

# Welcome!

This slide and more info at: <http://bit.ly/JPyBloom>

Before we get started please fill out this poll:

<http://etc.ch/gw2Z>

If you're feeling ambitious, install Anaconda with Jupyter at:  
<https://www.continuum.io/downloads>

IP[y]:  
IPython

# Learning Jupyter

A hands-on night with PyLadies  
at Bloomberg



Jamie Whitacre  
[whitacre@berkeley.edu](mailto:whitacre@berkeley.edu)  
Berkeley Institute for Data Science



Berkeley  
UNIVERSITY OF CALIFORNIA

BIDS  
BERKELEY INSTITUTE  
FOR DATA SCIENCE



# Tonight

A Jupyter overview + demo

Lab time to install Jupyter and try it out!

Jupyter is great for learning Python  
basic knowledge of Python is assumed for this talk

Survey and install info at: <http://bit.ly/JPyBloom>



# Live Resources

Live humans here tonight:

Jupyter Team Members

Friends of Jupyter

Extraordinary PyLadies

Survey and install info at: <http://bit.ly/JPyBloom>



# Other Resources

- **Our website:** [Jupyter.org](https://jupyter.org)
- **Online documentation:**
  - “Official” documentation written by the project: <https://jupyter.readthedocs.io/en/latest/>
  - Documentation written by the community:  
<http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/index.html>
- **Existing Notebooks:**  
<https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks>

Now you can also search for notebooks on GitHub as well



# Lots more! The IPython Gallery

## A gallery of interesting IPython Notebooks

Fernando Perez edited this page 8 days ago · 229 revisions

This page is a curated collection of IPython notebooks that are notable for some reason. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

**Important contribution instructions:** If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using [nbviewer](#), rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that [Matt Davis](#) has conveniently written a set of [bookmarklets and extensions](#) to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

## Table of Contents

1. Entire books or other large collections of notebooks on a topic
  - Introductory Tutorials
  - Programming and Computer Science
  - Statistics, Machine Learning and Data Science
  - Mathematics, Physics, Chemistry, Biology
  - Earth Science and Geo-Spatial data
  - Linguistics and Text Mining
  - Signal Processing
2. Scientific computing and data analysis with the SciPy Stack
  - General topics in scientific computing
  - Social data
  - Psychology and Neuroscience
  - Machine Learning
  - Physics, Chemistry and Biology
  - Economics
  - Earth science and geo-spatial data

## Reproducible academic publications

This section contains academic papers that have been published in the peer-reviewed literature or pre-print sites such as the [ArXiv](#) that include one or more notebooks that enable (even if only partially) readers to reproduce the results of the publication. If you include a publication here, please link to the journal article as well as providing the nbviewer notebook link (and any other relevant resources associated with the paper).

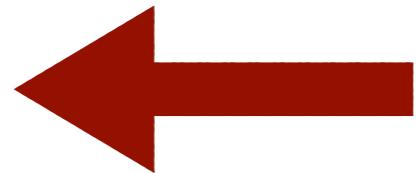
1. [Reply to 'Influence of cosmic ray variability on the monsoon rainfall and temperature': a false-positive in the field of solar-terrestrial research](#) by Benjamin Laken, 2015. Reviewed article will appear in JASTP. The [IPython notebook](#) reproduces the full analysis and figures exactly as they appear in the article, and is available on Github: link via [figshare](#).
2. [The probability of improvement in Fisher's geometric model: a probabilistic approach](#), by Yoav Ram and Lilach Hadany. ([Theoretical Population Biology](#), 2014). An [IPython notebook](#), allowing figure reproduction, was deposited as a [supplementary file](#).
3. [Stress-induced mutagenesis and complex adaptation](#), by Yoav Ram and Lilach Hadany ([Proceedings B](#), 2014). An [IPython notebook](#), allowing figures reproduction, was deposited as a [supplementary file](#).
4. [Automatic segmentation of odor maps in the mouse olfactory bulb using regularized non-negative matrix factorization](#), by J. Soelter et al. ([Neuroimage 2014, Open Access](#)). The [notebook](#) allows to reproduce most figures from the paper and provides a deeper look at the data. The [full code repository](#) is also available.
5. [Multi-tiered genomic analysis of head and neck cancer ties TP53 mutation to 3p loss](#), by A. Gross et al. ([Nature Genetics 2014](#)). The [full collection of notebooks](#) to replicate the results.
6. [powerlaw: a Python package for analysis of heavy-tailed distributions](#), by J. Alstott et al.. Notebook of examples in manuscript, [ArXiv link](#) and [project repository](#).
7. [Collaborative cloud-enabled tools allow rapid, reproducible biological insights](#), by B. Ragan-Kelley et al.. The [main notebook](#), the [full collection of related notebooks](#) and the [companion site](#) with the Amazon AMI information for reproducing the full paper.
8. [A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data](#), by C.T. Brown et al.. Full notebook, [ArXiv link](#) and [project repository](#).
9. [The kinematics of the Local Group in a cosmological context](#) by J.E. Forero-Romero et al.. The [Full notebook](#) and also all the data in a [github repo](#).

# Over 500k Notebooks on GitHub

Check out “Trending Repositories”

The screenshot shows the GitHub Trending page for open source repositories. The top navigation bar includes links for Personal, Open source, Business, Explore, Pricing, Blog, Support, a search bar, and a dashboard icon. Below the navigation, tabs for Showcases, Integrations, Trending (which is selected), and Stars are visible. The main section is titled "Trending in open source" with the sub-instruction "See what the GitHub community is most excited about this week." A dropdown menu for "Trending: this week" is open, showing options for All languages, Unknown languages, Python, and Other: Jupyter Notebook. A "ProTip!" box suggests searching for most starred Jupyter Notebooks. The list of repositories includes:

- guipsamora/pandas\_exercises**: Practice your pandas skills! (Jupyter Notebook, 142 stars this week, built by 1 person)
- aymericdamien/TensorFlow-Examples**: TensorFlow tutorials and code examples for beginners (Jupyter Notebook, 70 stars this week, built by 5 people)
- unnati-xyz/fifthel-2016-workshop**: Content for fifth elephant workshop 2016. Pandas, Luigi, Spark & Flask (Jupyter Notebook, 78 stars this week, built by 3 people)
- martinwicke/tensorflow-tutorial**: A tutorial on TensorFlow (Jupyter Notebook, 62 stars this week, built by 2 people)
- ellisonbg/altair**: Declarative statistical visualization library for Python (Jupyter Notebook, 48 stars this week, built by 5 people)



**ProTip!** Looking for most forked Jupyter Notebook repositories? [Try this search](#)

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



# About Me

- Jupyter Technical Project Manager based out of the Berkeley Institute for Data Science at UC Berkeley
- Background in Bioinformatics & Scientific Data Systems Engineering
- How data can help us tell stories about the earth and everything on it.





What is it?

**jupyter** Welcome to P

**jupyter**

Welcome to the

This Notebook Server was

**WARNING**

Don't rely on this serv

Your server is hosted than

**Run some Python**

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

import pandas as pd
import numpy as np
import matplotlib
```

# jupyter Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help

Python 3

## Exploring the Lorenz System

In this Notebook we explore the [Lorenz system](#) of differential equations:

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

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters ( $\sigma$ ,  $\beta$ ,  $\rho$ ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0., 360.),
               sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0));
```

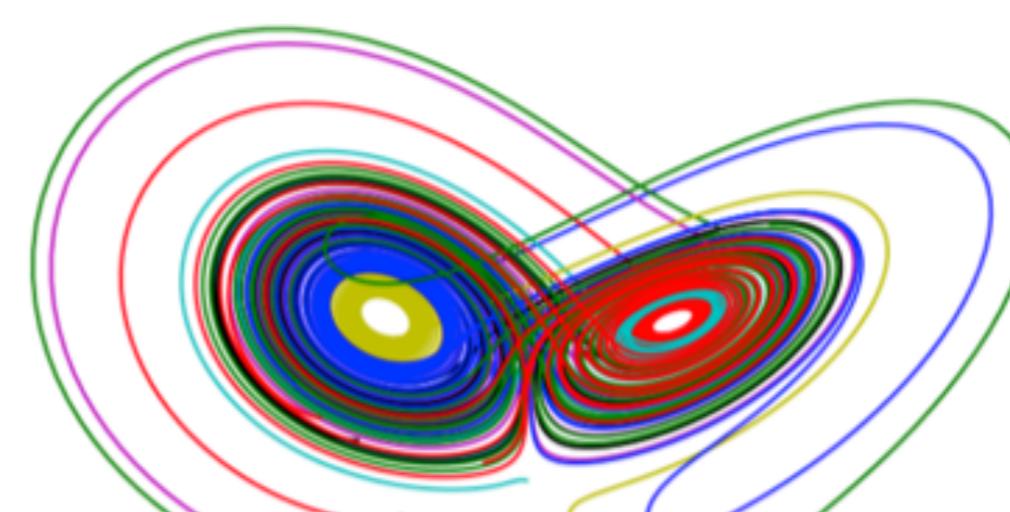
angle

max\_time

$\sigma$

$\beta$

$\rho$





# Jupyter in the wild

- Open source - which means anyone can download it and use it for free.
- Digital Journalism, Scientific research, Commercial products, Classrooms, Coding Bootcamps, Online Learning, Conferences, Data Engineering Pipelines, High Performance Computing (HPC) . . .
- Physics, Astronomy, Biology, Economics & Finance, Social Sciences, Geo Sciences, Digital Humanities . . .



# Jupyter Notebook

- A digital “document” that gives you a way to record your thoughts and CODE in a single place
- Provides a window into your data, serves as a tool to explore, understand, communicate, and tell stories about your data in a systematic and reproducible way
- Is the evolution of the IPython Notebook into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.
- Allows you to use the many Python libraries available to developers



# Jupyter Protocol is language agnostic

 Scala



 Julia

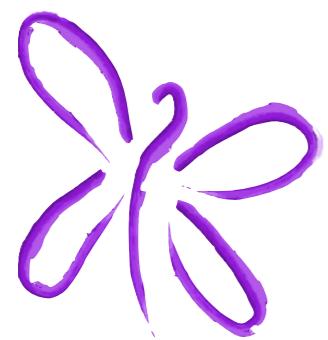


 perl

 R



 julia

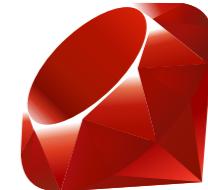


 Spark

 ERLANG



 python™





# Jupyter Notebook

- A digital “document” that gives you a way to record your thoughts and CODE in a single place
- Provides a window into your data, serves as a tool to explore, understand, communicate, and tell stories about your data in a systematic and reproducible way
- Is the evolution of the IPython Notebook into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.
- Allows you to use the many Python libraries available to the Python community



# Jupyter Notebook

- Notebook is JSON sequence of text cells & code cells
- Publicly documented
- ~~textcell~~ = markdown + latex
- Machine readable, easy
- ~~to understand~~ REP (input + output)
- Transformable
- ~~(meta)data are~~ everywhere

The screenshot shows a Jupyter Notebook interface with the title "Sampling\_Theorem (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Help, and Logout. A Python icon is in the top right. The main content area contains the following text:

### Investigating the Sampling Theorem

In this section, we investigate the implications of the sampling theorem. Here is the usual statement of the theorem from wikipedia:

*"If a function  $x(t)$  contains no frequencies higher than  $B$  hertz, it is completely determined by giving its ordinates at a series of points spaced  $1/(2B)$  seconds apart."*

Since a function  $x(t)$  is a function from the real line to the real line, there are uncountably many points between any two ordinates, so sampling is a massive reduction of data since it only takes a tiny number of points to completely characterize the function. This is a powerful idea worth exploring. In fact, we have seen this idea of reducing a function to a discrete set of numbers before in Fourier series expansions where (for periodic  $x(t)$ )

$$a_n = \frac{1}{T} \int_0^T x(t) \exp(-j\omega_n t) dt$$

with corresponding reconstruction as:

$$x(t) = \sum_k a_n \exp(j\omega_n t)$$

But here we are generating discrete points  $a_n$  by integrating over the **entire** function  $x(t)$ , not just evaluating it at a single point. This means we are collecting information about the entire function to compute a single discrete point  $a_n$ , whereas with sampling we are just taking individual points in isolation.



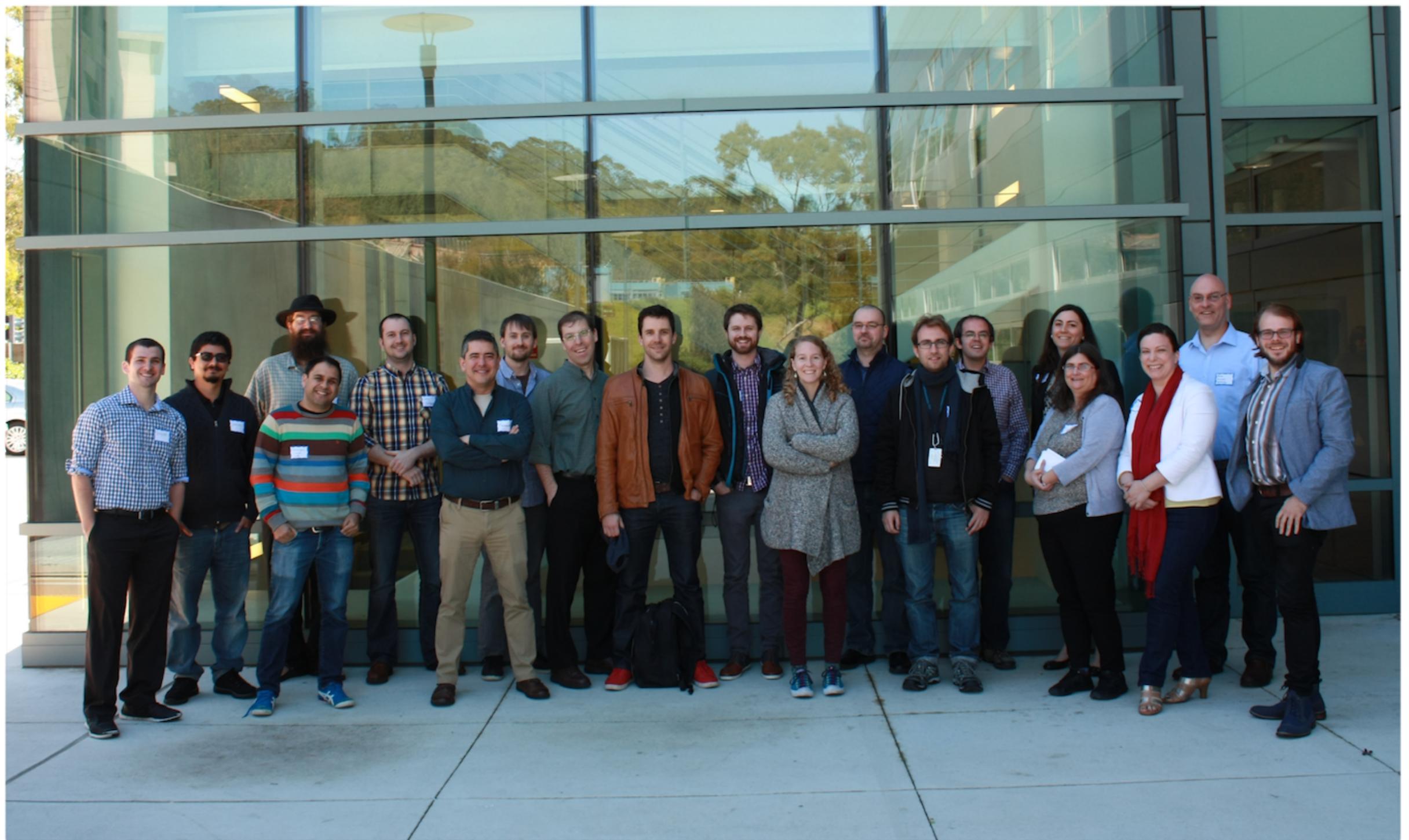
# A Tool for Interactive Computing

- A dialogue between the human and the computer.
- Assemble ideas using the computer as playground, as “data microscope”
- Building blocks for scientific computing and data science

# Project Jupyter

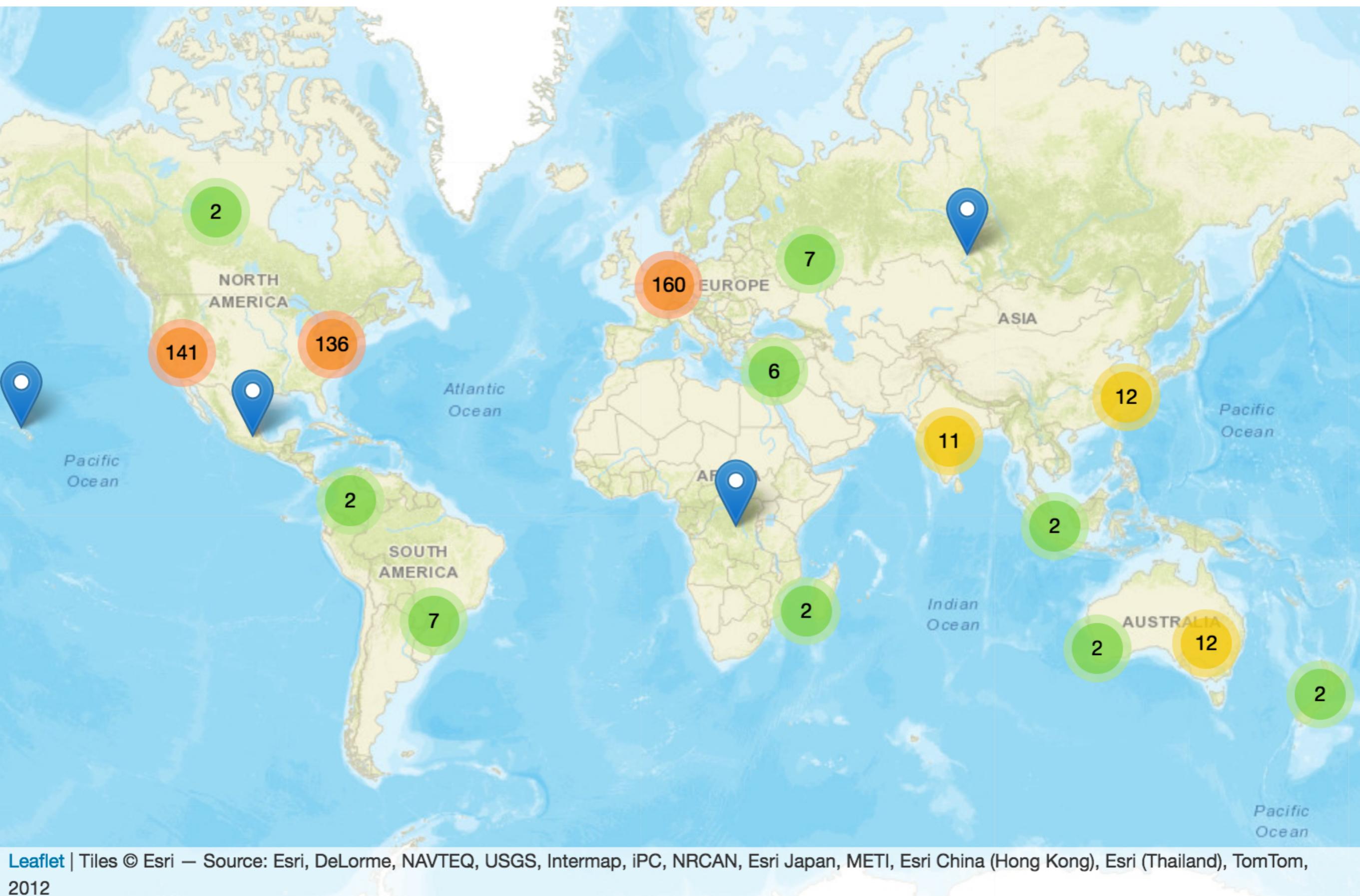


# Jupyter team: where *all the credit* goes



Plus ~ 500 more Open source contributors!

# Committers to IPython/Jupyter repositories



By Stuart Geiger, Jamie Whitacre, Matthias Bussonnier, and Nami Saghaei



# Funding and partnerships



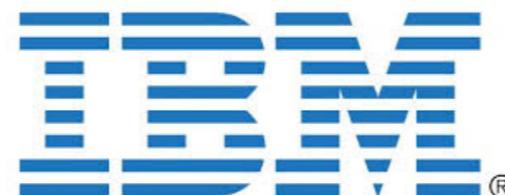
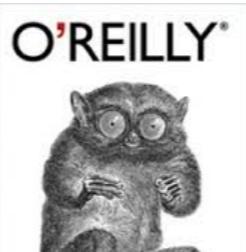
ALFRED P. SLOAN  
FOUNDATION

GORDON AND BETTY  
**MOORE**  
FOUNDATION



CONTINUUM  
ANALYTICS

 ENTHOUGHT  
SCIENTIFIC COMPUTING SOLUTIONS



Google

THE LEONA M. AND HARRY B.  
**HELMSLEY**  
CHARITABLE TRUST

U.S. DEPARTMENT OF  
**ENERGY**

NEW! We're hiring!

SIMONS FOUNDATION

 POWERED BY  
**rackspace**<sup>®</sup>  
*the open cloud company*



**Microsoft**  
**Bloomberg**

# Notebook Workflows: The Big Picture

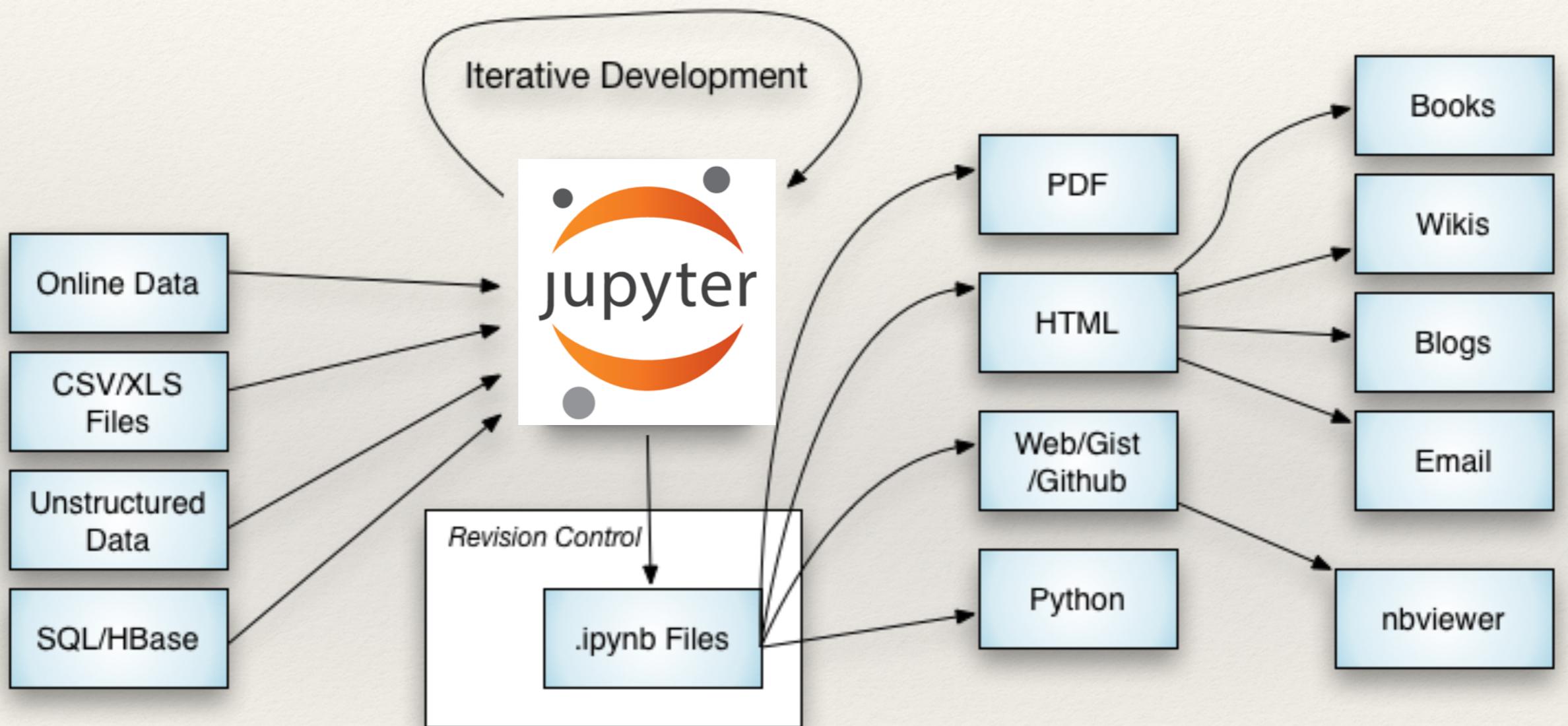


Image credit: [Joshua Barratt](#)



# Data Analysis Workflow

1. Pose a question or problem
2. Acquire data
3. Explore the data by writing & running code
4. Prepare and clean the data
5. Complete final analysis & visualize results
6. Write up analysis for publication (blogs, journals, etc.)
7. Share what you've done
8. Reproduce what other people have done



# Berkeley's *Foundations of Data Science*

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



The Foundations of Data Science — ds8

ABOUT 4 DISCUSSIONS

★ Star 3 Subsribe 1

About this book Readme Table of Contents Read PDF ePub Mobi

Last edit was 7 hours ago

## Arrays

**Interact**

Many experiments and data sets involve multiple values of the same type. An array is a collection of values that all have the same type. The `numpy` package, abbreviated `np` in programs, provides Python programmers with convenient and powerful functions for creating and manipulating arrays.

```
import numpy as np
```

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

```
baseline_high = 14.48
highs = np.array([baseline_high - 0.880,
                 baseline_high - 0.093,
                 baseline_high + 0.105,
                 baseline_high + 0.684])
```

`highs`

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

## Histograms

**Interact**

### Quantitative Data and Histograms

Many of the variables that data scientists study are *quantitative*. For instance, we can study the amount of revenue earned by movies in recent decades. Our source is the [Internet Movie Database](#), an online database that contains information about movies, television shows, video games, and so on.

The table `top` consists of [U.S.A.'s top grossing movies (<http://www.boxofficemojo.com/alltime/adjusted.htm>) of all time. The first column contains the title of the movie; *Star Wars: The Force Awakens* has the top rank, with a box office gross amount of more than 900 million dollars in the United States. The second column contains the name of the studio; the third contains the U.S. box office gross in dollars; the fourth contains the gross amount that would have been earned from ticket sales at 2016 prices; and the fifth contains the release year of the movie.

There are 200 movies on the list. Here are the top ten.

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



# 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

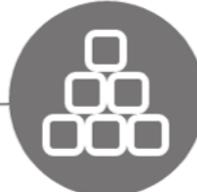
Manage users and authentication with PAM, OAuth or integrate with your own directory service system.

Collaborate with others through the Linux permission model.



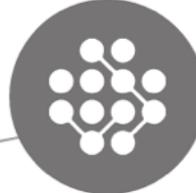
### Centralized deployment

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.



# LIGO: Open Science with Jupyter

The diagram illustrates the integration of three platforms:

- Microsoft Azure Notebooks**: A white card on the left shows the Azure Notebooks interface with a "PREVIEW" button.
- Jupyter**: A white card below the Azure card displays the text "Notebooks hosted on Microsoft Azure".
- LIGO Open Science Center**: A central browser window shows the LIGO website (<https://losc.ligo.org/tutorials/>). The "Tutorials" section features three examples:
  - Binary Black Hole Events**: Shows a waveform plot. Action buttons: "Run: Azure" (blue circle) and "mybinder" (orange circle). Options: "View: GW150914 | LVT151012 | GW151226" and "Download: zip file with data | IPython 4 | IPython 3 | python script".
  - Quickview Notebook**: Shows a spectrogram. Action buttons: "Run: Azure" (blue circle) and "mybinder" (orange circle). Option: "Download: IPython 4".
  - Signal Processing with GW150914**: Shows a waveform plot. Action buttons: "Run: Azure" (blue circle) and "mybinder" (orange circle). Options: "View: HTML" and "Download: zip file with data".
- binder**: A white card on the right shows the binder logo and the word "binder".

Blue arrows point from the Microsoft Azure Notebooks card to the "Run: Azure" buttons in the LIGO tutorials. Orange arrows point from the Jupyter card to the "mybinder" buttons in the LIGO tutorials.



# nbviewer: seamless notebook sharing

- ❖ Zero-install reading of notebooks
- ❖ Just share a URL
- ❖ [nbviewer.jupyter.org](http://nbviewer.jupyter.org)

The screenshot shows the nbviewer homepage. At the top, there's a navigation bar with links to nbviewer, FAQ, IPython, and Jupyter. Below that is the main title "nbviewer" and the subtitle "A simple way to share Jupyter Notebooks". A search bar at the top right accepts URLs, GitHub usernames, GitHub username/repo pairs, or Gist IDs, with a "Go!" button. The page is divided into several sections displaying different notebook examples:

- Programming Languages:** Examples for IPython, IRuby, and IJulia.
- Books:** Examples for "Python for Signal Processing" (O'Reilly), "Mining the Social Web" (O'Reilly 2nd Edition), and "Probabilistic Programming & Bayesian Methods for Hackers".
- Misc:** Examples for "Data Visualization with Lightning", "Interactive data visualization with Bokeh", and "Interactive plots with Plotly".

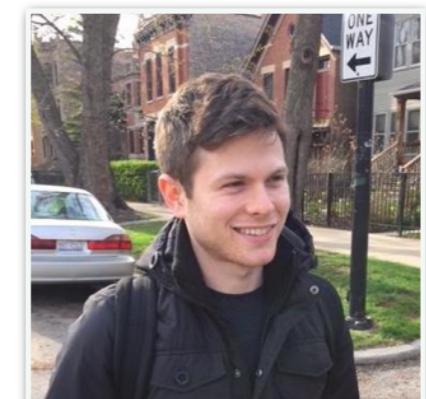


# mybinder.org

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



[github.com/freeman-lab](https://github.com/freeman-lab)



[github.com/andrewosh](https://github.com/andrewosh)

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

These foundations have  
become infrastructure



# Microsoft, IBM, Google, Continuum...



**Machine Learning Blog**  
Introducing Jupyter Notebooks in Azure ML Studio

of Program Management at Microsoft.

position of machine learning experiments and option. Although the Studio provides an easy to experiments, you sometimes need a good old script code and get a response. I am delighted tionality into ML Studio through Jupyter

**Related Links**

- Microsoft Azure ML
- Microsoft Data Platform Insider Blog
- Microsoft Big Data Solutions
- Data Science Dojo
- Azure Big Data Blog

**Tags**

- ADF
- asa
- Azure ML
- Azure Stream Analytics
- Cortana Analytics Suite
- Customers
- Data
- Science
- Machine
- Learning
- Partners
- Python
- r
- Webinar

More ▾ Less ▲

**WHAT IS JUPYTER?**

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

**ABOUT THIS SERVICE**

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

**Google Cloud Databar**

Notebooks Sessions

Notebook Folder Upload

/ databar / intro

Introduction to Notebooks.ipynb

Introduction to Python.ipynb

Working with Databar.ipynb

git Repository

Running

**IBM**

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

Prepare data effortlessly.

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

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

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

Analyze data interactively.

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

**Cloud Databar** BETA

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

TRY IT FREE

**Powerful Data Exploration**

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

# Google

**CONTINUUM ANALYTICS**

Gallery About Pricing Anaconda Help Download Anaconda Sign In

CONTINUUM ANALYTICS

Gallery About Pricing Anaconda Help Download Anaconda Sign In

Interactive Stock Prices Downsampling

Hover Over Points

My Gist Activity

Data Visualization in Python

Scientific Programming in Python

Texas Unemployment Choropleth

Boston area cities and towns with their population

Texas Unemployment 2009

# In Development



# Dashboards

<https://github.com/jupyter-incubator/dashboards>

The screenshot shows a Jupyter Notebook interface with the title "meetup-streaming" and the URL "jupyter.cloudet.xyz/user/AnY1jQel3oyG/notebooks/dashboards/stream\_demo/meetup-streaming.ipynb". The notebook is titled "jupyter meetup-streaming (autosaved)" and uses Python 3.

The dashboard contains the following components:

- Streaming Meetups Dashboard**: A title card with a "Stream" toggle switch.
- Park**: A section featuring a thumbnail image of people playing volleyball, the text "Open gym volleyball for all levels of play.", and the date "Wednesday, March 9, 2016, 8:30 PM".
- Filter**: A text input field for filtering data.
- Global Meetup Locations**: A world map showing numerous colored lines radiating from various locations, indicating the flow of data or RSVPs.
- Bar Chart**: A histogram showing the distribution of data over time. The y-axis ranges from 300.0 to 1,100.0. The x-axis has a single data point labeled "1". The chart includes options for "Grouped" and "Stacked" bar types.

# Jupyter Dashboards & Declarative Widgets

Will help users to:

- Pull interactive web component widgets into a notebook
- Arrange widgets in a grid- or report-like layout
- Bundle notebooks and widgets for deployment as dashboards
- Serve notebook-defined dashboards as standalone web apps

# nbdime

## notebook diff & merge

- tools for diffing and merging notebooks
- command-line rendering of diffs (outputs elided)  
`nbdiff`
- html rendering of rich diffs (JupyterLab)  
`nbdiff-web`
- git integration  
`git-nbdiffdriver`

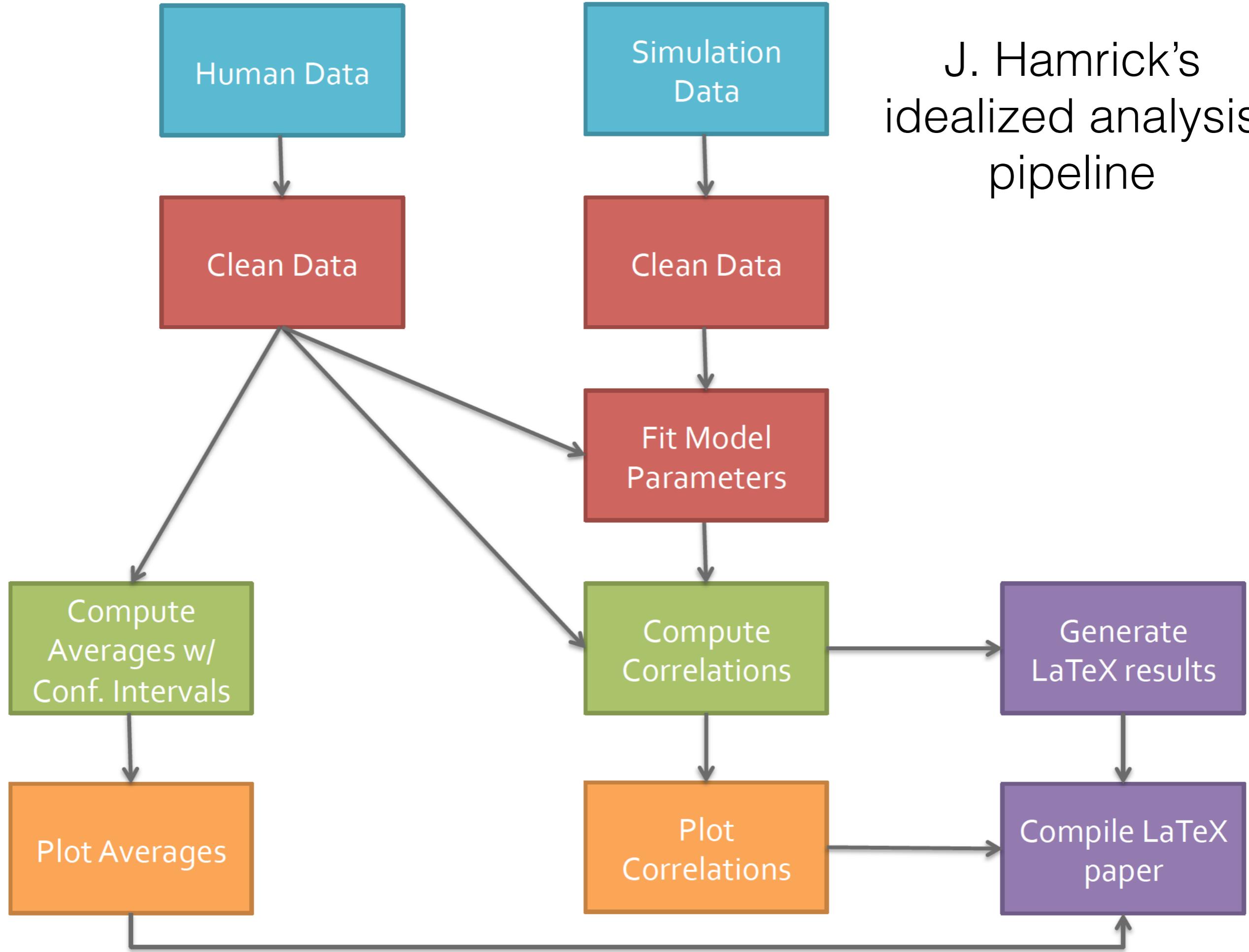
<https://github.com/jupyter/nbdime>

# nbflow

## Jess Hamrick

- Reproducible, One-Button Workflows with the Jupyter Notebook and SCons
- Ideal analysis pipelines
- Notebooks can be SCons commands  
(Python 2 only for now)
- Jess' nbflow repo:  
<http://tinyurl.com/nbflow-example>
- For Jess' SciPy 2016 talk:  
[youtube.com/watch?v=Fc2W930NJs8](https://www.youtube.com/watch?v=Fc2W930NJs8)

# J. Hamrick's idealized analysis pipeline



# JupyterLab:

## Building Blocks for Interactive Computing



Brian E. Granger, Cal Poly  
Jason Grout, Bloomberg LP  
Chris Colbert, Continuum  
Sylvain Corlay, Bloomberg  
Afshin Darian, Continuum

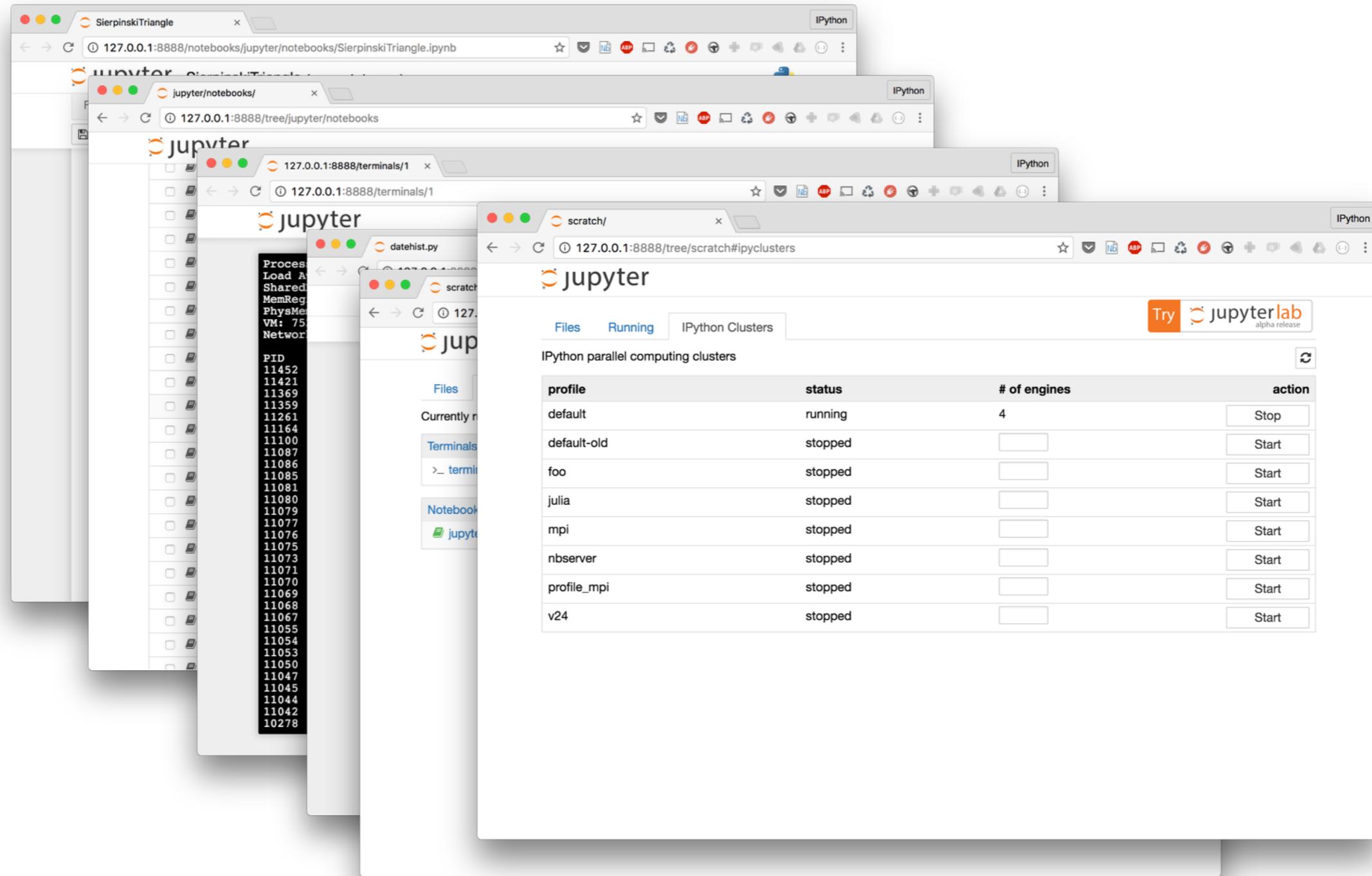
Cameron Oelsen, Cal Poly  
Fernando Perez, LBNL/  
Berkeley  
Steven Sylvester, Continuum  
David Willmer

# JupyterLab

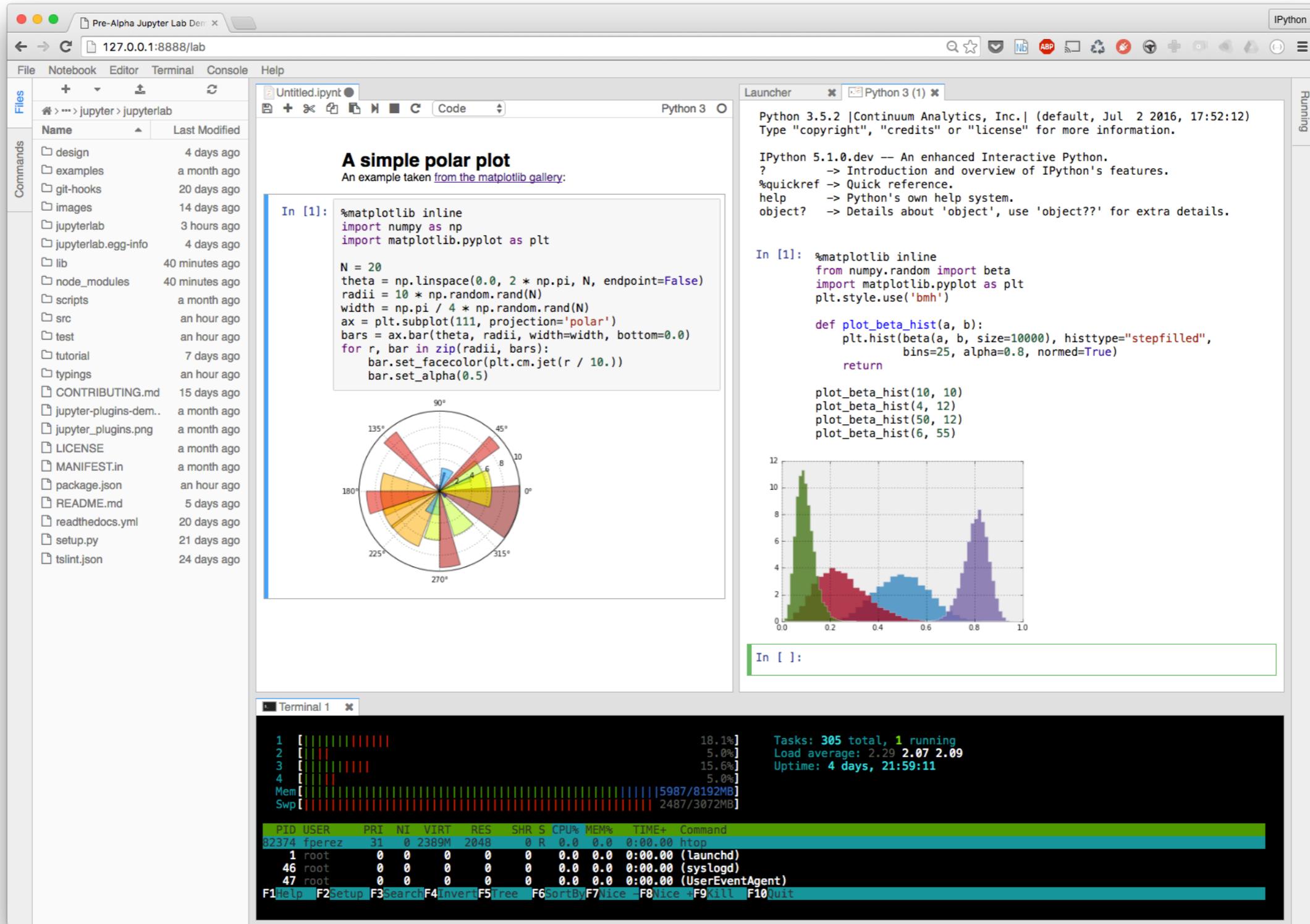
<https://github.com/jupyter/jupyterlab>

- JupyterLab is the natural evolution of the Jupyter Notebook user interface
- JupyterLab is an IDE: *Interactive* Development Environment
- Flexible user interface for assembling the fundamental building blocks of interactive computing
- Modernized JavaScript architecture based on npm/webpack, plugin system, model/view separation
- Built using PhosphorJS (<http://phosphorjs.github.io/>)
- Design-driven development process

# The “Notebook”?



# JupyterLab: unifying these ideas



<https://raw.githubusercontent.com/jupyter/jupyterlab/master/jupyter-plugins-demo.gif>



# GitHub

[jupyter/jupyterlab\\*](#)

\*JupyterLab is a early developer preview, and is not suitable for general usage yet. Features and implementation are subject to change.





# Learn More About Jupyter

- [jupyter.org](http://jupyter.org)
- [jupyter.readthedocs.org](http://jupyter.readthedocs.org)
- GitHub repos
- O'Reilly Tutorials
  - Advanced Jupyter Notebook Deployment by Jonathan Frederic
  - Jupyter Notebook for Data Science Teams by Jonathan Whitmore
- Mailing lists
- Conferences: PyData, SciPy, JupyterDays, PyCon, Strata
- YouTube
- Google it

Create tools for Data Science!  
Contribute to Jupyter!

jupyter.org

github.com/jupyter

# Thank You

[whitacre@berkeley.edu](mailto:whitacre@berkeley.edu)

[@jupyterJW](https://twitter.com/jupyterJW)

[@ProjectJupyter](https://twitter.com/ProjectJupyter)    [@IPythonDev](https://twitter.com/IPythonDev)

Try it out at

[try.jupyter.org](https://try.jupyter.org)



# Jupyter Walkthrough

- Layout of screens
  - Notebook
  - File directory
  - other
- Basic Features
  - Create notebook
  - Add Cell
  - Execute Cell
  - Delete Cell
  - Keyboard shortcuts
  - Cell Magic
- Working with Data
  - Plots
  - Tables
  - Widgets
  - Advanced
  - Multi-language
  - R



# Install Jupyter

- Please take this poll before we get started:

<http://etc.ch/gw2Z>

- Easiest way to install Jupyter is by going to [continuum.io/downloads](https://continuum.io/downloads) and downloading Anaconda
- Anaconda is a free, open source scientific computing package that will put all the good stuff on your computer without much trouble

🔍 | Spotlight Search



🔍 Terminal



TOP HIT

Terminal

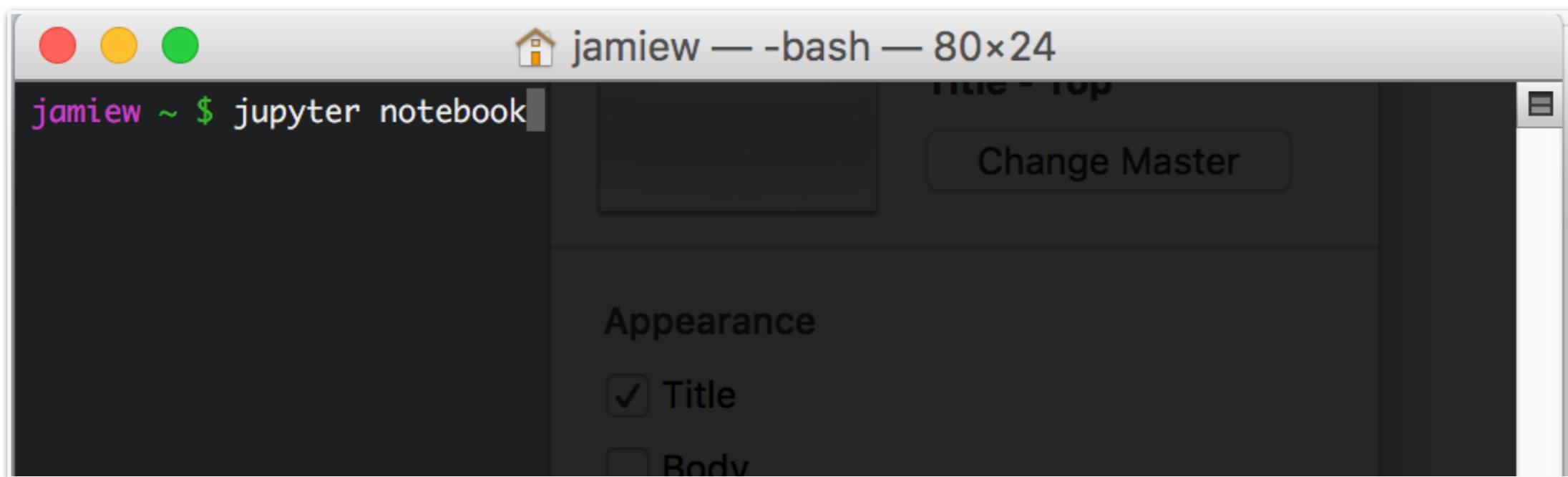
FOLDERS

- 📁 terminal - jamiew
- 📁 terminal - jamiew
- 📁 terminal - notebook-4.2.3-py27\_0
- 📁 terminal - jamiew
- 📁 terminal - src
- 📁 terminal - examples
- 📁 terminal - notebook-4.2.2-py27\_0



Terminal

Version: 2.6.1



localhost:8888/tree

jupyter

Files    Running    Clusters

Select items to perform actions on them.

Upload    New   

<input type="checkbox"/>	<input type="checkbox"/>	
2015		
<input type="checkbox"/>	anaconda	
<input type="checkbox"/>	Applications	
<input type="checkbox"/>	Biorepository Legacy Migration	
<input type="checkbox"/>	bokeh-notebooks-master	
<input type="checkbox"/>	book-exercises	
<input type="checkbox"/>	Coursera-HowToUseGitandGitHub	
<input type="checkbox"/>	datasciencecoursera	
<input type="checkbox"/>	Desktop	
<input type="checkbox"/>	development	
<input type="checkbox"/>	Documents	



Files    Running    Clusters

Select items to perform actions on them.

- 
- 
- 
- 

Upload

New ▾



Text File

Folder

Terminal

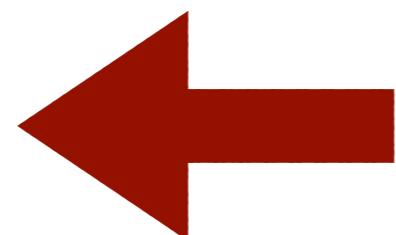
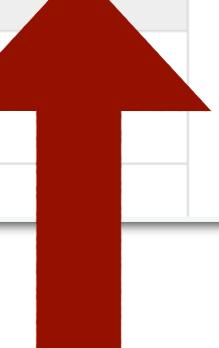
Notebooks

Python 2

Python 3

R

Upload New ▾





# jupyter Untitled1 Last Checkpoint: a few seconds ago (unsaved changes)



File Edit View Insert Cell Kernel Help

Python 2



In [ ]:

# jupyter Untitled1 Last Checkpoint: 2 minutes ago (autosaved)

File

Edit

View

Insert

Cell

Kernel

Help



Code



CellToolbar

In [ ]:



# General Commands

- Define a variable

```
x = 5
```

```
print(x)
```



# Making Charts

<http://matplotlib.org/users/screenshots.html#simple-plot>

- Simple plot
- Histograms
- Bar Charts
- Scatter plot for fun

Other great plotting libraries: plot.ly, Seaborn, Altair, Bokeh



# Data Frames

<http://pandas.pydata.org/pandas-docs/stable/10min.html>

[github.com/jehuston/pandas\\_tutorial/blob/master/pandas\\_tutorial.ipynb](https://github.com/jehuston/pandas_tutorial/blob/master/pandas_tutorial.ipynb)

<https://github.com/brandon-rhodes/pycon-pandas-tutorial>

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

# R kernel for Jupyter

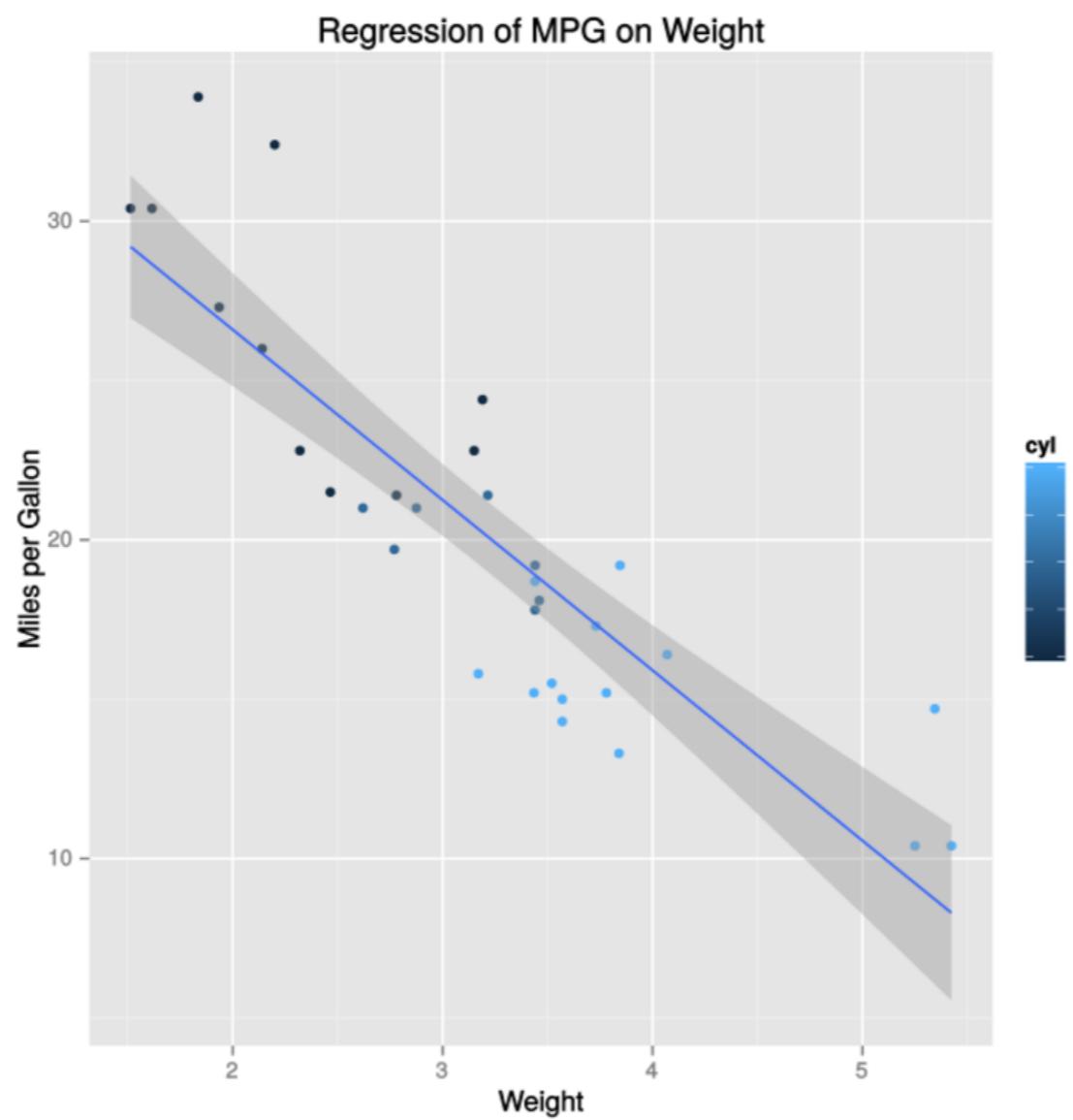


<https://irkernel.github.io/>

```
In [1]: library(ggplot2)

In [2]: qplot(wt, mpg, data=mtcars, geom=c('point', 'smooth'),
           method='lm', formula=y~x, color=cyl,
           main='Regression of MPG on Weight',
           xlab='Weight', ylab='Miles per Gallon')
```

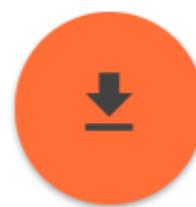
<https://try.jupyter.org/>



```
In [ ]:
```



REQUIREMENTS



INSTALLATION



RUNNING



FAQ

# Requirements

To run Jupyter with an R kernel, you need at least the following:

- [Jupyter](#). If you already have a Python environment set up, install Jupyter using your preferred tools. If not, installing [Anaconda](#) is the quickest way to get everything you need.
- A current [R installation](#).



[Home]

[Download](#)

[CRAN](#)

[R Project](#)

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Development Site](#)

[Conferences](#)

[Search](#)

[R Foundation](#)

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

[Help With R](#)

[Getting Help](#)

[Documentation](#)

[Manuals](#)

[FAQs](#)

[The R Journal](#)

[Books](#)

[Certification](#)

[Other](#)

# The R Project for Statistical Computing

<https://www.r-project.org/>

## Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

## News

- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **R version 3.3.2 (Sincere Pumpkin Patch)** has been released on Monday 2016-10-31.
- **The R Journal Volume 8/1** is available.
- The **useR! 2017** conference will take place in Brussels, July 4 - 7, 2017, and details will be appear here in due course.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, has taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

# Installation

BINARY

SOURCE

## 1/2) Installing via supplier (default on Windows + Mac)

You can install all packages using the following lines in an R session:

```
install.packages(c('repr', 'IRdisplay', 'IRkernel'))  
devtools::install_github('IRkernel/IRkernel')  
# Don't forget step 2/2!
```

**Important!**



On OS X, be sure to execute this in R started from the Terminal, not the R App!  
(This is because the R app doesn't honor \$PATH changes in ~/.bash\_profile)



To update the IRkernel package, which is not yet on CRAN, you have to rerun the devtools:: line. For the other packages, a simple update.packages() is sufficient.

## 2/2) Making the kernel available to Jupyter

If you haven't done this already, you will have to make Jupyter see the newly installed R kernel by installing a kernel spec.

The kernel spec can be installed for the current user with the following line from R:

```
IRkernel::installspec()
```

To install system-wide, set user to False in the installspec command:

```
IRkernel::installspec(user = FALSE)
```



# Exercises

```
import matplotlib.pyplot as plt  
plt.plot(range(4))  
plt.show() —> [shift + enter]
```

Find a Notebook you like in the *Gallery of Interesting IPython Notebooks* and try to reproduce the analysis. Make subtle changes or import a completely different data set to see what happens. Share with your neighbors.

Additional resources including Notebooks by PyLadies:  
<https://github.com/willingc/workshop-help>

Thanks!  
Happy Jupytering!

