jupyter

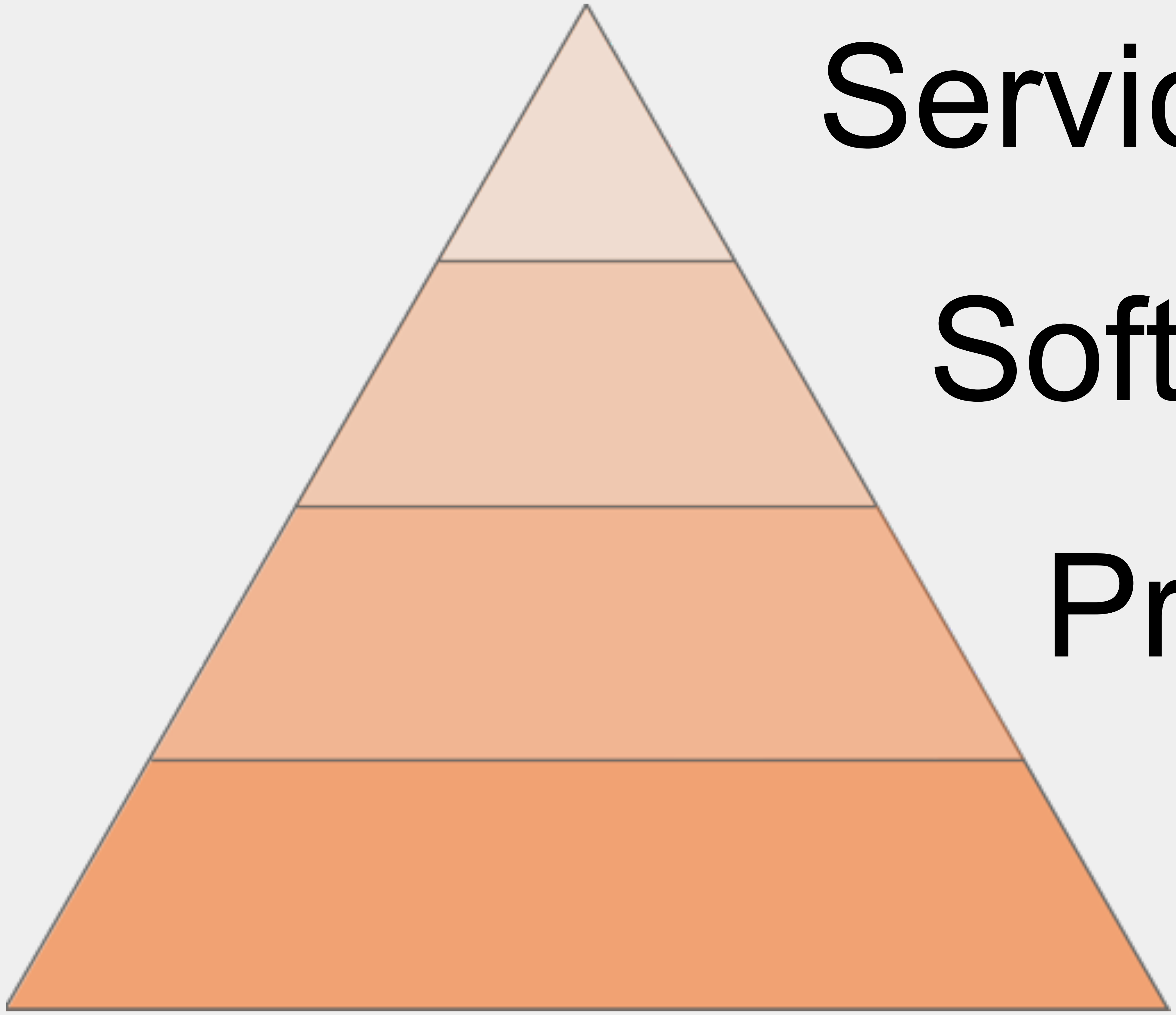
A practical introduction to Project Jupyter and JupyterLab

IceSat 2 Hack Week

Slide credits from multiple Project Jupyter teams and individuals.

@ProjectJupyter on Twitter

What is jupyter ?



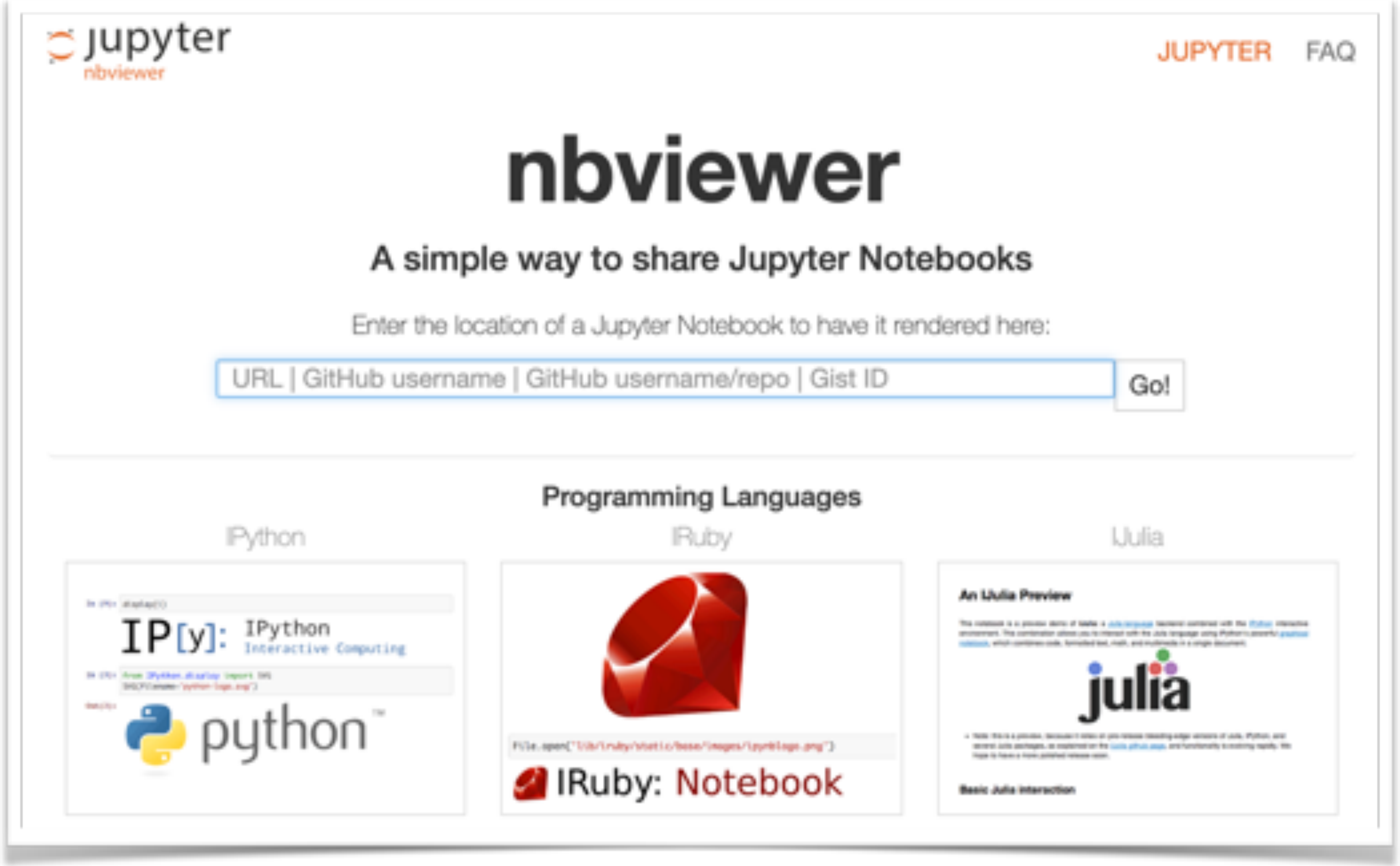
Services

Software

Protocols & Standards

Community

Services (more later…)



nbviewer

A simple way to share Jupyter Notebooks

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

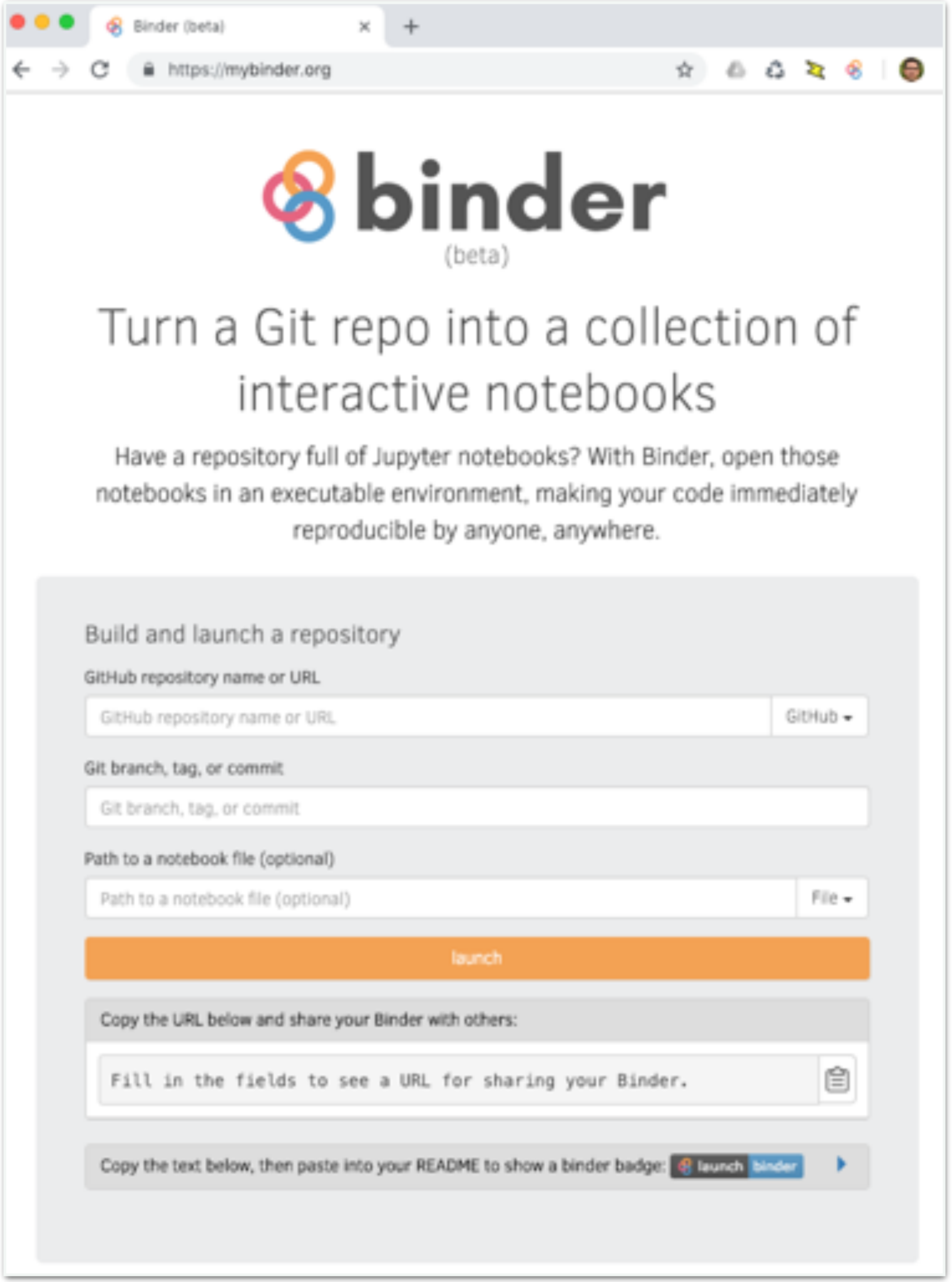
URL | GitHub username | GitHub username/repo | Gist ID

Programming Languages

IPython

IRuby

IJulia



Turn a Git 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 repository name or URL

Git branch, tag, or commit

Git branch, tag, or commit

Path to a notebook file (optional)

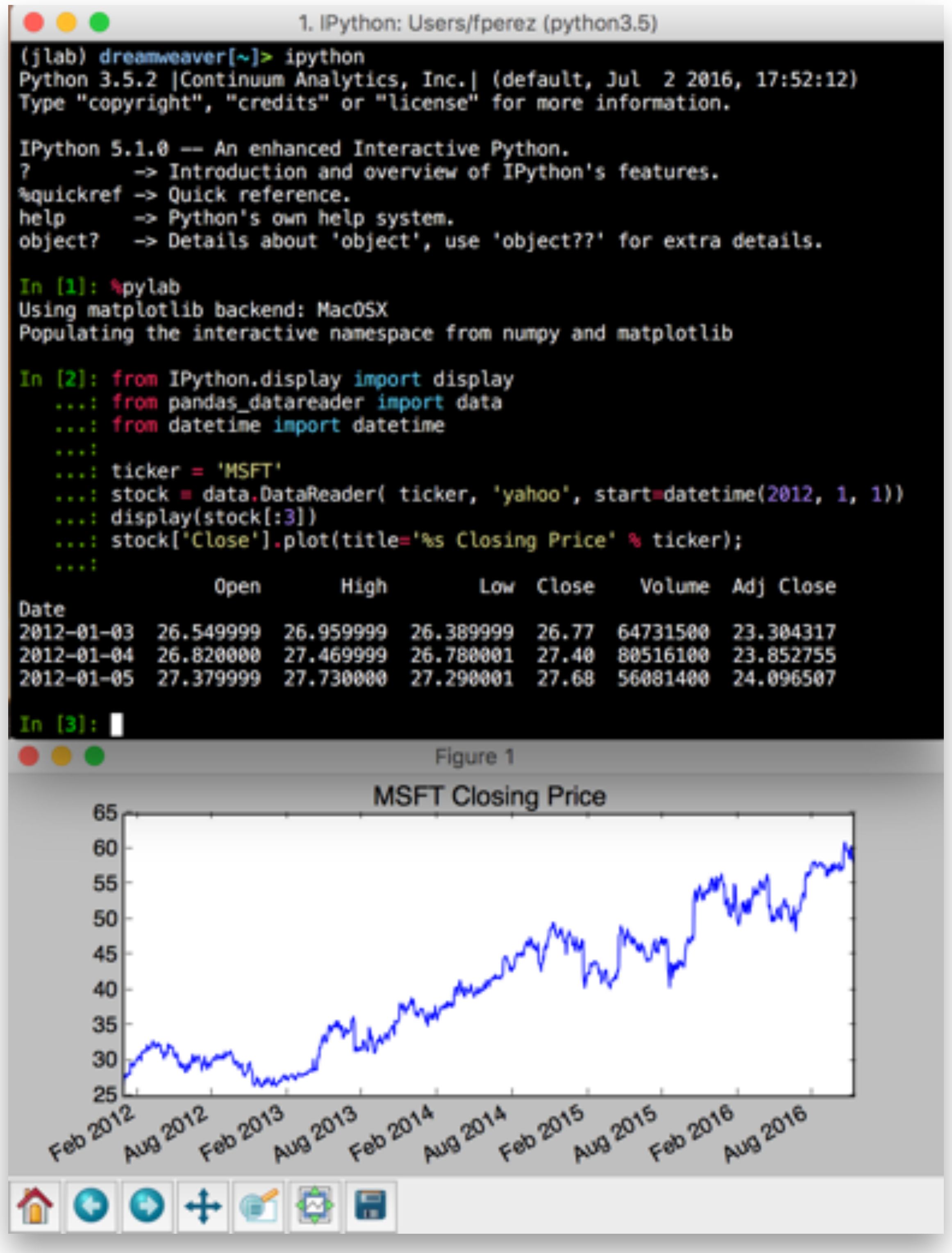
Path to a notebook file (optional)

Copy the URL below and share your Binder with others:

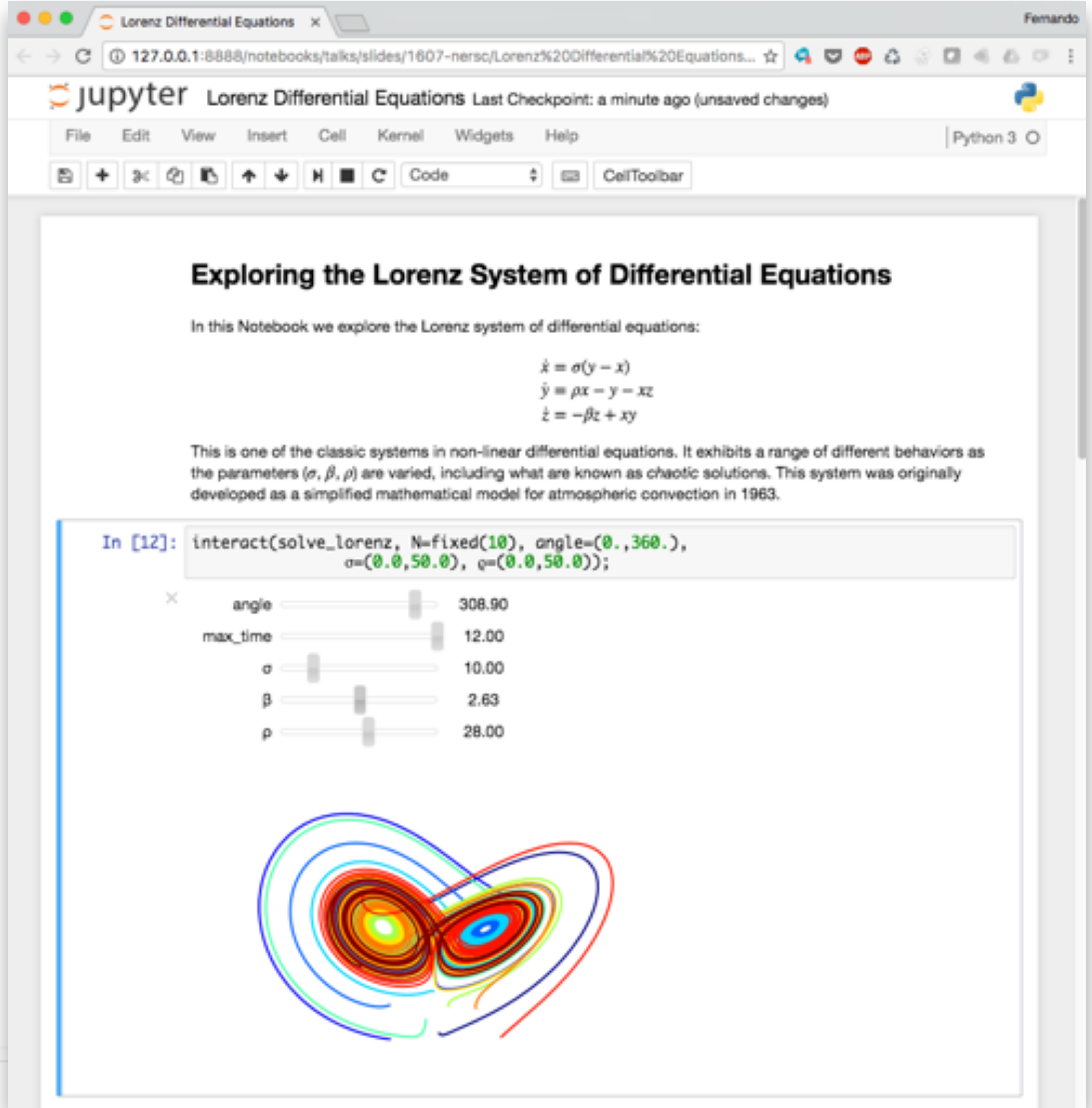
```
Fill in the fields to see a URL for sharing your Binder.
```

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

Software



```
(jlab) dreamweaver[~]> ipython
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: %pylab
Using matplotlib backend: MacOSX
Populating the interactive namespace from numpy and matplotlib

In [2]: from IPython.display import display
   ...: from pandas_datareader import data
   ...: from datetime import datetime
   ...:
   ...: ticker = 'MSFT'
   ...: stock = data.DataReader( ticker, 'yahoo', start=datetime(2012, 1, 1))
   ...: display(stock[:3])
   ...: stock['Close'].plot(title='%s Closing Price' % ticker);
   ...:
                 Open       High        Low  Close    Volume  Adj Close
Date
2012-01-03  26.549999  26.959999  26.389999  26.77  64731500  23.304317
2012-01-04  26.820000  27.469999  26.780001  27.40  80516100  23.852755
2012-01-05  27.379999  27.730000  27.290001  27.68  56081400  24.096507

In [3]:
```


Lorenz Differential Equations
Fernando
127.0.0.1:8888/notebooks/talks/slides/1607-nersc/Lorenz%20Differential%20Equations...
jupyter
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python 3
Code
CellToolbar

Lorenz Differential Equations Last Checkpoint: a minute ago (unsaved changes)

Exploring the Lorenz System of Differential Equations

In this Notebook we explore the Lorenz system of differential equations:

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

This is one of the classic systems in non-linear differential equations. It exhibits a range of different behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic* solutions. This system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [12]: interact(solve_lorenz, N=fixed(10), angle=(0.,360.),
                  σ=(0.0,50.0), ρ=(0.0,50.0));
```

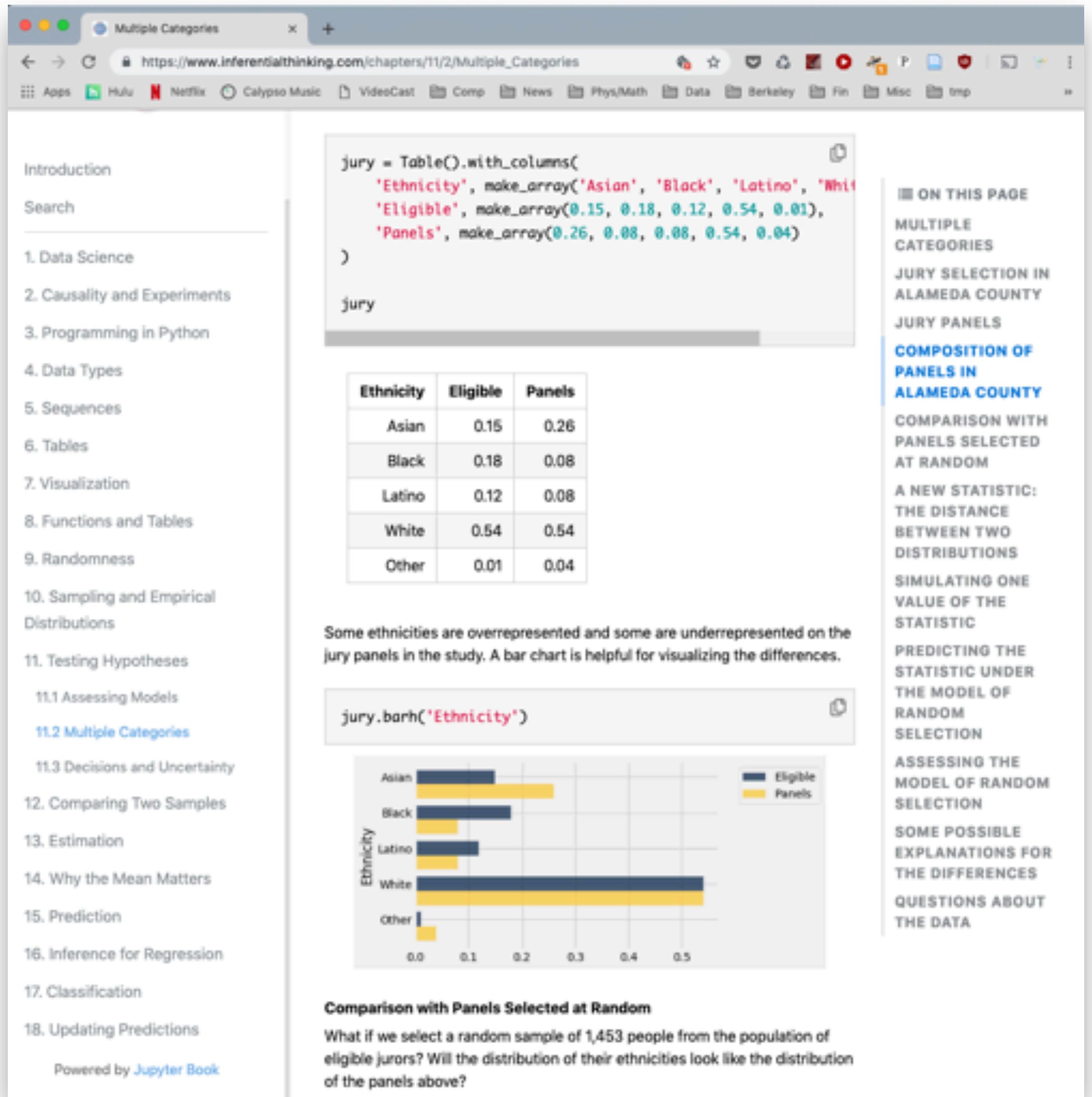
angle 308.90

max_time 12.00

σ 10.00

β 2.63

ρ 28.00



Introduction

Search

1. Data Science
2. Causality and Experiments
3. Programming in Python
4. Data Types
5. Sequences
6. Tables
7. Visualization
8. Functions and Tables
9. Randomness
10. Sampling and Empirical Distributions
11. Testing Hypotheses
11.1 Assessing Models
11.2 Multiple Categories
11.3 Decisions and Uncertainty
12. Comparing Two Samples
13. Estimation
14. Why the Mean Matters
15. Prediction
16. Inference for Regression
17. Classification
18. Updating Predictions

Powered by Jupyter Book

```
jury = Table().with_columns(
    'Ethnicity', make_array('Asian', 'Black', 'Latino', 'Whi
    'Eligible', make_array(0.15, 0.18, 0.12, 0.54, 0.01),
    'Panels', make_array(0.26, 0.08, 0.08, 0.54, 0.04)
)

jury
```

Ethnicity	Eligible	Panels
Asian	0.15	0.26
Black	0.18	0.08
Latino	0.12	0.08
White	0.54	0.54
Other	0.01	0.04

Some ethnicities are overrepresented and some are underrepresented on the jury panels in the study. A bar chart is helpful for visualizing the differences.

```
jury.barh('Ethnicity')
```

Comparison with Panels Selected at Random

What if we select a random sample of 1,453 people from the population of eligible jurors? Will the distribution of their ethnicities look like the distribution of the panels above?

ON THIS PAGE

MULTIPLE CATEGORIES

JURY SELECTION IN ALAMEDA COUNTY

JURY PANELS

COMPOSITION OF PANELS IN ALAMEDA COUNTY

COMPARISON WITH PANELS SELECTED AT RANDOM

A NEW STATISTIC: THE DISTANCE BETWEEN TWO DISTRIBUTIONS

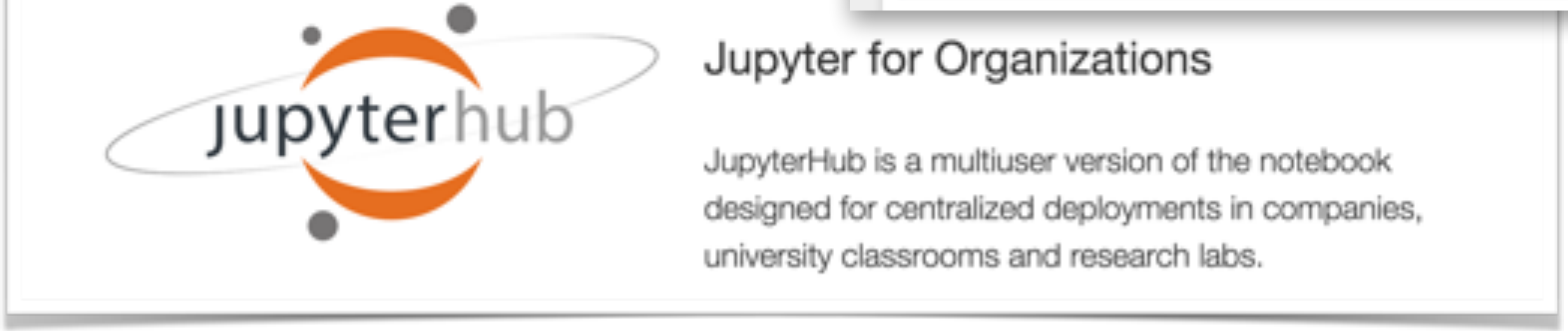
SIMULATING ONE VALUE OF THE STATISTIC

PREDICTING THE STATISTIC UNDER THE MODEL OF RANDOM SELECTION

ASSESSING THE MODEL OF RANDOM SELECTION

SOME POSSIBLE EXPLANATIONS FOR THE DIFFERENCES

QUESTIONS ABOUT THE DATA



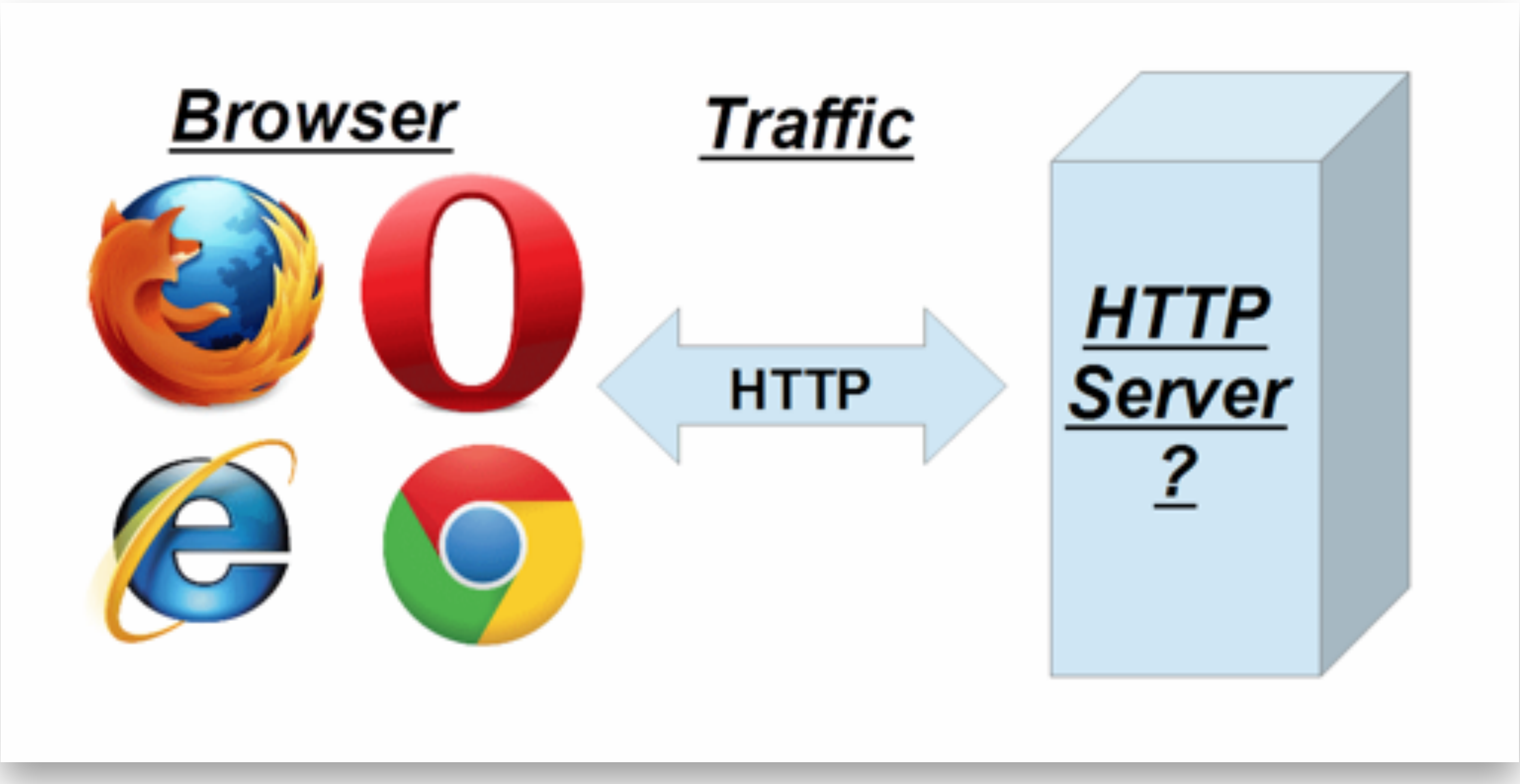
Jupyter for Organizations

JupyterHub is a multiuser version of the notebook designed for centralized deployments in companies, university classrooms and research labs.

jupyter

Standards and Protocols

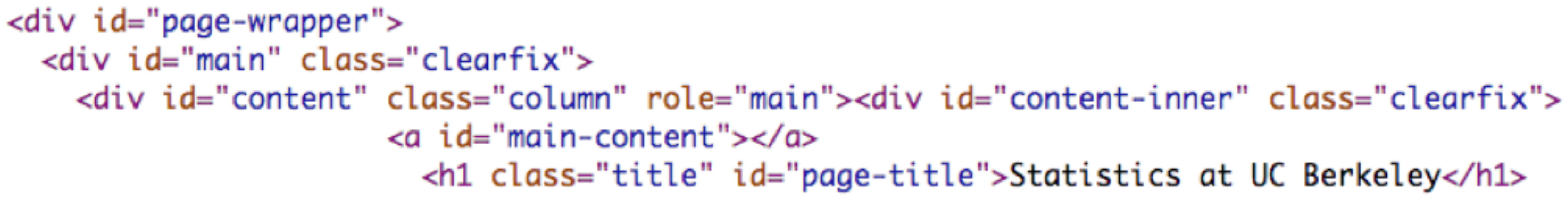
Core ideas of the web: HTTP & HTML



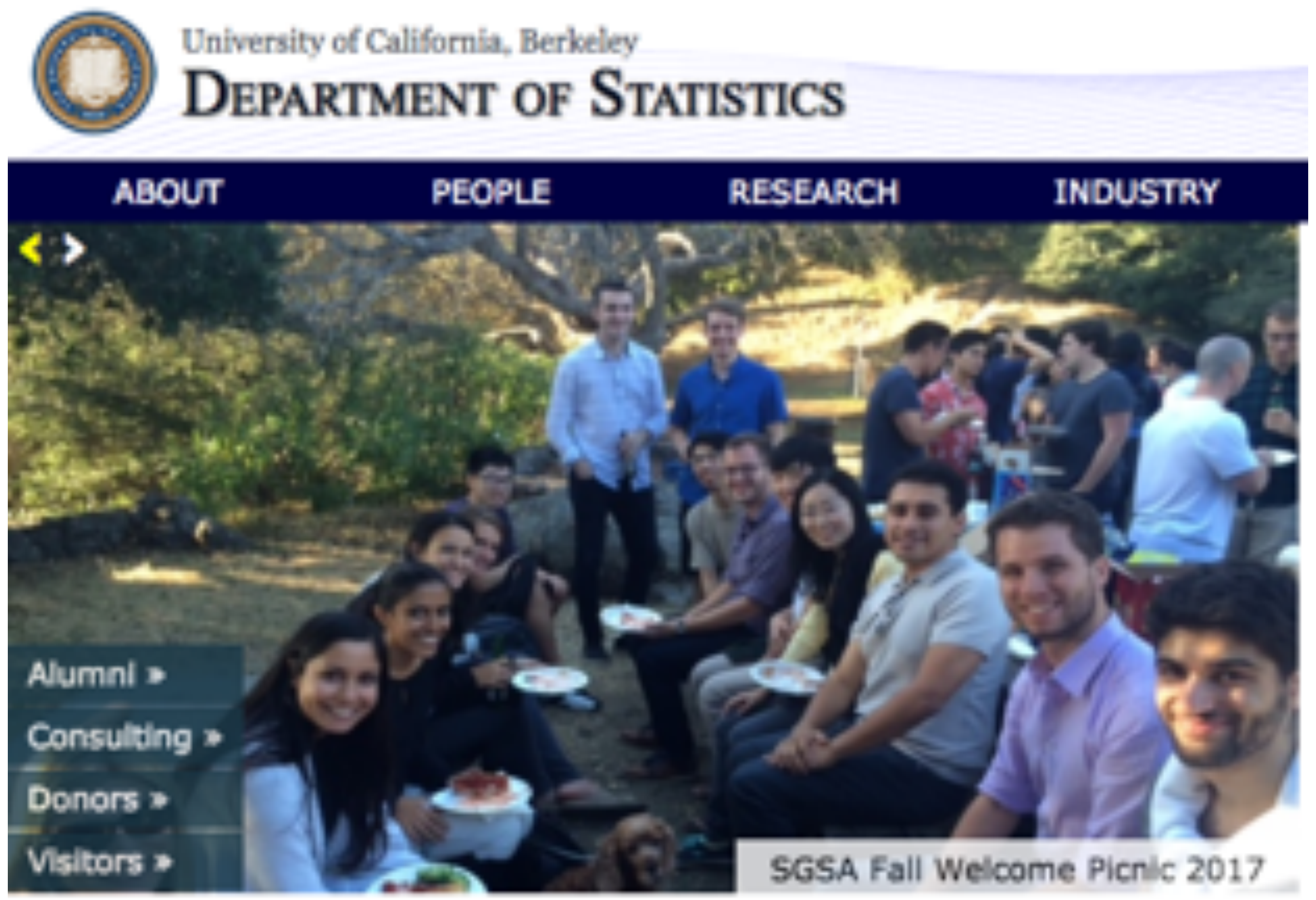
HTTP: protocol to connect clients and servers

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>
```

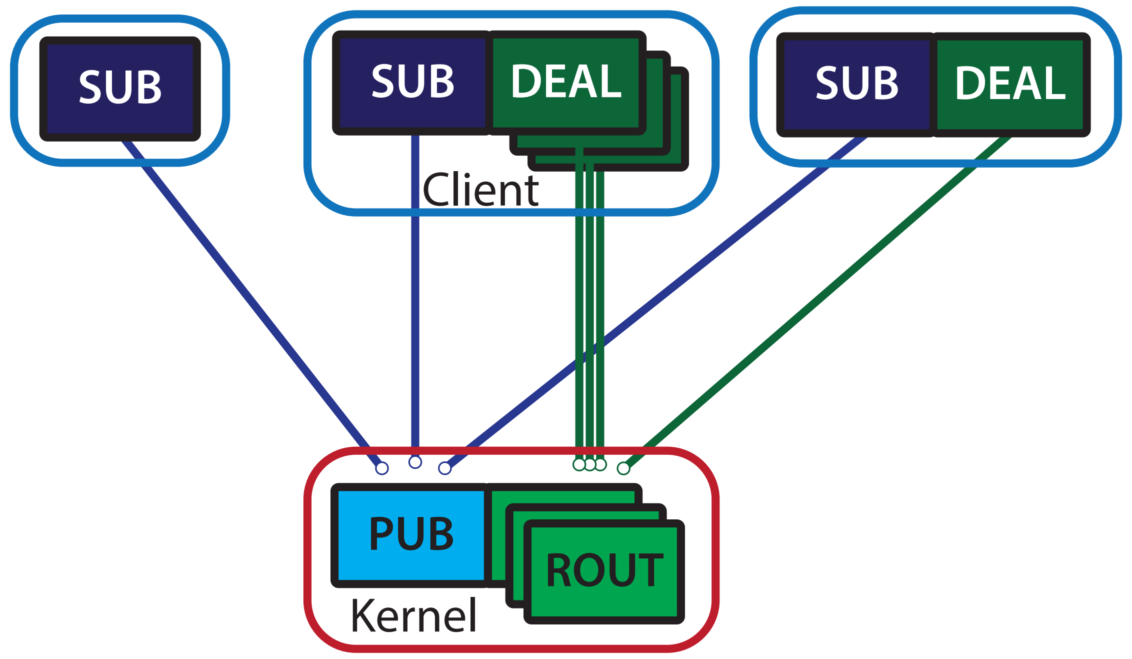


HTML: format to represent content

HyperText Markup Language

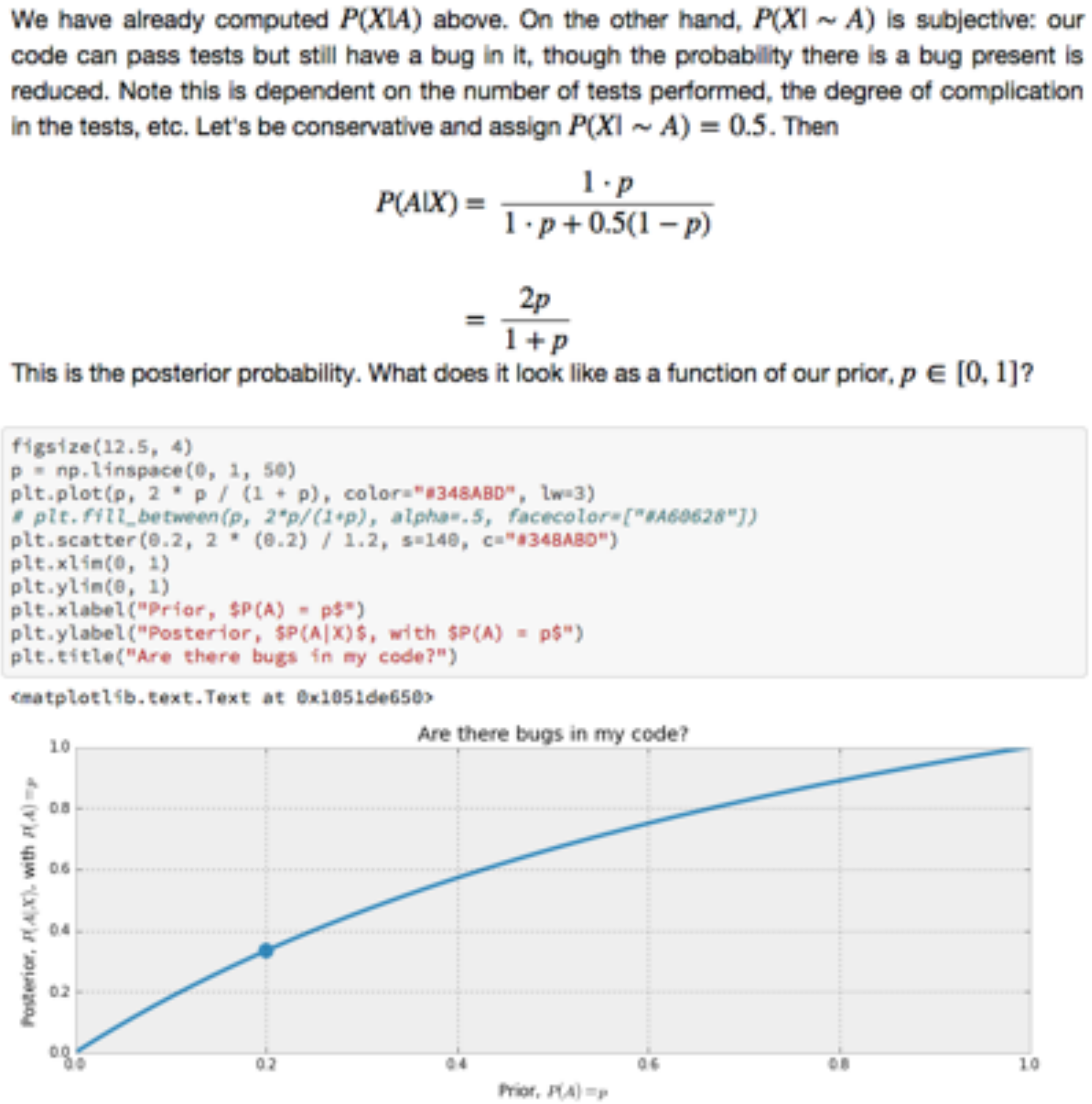
Core ideas of Jupyter

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?")
```

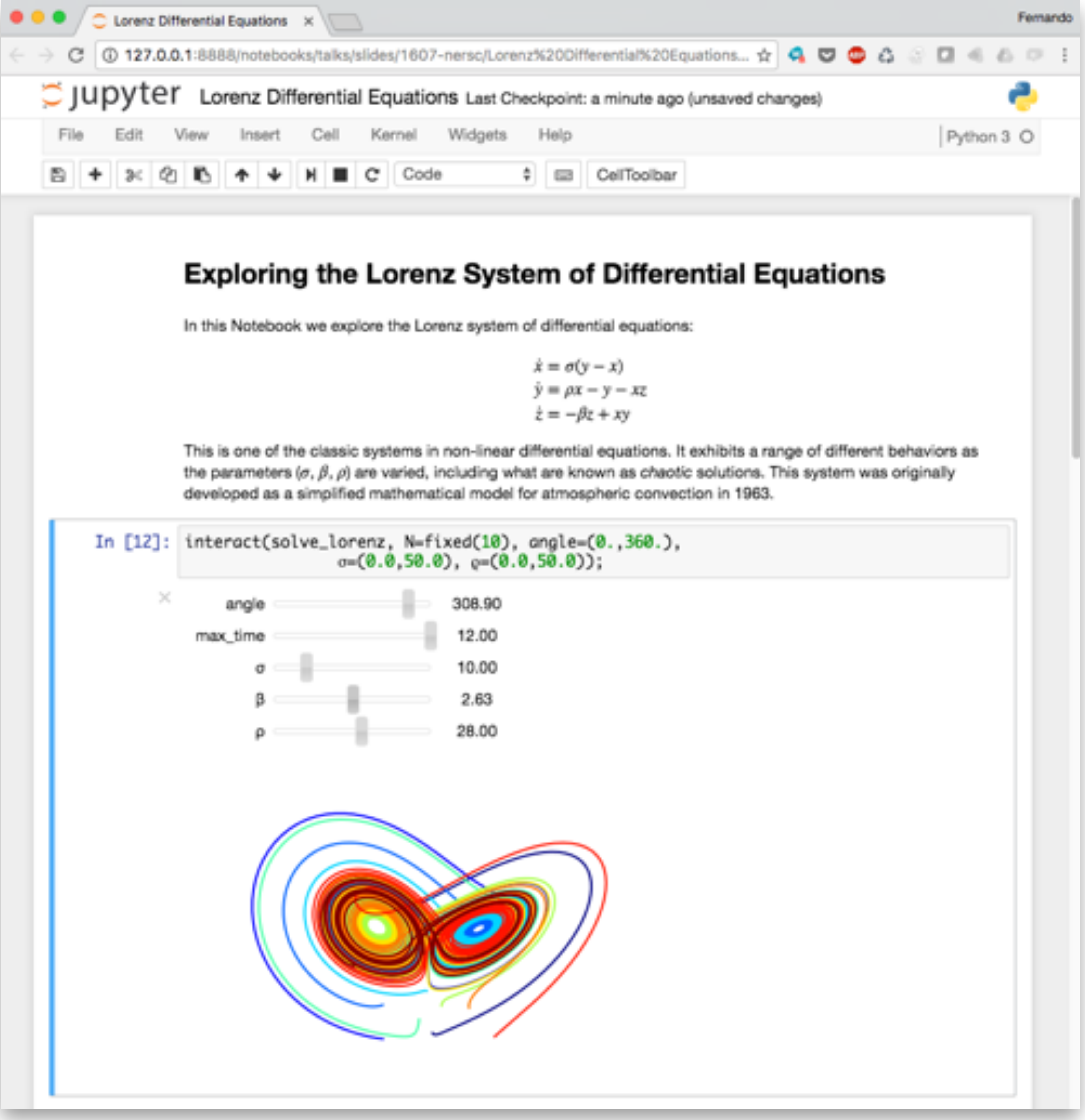
```
<matplotlib.text.Text at 0x1051de650>
```

Are there bugs in my code?
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Posterior, $P(A|X)$, with $P(A) = p$
Prior, $P(A) = p$

https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-

The Notebook as document & format

- JSON specification
- Machine readable
- Metadata-rich


Lorenz Differential Equations
Fernando
127.0.0.1:8888/notebooks/talks/slides/1607-nersc/Lorenz%20Differential%20Equations...
jupyter
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python 3
Code
CellToolbar

Lorenz Differential Equations Last Checkpoint: a minute ago (unsaved changes)

Exploring the Lorenz System of Differential Equations

In this Notebook we explore the Lorenz system of differential equations:

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

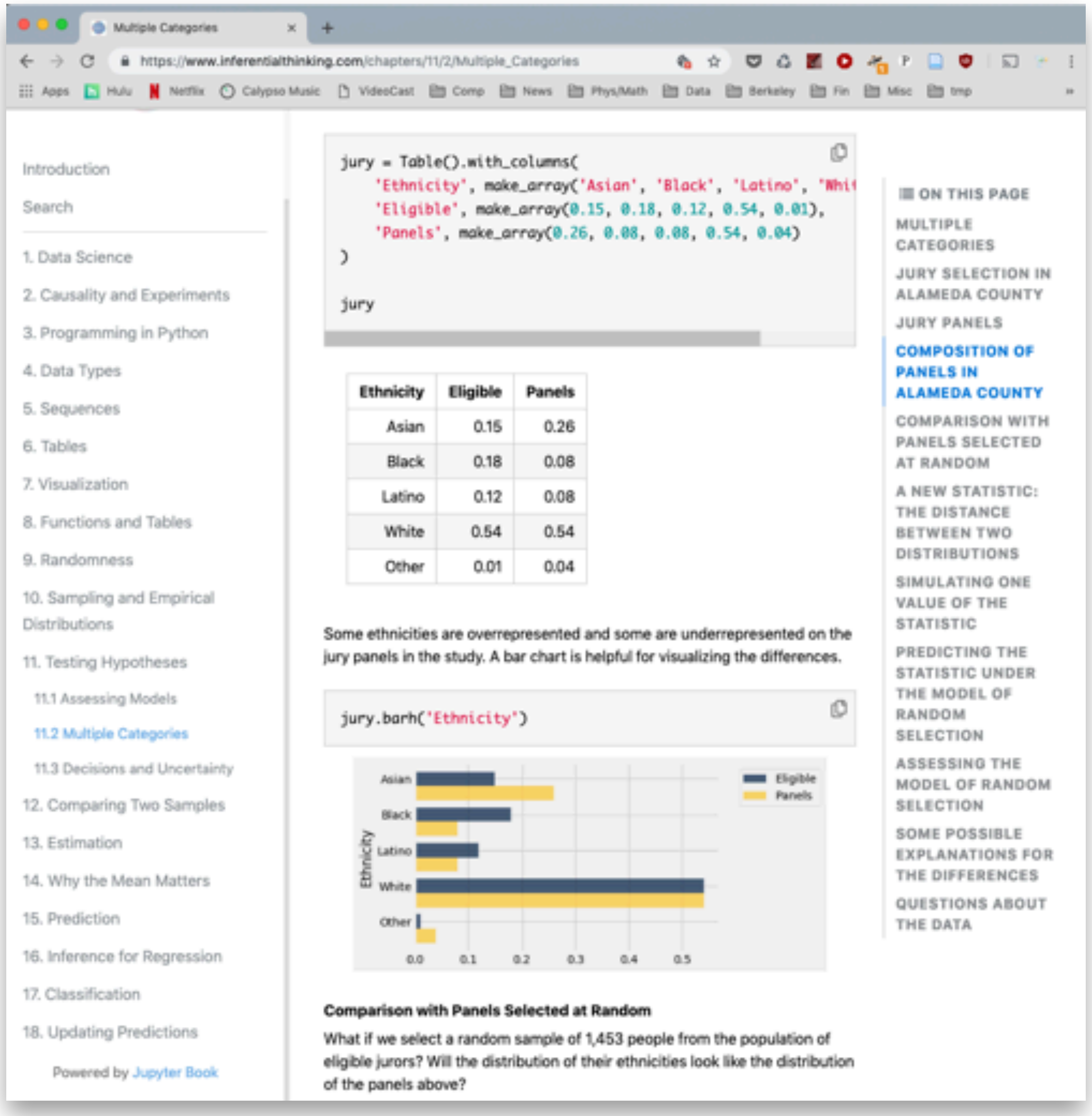
This is one of the classic systems in non-linear differential equations. It exhibits a range of different behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic* solutions. This system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [12]: interact(solve_lorenz, N=fixed(10), angle=(0.,360.),
                  σ=(0.0,50.0), ρ=(0.0,50.0));
```

```
angle     308.90
max_time   12.00
σ          10.00
β           2.63
ρ          28.00
```

New tools built atop the format

Jupyter Book



Powered by Jupyter Book

```
jury = Table().with_columns(
    'Ethnicity', make_array('Asian', 'Black', 'Latino', 'Whi
    'Eligible', make_array(0.15, 0.18, 0.12, 0.54, 0.01),
    'Panels', make_array(0.26, 0.08, 0.08, 0.54, 0.04)
)

jury
```

Ethnicity	Eligible	Panels
Asian	0.15	0.26
Black	0.18	0.08
Latino	0.12	0.08
White	0.54	0.54
Other	0.01	0.04

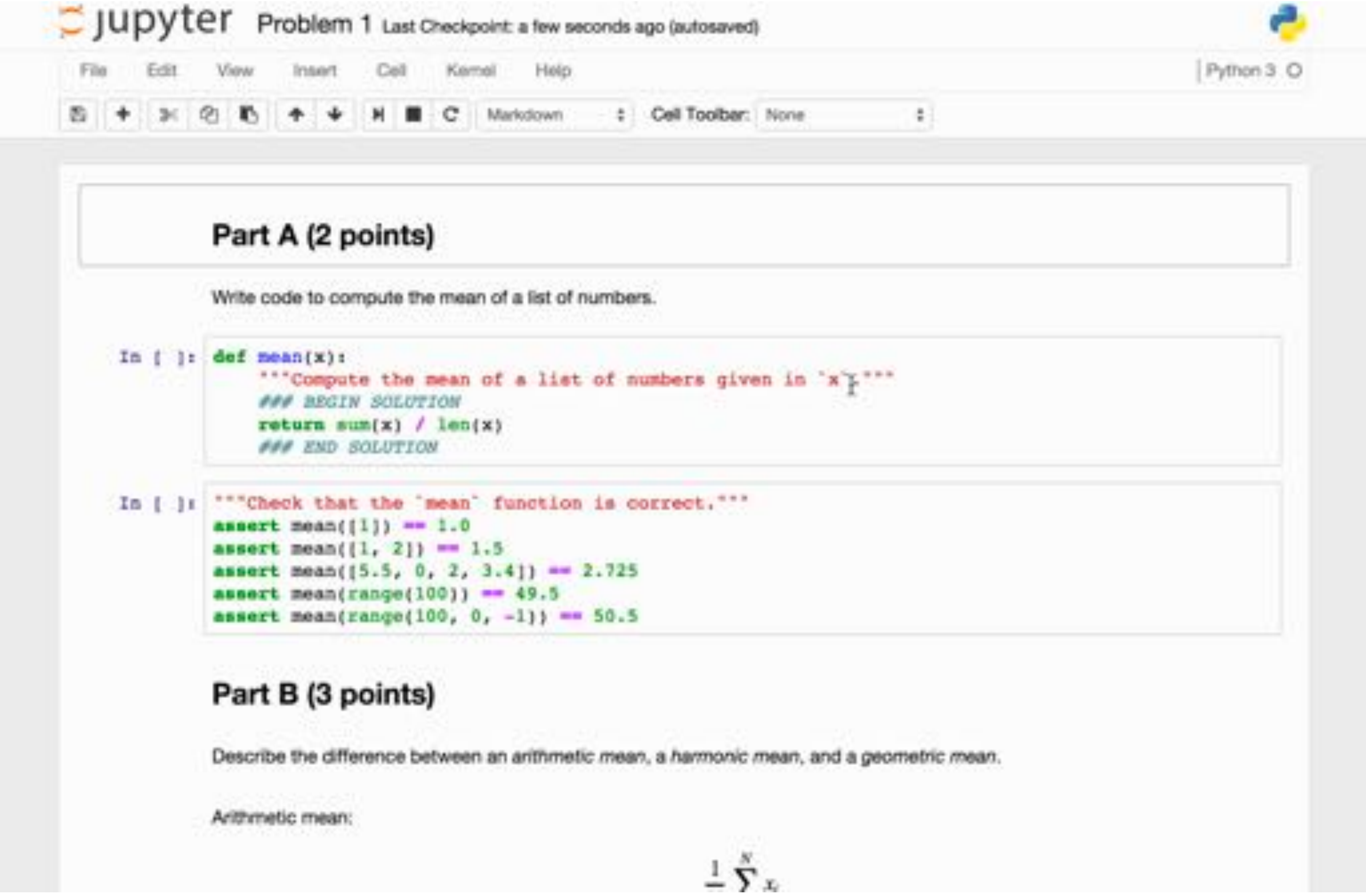
Some ethnicities are overrepresented and some are underrepresented on the jury panels in the study. A bar chart is helpful for visualizing the differences.

```
jury.barh('Ethnicity')
```

Comparison with Panels Selected at Random

What if we select a random sample of 1,453 people from the population of eligible jurors? Will the distribution of their ethnicities look like the distribution of the panels above?

nbgrader: homework assignments



Part A (2 points)

Write code to compute the mean of a list of numbers.

```
In [ ]: def mean(x):
            """Compute the mean of a list of numbers given in `x`."""
            ### BEGIN SOLUTION
            return sum(x) / len(x)
            ### END SOLUTION
```

```
In [ ]: """Check that the `mean` function is correct."""
        assert mean([1]) == 1.0
        assert mean([1, 2]) == 1.5
        assert mean([5.5, 0, 2, 3.4]) == 2.725
        assert mean(range(100)) == 49.5
        assert mean(range(100, 0, -1)) == 50.5
```

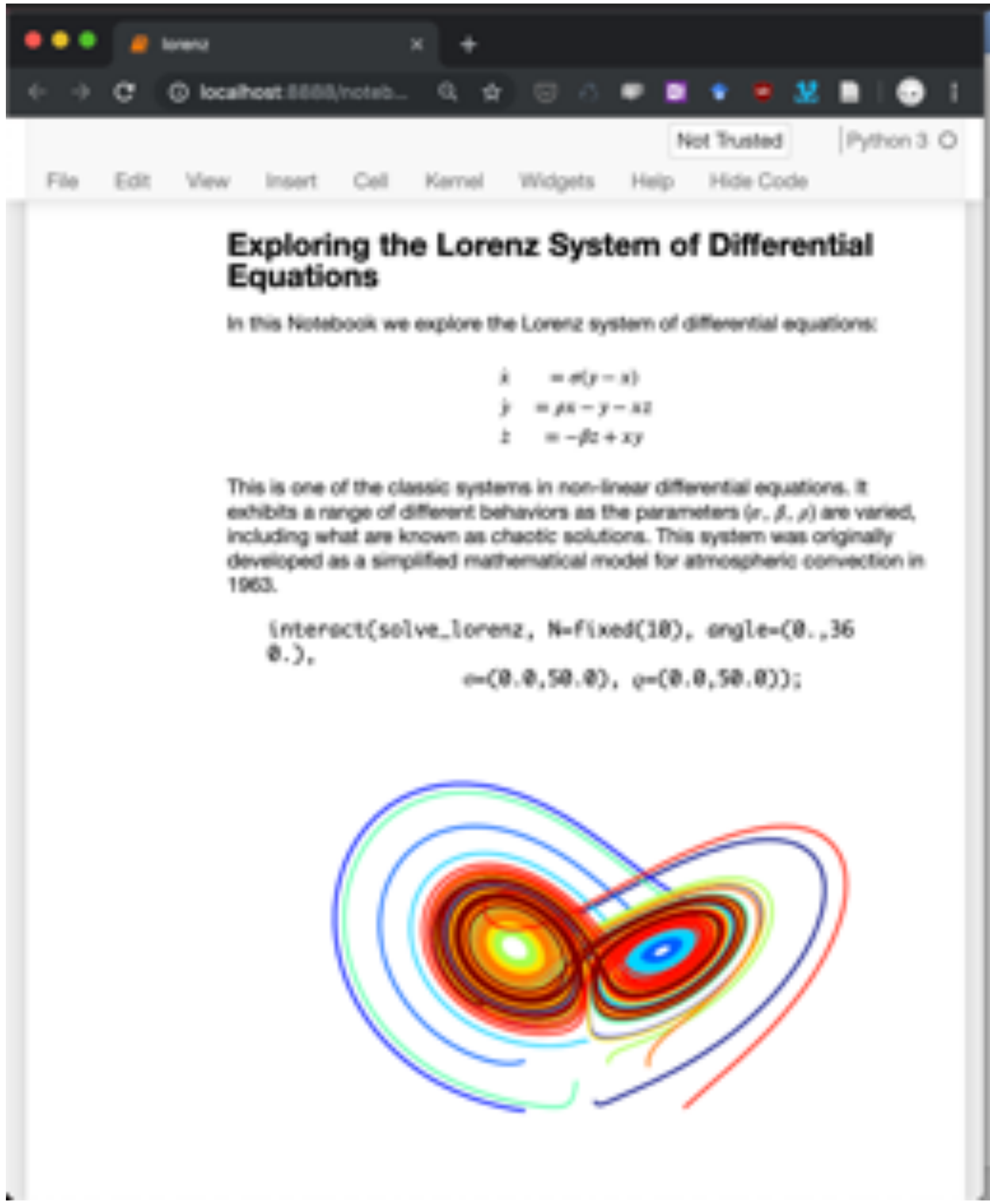
Part B (3 points)

Describe the difference between an *arithmetic mean*, a *harmonic mean*, and a *geometric mean*.

Arithmetic mean:

$$\frac{1}{\ }\sum^{N} x_i$$

pandoc: ipynb to word (and more)



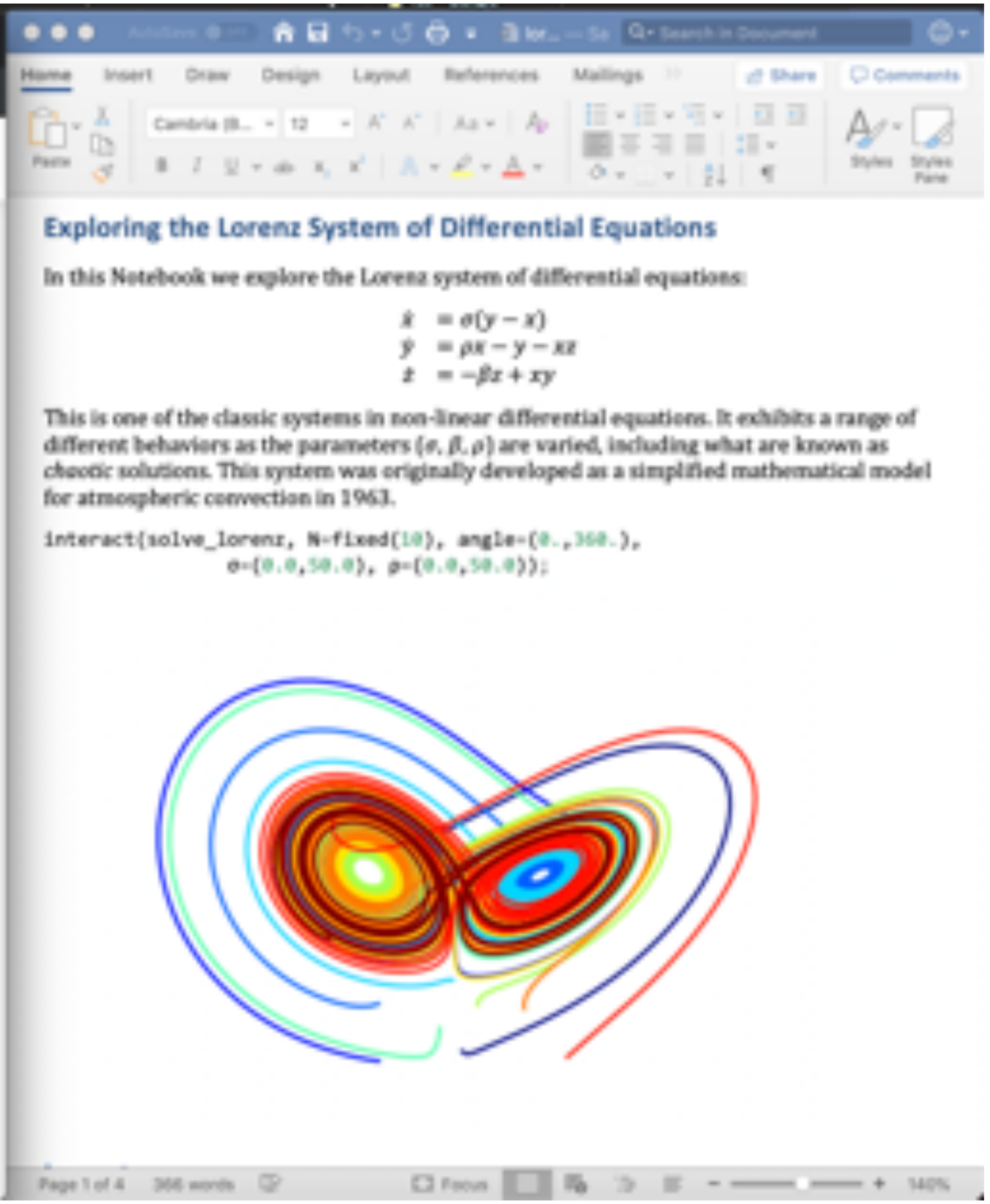
Exploring the Lorenz System of Differential Equations

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 different behaviors as the parameters (σ, β, ρ) are varied, including what are known as chaotic solutions. This system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
interact(solve_lorenz, N=fixed(10), angle=(0.,36
0.),
                      σ=(0.0,50.0), ρ=(0.0,50.0));
```



Exploring the Lorenz System of Differential Equations

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 different behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic* solutions. This system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

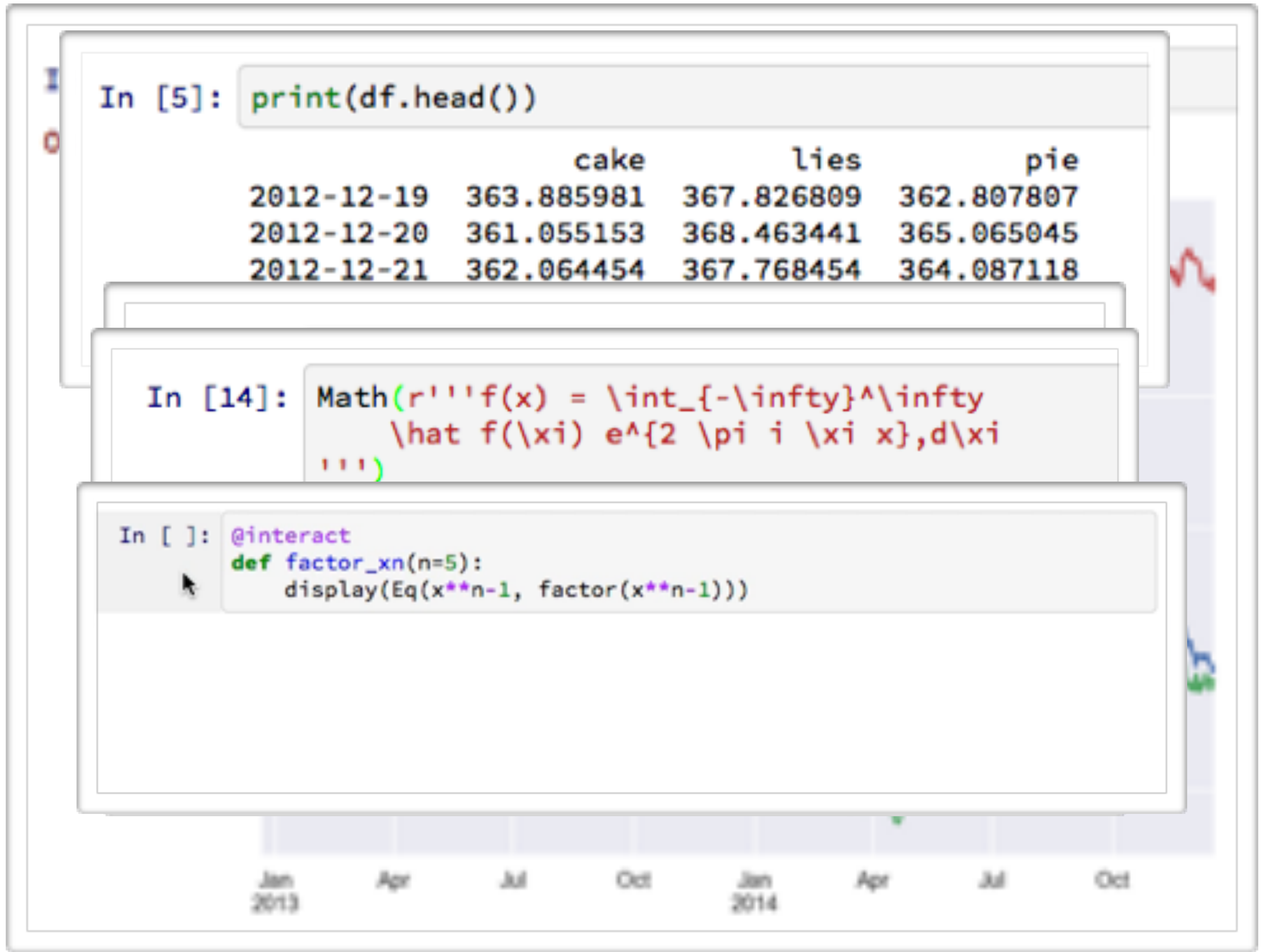
```
interact(solve_lorenz, N=fixed(10), angle=(0.,360.),
                σ=(0.0,50.0), ρ=(0.0,50.0));
```

```
pandoc lorenz.ipynb -o lorenz.docx
```

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



```
In [5]: print(df.head())
                  cake        lies         pie
2012-12-19  363.885981  367.826809  362.807807
2012-12-20  361.055153  368.463441  365.065045
2012-12-21  362.064454  367.768454  364.087118
```

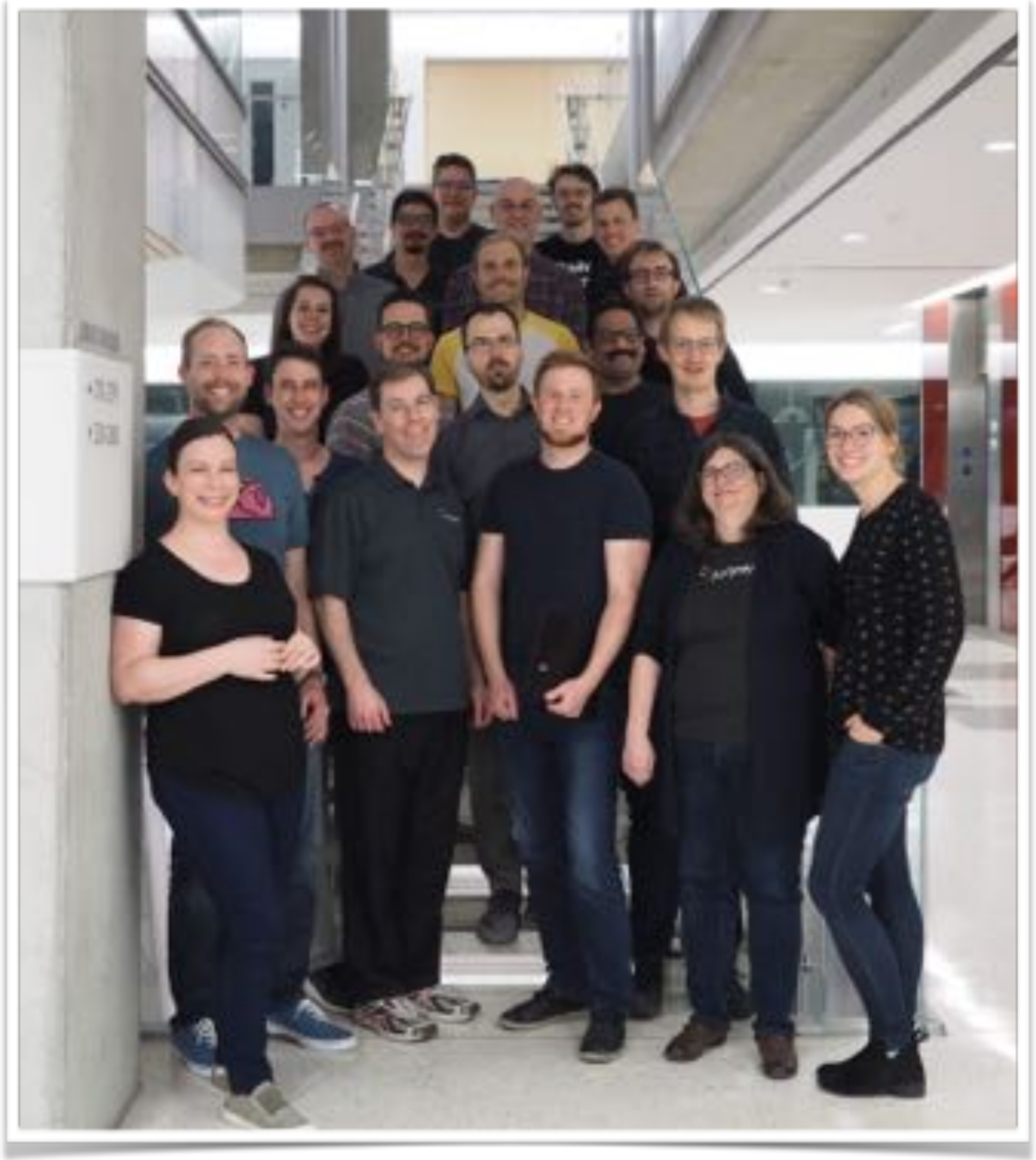
```
In [14]: Math(r'''f(x) = \int_{-\infty}^\infty
             \hat f(\xi) e^{2 \pi i \xi x},d\xi
         ''')
```

```
In [ ]: @interact
        def factor_xn(n=5):
            display(Eq(x**n-1, factor(x**n-1)))
```

A language agnostic protocol

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

Community: a true team effort



Plus ~ 1500 more Open source contributors!

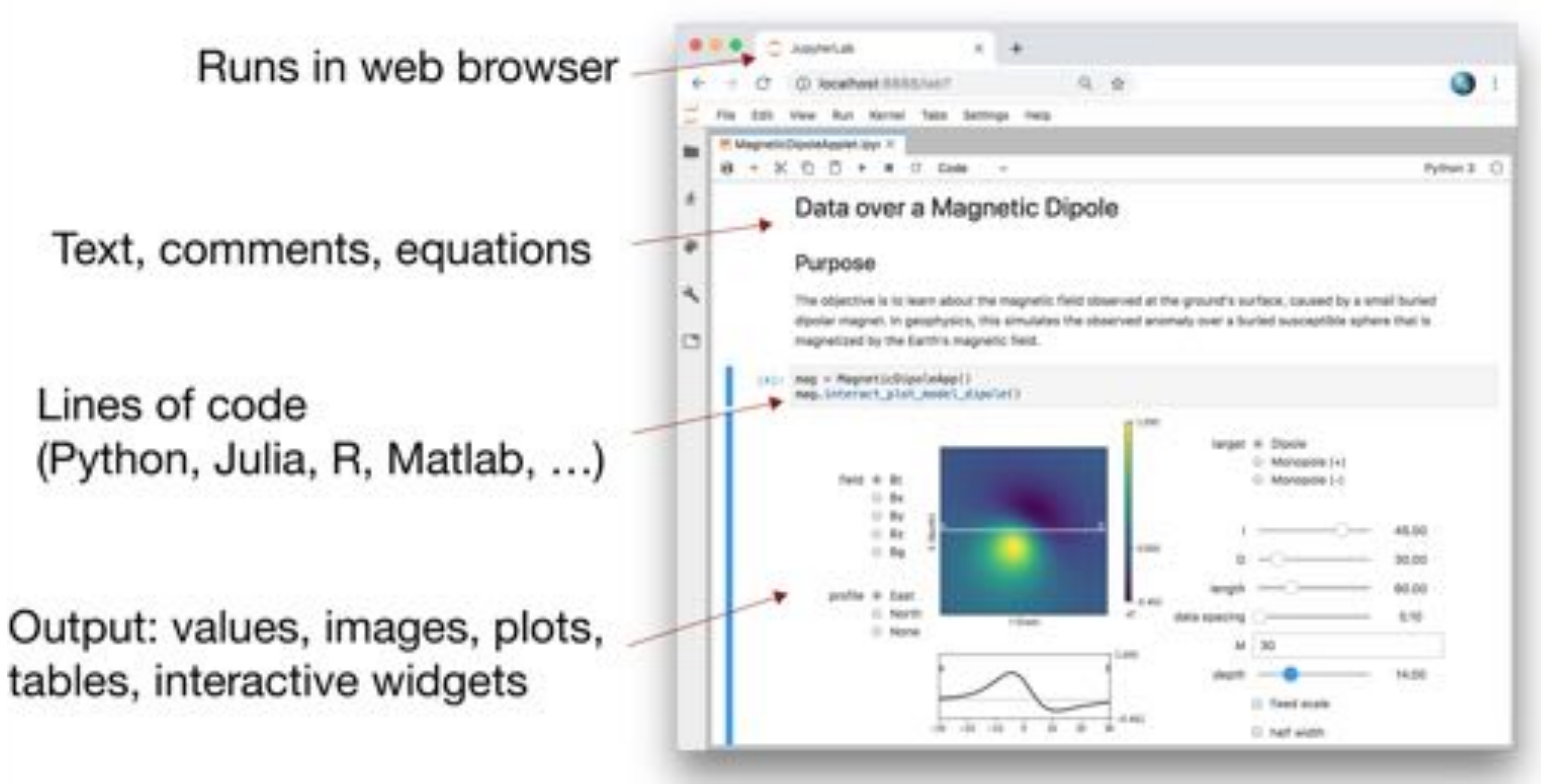
The Jupyter Notebook

Jupyter Notebook

Interactive, Exploratory, Reproducible

- Interactive, browser-based computing environment
- Exploratory data science, ML, visualization, analysis, stats
- Reproducible document format:
  - Code
  - Narrative text (markdown)
  - Equations (LaTeX)
  - Images, visualizations
- Over 100 programming languages
- Everything open-source (BSD license)

Jupyter Notebooks:
Live computational narratives



Runs in web browser

Text, comments, equations

Lines of code
(Python, Julia, R, Matlab, …)

Output: values, images, plots,
tables, interactive widgets

Data over a Magnetic Dipole

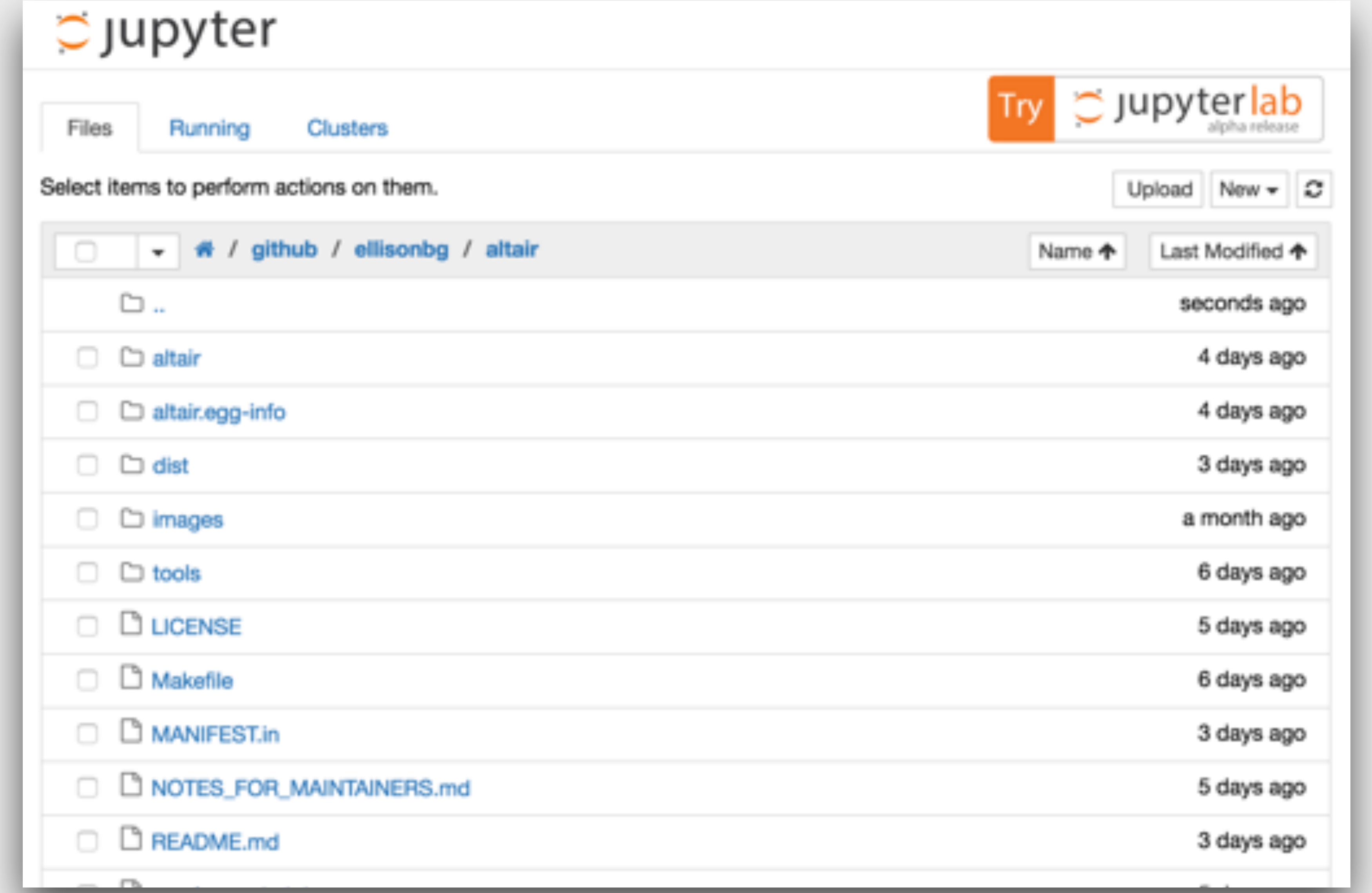
Purpose

The objective is to learn about the magnetic field observed at the ground's surface, caused by a small buried dipolar magnet. In geophysics, this simulates the observed anomaly over a buried susceptible sphere that is magnetized by the Earth's magnetic field.

```
mag = MagneticDipoleApp()
mag.interact_plot_model_dipole()
```

Building Blocks for Interactive Computing

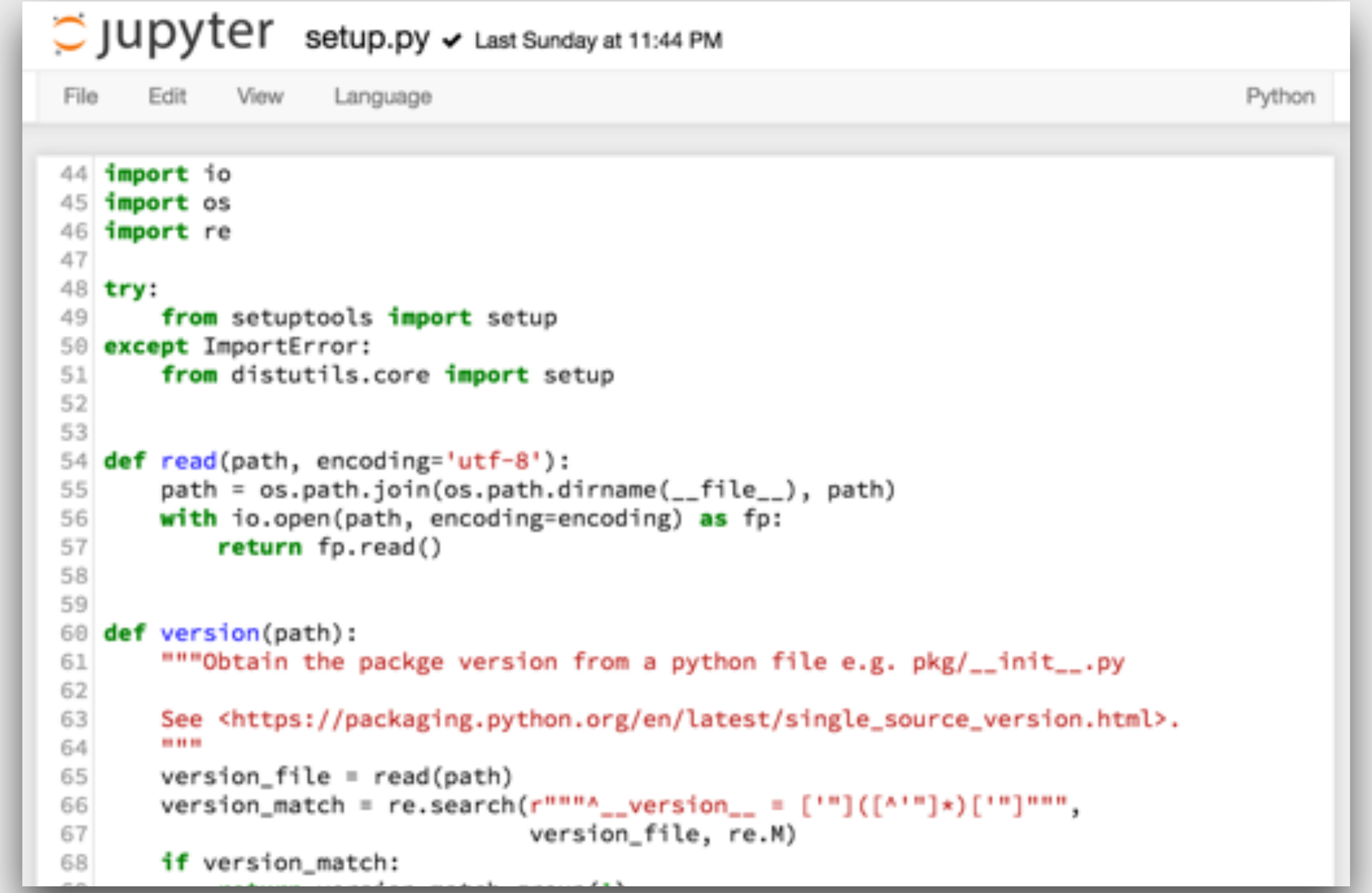
Classic Jupyter: More Than Just Notebooks



Select items to perform actions on them.

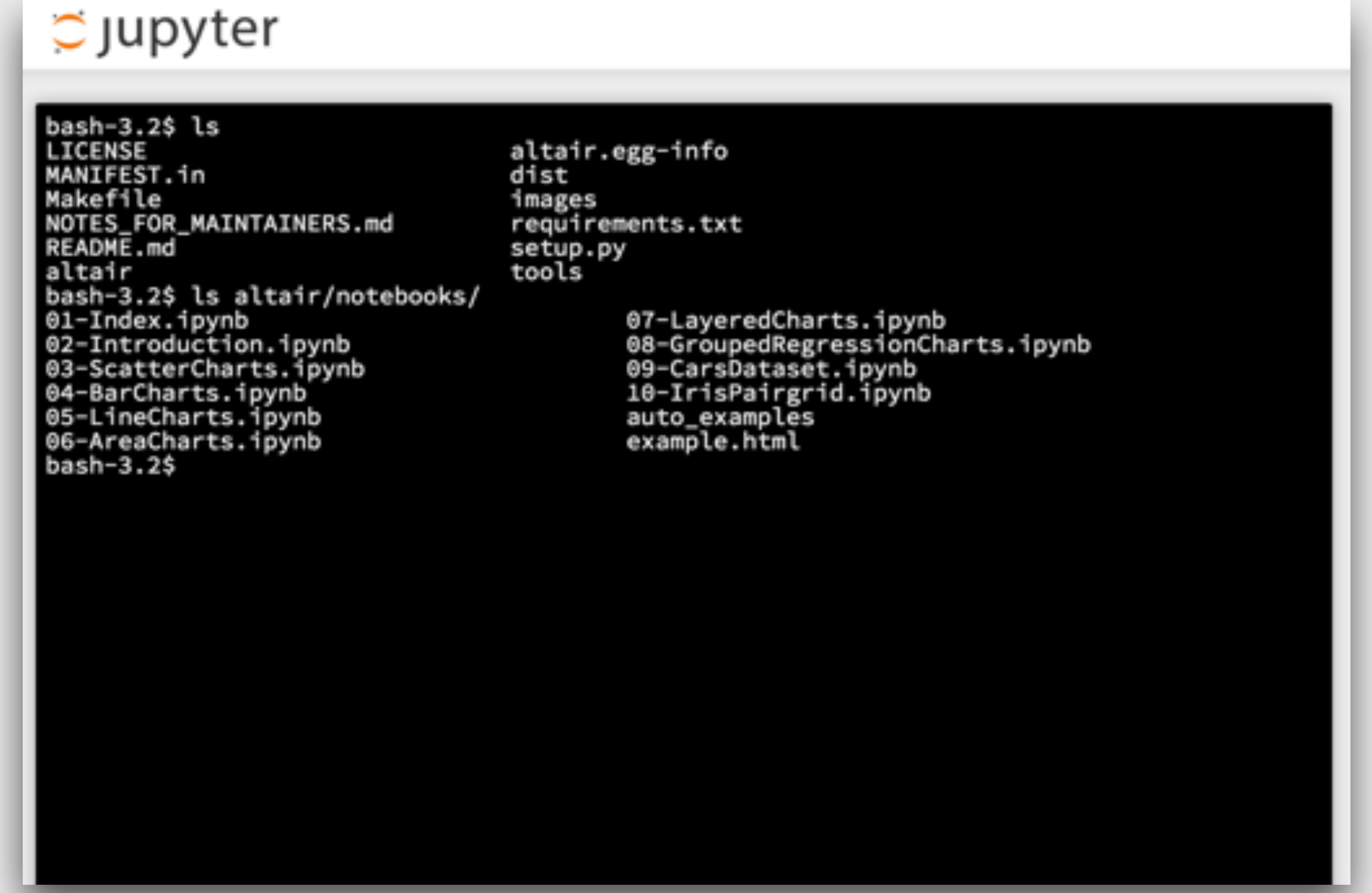
🏠 / github / ellisonbg / altair

Name ↑	Last Modified ↑
..	seconds ago
altair	4 days ago
altair.egg-info	4 days ago
dist	3 days ago
images	a month ago
tools	6 days ago
LICENSE	5 days ago
Makefile	6 days ago
MANIFEST.in	3 days ago
NOTES_FOR_MAINTAINERS.md	5 days ago
README.md	3 days ago

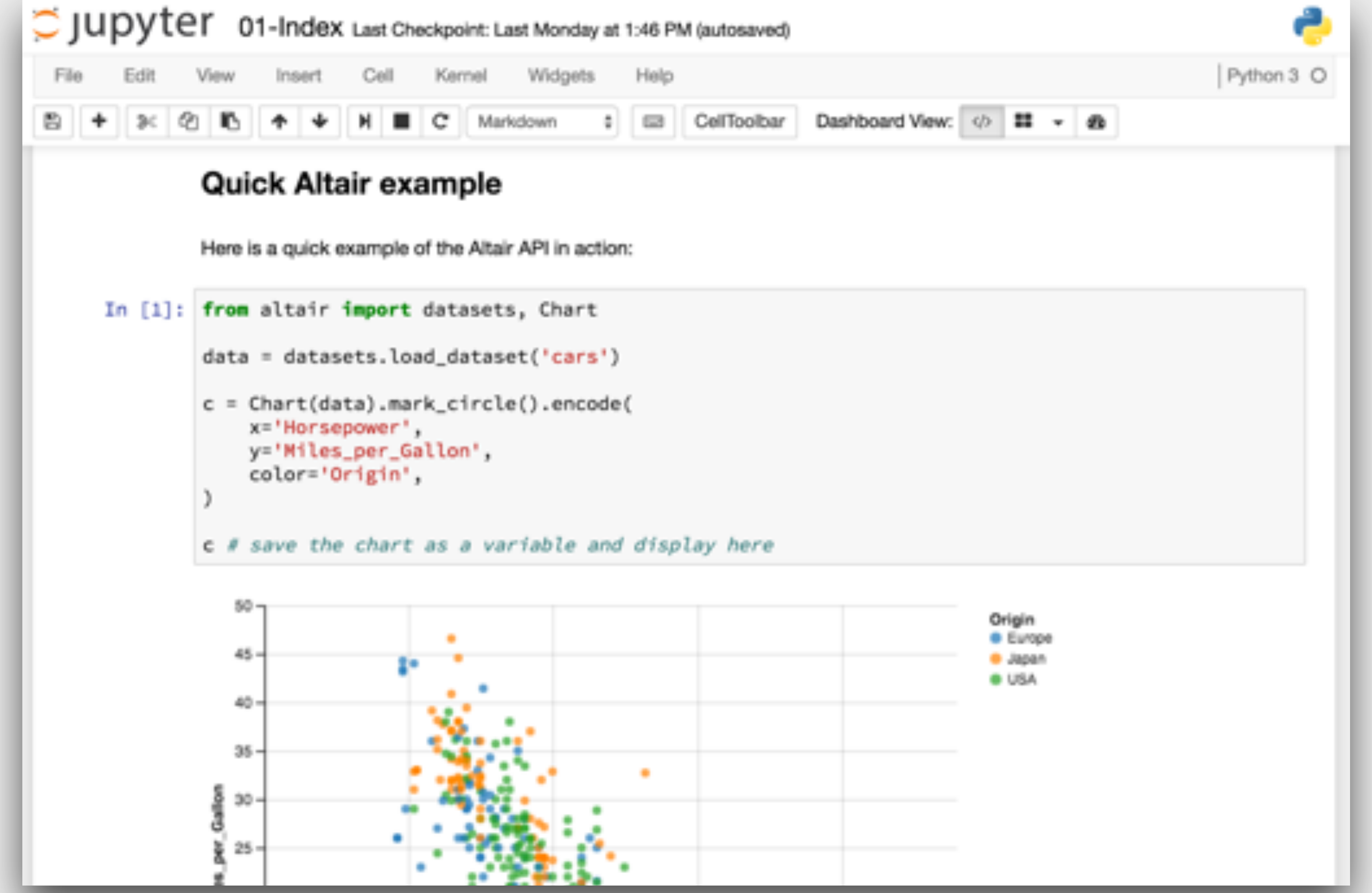


setup.py ✔ Last Sunday at 11:44 PM

```python
import io
import os
import re

try:
    from setuptools import setup
except ImportError:
    from distutils.core import setup


def read(path, encoding='utf-8'):
    path = os.path.join(os.path.dirname(__file__), path)
    with io.open(path, encoding=encoding) as fp:
        return fp.read()


def version(path):
    """Obtain the packge version from a python file e.g. pkg/__init__.py

    See <https://packaging.python.org/en/latest/single_source_version.html>.
    """
    version_file = read(path)
    version_match = re.search(r"""^__version__ = ['"]([^'"]*)['"]""",
                              version_file, re.M)
    if version_match:
```



```
bash-3.2$ ls
LICENSE                                altair.egg-info
MANIFEST.in                            dist
Makefile                               images
NOTES_FOR_MAINTAINERS.md               requirements.txt
README.md                              setup.py
altair                                 tools
bash-3.2$ ls altair/notebooks/
01-Index.ipynb                                 07-LayeredCharts.ipynb
02-Introduction.ipynb                          08-GroupedRegressionCharts.ipynb
03-ScatterCharts.ipynb                         09-CarsDataset.ipynb
04-BarCharts.ipynb                             10-IrisPairgrid.ipynb
05-LineCharts.ipynb                            auto_examples
06-AreaCharts.ipynb                            example.html
bash-3.2$
```



01-Index Last Checkpoint: Last Monday at 1:46 PM (autosaved)

Quick Altair example

Here is a quick example of the Altair API in action:

```python
In [1]: from altair import datasets, Chart

        data = datasets.load_dataset('cars')

        c = Chart(data).mark_circle().encode(
            x='Horsepower',
            y='Miles_per_Gallon',
            color='Origin',
        )

        c # save the chart as a variable and display here
```

Building Blocks

File Browser

Notebooks

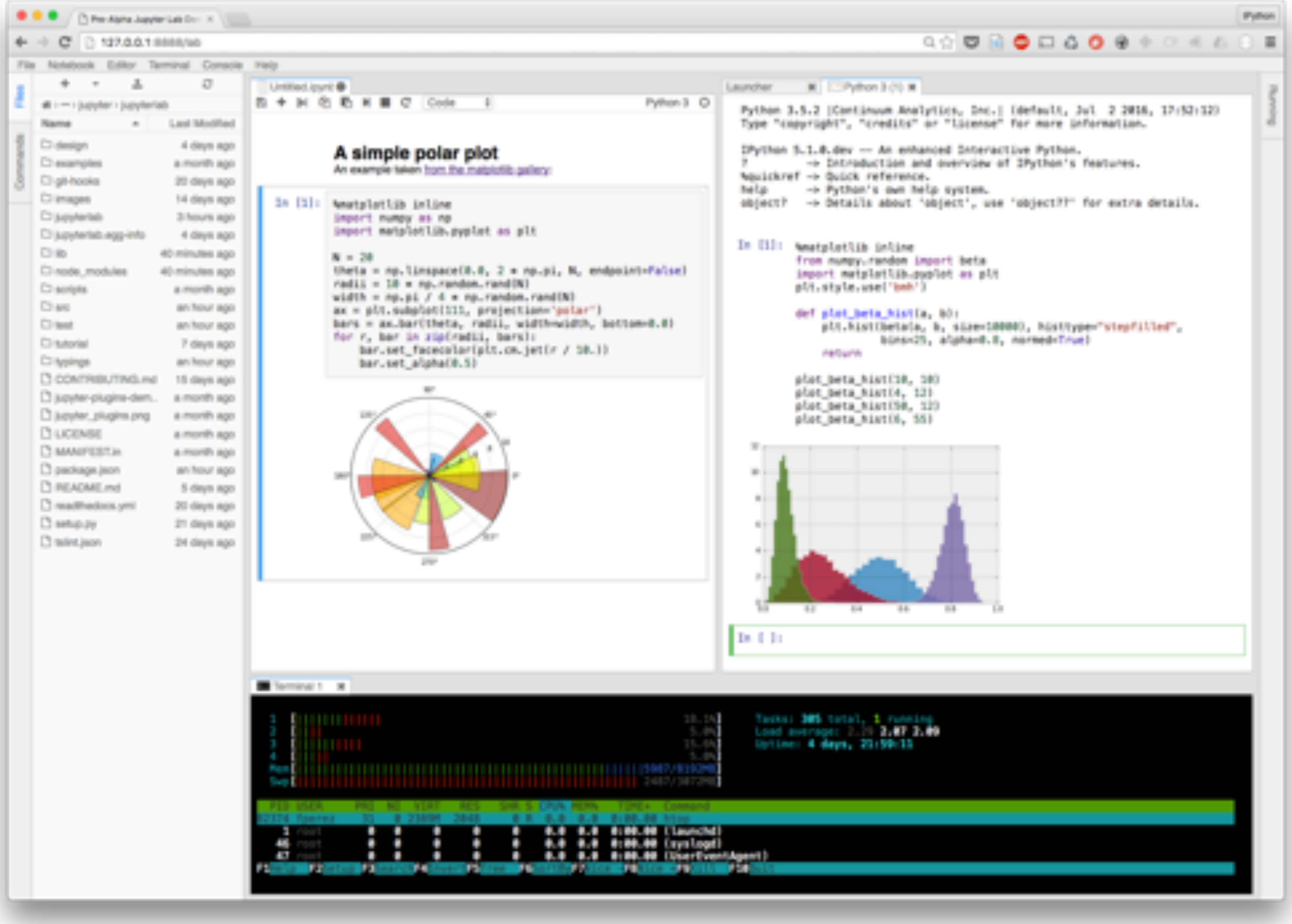
Terminal

Text Editor

Kernels

Output

JupyterLab: a grand unified theory of Jupyter



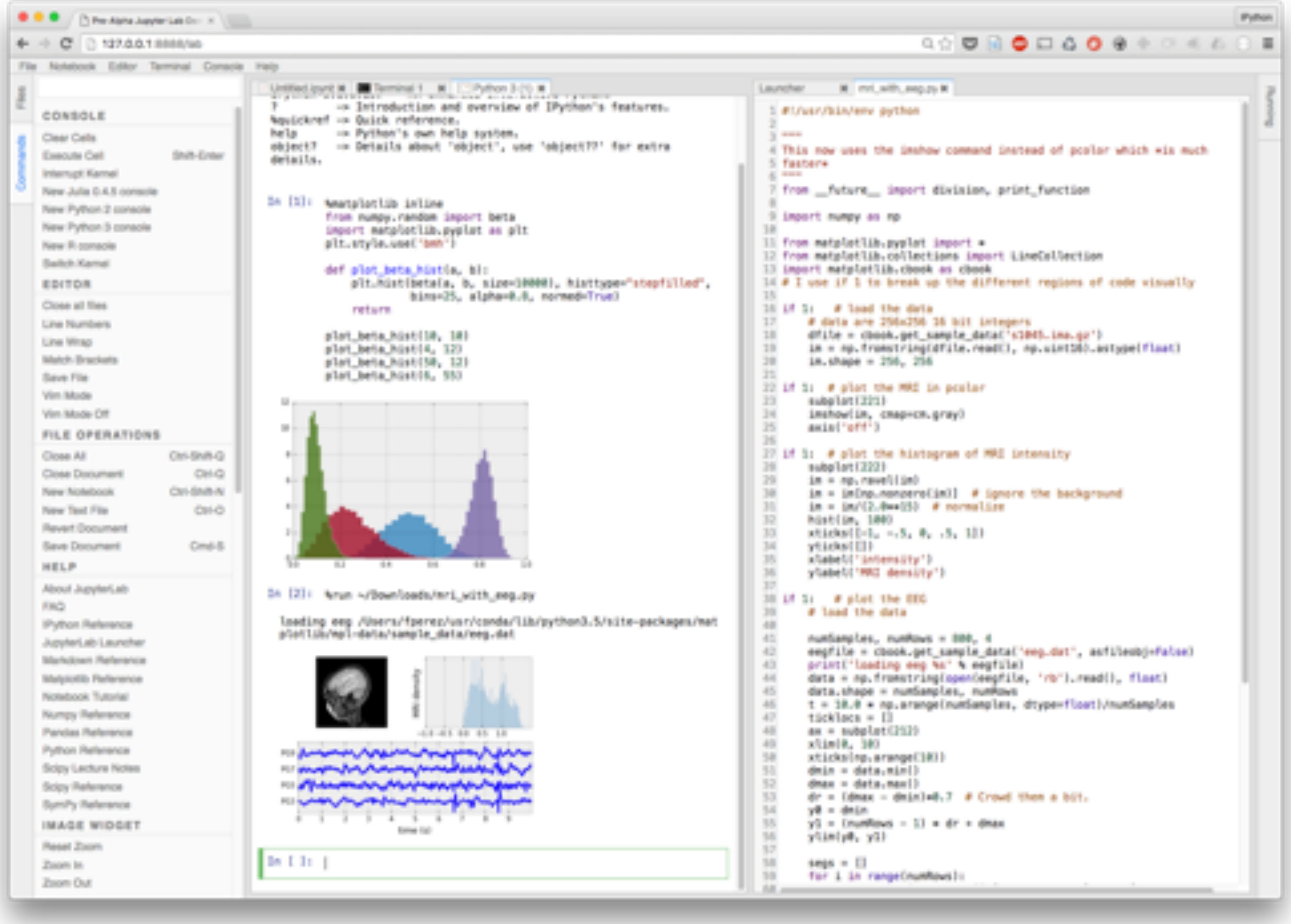
A simple polar plot

An example taken from the matplotlib gallery:

```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt

        N = 20
        theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
        radii = 10 * np.random.rand(N)
        width = np.pi / 4 * np.random.rand(N)
        ax = plt.subplot(111, projection='polar')
        bars = ax.bar(theta, radii, width=width, bottom=0.0)
        for r, bar in zip(radii, bars):
            bar.set_facecolor(plt.cm.jet(r / 10.))
            bar.set_alpha(0.5)
```

```
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0.dev -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: %matplotlib inline
        from numpy.random import beta
        import matplotlib.pyplot as plt
        plt.style.use('bmh')

        def plot_beta_hist(a, b):
            plt.hist(beta(a, b, size=10000), histtype="stepfilled",
                     bins=25, alpha=0.8, normed=True)
            return

        plot_beta_hist(10, 10)
        plot_beta_hist(4, 12)
        plot_beta_hist(50, 12)
        plot_beta_hist(6, 55)

In [ ]:
```

```
Tasks: 305 total, 1 running
Load average: 2.29 2.07 2.09
Uptime: 4 days, 21:59:15
```



```
In [2]: %run ~/Downloads/mri_with_eeg.py

loading eeg /Users/fperez/usr/conda/lib/python3.5/site-packages/mat
plotlib/mpl-data/sample_data/eeg.dat

In [ ]:
```

```
#!/usr/bin/env python

"""
This now uses the imshow command instead of pcolor which *is much
faster*
"""
from __future__ import division, print_function

import numpy as np

from matplotlib.pyplot import *
from matplotlib.collections import LineCollection
import matplotlib.cbook as cbook
# I use if 1 to break up the different regions of code visually

if 1:   # load the data
    # data are 256x256 16 bit integers
    dfile = cbook.get_sample_data('s1045.ima.gz')
    im = np.fromstring(dfile.read(), np.uint16).astype(float)
    im.shape = 256, 256

if 1:  # plot the MRI in pcolor
    subplot(221)
    imshow(im, cmap=cm.gray)
    axis('off')

if 1:  # plot the histogram of MRI intensity
    subplot(222)
    im = np.ravel(im)
    im = im[np.nonzero(im)]  # ignore the background
    im = im/(2.0**15)  # normalize
    hist(im, 100)
    xticks([-1, -.5, 0, .5, 1])
    yticks([])
    xlabel('intensity')
    ylabel('MRI density')

if 1:   # plot the EEG
    # load the data

    numSamples, numRows = 800, 4
    eegfile = cbook.get_sample_data('eeg.dat', asfileobj=False)
    print('loading eeg %s' % eegfile)
    data = np.fromstring(open(eegfile, 'rb').read(), float)
    data.shape = numSamples, numRows
    t = 10.0 * np.arange(numSamples, dtype=float)/numSamples
    ticklocs = []
    ax = subplot(212)
    xlim(0, 10)
    xticks(np.arange(10))
    dmin = data.min()
    dmax = data.max()
    dr = (dmax - dmin)*0.7  # Crowd them a bit.
    y0 = dmin
    y1 = (numRows - 1) * dr + dmax
    ylim(y0, y1)

    segs = []
    for i in range(numRows):
```

Huge Team Effort!

ANACONDA

Bloomberg

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

Roadmap

JupyterLab Beta: Use It Today

```
conda install -c conda-forge jupyterlab
```

or (`--pre` only needed for a bit)

```
pip install --pre jupyterlab
```

- 1.0 coming in a few days/weeks!
  - For all users, extension developers
- Eventually:
  - Classic notebook will be retired

Let’s play live!