

# Introduction to Jupyter

Presented by Ben Winjum

IDRE

October 25, 2018

# Outline

- Briefly, what is the Jupyter notebook?
- Using the notebook: text and code
- The growing use of Jupyter
- Using the notebook: output and interactive widgets
- Jupyter notebooks as part of a larger Jupyter ecosystem
- Methods for sharing notebooks
  - And words of caution
- Intro to JupyterLab

```
Python 3.6.3 | packaged by conda-forge | (default, Nov  4 2017, 10:13:32)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.0.0.dev -- An enhanced Interactive Python. Type '?' for help.
```

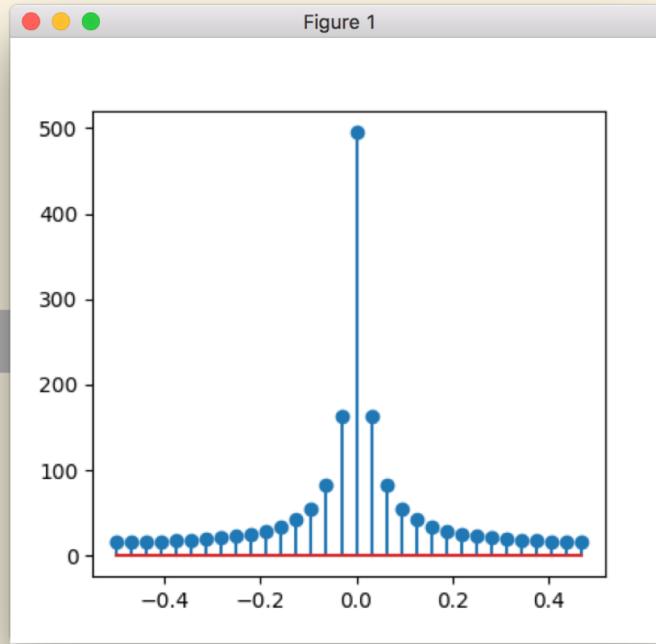
```
In [1]: from numpy.fft import *
...: from numpy import arange
...: a = arange(32)
...: A = fft(a)
...: f = fftfreq(32)

In [2]: %matplotlib tk

In [3]: from matplotlib.pyplot import stem

In [4]: stem(f, abs(A))
Out[4]: <Container object of 3 artists>

In [5]: .
       add_callback    eventson
       baseline        get_children
       count()         get_label
```



Jupyter grew out of IPython, an interactive shell for computing with Python.

File Edit View Insert Cell Kernel Help

Python 3



## Simple spectral analysis

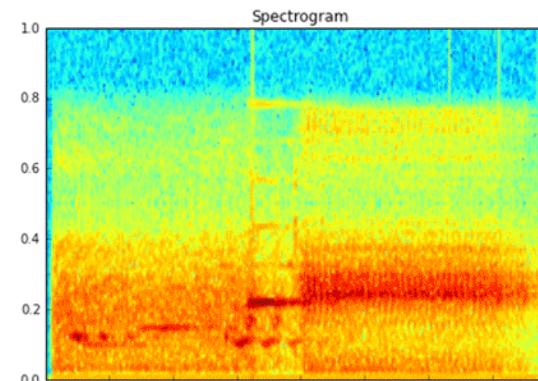
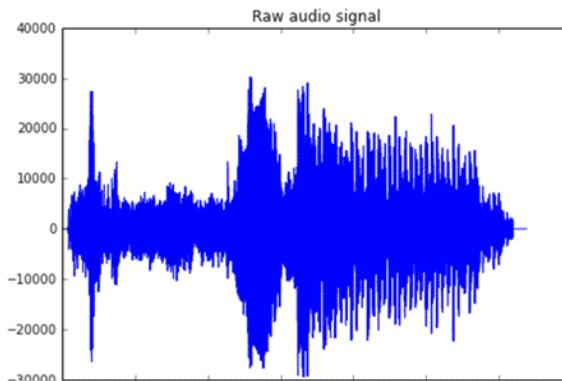
An illustration of the [Discrete Fourier Transform](#)

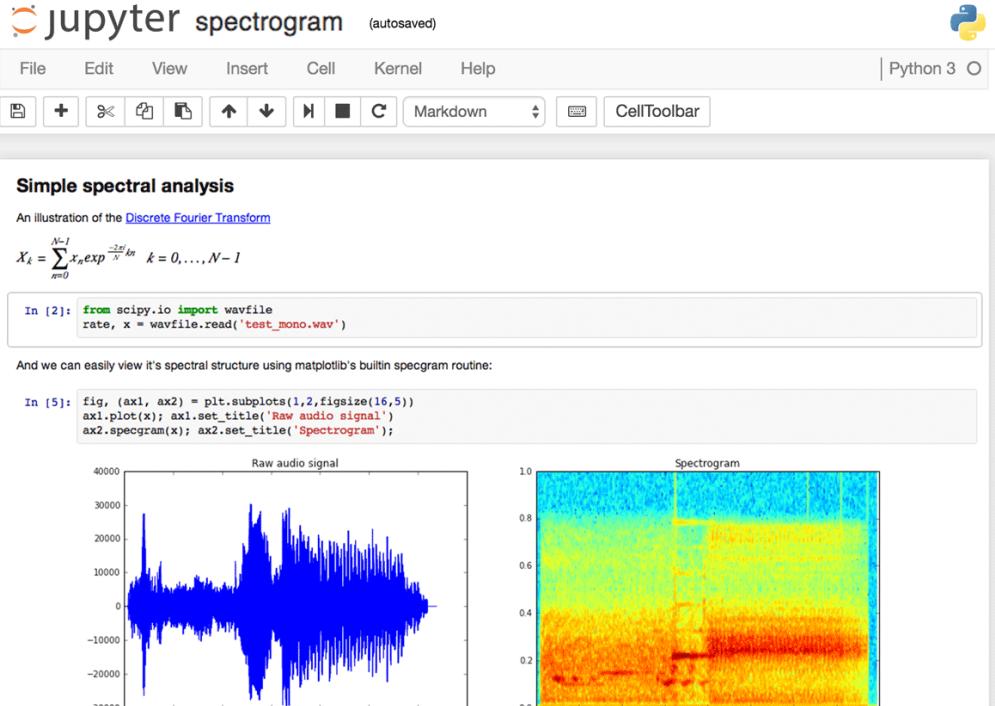
$$X_k = \sum_{n=0}^{N-1} x_n \exp^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

```
In [2]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

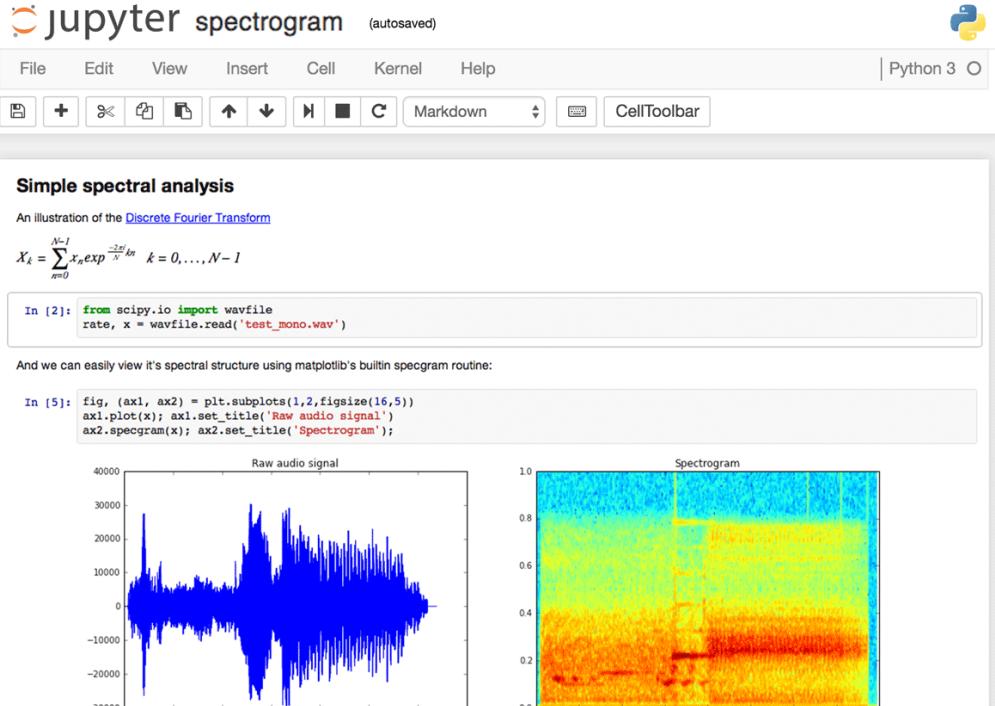
```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize=(16,5))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```





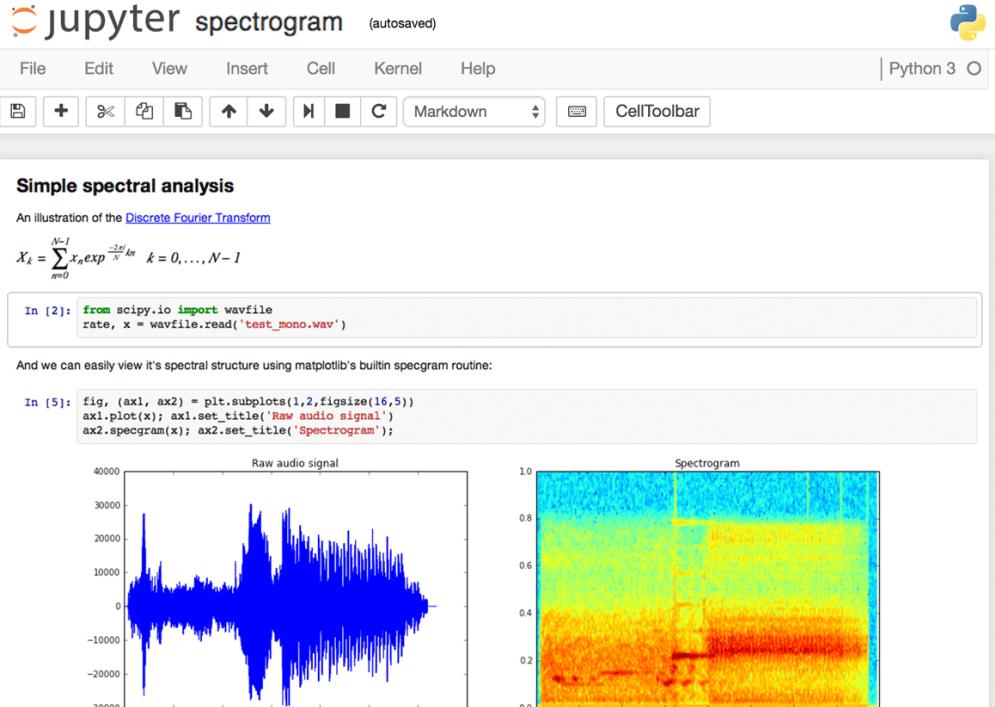
## Main features of the web application

- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.



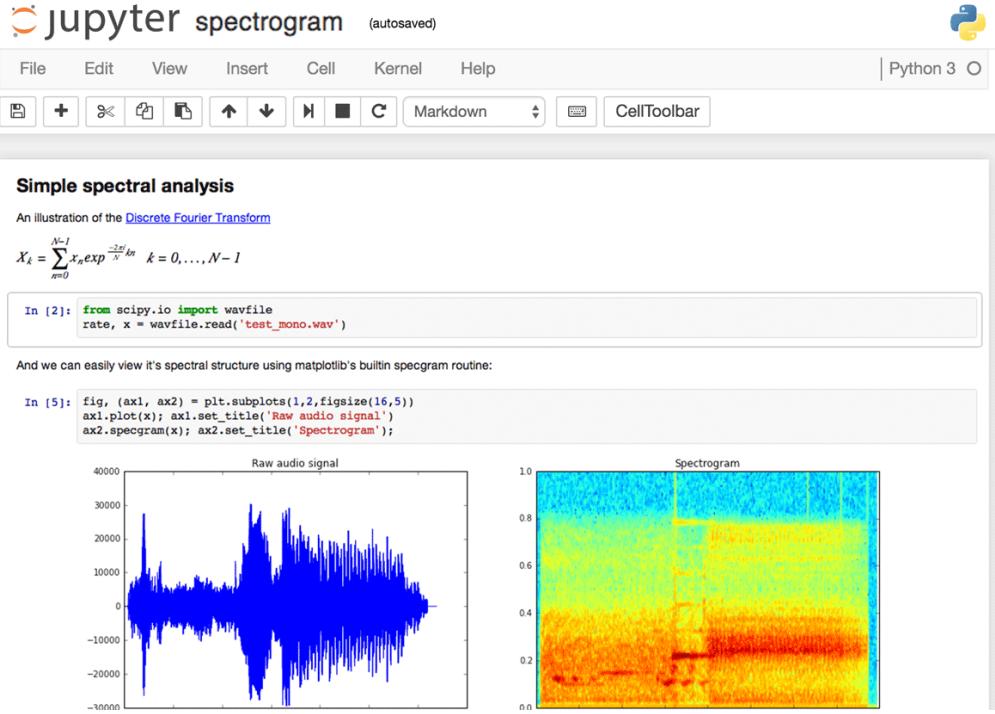
## Main features of the web application

- The ability to execute code from the browser, with the results of computations attached to the code which generated them.



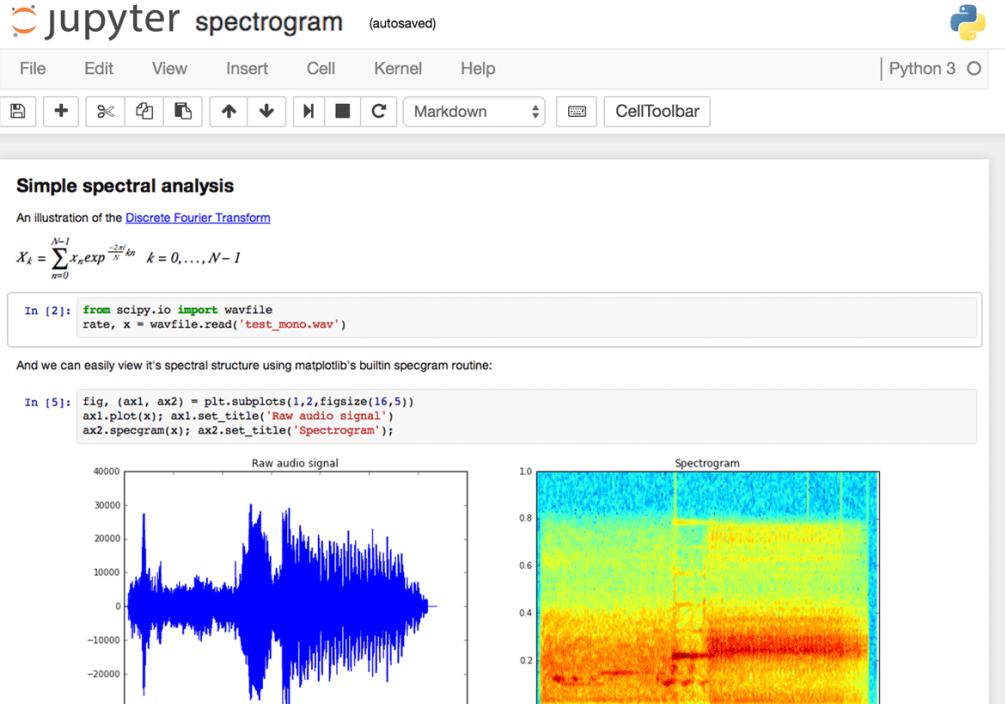
## Main features of the web application

- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc. For example, publication-quality figures rendered by the matplotlib library, can be included inline.



## Main features of the web application

- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.



## Main features of the web application

- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.



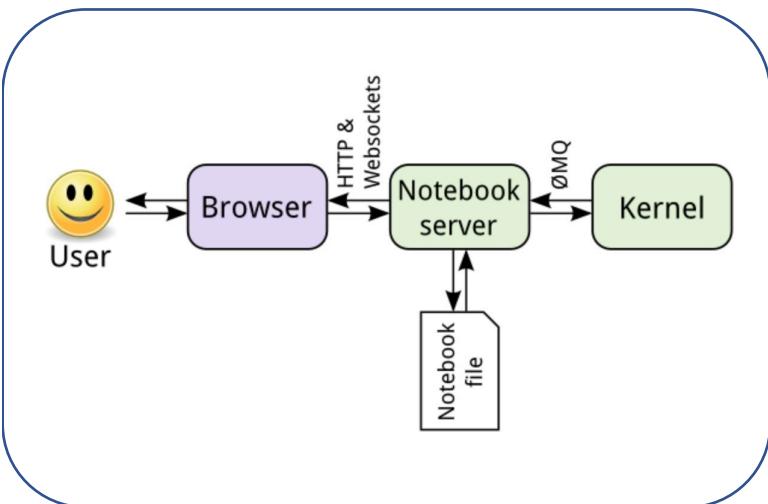
**Julia, Python, R**  
and now over 100 other languages

# Notebook basics

go to <https://jupyter.idre.ucla.edu>

```
$ cd work  
$ git clone https://github.com/benjum/idre-intro-to-jupyter.git
```

# The Jupyter Notebook



A screenshot of the Jupyter Notebook interface titled "jupyter spectrogram (autosaved)". The top navigation bar includes File, Edit, View, Insert, Cell, Kernel, Help, and a Python 3 logo. Below the title is a toolbar with various icons for file operations like Open, Save, and Print, along with buttons for Cell and CellToolbar.

The main content area displays a notebook cell titled "Simple spectral analysis". The cell contains the text "An illustration of the Discrete Fourier Transform" and a mathematical formula:

$$X_k = \sum_{n=0}^{N-1} x_n \exp^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

Below the formula is another code cell:

```
In [2]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

Text below the cell says "And we can easily view its spectral structure using matplotlib's builtin specgram routine:" followed by:

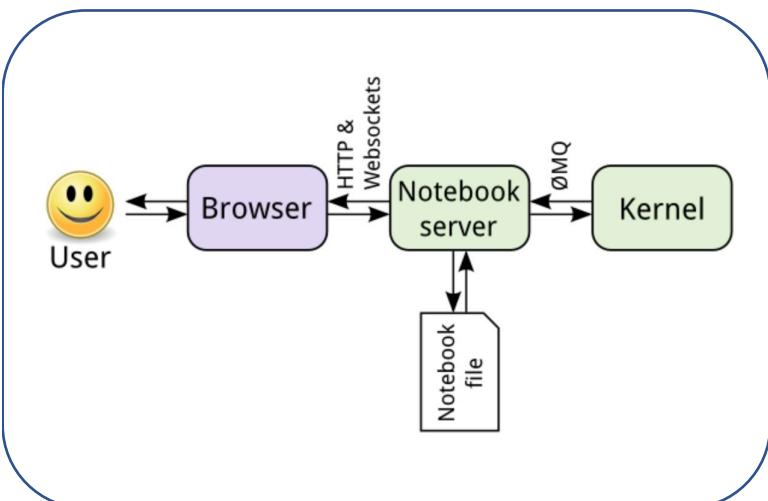
```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize=(16,5))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```

Two plots are shown side-by-side. The left plot, titled "Raw audio signal", shows a blue line graph of raw audio data with axes ranging from -30000 to 40000. The right plot, titled "Spectrogram", is a heatmap showing frequency over time with axes ranging from 0.0 to 1.0.

# The Jupyter Notebook

Jupyter is comprised of several components

## Front-end:



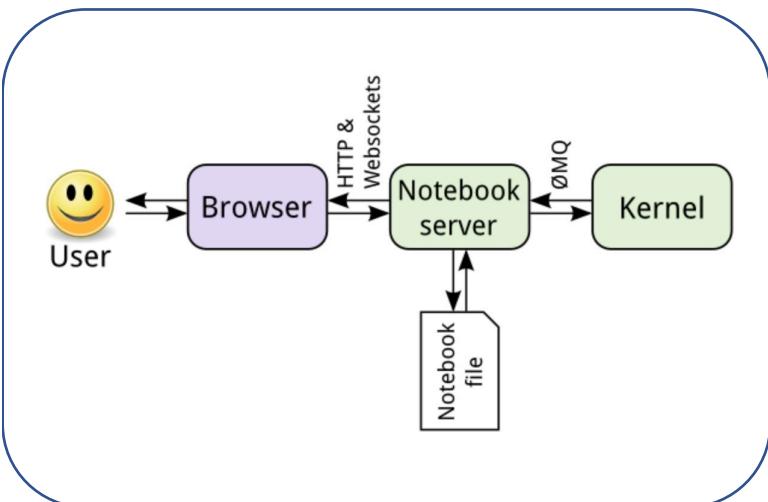
1. **Web Application:** Browser-based tool for interactive development of notebook documents
2. **Notebook Document:** A representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects. These documents are internally JSON files and are saved with the .ipynb extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues.

# The Jupyter Notebook

Jupyter is comprised of several components

## Back-end:

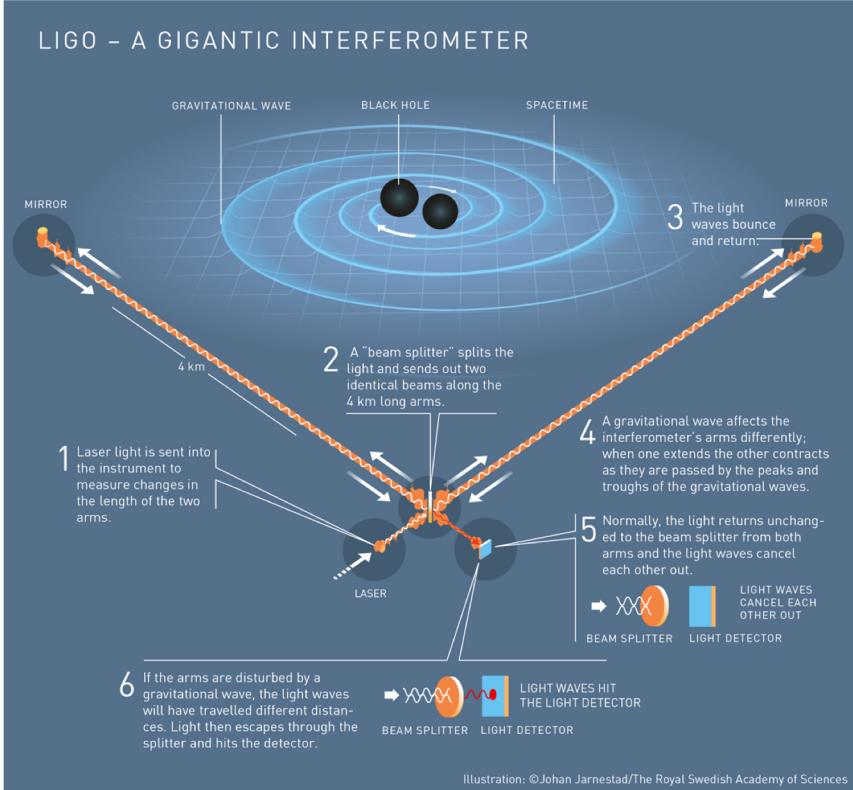
1. **Kernel:** A separate process responsible for running user code. Jupyter is capable of interfacing with many programming languages.
2. **Notebook Server:** Communicates with kernel and routes the programming language to the web browser.





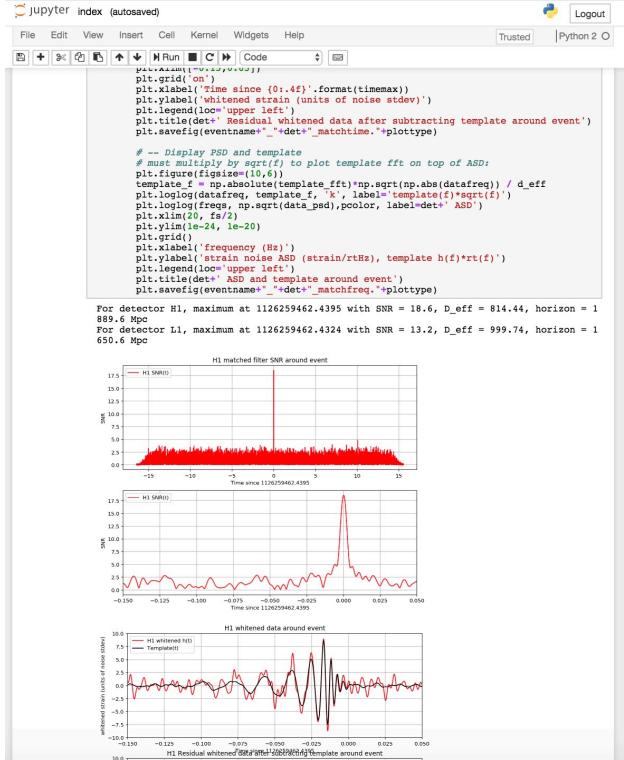
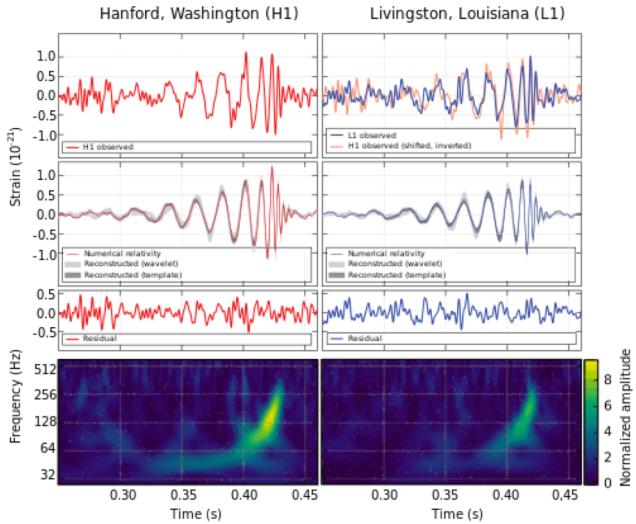
## Supported kernels

# Jupyter Exploration of 2017 Physics Nobel Work



<http://physics.aps.org/featured-article-pdf/10.1103/PhysRevLett.116.061102>  
<https://www.gw-openscience.org/tutorials/>

# LIGO & Open Science



<http://physics.aps.org/featured-article-pdf/10.1103/PhysRevLett.116.061102>  
<https://www.gw-openscience.org/tutorials/>

# Jupyter notebooks engage learners

**Convert from kern format to MusicXML**

```
In [12]: c = converter.parse('/Users/carol/Downloads/duet.edokomurui.krn')
c.show()
```

Out[12]:

5

10

music21: a toolkit for computer-aided musicology

What is music21?

music21 is a toolkit for doing research and other activities answer questions about music quickly and simply. If you've ever used Python in the lab, "Regular how often does that?" or "I wish I knew which hand was first to use these chords in this order," or "I'll bet we'd know more about Indian music's counterparts (or Indian ragas or postal pitch engines) or whatnot. I wrote a program to let me do that" or "I want to automatically write more of them." That's what music21 can help you with your work.

How simple is music21 to use?

Extremely. After starting Python and typing "from music21 import \*" you can do all of these things with only a single line of music21 code:

```
Display a short melody in musical notation:
converter.parse("J:/kern/composition.krn").show()

Print the twelve-tone matrix for a tone row (In this case the opening of Schoenberg's Fourth String Quartet):
print(music21.sets.resonance((2,1,0,1,3,0,0,2,7,1,1)) )

or since all the classical school rows are already convertible to objects, you can type:
print(music21.sets.classicalSchools.mozart("string quartet971").matrix[1])
```

Convert a file from Hindmarsh's "kern" data format to MusicXML (for editing in Sibelius):

```
converter.parse("J:/kern/composition.krn").write("musicxml")
```

```
def closedPosition(self):
    return self._newChordObject with:
        >>> chord = Chord("C4", "G5")
        >>> chord2 = chord.closedPosition
        >>> print(chord2.lyricValue)
        'C4' g4
```

```
newChord = copy.deepcopy(self)
tempChord.pitches = newChord.pitches
chordSaps = self._newSaps()
for thisPitch in chord.pitches:
    while thisPitch in chordSaps:
        thisPitch = thisPitch.octave + 1
    tempChord.pitches.append(thisPitch)
```

- Get Started with music21
- Browse the music21 documentation
- Download the music21 Google Code
- Get our latest news and updates at the music21 blog
- Read the Frequently Asked Questions list
- Sign up for the music21-discuss mailing list through Google Groups.

The figure shows a Jupyter Notebook interface. Cell [10] contains Python code to parse a file and show its contents. Cell [10] displays the parsed data, which includes musical notation on a staff and lyrics in Russian. The notation consists of two measures of music in common time, G major, with notes and rests. The lyrics are in Russian, with some words in German (e.g., 'Gott sei Dank'). A screenshot of a person's face is visible in the bottom right corner.

```
In [10]: sBach = corpus.parse('bach/bwv7.7')
sBach.show()

Out[10]:
```

**bwv7.7.mxl**

Movement Name: bwv7.7.mxl

1. Господи, яко ты ведаеши око икою икою  
и честною кровию икою икою  
и честною кровию икою икою

2. Господи, яко ты ведаеши око икою икою  
и честною кровию икою икою  
и честною кровию икою икою

3. Господи, яко ты ведаеши око икою икою  
и честною кровию икою икою  
и честною кровию икою икою

4. Господи, яко ты ведаеши око икою икою  
и честною кровию икою икою  
и честною кровию икою икою

5. Господи, яко ты ведаеши око икою икою  
и честною кровию икою икою  
и честною кровию икою икою



**Michael Scott Cuthbert** ([cuthbert \[at\] mit.edu](mailto:cuthbert@mit.edu)) is Associate Professor of Music and Homer A. Burnell Career Development Professor at M.I.T.

1

From C. Willing, JupyterCon 2017

R. Connelly and V. Gayle

## An investigation of Social Class Inequalities in General Cognitive Ability in Two British Birth Cohorts

“Concern about the lack of the reproducibility of research persists across a range of academic disciplines (Nature [Editorial], 2016). There is a general appeal for extra materials to be routinely provided alongside research publications which include sufficient information for a third party to reproduce results without any additional information from the authors (Diggle, 2015; King, 1995; King, 2003). An innovative aspect of this work is that we go beyond providing the usual supplementary material and make the complete workflow openly available. We take the trailblazing step of rendering the complete workflow accessible and reproducible within a Jupyter notebook .... Jupyter notebooks have been used in Nobel Prize winning high-profile big science applications but are rarely used in Sociology.”

<https://github.com/RoxanneConnelly/Social-Class-Inequalities-in-General-Cognitive-Ability-in-Two-British-Birth-Cohorts>

# Novel uses of Jupyter: Digital Publication (takes a long time for adoption to actually occur)

## The "Paper" of the Future

-  Alyssa Goodman (Harvard University)
-  Josh Peek (Space Telescope Science Institute)
-  Alberto Accomazzi (Harvard-Smithsonian Center for Astrophysics (CFA))
-  CB Chris Beaumont (Harvard-Smithsonian Center for Astrophysics (CFA))
-  CB Christine L. Borgman (UCLA - University of California, Los Angeles)
-  Hope How-Huan Chen (Harvard University)
-  Merce Crosas (Harvard University)
-  CE Christopher Erdmann (North Carolina State University)
-  August Muench
-  Alberto Pepe (Authorea Team)
-  CW Curtis Wong (Microsoft)

show less

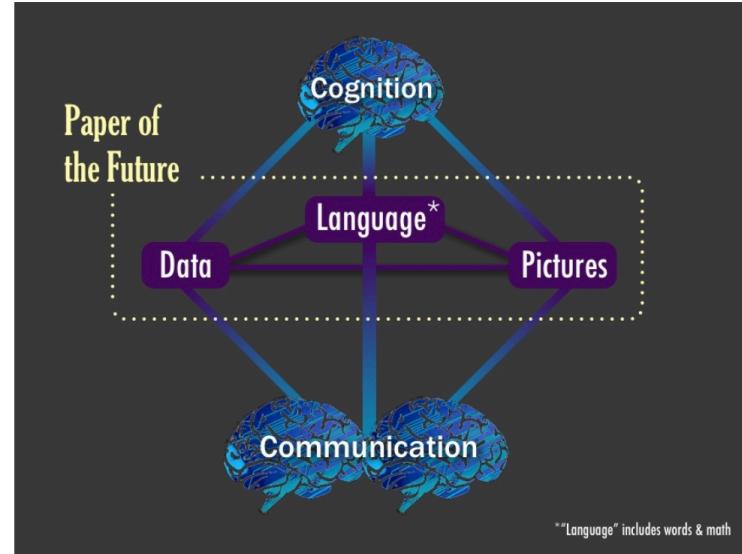


Fig. 1

*The Paper of the Future should include seamless linkages amongst **data**, **pictures**, and **language**, where "language" includes both words and math. When an individual attempts to understand each of these kinds of information, different cognitive functions are utilized: communication is inefficient if the channel is restricted primarily to language, without easy interconnection to data and pictures.*

Notebooks: output and widgets

# One more note on output and widgets

## Jupyter widgets as a framework

Jupyter widgets forms a framework for representing python objects interactively. Some large open-source interactive controls based on Jupyter widgets include:

- bqplot - 2d plotting library
- pythreejs - low-level 3d graphics library
- ipyvolume - high-level 3d graphics library
- ipyleaflet - maps

# The number of Jupyter users is growing very quickly

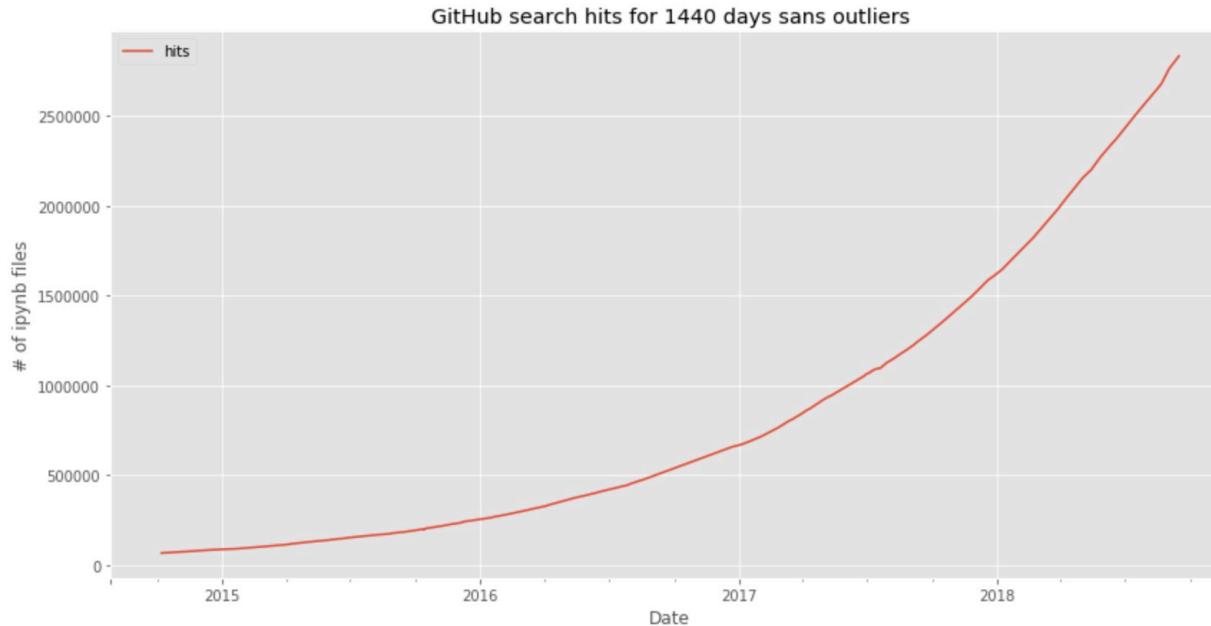


Image: <http://nbviewer.jupyter.org/github/parente/nbestimate/blob/master/estimate.ipynb>

Nevertheless, Jupyter tools are being used by some major companies, and they are finding it in their best interest to contribute to open source tools related to Jupyter



M Beyond Interactive: Notebook | X +

A Medium Corporation [US] | https://medium.com/netflix-techblog/notebook-innovation-591ee3221233

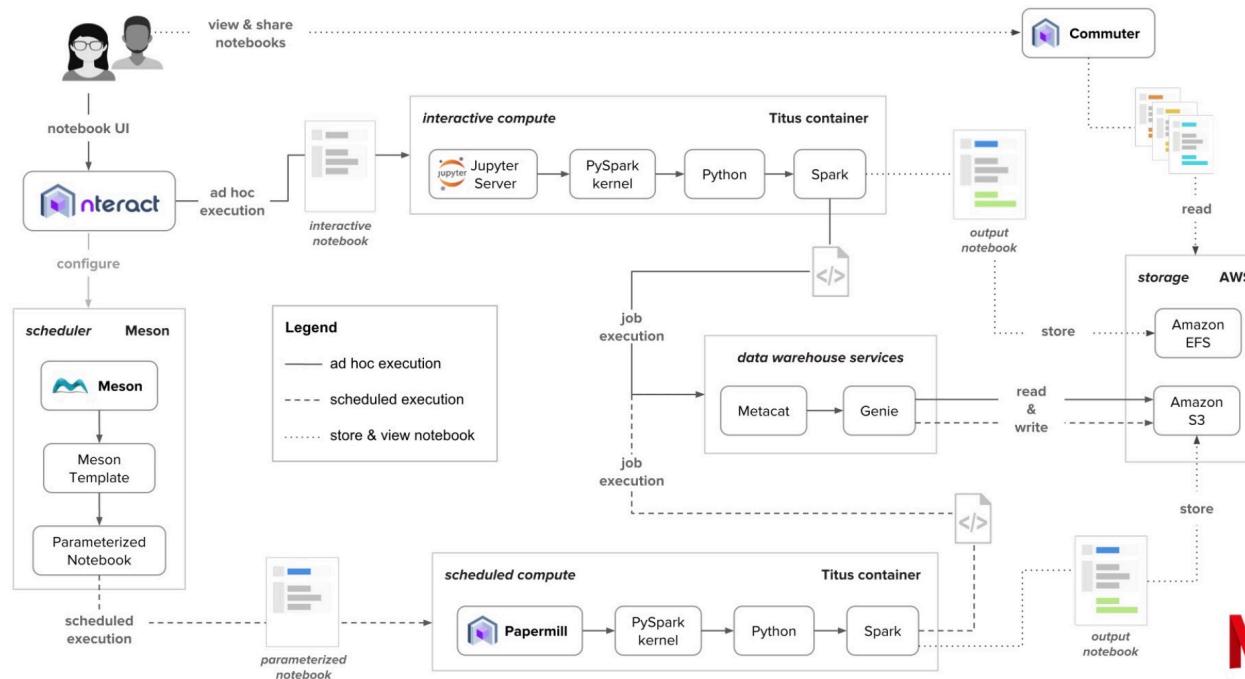
Apps bookmarks science gateways r docker jupyterhub Software Carpentry Computational Inf... PLOS Biology: Bes... » Other Bookmarks

**THE NETFLIX TECH BLOG**

Follow

Search

Profile

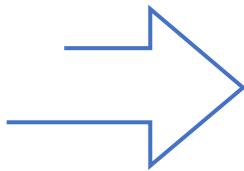


Notebook Infrastructure at Netflix

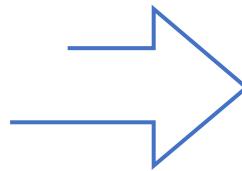
<https://medium.com/netflix-techblog/notebook-innovation-591ee3221233>

# PayPal Notebooks Open Source Plans

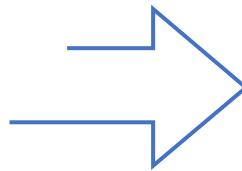
Slide from JupyterCon 2018



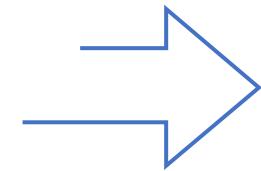
- PPMagics: available **now**
- Gimel: available **now**



- Airflow integration
- Github sharing
- Project collaboration
- Tableau integration



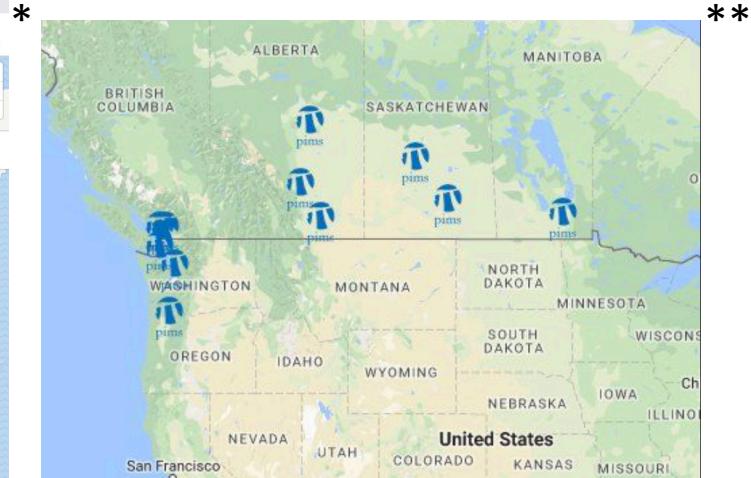
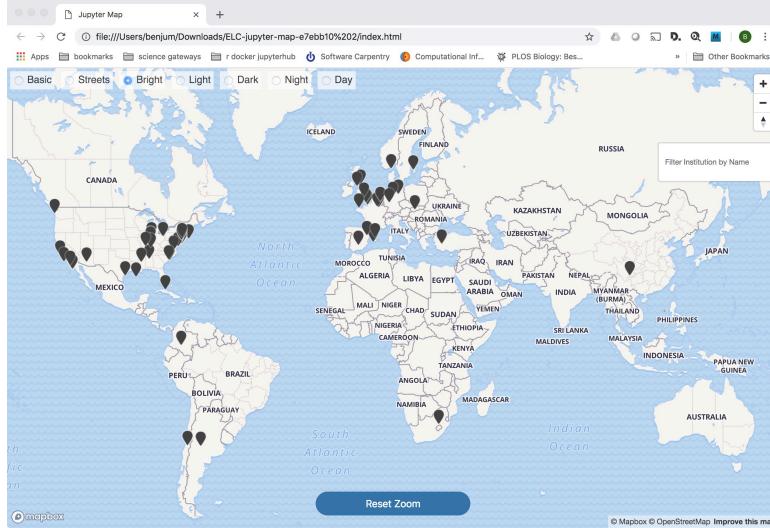
- Config UI
- Dataset browser
- Pipelines



- Data Science Workbench

Jupyter is being used by academic institutions of very high caliber all around the world and in many different disciplines

Institutional JupyterHub deployments, as well as a federation deployment (SYZYGY)



It's actually developing so quickly that many institutions are not on this map.... including UCLA and UC Berkeley!

\* Institutional deployment map generated from data at <https://github.com/ELC/jupyter-map/>

\*\* Canadians Land on Jupyter, JupyterCon 2018, <https://conferences.oreilly.com/jupyter/jup-ny/public/schedule/detail/68396>

# Jupyter Projects

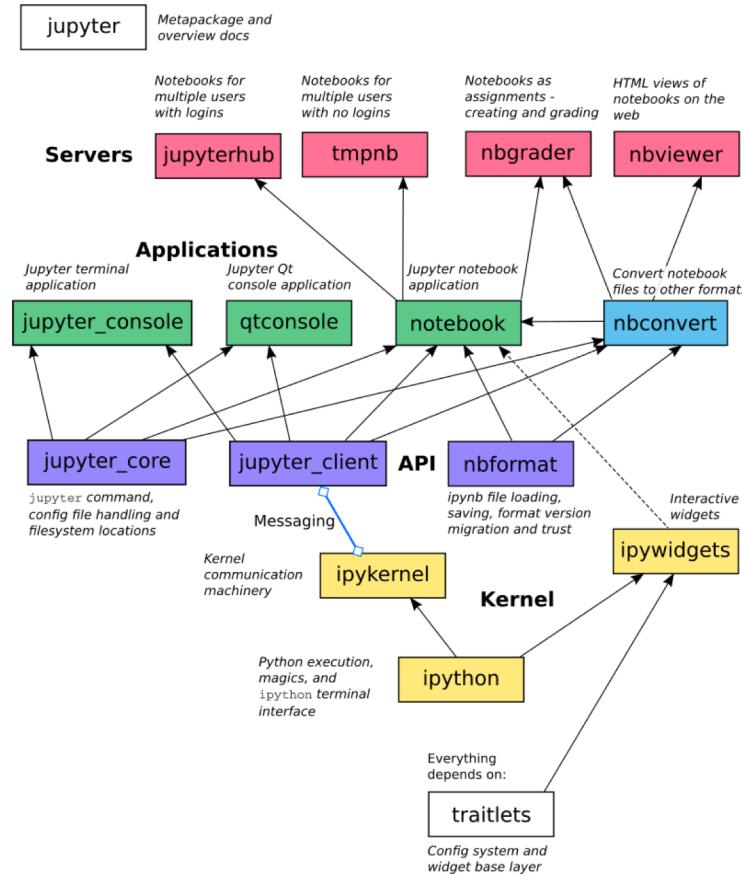


Image: [https://jupyter.readthedocs.io/en/latest/architecture/visual\\_overview.html](https://jupyter.readthedocs.io/en/latest/architecture/visual_overview.html)

# How to share

- jupyter nbconvert --to html mynotebook.ipynb
- Or pdf or .py
- Via menu
- Binder
- nbviewer.jupyter.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 repository name or URL  
 GitHub ▾

Git branch, tag, or commit  
 Path to a notebook file (optional)  
 File ▾

Copy the URL below and share your Binder with others:

Example taken from <https://mybinder.org/>

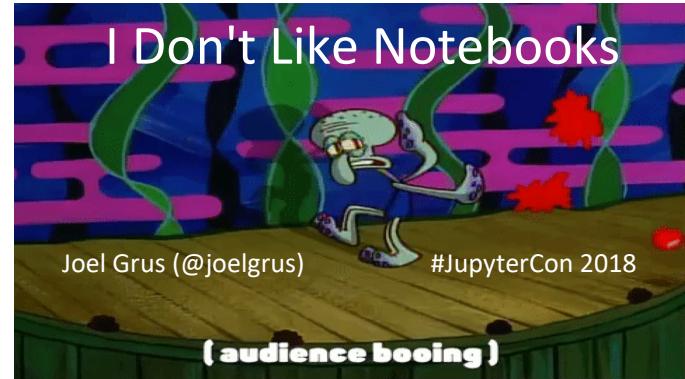
# Not all uses of Jupyter are for clearly documented research and education

A. Rule, A. Tabard, J. Hollan. *Exploration and Explanation in Computational Notebooks*. ACM CHI Conference on Human Factors in Computing Systems, Apr 2018, Montréal, Canada.

- In a scan of > 1 million notebooks on GitHub, 1/4 had no explanatory text
- In analyzing 200 academic notebooks, < 40% discussed reasoning or explained results
- In interviews with 15 academic data analysts, most considered computational notebooks personal, exploratory, and messy

# “Why not to use Jupyter”

- Hidden state and out-of-order execution
- Notebooks are difficult for beginners
- Notebooks encourage bad habits
- Notebooks discourage modularity and testing
- Jupyter’s autocomplete, linting, and way of looking up the help are awkward
- Notebooks encourage bad processes
- Notebooks hinder reproducible + extensible science
- Notebooks make it hard to copy and paste into Slack/Github issues
- Errors will always halt execution
- Notebooks make it easy to teach poorly
- Notebooks make it hard to teach well



# So Beware!

- If you interact with others' notebooks (or even your own past notebooks), there can be hidden state due to:
  - Sloppiness
  - Out-of-order execution
  - Different libraries
  - ....

# Different ways to view notebooks

- Lab notebook
  - Single author... and meant primarily for single viewer
  - Can be split into parts before they get too long
  - Can be split into different topics
  - Not meant to be anything other than place for experimentation and development
- Deliverable report
  - Single- or team-authored
  - Meant to share
  - Fully polished
  - Store the final analysis and outputs
- Interactive playground
  - Carefully crafted educational narratives to invite interaction
  - Interleaving regular notebook cells with test cells

# Novel uses of Jupyter: Can Enable Data Re-use

## Simultaneous Matrix Diagonalization for Structural Brain Networks Classification

*“... We also tried the proposed approach on UCLA Autism dataset [23] and UCLA APOE-4 dataset [2,3]. We note that ...”*

*“For the numerical experiments, we used Python programming language and Jupyter notebook environment. We did matrix calculations and numerical analysis using NumPy, SciPy, NetworkX and igraph libraries. The main classification pipeline was implemented using scikit-learn library [20], which we used for all the classifiers except GBDT. For Gradient Boosted Decision Trees, we used xgboost library [5]. The code is available from authors upon request.”*

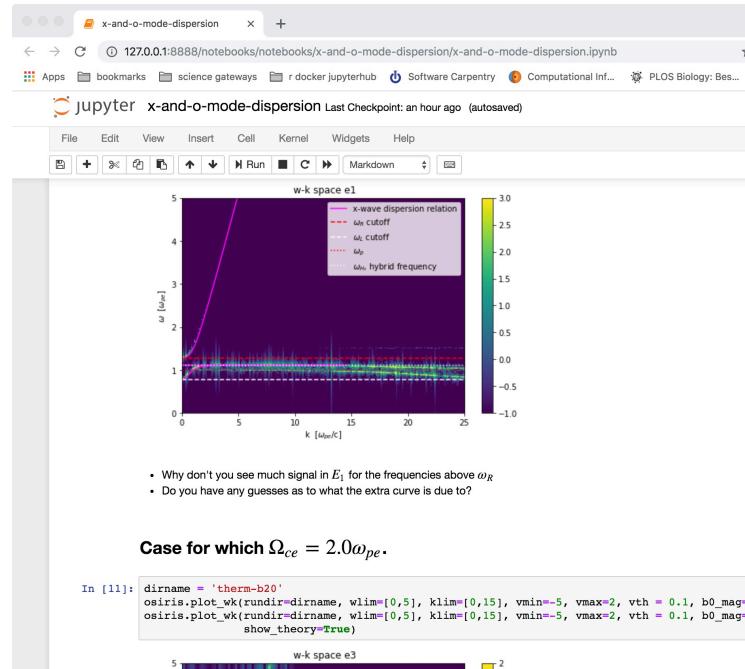
# Novel uses of Jupyter: Integration with Research Software

Used as a data analysis tool of research results, or as a medium in which to run simulations

```
In [17]: import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

and simply plot the data

In [18]: data.plot(x="pos")
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1f5a1cb446d0>
```



# Jupyter and Education



<https://data.berkeley.edu/news/coursefuture>

# Educational Examples

- Interactive notes
  - [12 Steps to Bioinformatics](#), Lorena Barba
- Online textbooks
  - [An Introduction to Applied Bioinformatics](#), Greg Caporaso
- Notebook projects (using research software for education)
  - [JupyterPIC](#) for UCLA EE/Phys M185

# Jupyter and HPC



<https://ccit.clemson.edu/research/jupyter/>

# Connecting Notebooks with HPC Clusters

- University HPC Clusters
  - Like UCLA's Hoffman2
  - <https://www.hoffman2.idre.ucla.edu/access/jupyter-notebook/>
- National Supercomputing Centers
  - Like DOE's NERSC (National Energy Research Scientific Computing Center)
  - <http://www.nersc.gov/users/data-analytics/data-analytics-2/jupyter-and-rstudio/>

# JupyterHub

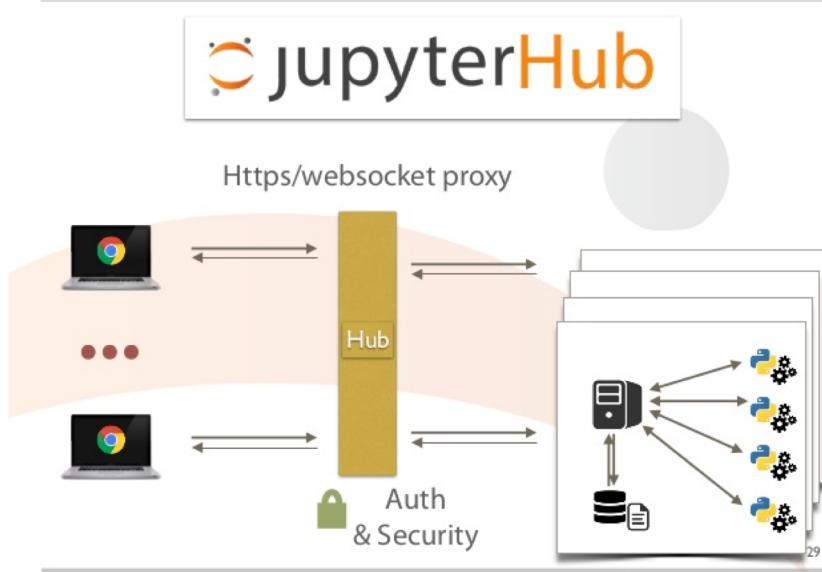
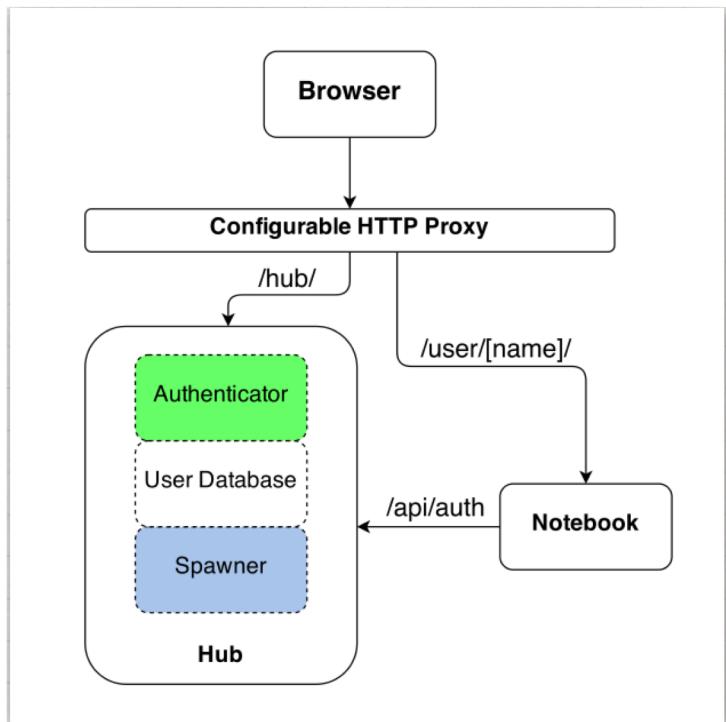


Image: <https://jupyterhub.readthedocs.io/en/stable/>

# Next Generation: JupyterLab

The screenshot shows the JupyterLab interface. On the left is a sidebar with tabs for Files, Running, Commands, Cell Tools, and Tabs. The Files tab shows a list of notebooks: Data.ipynb (an hour ago), Fasta.ipynb (a day ago), Julia.ipynb (a day ago), Lorenz.ipynb (seconds ago, currently selected), R.ipynb (a day ago), iris.csv (a day ago), lightning.json (9 days ago), and lorenz.py (3 minutes ago). The Running tab shows no active processes. The Commands tab lists R.ipynb, iris.csv, lightning.json, and lorenz.py. The Cell Tools tab is empty. The Tabs tab is also empty.

The main area has tabs for Lorenz.ipynb, Terminal 1, Console 1, Data.ipynb, README.md, and Code. The Lorenz.ipynb tab is active. It contains text about the Lorenz system of differential equations:

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

Below this, it says: "Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors."

In [4]:

```
from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```

Output View shows sliders for sigma (10.00), beta (2.67), and rho (28.00). The corresponding Lorenz attractor plot is displayed.

Code view shows the Python code for the Lorenz system:

```
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim(5, 55)

def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
    """Compute the time-derivative of a Lorenz system."""
    x, y, z = x_y_z
    return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

# Choose random starting points, uniformly distributed from -15 to 15
np.random.seed(1)
x0 = -15 + 30 * np.random(N, 3)
```

Image: <https://jupyterlab.readthedocs.io/en/stable/>

Notebooks: in JupyterLab