

# Project Jupyter: From Computational Notebooks to Large Scale Data Science with Sensitive Data

Brian Granger  
Cal Poly, Physics/Data Science  
Project Jupyter, Co-Founder

ACM Learning Seminar  
September 2018



Cal Poly

# Outline

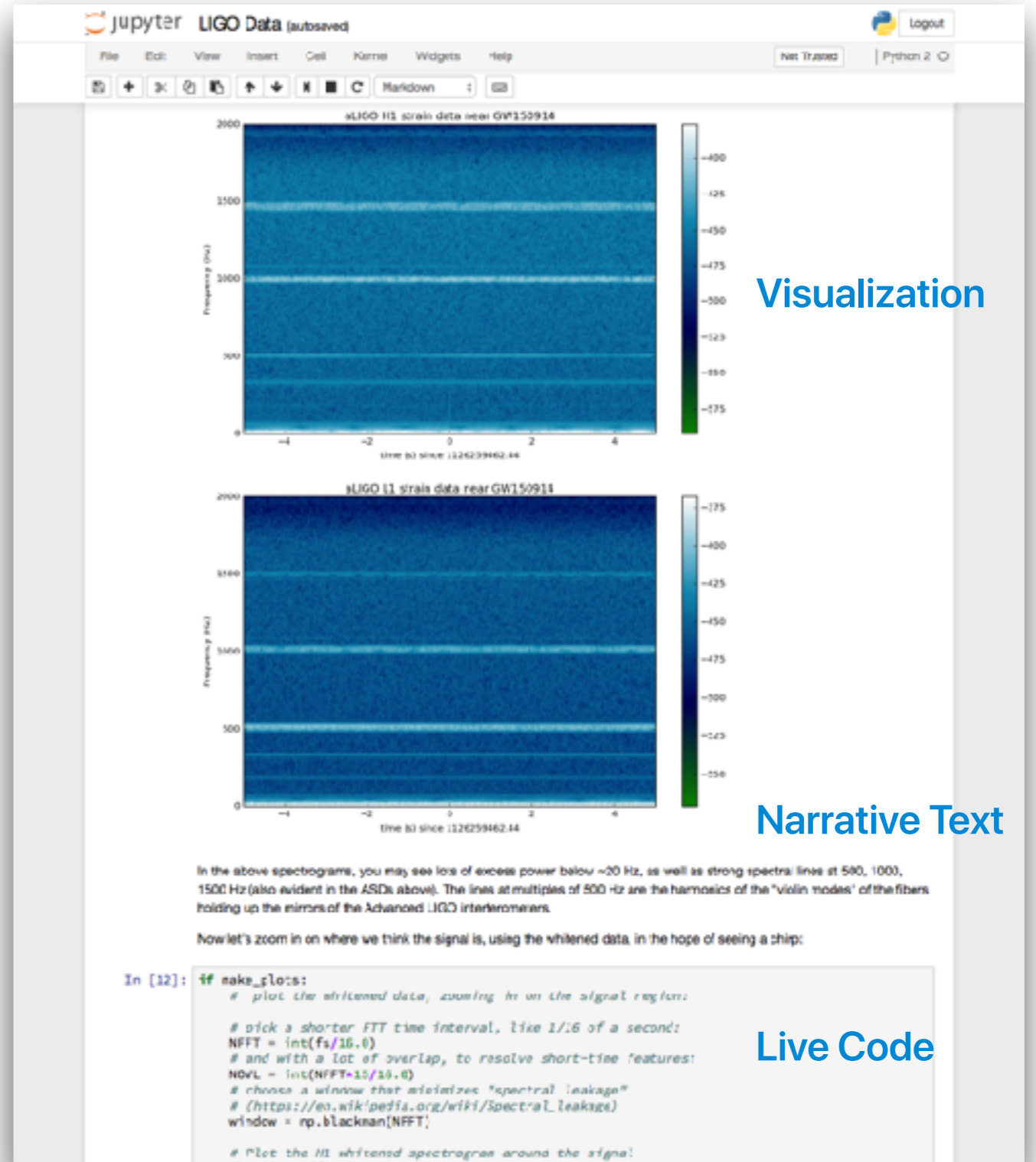
- Jupyter + Computational Notebooks
- Data Science in Large, Complex Organizations
  - JupyterLab
  - JupyterHub



Project Jupyter exists to develop **open-source** software, open-standards and services for interactive and reproducible computing.

# The Jupyter Notebook

- Project Jupyter (<https://jupyter.org>) started in 2014 as a spinoff of IPython
- Flagship application is the Jupyter Notebook
- Interactive, exploratory, browser-based computing environment for data science, scientific computing, ML/AI
- Notebook document format (**.ipynb**):
  - Live code, narrative text, equations (LaTeX), images, visualizations, audio
  - Reproducible Computational Narrative
- ~100 programming languages supported
- Over 500 contributors across 100s of GitHub repositories.
- 2017 ACM Software System Award.



Example notebook from the LIGO Collaboration

# Before Moving On: Attribution?

# Who Builds Jupyter?

- Jupyter Steering Council:
  - Fernando Perez, Brian Granger, Min Ragan-Kelley, Paul Ivanov, Thomas Kluyver, Jason Grout, Matthias Bussonnier, Damian Avila, Steven Silvester, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Carol Willing, Sylvain Corlay, Peter Parente, Ana Ruvalcaba, Afshin Darian, M Pacer.
- Other Core Jupyter Contributors:
  - Chris Holdgraf, Yuvi Panda, M Pacer, Ian Rose, Tim Head, Jessica Forde, Jamie Whitacre, Grant Nestor, Chris Colbert, Cameron Oelsen, Tim George, Maarten Breddels, **100s others**.
- Dozens of interns at Cal Poly
- Funding
  - Alfred P. Sloan Foundation, Moore Foundation, Helmsley Trust, Schmidt Foundation
- NumFOCUS: Parent 501(c)3 for Project Jupyter and other open-source projects

**How to think about the contributions of different people? What is the right narrative?**

# Attribution Narrative: Not This!



Jupyter is not the heroic work of one person, or even a small number of people.

# Attribution Narrative: More Like This!













Jupyter is created by a large number of people with different strengths working in diverse teams.



Onwards!

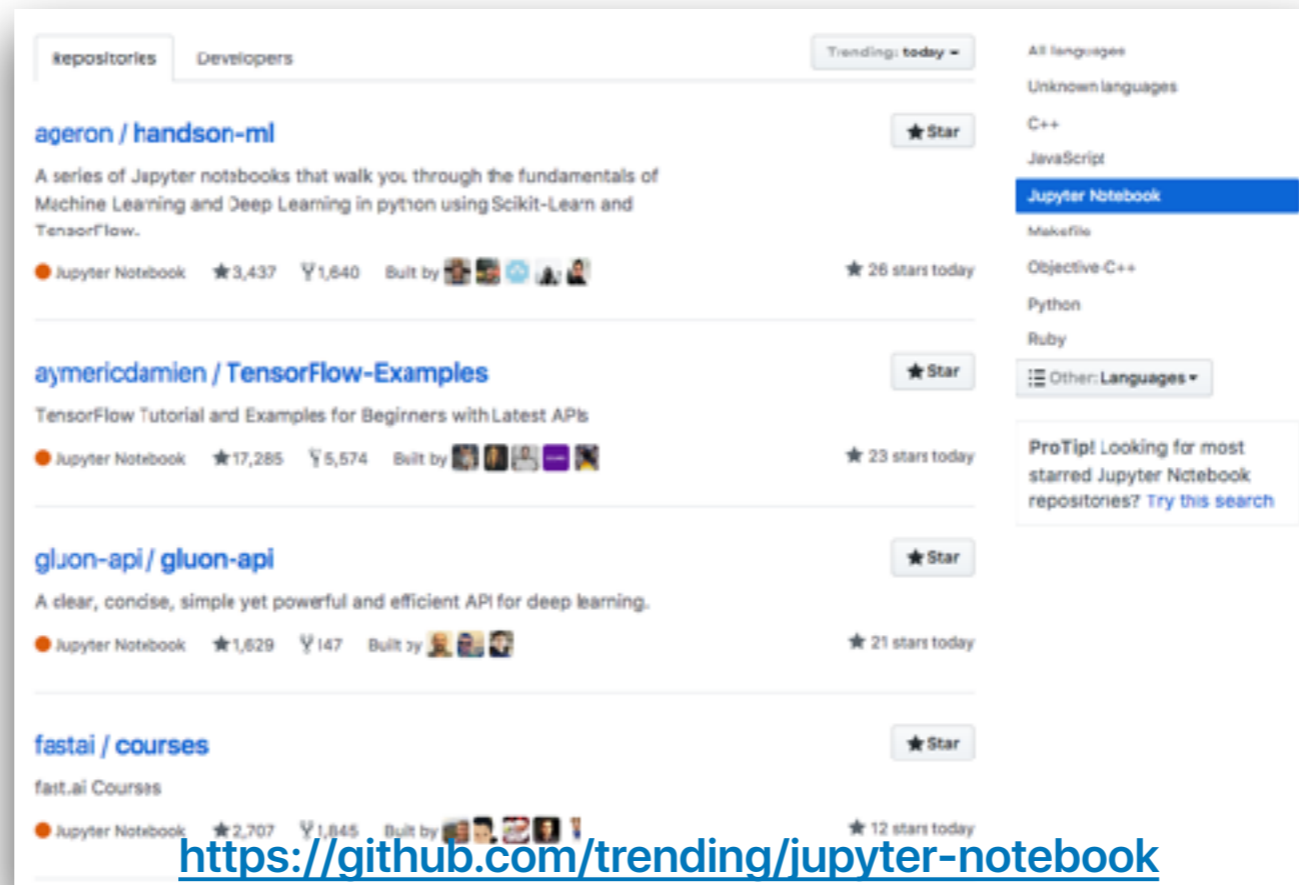
# International User Community of Millions

As of Summer 2018, Asia is the most represented continent in Jupyter's web traffic.

Country ?	Acquisition		
	Sessions ? ↓	% New Sessions ?	New Users ?
	<b>1,037,098</b> % of Total: 100.00% (1,037,098)	<b>54.00%</b> Avg for View: 53.92% (0.16%)	<b>560,065</b> % of Total: 100.16% (559,154)
1.  <b>United States</b>	<b>326,770</b> (31.51%)	53.90%	<b>176,128</b> (31.45%)
2.  <b>China</b>	<b>82,152</b> (7.92%)	58.00%	<b>47,652</b> (8.51%)
3.  <b>India</b>	<b>77,828</b> (7.50%)	52.25%	<b>40,664</b> (7.25%)
4.  <b>United Kingdom</b>	<b>44,005</b> (4.24%)	55.12%	<b>24,255</b> (4.33%)
5.  <b>Germany</b>	<b>37,732</b> (3.64%)	56.24%	<b>21,221</b> (3.79%)
6.  <b>Russia</b>	<b>37,502</b> (3.62%)	43.94%	<b>16,479</b> (2.94%)
7.  <b>Canada</b>	<b>31,976</b> (3.08%)	57.75%	<b>18,465</b> (3.30%)
8.  <b>France</b>	<b>26,359</b> (2.54%)	59.48%	<b>15,678</b> (2.80%)
9.  <b>Japan</b>	<b>24,004</b> (2.31%)	56.51%	<b>13,564</b> (2.42%)
10.  <b>Brazil</b>	<b>23,434</b> (2.26%)	54.62%	<b>12,799</b> (2.29%)

Google Analytics for [jupyter.org](http://jupyter.org) for September 2017

# Trending Notebooks on GitHub



A screenshot of the GitHub trending page for Jupyter Notebooks. The page shows a list of repositories with their names, descriptions, star counts, and daily star counts. The repositories listed are:

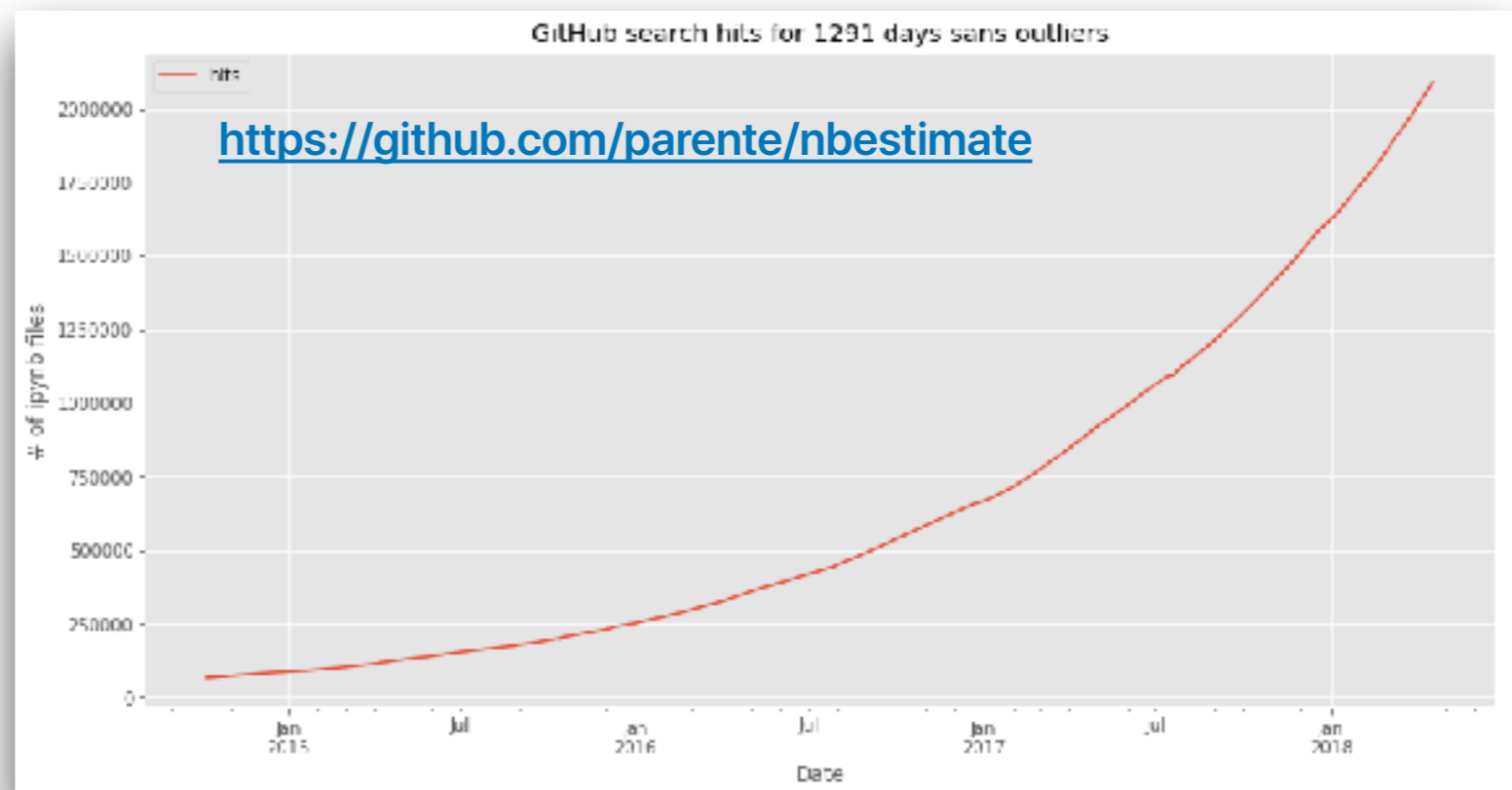
- ageron / hands-on-ml**: A series of Jupyter notebooks that walk you through the fundamentals of Machine Learning and Deep Learning in python using Scikit-Learn and TensorFlow. 3,437 stars, 1,640 forks, 26 stars today.
- aymericdamien / TensorFlow-Examples**: TensorFlow Tutorial and Examples for Beginners with Latest APIs. 17,285 stars, 5,574 forks, 23 stars today.
- gluon-api / gluon-api**: A clear, concise, simple yet powerful and efficient API for deep learning. 1,629 stars, 147 forks, 21 stars today.
- fastai / courses**: fast.ai Courses. 2,707 stars, 1,845 forks, 12 stars today.

On the right side, there is a filter menu for languages, with 'Jupyter Notebook' selected. A 'ProTip!' box suggests searching for the most starred Jupyter Notebook repositories.

<https://github.com/trending/jupyter-notebook>

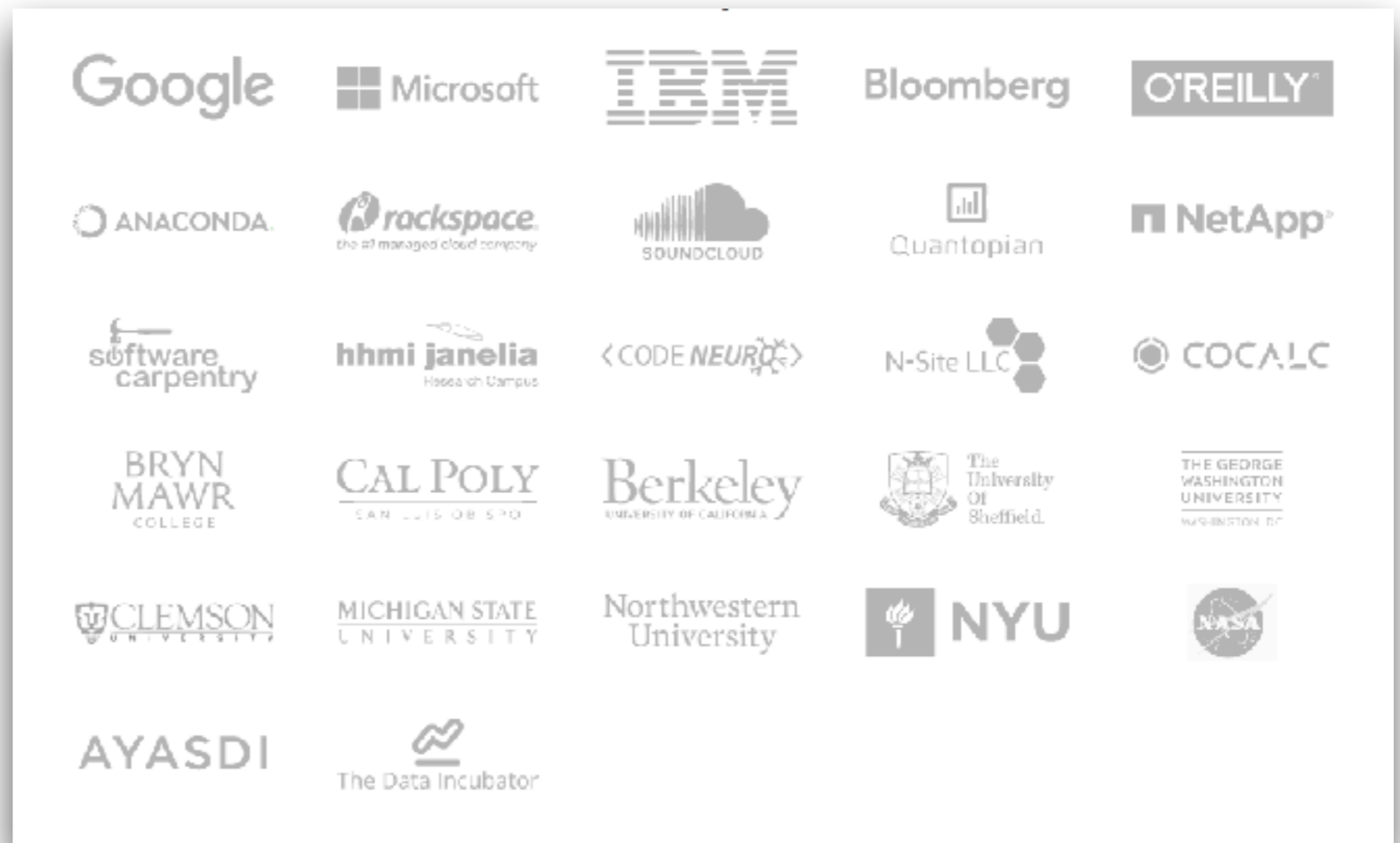
Over 2.5M Public Notebooks on GitHub

# of Public Notebooks on GitHub



# Organizational Usage

We are seeing strong organizational adoption, driven by JupyterHub and other cloud based deployments



... and 100s - 1000s more

- Data science platforms (Teradata, Google, Microsoft, IBM, AWS, Anaconda, Domino, CoCalc, Dataiku, data.world, Kaggle,...)
- Data journalism (LA Times, Chicago Tribune, BuzzFeedNews,...)
- Publishing (Springer, O'Reilly)
- K-12, University Education (Berkeley, Cal Poly,...)
- Data Science/ML/AI Teams (1000's)
- Large scale scientific collaborations (LSST, CERN, LIGO/VIRGO, PIMS, NASA JPL, Pangeo,...)

# An Amazing Community of Users

LIGO Open Science Center  
LIGO is operated by California Institute of Technology and Massachusetts Institute of Technology and supported by the U.S. National Science Foundation.

Getting Started  
Tutorials  
Data  
Events  
Duke University  
Timeline  
My Source  
Subscribers  
Open Access  
About LIGO  
Data Analysis Projects  
Acknowledgments

nature genetics  
home > archive > issue > analysis > abstract

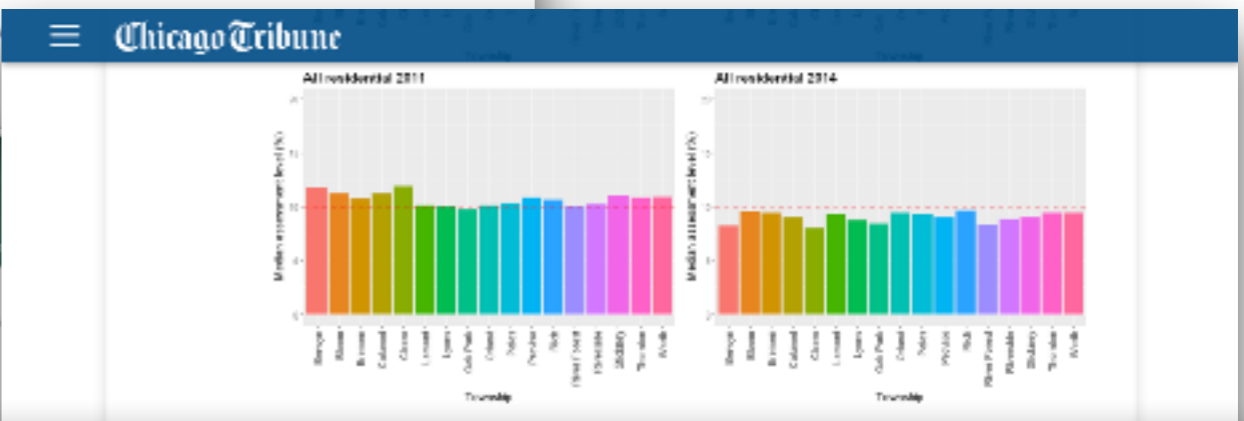
ARTICLE PREVIEW  
view full access options >

NATURE GENETICS | ANALYSIS  
日本語版

Multi-tiered genomic analysis  
cancer ties *TP53* mutation to

Andrew M Gross, Ryan K Orsco, John P Shen, Ann Holree, Michel Choucri, Charles S Coffey, Scott M L Jennifer R Grandis, Quyen T Nguyen & Trey Ideker

Affiliations | Contributions | Corresponding author



GenePattern  
A Platform for Reproducible Bioinformatics

Use GenePattern  
GenePattern Basics  
10-minute Tutorial  
Community

O'REILLY

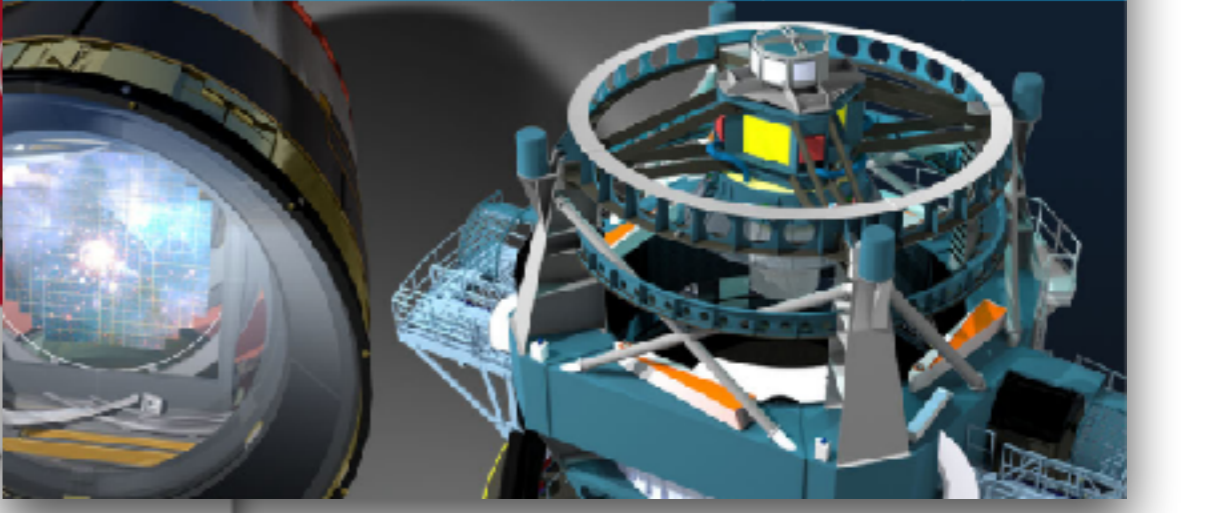
Python  
Data Science  
Handbook

ESSENTIAL TOOLS FOR WORKING WITH DATA

LSST  
Large Synoptic Survey Telescope  
Opening a Window of Discovery on the Dynamic Universe

ABOUT | SCIENCE GOALS | PARTICIPATE | GALLERY | NEWS

Blog - GP updates  
• GenePattern 1.8.10 is released!  
• GenePattern Notebook Repository Update Released - v1.7.0



# Example: LSST

- Large Synoptic Survey Telescope (<https://www.lsst.org/>)
- 27ft primary mirror
- 10 year operating period
- Each image covers 40 moons worth of the sky
- 15 TB of data every night!
- Computational platform based on JupyterHub + JupyterLab:
  - User base: "every astronomer on the planet" (~7,500)
  - "Next-to-the-data" analysis
  - Data access (3 PB Database, 4 PB files)
  - Scalable compute (2,400 cores)
  - Interactive analysis, modeling, simulation, visualization
  - Collaboration



# Open-Standards for Interactive Computing

- The foundation of Jupyter is a set of open standards for interactive computing.
- Jupyter Notebook format (<https://github.com/jupyter/nbformat>)
  - JSON based document format for code, data, narrative text, equations, output
  - Independent of user interface, programming language
- Jupyter Message Specification ([https://github.com/jupyter/jupyter\\_client](https://github.com/jupyter/jupyter_client))
  - JSON based network protocol for interactive computing **user interfaces** (Jupyter Notebook) to talk to **kernels** that runs code interactively in a given programming language.
  - Transport layer over ZeroMQ or WebSockets.
- Jupyter Notebook Server ([https://github.com/jupyter/jupyter\\_server](https://github.com/jupyter/jupyter_server))
  - A set of WebSocket and HTTP APIs for remote access to building blocks of interactive computing:
    - File system
    - Terminal
    - Kernels

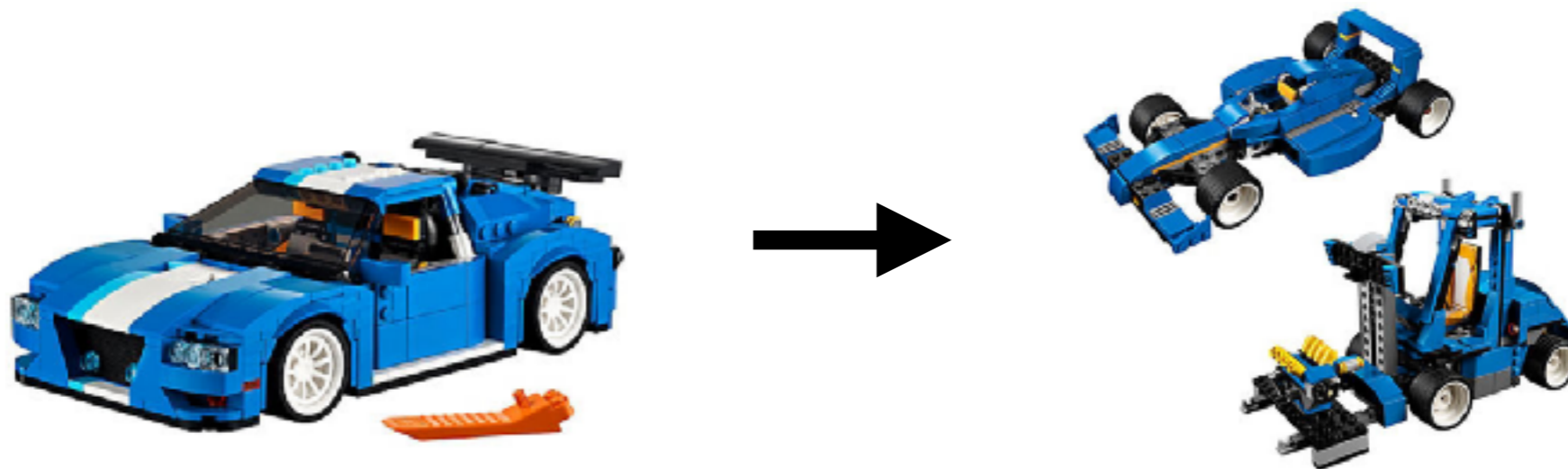
# Open-Source Software for Interactive Computing

- **Jupyter Notebook**: the original Jupyter notebook server and user interface.
- **JupyterLab**: next generation user interface for Jupyter notebooks.
- **JupyterHub**: deploy Jupyter to large organizations in a scalable, secure and maintainable manner.
- **IPython**: the Python kernel for Jupyter.
- **Jupyter Widgets**: interactive user interfaces within Jupyter notebooks.
- **nbconvert**: convert notebooks to other formats (HTML, Markdown, LaTeX).



# Building Blocks for Interactive Computing

- Jupyter's open standards and open-source software provides a set of building blocks that can be used to build a wide range of interactive computing systems.
- LEGO for interactive computing!



- Examples: JupyterLab, nteract, Google Colaboratory, Binder

# JupyterLab

JupyterLab is Jupyter's next-generation user interface. It uses the same notebook format, server and network protocols.

The screenshot displays the JupyterLab interface. On the left, a sidebar shows a file browser with a list of files and notebooks, including 'Data.ipynb', 'Fasta.ipynb', 'Julia.ipynb', 'Lorenz.ipynb' (selected), 'R.ipynb', 'iris.csv', 'lightning.json', and 'lorenz.py'. The main area is divided into several panes:

- Code Editor:** Shows the 'Lorenz.ipynb' notebook. The text reads: "In this Notebook we explore the Lorenz system of differential equations:" followed by the equations:
$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$
Below the equations, 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." The code cell contains: 

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```
- Output View:** Displays three sliders for parameters: sigma (set to 10.00), beta (set to 2.07), and rho (set to 28.00). Below the sliders is a 3D plot of the Lorenz attractor, showing its characteristic butterfly shape.
- Terminal/Code Editor:** Shows the 'lorenz.py' file with the following code:

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x,y,z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```

# ninteract

ninteract is an alternate user interface for working with Jupyter notebooks, focused on simplicity.

Open-source and sponsored by Netflix.

Uses the same notebook document format, server and network protocols.

```
~/linked-charts.ipynb - idle
[13] from vega_datasets import data
import pandas as pd
df = pd.read_json(data.movies.url)
df = df[df["MPAA_Rating"].isin(["G", "PG"])]

[17] df.sample(3)
```

	Creative_Type	Director	Distributor	IMDB_Rating	IMDB_Votes	MPAA_Rating	Major_Genre	Production_Budget	Release_Date
2989	Kids Fiction	None	Walt Disney Pictures	6.6	12099.0	PG	Adventure	100000000.0	27-Nov-02
2905	Science Fiction	None	Warner Bros.	5.4	17513.0	PG	Adventure	85000000.0	15-Aug-08
2486	Historical Fiction	Michael O. Sajbel	Rocky Mountain Pictures	6.0	2993.0	PG	Drama	20000000.0	13-Oct-06

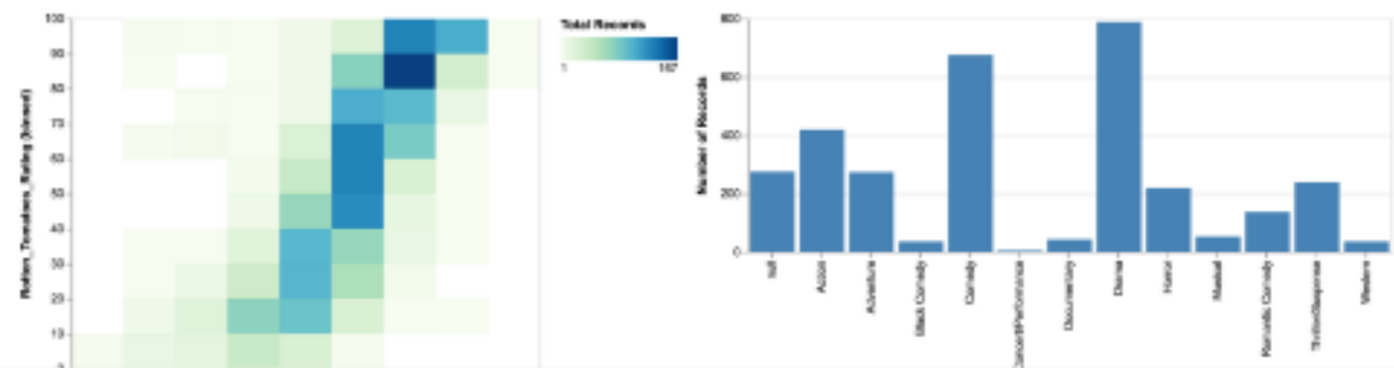
## Creating Interactive Visualizations with Altair

We can combine charts, linking them via an altair.selection

```
[13] import altair as alt

pts = alt.selection(type="single", encodings=['x'])

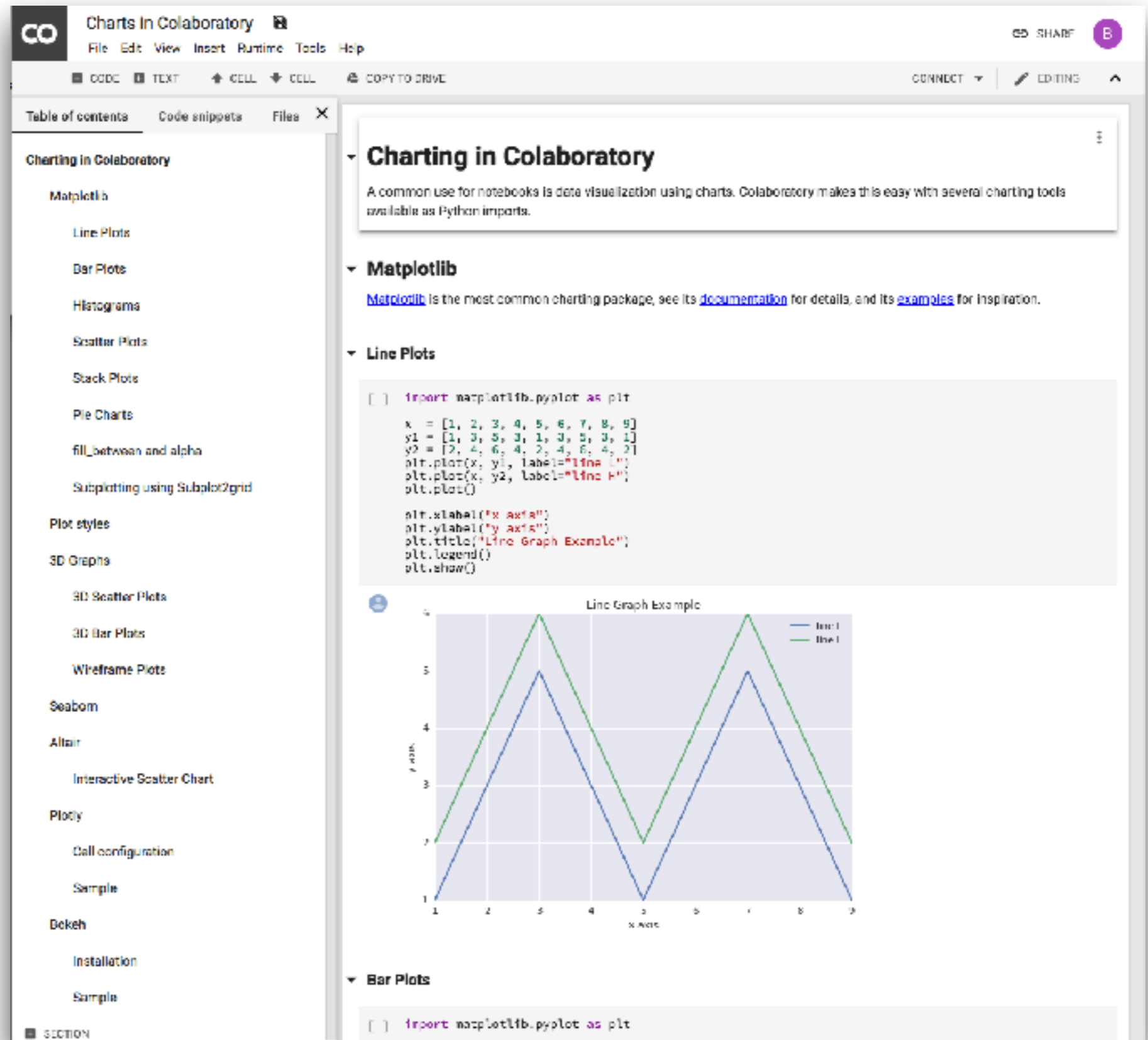
alt.hconcat(
  alt.Chart(df).mark_rect().encode(
    alt.X('IMDB_Rating:Q', bin=True),
    alt.Y('Rotten_Tomatoes_Rating:Q', bin=True),
    alt.Color('count()'),
    scale=alt.Scale(scheme='greenblue'),
    legend=alt.Legend(title='Total Records')
  ),
  alt.Chart(df).mark_bar().encode(
    x='Major_Genre:N', y='count()', color=alt.condition(pts, alt.ColorValue("steelblue"), alt.ColorValue("grey")),
  ).properties(
    selection=pts, width=550, height=200
  )
).resolve_legend(
  color="independent", size="independent"
)
```



# Google Colaboratory

Colaboratory is an alternate user interface for working with Jupyter notebooks, integrated with Google Drive.

Uses the same notebook format and network protocols.



The screenshot displays the Google Colaboratory interface. On the left is a table of contents for a notebook titled "Charting in Colaboratory". The main area shows the notebook content, which includes a title, an introductory paragraph, a code cell for Matplotlib, and a line plot.

**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**

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 1")
plt.plot(x, y2, label="line 2")
plt.plot()

plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("Line Graph Example")
plt.legend()
plt.show()
```

The plot, titled "Line Graph Example", shows two lines on a grid. The x-axis is labeled "x axis" and ranges from 1 to 9. The y-axis is labeled "y axis" and ranges from 1 to 6. The blue line (line 1) has values [1, 3, 5, 3, 1, 3, 5, 3, 1] and the green line (line 2) has values [2, 4, 6, 4, 2, 4, 6, 4, 2]. Both lines show a repeating triangular pattern.

**Bar Plots**

```
[ ] import matplotlib.pyplot as plt
```

<https://colab.research.google.com/>

# Binder

Binder turns any Git repo with notebooks into a live notebook server for anyone in the world. It works with any Jupyter user interface and programming language (kernel).

The screenshot shows the Binder website's main interface. At the top is the Binder logo with the text "(beta)". Below it is the heading "Turn a GitHub repo into a collection of interactive notebooks". A sub-heading reads: "Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere." The main form is titled "Build and launch a repository" and contains several input fields: "GitHub repository name or URL" with a dropdown menu set to "GitHub", "Git branch, tag, or commit", "Path to a notebook file (optional)" with a dropdown menu set to "file", and a "launch" button. Below the form, there are instructions to copy the URL and share it, and a section for generating a Binder badge for a README file.

The screenshot shows a GitHub repository file list. The files listed are: "slides" (Updated slides an...), ".gitignore" (More cleanup), ".travis.yml" (try latest pip), "LICENSE" (Merge pull request), "README.md" (Tiny typo fix in RE...), "appveyor.yml" (Update appveyor), "talks.yml" (Move the Lorenz), and "tasks.py" (Update for jlab 0...). Below the file list is a section for "README.md" which contains the heading "JupyterLab Demo" and two badges: "build passing" and "launch binder". An arrow points from the "launch" button in the Binder form to the "launch binder" badge in the README.

<https://mybinder.org/>

# Data Science in Large, Complex Organizations

# Human Centered Design

- If you don't design for humans, you will design for computers and humans will be miserable.
- Examples of such failures:
  - The primary "user interface" for working on a remote computer is still SSH
  - Tracebacks used to communicate to users when a program raises an exception
- See Alan Cooper's "The Inmates Are Running the Asylum"
- Scientific computing and data science, are, by definition, **human-centered activities** that involve iterative exploration, analytical reasoning, visualization, mathematical abstraction, model building, moral and ethical reasoning, and decision making.
- In large organizations, there are a **diverse range of individuals working with code and data**: data scientists, data engineers, analytics, marketing, sales, product managers, university administrators, teachers, statisticians, etc.
- **Not everyone who works with data wants or needs to write or look at code.**

# Collaboration is Essential

- Large organizations have complex human networks of people that need to work together.
- Individuals have different skill sets, responsibilities, access permissions, roles, priorities.
- Yet everyone needs to **look at** and **make decisions** based on the same overall data.
- GitHub is an effective collaboration tools only for people that live and breath code.



# Datasets are Often Sensitive, Confidential

- The development of data science, ML/AI have been driven by open-source software and freely available, open, public datasets.
- However, most datasets of value to organizations are sensitive and confidential and require differing levels of protection
- A range of different regulations: HIPAA, FERPA, GDPR, FedRAMP, Title 13, Title 26, SOX, GLBA, California Consumer Privacy Act, [A.B. 375](https://www.caprivacy.org/) (<https://www.caprivacy.org/>)
- Five Safes (Desai, Ritchie, Welpton 2016)
  - <http://www2.uwe.ac.uk/faculties/BBS/Documents/1601.pdf>
  - Framework for “designing, describing and evaluating access systems for data, used by data providers, data users, and regulators.”
  - Safe Projects, Safe People, Safe Data, Safe Settings, Safe Outputs
- Open-source tools can't take a “not our problem” attitude.
  - Jupyter and other open-source tools were almost certainly used by Cambridge Analytica, SCLElections, to build models with Facebook user profiles for the 2016 US election.

# How is Jupyter Tackling These Challenges?

# JupyterLab

JupyterLab is the next-generation web-based user interface for Project Jupyter

# JupyterLab

- Next-generation user-interface for Project Jupyter
- Full support for Jupyter Notebooks
- Notebooks, terminals, text editor, file browser, code console
- Extension architecture enables anyone to add capabilities to JupyterLab using modern web technologies (npm, react,...)
- Integration between builtin components and extensions through public APIs
- Rich handling of different data types
- Ready for use! JupyterLab is now out of Beta.
- <http://jupyterlab.readthedocs.io/>
- Real-time collaboration on the way!

# JupyterLab Demo

The screenshot displays the JupyterLab environment. On the left, a sidebar shows a file browser with a list of notebooks and files, including 'Lorenz.ipynb' which is currently selected. The main workspace is divided into several panes:

- Code Editor:** Contains the text "In this Notebook we explore the Lorenz system of differential equations:" followed by the Lorenz equations:
$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$
Below the equations, 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." A code cell is shown with the following code:

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```
- Output View:** Shows three interactive sliders for parameters: sigma (set to 10.00), beta (set to 2.67), and rho (set to 28.00). Below the sliders is a 3D plot of the Lorenz attractor, a complex, butterfly-shaped trajectory.
- Source Code Editor (lorenz.py):** Shows the implementation of the Lorenz system solver:

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x_dot, y_dot, z_dot = x,y,z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```

# JupyterHub

Scaling interactive computing with Jupyter to organizations

# JupyterHub

- In the Jupyter architecture, each user gets a dedicated Notebook/JupyterLab server, with containerized\* compute and persistent\* storage for files.
- JupyterHub scales this model to multiple users and large organizations:
  - Authenticator: extensible API for identifying and authenticating users (OAuth, LDAP, PAM,...)
  - Spawner: extensible API for managing single user servers (subprocess, docker, kubernetes,...)
  - Proxy: Dynamically map URLs to single user servers
- UC Berkeley, *Foundations of Data Science*, edX, 100k users on JupyterHub.

\*Usually, not required

# JupyterHub for Sensitive Data

- Organizational Data Model
  - Users, groups, roles, resources (compute, docker images, datasets,...)
  - Integration with directory services (Keycloak, Active Directory, LDAP), SAML, OIDC)
- Projects for JupyterHub
  - Shared workspace for text files, compute, Jupyter Notebooks
  - Well defined scope for collaboration and data access/security
- Telemetry and event logging
  - Needed for monitoring, auditing and compliance
- Reliable, Secure, Maintainable Deployments
  - Encryption in-transit and at-rest in the Jupyter architecture
  - Declarative, immutable, continuous deployments using Helm, Kubernetes
- With Julia Lane (NYU), Fernando Perez (Berkeley), funded by the Sloan and Schmidt Foundations.



Thank you!

Questions?