



# In a Nutshell

JupyterDay in the Triangle  
November 13, 2018

Peter Parente (@parente)  
Project Jupyter Steering Council

# What is Jupyter?



- The Project
- The Technology
- The Community
- And You

# But first ... what can we do with Jupyter?



# The Project



# Motivations

- Open science
- Reproducible research
- Computational thinking
- Computational narratives



The collage includes:

- A Jupyter Notebook interface showing code to import pandas and read a CSV file named "guests.csv".
- A Jupyter Notebook interface showing a DataFrame with columns "game\_of\_thrones", "tacos", and "grape\_cotton\_candy".
- A screenshot of a Twitter post from Mark Cuban (@mcuban) titled "The new NBA" featuring a photo of a laptop displaying a Jupyter Notebook.
- A Jupyter Notebook interface titled "Regex Golf with Arbitrary Lists" showing a function definition for "report" and its usage.
- A Project Jupyter tweet from @ProjectJupyter dated 10:46 AM - 12 Feb 2016, announcing progress towards open science and reproducibility, specifically mentioning LIGO's release of data and analysis as a Jupyter notebook.

# Project Jupyter



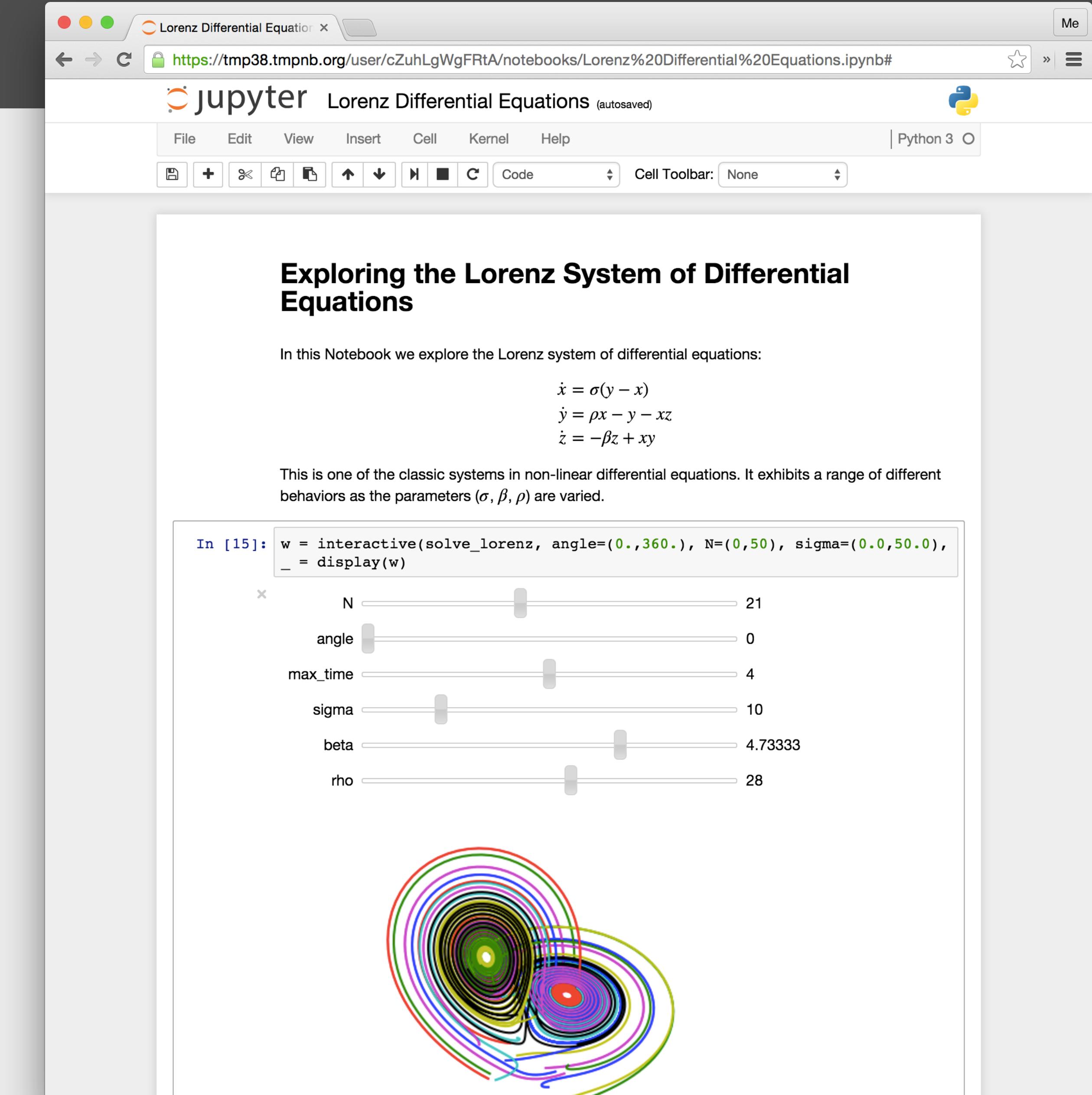
- Born out of the IPython project circa 2011
- Open source software to support interactive computing
- Open formats, protocols, APIs to build an ecosystem
- Open community to promote use and contribution

# The Technology



# Jupyter Notebooks

Text  
+  
Code  
+  
Visualizations  
+  
Widgets  
+  
Compute  
+  
Data



# Building Blocks



Services

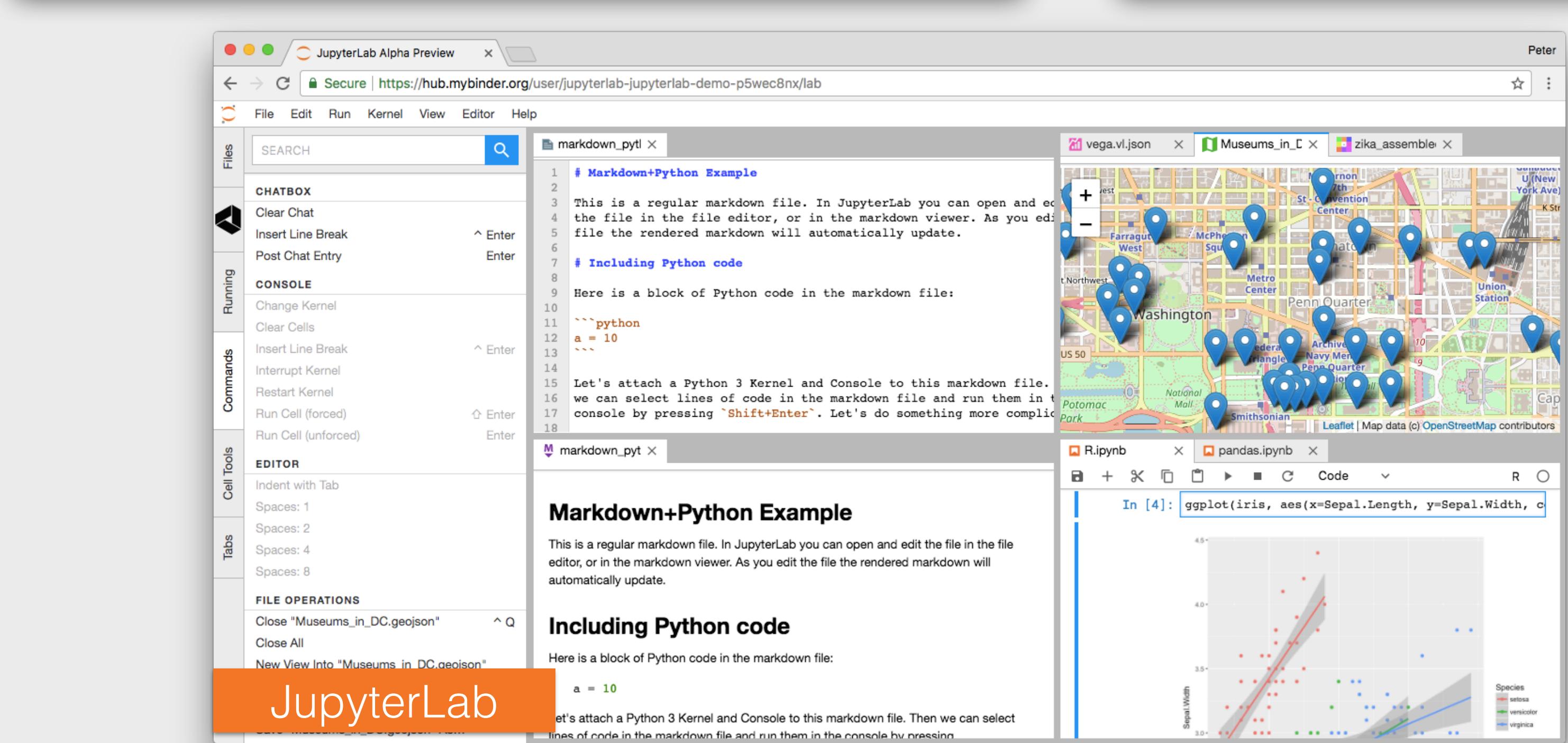
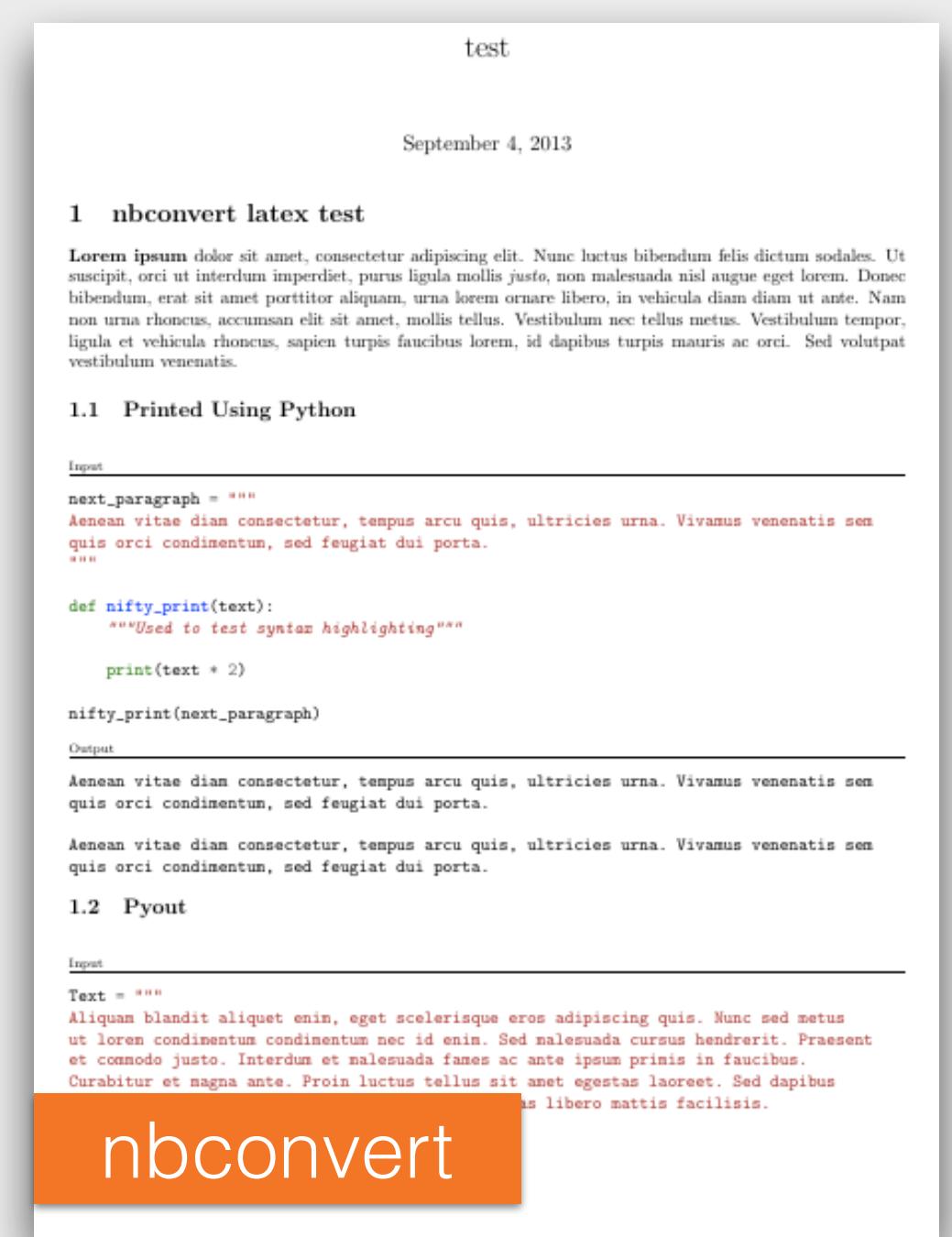
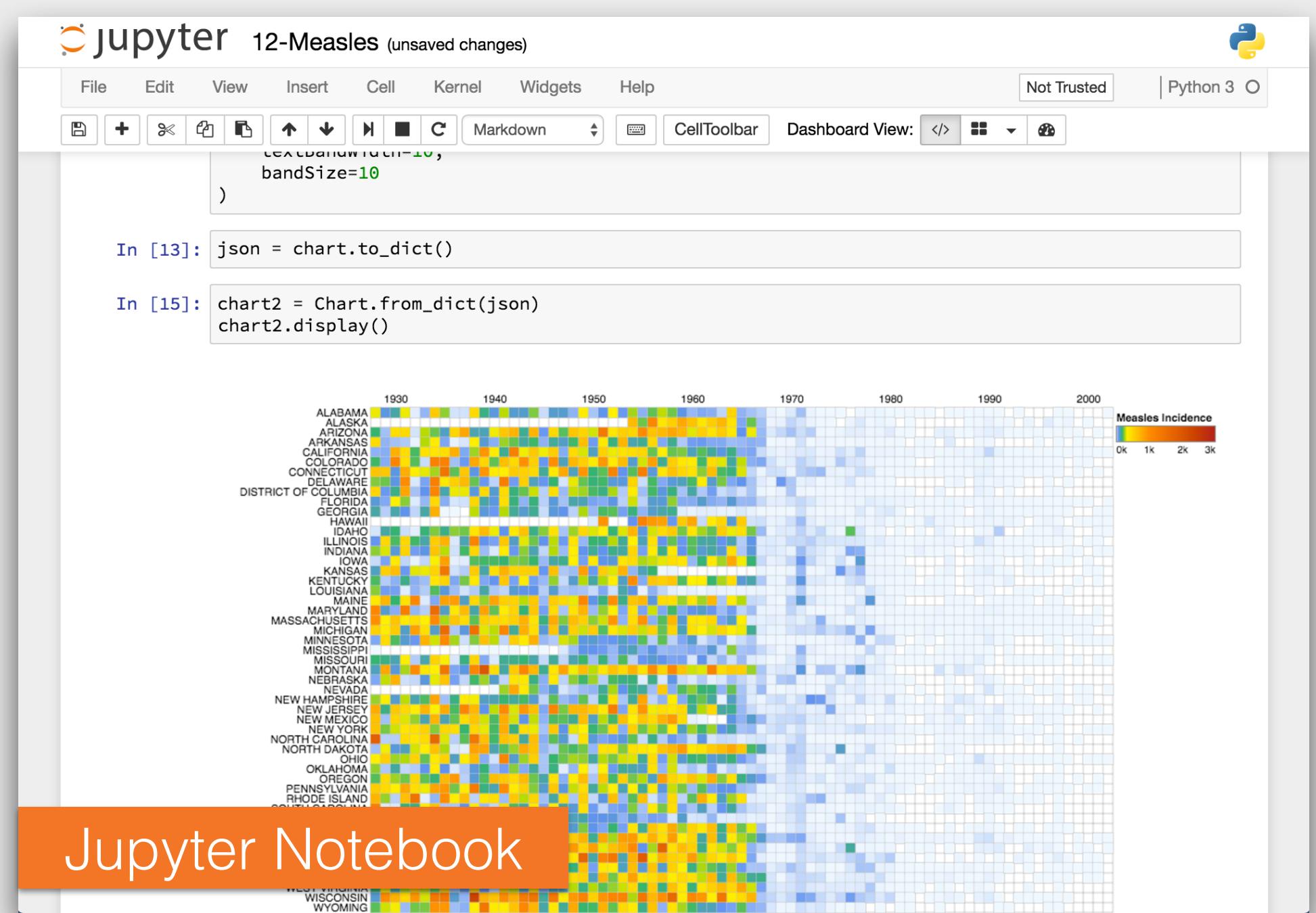
Applications

Libraries

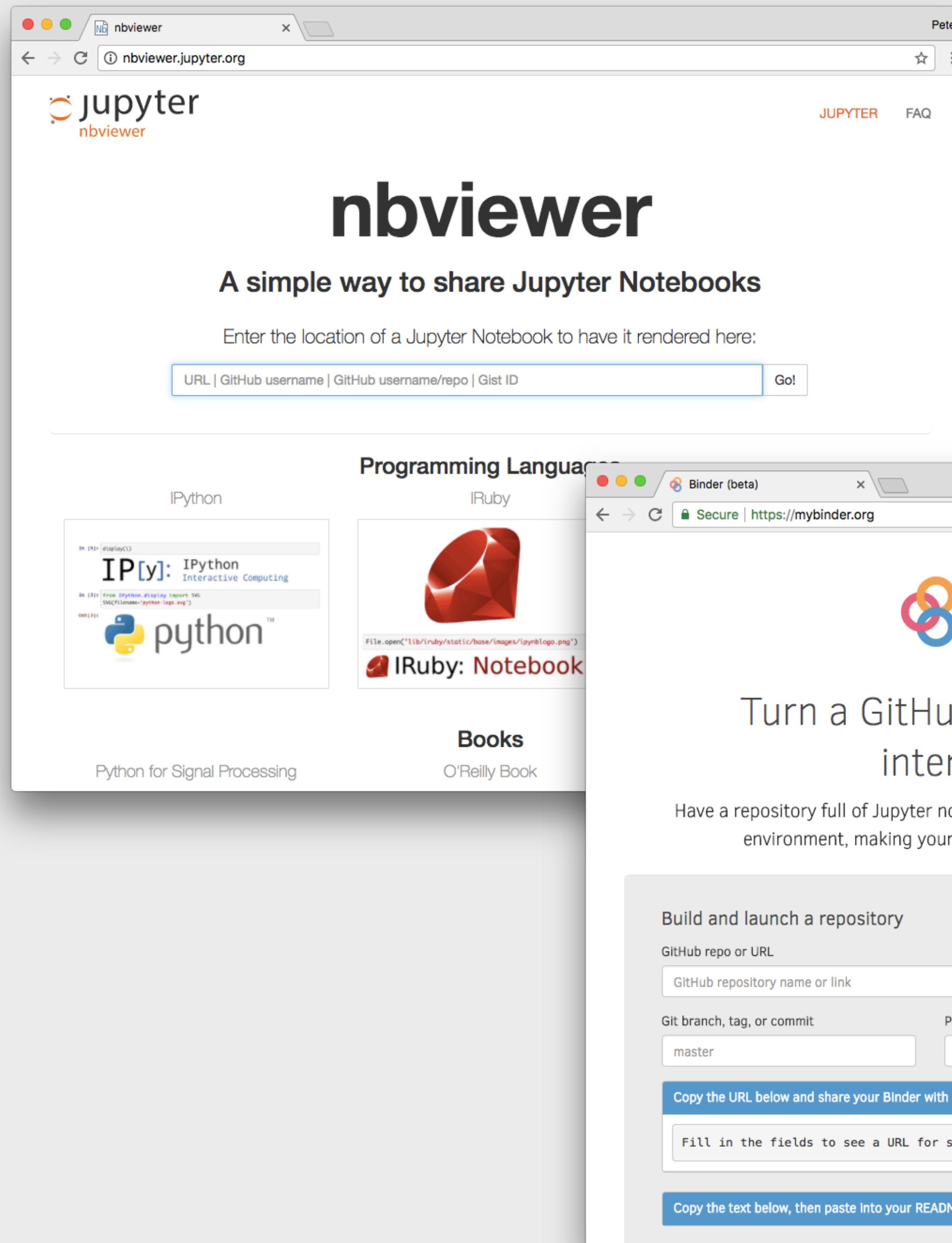
Standards



# Applications



# Services



The screenshot shows two web browser windows side-by-side. The left window is titled 'nbviewer' and displays the nbviewer homepage. It features the Jupyter logo, navigation links for 'JUPYTER' and 'FAQ', and a large title 'nbviewer' with the subtitle 'A simple way to share Jupyter Notebooks'. Below this is a text input field with placeholder 'Enter the location of a Jupyter Notebook to have it rendered here:' and a 'Go!' button. The right window is titled 'Binder (beta)' and displays the Binder homepage. It features the Binder logo, navigation links for 'Secure | https://mybinder.org', and a main heading 'Turn a GitHub repo into a collection of interactive notebooks'. Below this is a text input field for 'GitHub repo or URL' with placeholder 'GitHub repository name or link', a dropdown for 'Git branch, tag, or commit' set to 'master', a dropdown for 'Path to a notebook file (optional)', and a 'launch' button. A blue bar at the bottom contains the text 'Copy the URL below and share your Binder with others:' and a text input field with placeholder 'Fill in the fields to see a URL for sharing your Binder.' There is also a 'Copy the text below, then paste into your README to show a binder badge:' section with a 'launch binder' button.

**nbviewer**

JUPYTER FAQ

# nbviewer

A simple way to share Jupyter Notebooks

Enter the location of a Jupyter Notebook to have it rendered here:

URL | GitHub username | GitHub username/repo | Gist ID Go!

**Programming Languages**

IPython

In [1]: In [1]: display(

IP[y]: IPython Interactive Computing

In [2]: from IPython.display import Image

In [3]: Image('python-logo.png')

Out[3]: python™

IRuby

File.open('lib/ruby/static/base/images/pythonlogo.png')

IRuby: Notebook

**Books**

Python for Signal Processing

O'Reilly Book

**Binder (beta)**

Secure | https://mybinder.org

# binder

(beta)

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

GitHub repository name or link

Git branch, tag, or commit

Path to a notebook file (optional)

master Path to a notebook file (optional) File ▾ launch

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

Copy the text below, then paste into your README to show a binder badge: [launch binder](#)

# Ecosystem



- Kernels for 100+ languages
- Extensions for slideshows, GeoJSON, WebGL, code folding, SQL magics, notifications, Git, widgets, ...
- Support for Python, R, Julia, C++, Scala,, ... and their libraries
- Alternative open source frontends
- Commercial offerings

Hydrogen for Atom

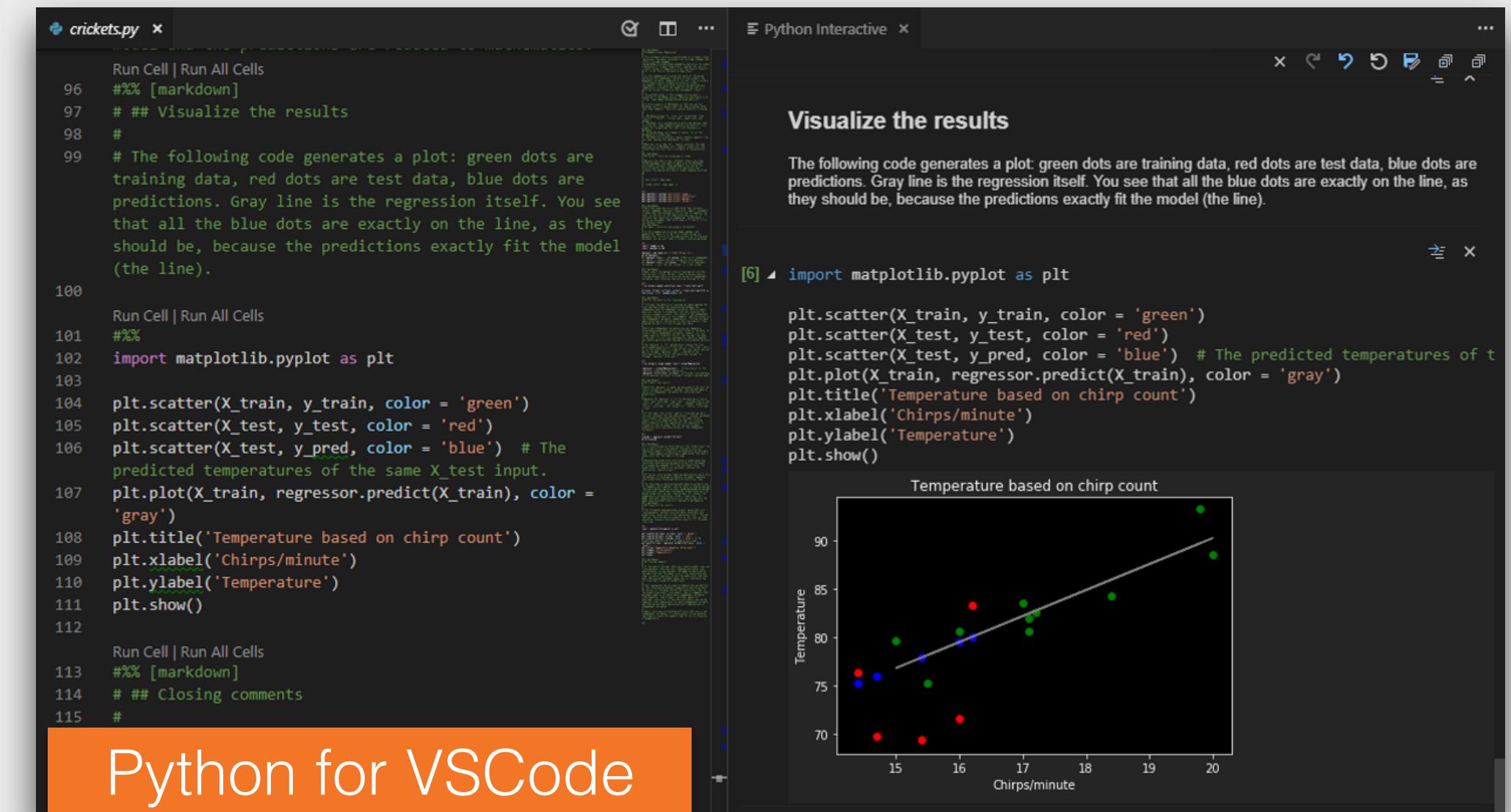
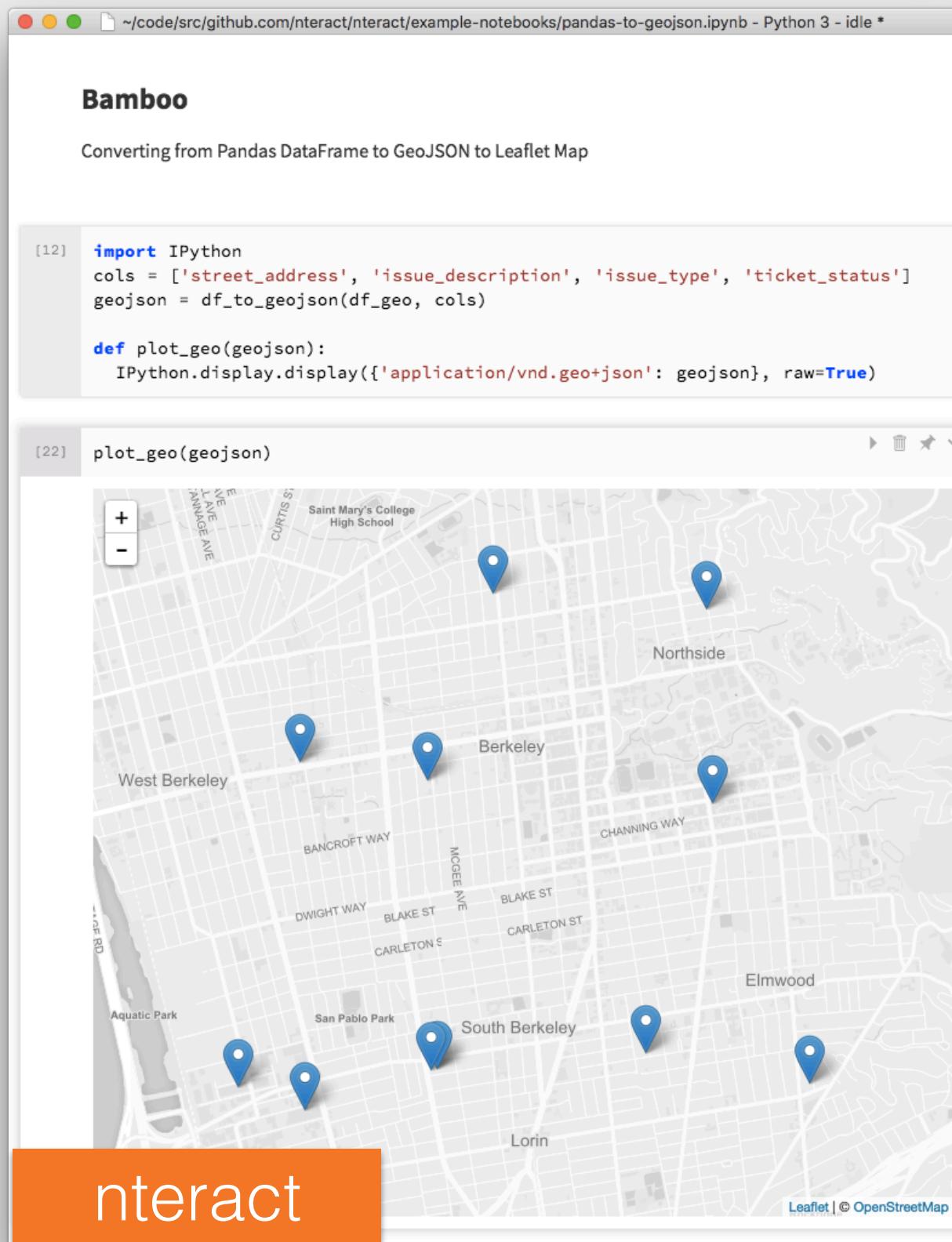
```

demo.py — ~/Desktop/hydrogen

11 # <codecell> One line outputs
12 print('Hello World!') Hello World!
13
14 # %% Render LaTeX
15 x, y, z = sp.symbols('x y z')
16 f = sp.sin(x * y) + sp.cos(y * z)
17 sp.integrate(f, x)

$$x \cos(yz) + \begin{cases} 0 & \text{for } y = 0 \\ -\frac{1}{y} \cos(xy) & \text{otherwise} \end{cases}$$

18
19 # In[1]: Display arrays
20 t = np.linspace(0, 20, 500)
21 t
array([ 0.        ,  0.04008016,  0.08016032,  0.12024048,
       0.16032064,  0.2004008 ,  0.24048096,  0.28056112,
       0.32064128,  0.36072144,  0.4008016 ,  0.44088176,
       0.48096192,  0.52104208,  0.5611224 ,  0.6012024 ,
       0.64128257,  0.68136273,  0.72144289,  0.76152305,
       0.80160321,  0.84168337,  0.88176353,  0.92184369,
       0.96192385,  1.00200401,  1.04208417,  1.08216433,
      1.12224449,  1.16232465,  1.20240481,  1.24248497,
      1.28256513,  1.32264529,  1.36272545,  1.40280561,
      1.44288577,  1.48296593,  1.52304609,  1.56312625,
      1.60320641,  1.64328657,  1.68336673,  1.72344689,
      1.76352705,  1.80360721,  1.84368737,  1.88376754,
      1.9238477 ,  1.96392786,  2.00400802,  2.04408818,
      2.08416834,  2.1242485 ,  2.16432866,  2.20440882,
      2.24448898,  2.28456914,  2.3246493 ,  2.36472946,
      2.40480962,  2.44488978,  2.48496994,  2.5250501 .
```



**Google Colaboratory**

Charts in Colaboratory

**Charting in Colaboratory**

A common use for notebooks is data visualization using charts. Colaboratory makes this easy with several charting tools available as Python imports.

**Matplotlib**

- Line Plots
- Bar Plots
- Histograms
- Scatter Plots
- Stack Plots
- Pie Charts
- fill\_between and alpha
- Subplotting using Subplot2grid
- Plot styles
- 3D Graphs
- 3D Scatter Plots
- 3D Bar Plots
- Wireframe Plots

**Matplotlib**

Matplotlib is the most common charting package, see its [documentation](#) for details, and its [examples](#) for inspiration.

**Line Plots**

```
[ ] import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
plt.plot(x, y1, label="line L")
plt.plot(x, y2, label="line H")
plt.plot()
```

Line Graph Example

Line Graph Example

**Microsoft Azure Notebooks**

jupyter 01-How-to-Run-Python-Code (unsaved changes)

This notebook contains an excerpt from the [Whirlwind Tour of Python](#) by Jake VanderPlas; the content is available on [GitHub](#). The text and code are released under the [CC0](#) license; see also the companion project, the [Python Data Science Handbook](#).

< 1. Introduction | Contents | A Quick Tour of Python Language Syntax >

## How to Run Python Code

Python is a flexible language, and there are several ways to use it depending on your particular task. One thing that distinguishes Python from other programming languages is that it is *interpreted* rather than *compiled*. This means that it is executed line by line, which allows programming to be interactive in a way that is not directly possible with compiled languages like Fortran, C, or Java. This section will describe four primary ways you can run Python code: the *Python interpreter*, the *IPython interpreter*, via *Self-contained Scripts*, or in the *Jupyter notebook*.

### The Python Interpreter

The most basic way to execute Python code is line by line within the *Python interpreter*. The Python interpreter can be started by installing the Python language (see the previous section) and typing `python` at the command prompt (look for the Terminal on Mac OS X and Unix/Linux systems, or the Command Prompt application in Windows).

**Amazon SageMaker**

Success! You created your notebook instance.

Open the notebook instance when status is `InService` and open a template notebook to get started.

Amazon SageMaker > Notebook instances

Name	Instance	Creation time	Status	Actions
my-first-instance	mL.t2.medium	Jan 14, 2018 10:59 UTC	<code>InService</code>	<a href="#">Open   Stop</a>

Amazon SageMaker

Feedback English (US)

© 2018 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

**IBM Data Science Experience**

Projects Tools Data Services Community

My Projects > DSX Demos > Spark Streaming NB

File Edit View Insert Cell Kernel Help

Start streaming job

```
In [9]: ssc.start()
```

SPARK JOB PROGRESS

JOB	PROGRESS	DURATION	STATUS
0	2 stages	8.22 sec	
1	1 stage	8.56 sec	
2	1 stage	0.6 sec	
3	1 stage	0.12 sec	
4	1 stage	0.13 sec	
5	1 stage	0.12 sec	
Stage 6	4/4 tasks		Completed

[350, 0, 345, 339, 335]  
max = 350 min = 0

Stop streaming job

```
In [10]: ssc.stop()
```

Access files in object storage with R

NOTEBOOK Apache Spark™ 2.0: Extend Structured...

SOURCE SparkTC DATE Nov 15, 2018

TOPIC Machine Learning FORMAT Web page

NOTEBOOK Access files in object storage with R

SOURCE IBM DATE Oct 04, 2016

NOTEBOOK Apache Spark™ 2.0: Extend Structured...

SOURCE Stack Exchange DATE Oct 04, 2016

**Anaconda Enterprise**

ANACONDA ENTERPRISE Projects Deployments Packages

markowitz\_notebook

**jupyter markowitz**

File Edit View Cell Kernel Help

CellToolbar

View Share Logs Advanced Terminate

10<sup>1</sup>

10<sup>0</sup>

10<sup>-1</sup>

lowest return highest return BAC VWO

Getting started

Portfolio Analysis using pyfolio

There are many ways to evaluate and analyze an algorithm. While we already provide you with some of these measures like a cumulative returns plot in the Quantopian backtester, you may want to dive deeper into what your algorithm is doing. For example, you might want to look at how your portfolio allocation changes over time, or what your exposure to certain risk-factors is.

At Quantopian, we built and open-sourced [pyfolio](#) for exactly that purpose. In this notebook you will learn how you can use this library from within the Quantopian research environment (you can also use this library independently, see the [pyfolio website](#) for more information on that).

At the core of pyfolio, we have tear sheets that summarize information about a backtest. Each tear sheet returns a number of plots, as well as other information, about a given topic. There are five main ones:

- Cumulative returns tear sheet
- Shock event returns tear sheet
- Positional tear sheet
- Transactional tear sheet
- Bayesian tear sheet

We have added an interface to the object returned by `get_backtest()` to create these various tear sheets. To generate all tear sheets at once, it's as simple as generating a backtest object and calling `create_full_tear_sheet` on it:

**Quantopian**

7. Tutorial Analyzing Backtest Results (read only)

Notebook Insert Edit Run

Capital Research Community Learn Help

7. Tutorial Analyzing Backtest Results (read only)

Cell Type: Markdown Memory: 4%

Getting started

Portfolio Analysis using pyfolio

There are many ways to evaluate and analyze an algorithm. While we already provide you with some of these measures like a cumulative returns plot in the Quantopian backtester, you may want to dive deeper into what your algorithm is doing. For example, you might want to look at how your portfolio allocation changes over time, or what your exposure to certain risk-factors is.

At Quantopian, we built and open-sourced [pyfolio](#) for exactly that purpose. In this notebook you will learn how you can use this library from within the Quantopian research environment (you can also use this library independently, see the [pyfolio website](#) for more information on that).

At the core of pyfolio, we have tear sheets that summarize information about a backtest. Each tear sheet returns a number of plots, as well as other information, about a given topic. There are five main ones:

- Cumulative returns tear sheet
- Shock event returns tear sheet
- Positional tear sheet
- Transactional tear sheet
- Bayesian tear sheet

We have added an interface to the object returned by `get_backtest()` to create these various tear sheets. To generate all tear sheets at once, it's as simple as generating a backtest object and calling `create_full_tear_sheet` on it:

#090716c383b7'

# The Community



# Users and Contributors



A screenshot of a GitHub repository page for "jupyterhub/outreachy". The repository title is "Tasks & management for Outreachy for JupyterHub". It shows 51 commits, 2 branches, 0 releases, and 5 contributors. The last commit was 5 days ago. The repository has 10 watchers, 5 stars, and 9 forks.

Code Issues Pull requests Projects Wiki Insights Settings

51 commits 2 branches 0 releases 5 contributors BSD-3-Clause

Branch: master New pull request Create new file Upload files Find file Clone or download

yuvipanda Delete HTML rendering of traefik idea ... Latest commit 7a31023 5 days ago

Authenticator Add return 11 days ago

etcd-client Fixed etcd version 11 days ago

ideas Delete HTML rendering of traefik idea 5 days ago

traefik Remove traefik defaults 19 days ago

.gitignore Add logging to files 12 days ago

.travis.yml Added second test and fixed some issues 11 days ago

LICENSE Initial commit 2 months ago

README.rst Add JupyterHub traefik proxy idea a month ago

authentication.csv Add authentiactor 19 days ago

requirements.txt Add Requirements 19 days ago

<https://github.com/jupyterhub/outreachy/graphs/contributors>

# PANGEOT

Geoscience Use Cases — Pang x +

Not Secure | pangeo.io/use\_cases/index.html

PANGEOT

## GEOSCIENCE USE CASES

We are developing a collection of Jupyter Notebooks enabled using Pangeo. If you have a use case or issue on the [Pangeo GitHub issue tracker](#) to p...

- Climate modeling
  - CESM Large Ensemble Tracer Budgets
    - Open the Dataset
    - Convert Units
    - Calculate Divergence
    - Vertical Divergence: Resolved
    - Vertical Divergence: Submesoscales
    - Vertical Divergence: Diapycnal
    - Horizontal Divergence: Resolved
    - Horizontal Divergence: Submesoscales
    - Write to Disk
- Meteorology
  - US Precipitation and Temperature
    - Connect to Dask Distributed
    - Open the dataset using Xarray
      - Metadata
      - Figure: Domain mask
      - Look! All these arrays are dask arrays
      - Intra-ensemble range
      - Calling compute
        - Figure: Intra-ensemble range
      - Persisting data on the cluster

SOSE Heat and Salt Budgets — x +

Not Secure | pangeo.io/use\_cases/physical-oceanography/SOSE.html#connect-to-dask-cluster

PANGEOT Blog Site Page

SOSE Heat and Salt Budgets

Connect to Dask Cluster

Open SOSE Dataset from Cloud Storage

Visualize Some Data

Create xgcm grid

Tracer Budgets

Flux Divergence

Surface Fluxes

Shortwave Flux

Add it all up

Include the "truth"

Validate Budget

Vertical and Horizontal Integrals of Budget

Histogram Analysis of Error

Weddell Sea Budget Timeseries

Removing Climatology

```
plt.figure(figsize=(18,8))
for v in budget_slt_weddell.data_vars:
    budget_slt_weddell[v].rolling(time=30).mean().plot(label=v)
plt.ylim([-4e6, 4e6])
plt.legend()
plt.grid()
plt.title('Weddell Sea Salt Budget')

Text(0.5,1,'Weddell Sea Salt Budget')
```

Weddell Sea Salt Budget

These figures show that, while advective terms are the largest ones in the budget, the actual variability in salinity is driven primarily by the surface fluxes.

### Removing Climatology

The timeseries is pretty short, but nevertheless we can try to remove the climatology to get a better idea of what drives the interannual variability

```
budget_slt_weddell_mm = budget_slt_weddell.groupby('time.month').mean(dim='time')
budget_slt_weddell_anom = budget_slt_weddell.groupby('time.month') - budget_slt_weddell_mm
```

```
plt.figure(figsize=(18,8))
for v in budget_slt_weddell.data_vars:
    budget_slt_weddell_anom[v].rolling(time=30).mean().plot(label=v)
plt.ylim([-2.5e7, 2.5e7])
```

# Berkeley Data8



# Columbia Journalism School



The New York Times

"All the News That's Fit to Print"

Late Edition

TODAY, rain early, afternoon sunshin, mild, high 51. Tonight, partly cloudy, low 37. Tomorrow, some sunshin, then cloudy, chillier, high 43. Details in SportsSunday, Page 8.

VOL CLXVII ... No. 57,856 © 2018 The New York Times Company NEW YORK, SUNDAY, JANUARY 28, 2018 \$6.00

## Buying Online Influence From a Shadowy Market

### Fake Followers Are Counterfeit Coins in a Booming Social Media Economy

This article is by Nicholas Confrey, J.X. Dene, Richard L. Mark Hosen.

Jessica Rychly is a Minnesotan with a broad, wavy hair. She likes the rapper Post Malone; she goes on Facebook, she sometimes muses, bored or trades jokes. Occasionally, like others, she posts a duck.

Twitter, there is a version that none of her friends would recognize. While Jessicas share a name, it and whimsical bio — "she" — the other Jessica accounts hawking California estate investments, residency and a radio station. The fake Jessica follows accounts using Indonesian languages Jessicas does not speak. She was a 17-year-old high schooler; her fake counterpart promoted graphic sex, retweeting accounts of pornography and Porno.

These accounts are counterfeit coins in the booming economy of online influence, reaching into virtually any industry where a mass audience — or the illusion of it — can be monetized. Fake accounts, deployed by governments, criminals and entrepreneurs, now infest social media networks. By some calculations, as many as 48 million of Twitter's reported active users — nearly 15 percent — are automated accounts designed to simulate real people, though the company claims that number is far lower.

In November, Facebook disclosed to investors that it had at least 100 million fake accounts on its platform, up from 60 million in July. The company has been trying to combat the problem by cracking down on accounts that post links to fake news stories, which have become a major source of revenue for fake accounts.

**Peter**

**The Follower Factory - The New York Times**

Secure | https://www.nytimes.com/interactive/2018/01/27/technology/social-media-bots.html

2018

2017

April 2016

2016

2015

2014

2013

2012

2011

2010

2009

2008

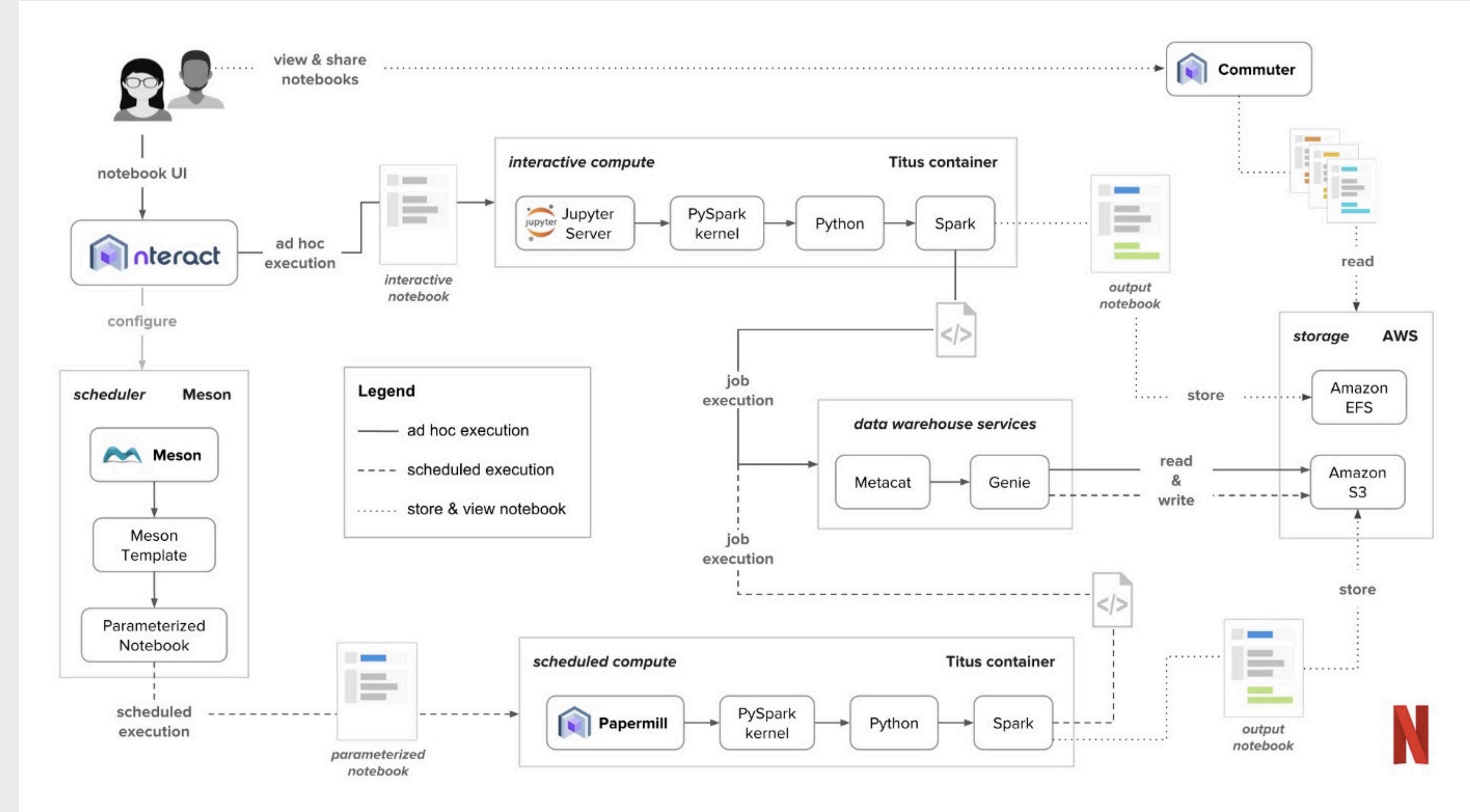
2007

Join date

50K 100K 150K 200K

**Martha Lane Fox**, a businesswoman and member of Britain's House of Lords, blamed a rogue employee for at least seven Devumi purchases made using Ms. Lane Fox's email address. The biggest — 25,000 followers — was made days after she became a Twitter board member in April 2016.

# Netflix



“... we’re currently in the process of migrating all 10,000 of the scheduled jobs running on the Netflix Data Platform to use notebook-based execution. When we’re done, more than 150,000 Genie jobs will be running through notebooks on our platform every single day.”



# Many, Many, More



GitHub, Inc. [US] | <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>

This repository Search Pull requests Issues Marketplace Explore Watch 294 Star 3,435 Fork 773

jupyter / jupyter Code Issues 83 Pull requests 0 Projects 0 Wiki Insights Settings Edit New Page

## A gallery of interesting Jupyter Notebooks

Ryan Francis edited this page 23 days ago · 51 revisions

This page is a curated collection of Jupyter/IPython notebooks that are notable. 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

Pages 6

 **Jupyter**

- Home
- A gallery of interesting Jupyter Notebooks
- Jupyter kernels
- Jupyter Notebook Server API

Clone this wiki locally

And You



# Get Started



- jupyter.org
- blog.jupyter.org
- groups.google.com/forum/#!forum/jupyter
- gitter.im/jupyter/jupyter
- mybinder.org
- *conda install -c conda-forge notebook jupyterlab*
- *Until 6 PM, here, today!*

# Get Involved



- Bug reports
- User feedback
- New ideas
- Documentation
- Graphic design
- Security audits
- Automation
- Test coverage
- Examples
- Plug-ins
- Code review
- Outreach
- User surveys
- Coordination
- Fundraising
- Release mgmt.
- Translations
- Accessibility
- Training
- Bug fixes
- New features

# JupyterCon, August 2019, NYC

<http://jupytercon.com>

Project Jupyter  
@ProjectJupyter

Following

Next Year **#JupyterCon** 2019 in NYC (same place) August 19th to 22nd reserve the dates.

10:20 AM - 24 Aug 2018

7 Retweets 30 Likes

Reply Retweet Like Message



# Q&A

