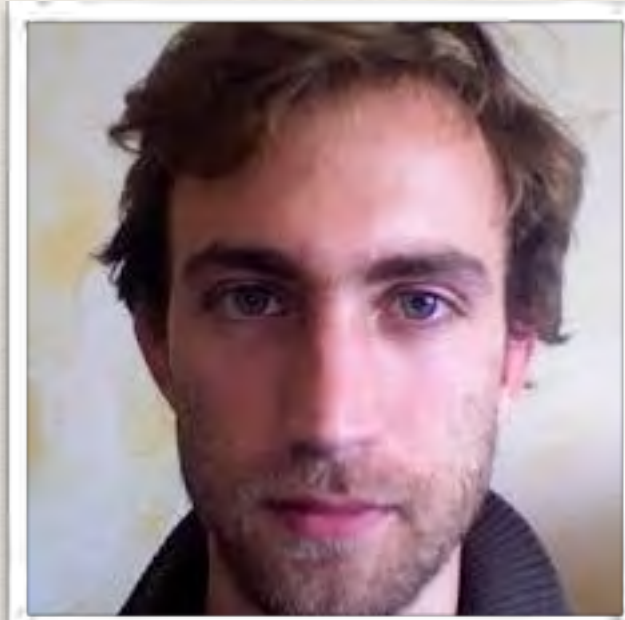
Jupyter @ Berkeley and LBNL



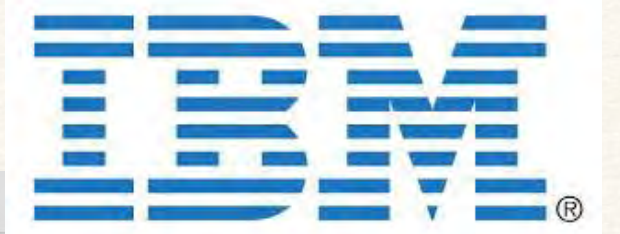
University of California, Berkeley
DEPARTMENT OF STATISTICS



Bloomberg



Wide industrial adoption



Data Science Experience

datascience.ibm.com

A Growing Set of Data Science Tools

- Jupyter Notebooks**
Create and collaborate on Python, R, and Scala notebooks that contain code and visualizations.
- RStudio**
Jumpstart your R experience with a free, open-source RStudio tool.
- Machine Learning (Coming Soon)**
Create, train and deploy machine learning models.

See All Features

GRYD



Amazon SageMaker

Notebook instance → Jobs → Models → Endpoint

- Notebook instance**
Explore AWS data in your notebooks, and use algorithms to create models via training jobs.
[Create notebook instance](#)
- Jobs**
Track training jobs at your desk or remotely. Leverage high-performance AWS algorithms.
[View jobs](#)
- Models**
Create models for hosting from job outputs, or import externally trained models into Amazon SageMaker.
[View models](#)
- Endpoint**
Deploy endpoints for developers to use in production. A/B Test model variants via an endpoint.
[View endpoints](#)

Microsoft Azure Notebooks

Overview Libraries FAQ/Support What's New Sign in

jupyter
Notebooks hosted on Microsoft Azure

[Go to my Notebook Server](#)
[Show me some samples](#)

WHAT IS JUPYTER?
Interactive Notebooks for Data Science
Browser-based REPL with Markdown
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 Storage
Enjoy some free cycles on us

Google Cloud Platform

Overview of Colaboratory Features

Table of contents

- Cells
 - Code cells
 - Text cells
 - Adding and moving cells
- Working with python
 - System aliases
 - Magics
 - Tab-completion and exploring code
- Rich, interactive outputs
- Integration with Drive
 - Commenting on a cell

Rich, interactive outputs

Until now all of the generated outputs have been text, but they can be more interesting, like the chart below.

```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
plt.title("Fills and Alpha Example")
plt.show()
```

Integration with Drive

Colaboratory is integrated with Google Drive. It allows you to share, comment, and collaborate on the same document with multiple people:

- The **SHARE** button (top-right of the toolbar) allows you to share the notebook and control permissions set on it.
- File->Make a Copy** creates a copy of the notebook in Drive.
- File->Save** saves the File to Drive. **File->Save and checkpoint** pins the version so it doesn't get deleted from the revision history.
- File->Revision history** shows the notebook's revision history.
- Multiple people can **collaboratively edit** the same notebook at the same time. Like Google Docs, you can see collaborators both within the document (top right, left of the comments button) and within a cell (right of the cell).



Funding and resources



**ALFRED P. SLOAN
FOUNDATION**

GORDON AND BETTY
MOORE
FOUNDATION

THE LEONA M. AND HARRY B.
HELMSLEY
CHARITABLE TRUST



U.S. DEPARTMENT OF
ENERGY



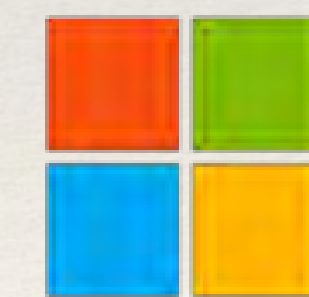
SIMONS FOUNDATION



ANACONDA®



POWERED BY
rackspace®
the open cloud company



Microsoft



Google

Bloomberg

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>



Thank You!

Contact: scholia@lbl.gov

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

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>

