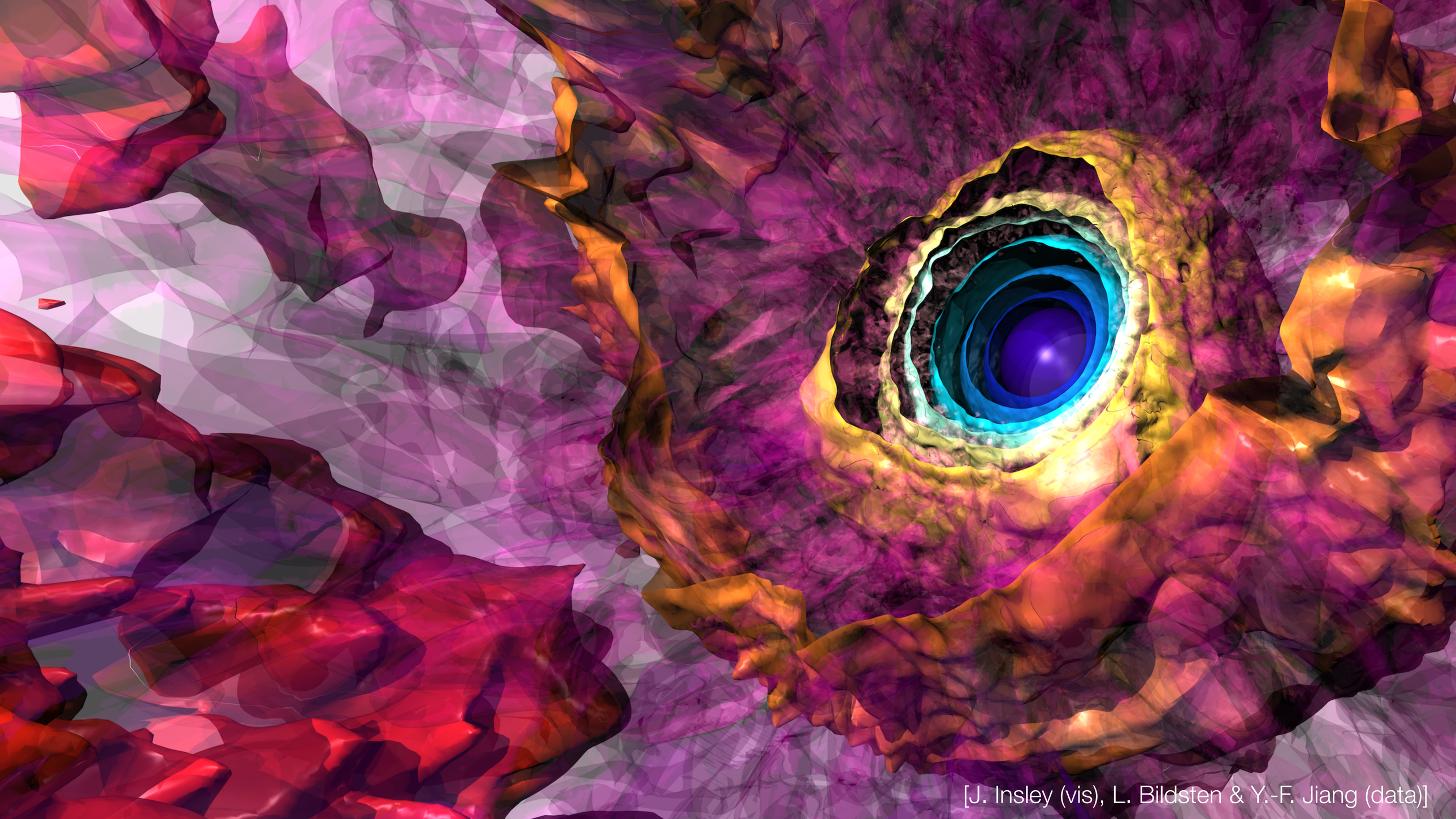
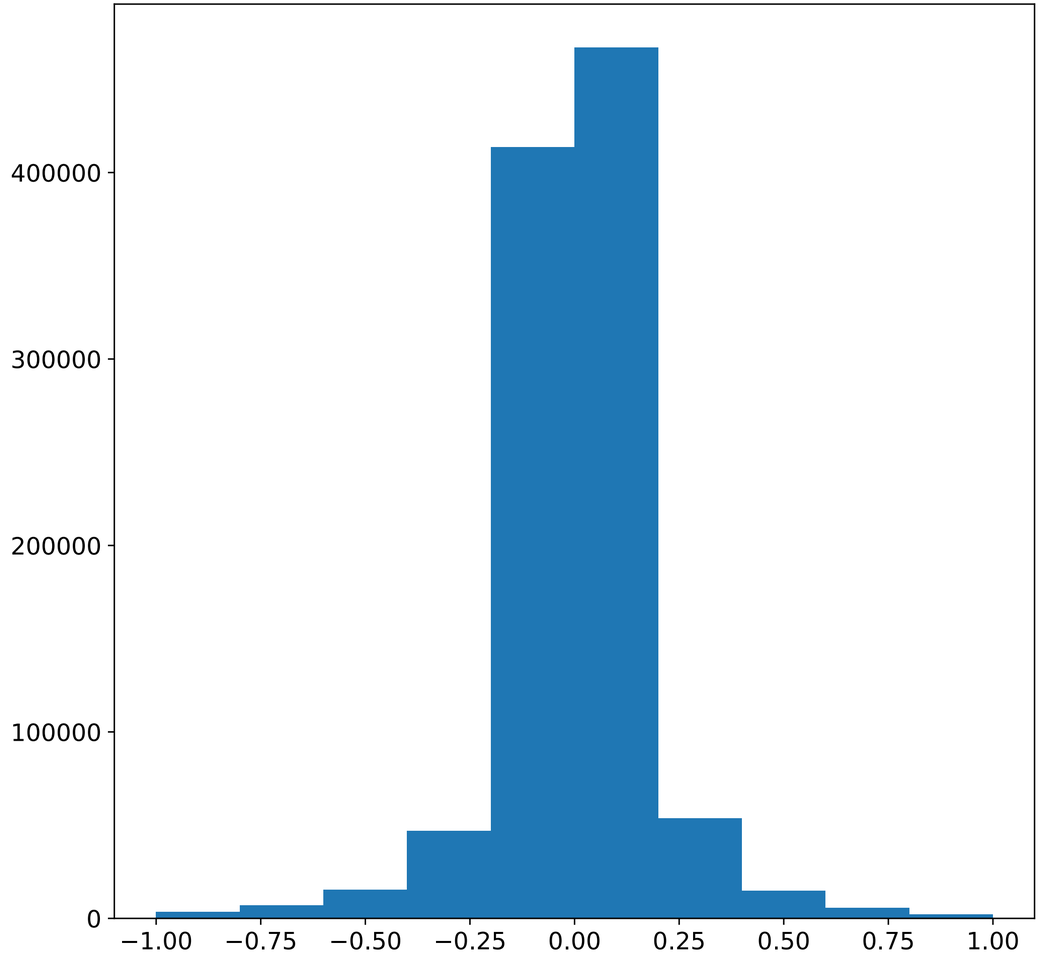
Data Visualization in Notebooks

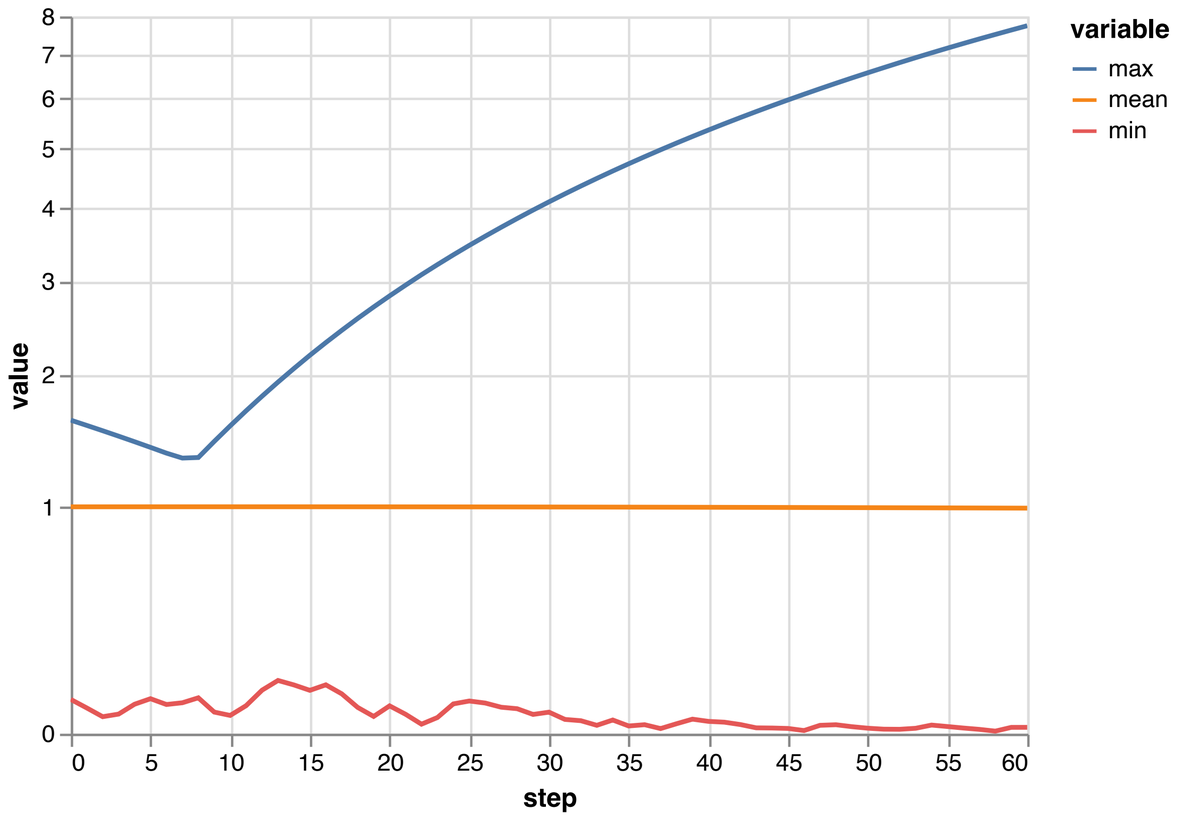
Dr. David Koop
Northern Illinois University



[J. Insley (vis), L. Bildsten & Y.-F. Jiang (data)]

Also Visualizations





"The purpose of visualization is **insight**, not pictures"

– B. Schneiderman

Notebooks

File Edit View Run Kernel Tabs Settings Help

Files | Running | Commands | Cell Tools | Tabs

Lorenz.ipynb

Code | Python 3

We explore the Lorenz system of differential equations:

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

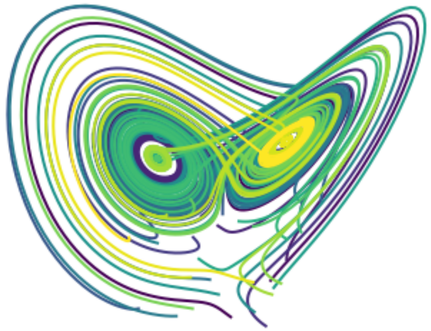
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

In [2]:
```
from lorenz import solve_lorenz
w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
```

sigma 10.00

beta 2.67

rho 28.00



lorenz.py

```
6  def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
7      """Plot a solution to the Lorenz differential equations."""
8
9      max_time = 4.0
10     N = 30
11
12     fig = plt.figure()
13     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
14     ax.axis('off')
15
16     # prepare the axes limits
17     ax.set_xlim((-25, 25))
18     ax.set_ylim((-35, 35))
19     ax.set_zlim((5, 55))
20
21     def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
22         """Compute the time-derivative of a Lorenz system."""
23         x, y, z = x_y_z
24         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
25
26     # Choose random starting points, uniformly distributed from -15 to 15
27     np.random.seed(1)
28     x0 = -15 + 30 * np.random.random((N, 3))
29
30     # Solve for the trajectories
31     t = np.linspace(0, max_time, int(250*max_time))
32     x_t = np.asarray([integrate.odeint(lorenz_deriv, x0i, t)
33                       for x0i in x0])
34
35     # choose a different color for each trajectory
36     colors = plt.cm.viridis(np.linspace(0, 1, N))
37
38     for i in range(N):
39         x, y, z = x_t[i,:,:].T
40         lines = ax.plot(x, y, z, '-', c=colors[i])
41         plt.setp(lines, linewidth=2)
42     angle = 104
43     ax.view_init(30, angle)
```

[JupyterLab]

File Edit View Run Kernel Tabs Settings Help

Lorenz.ipynb

Code

Python 3

The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

[1]:
```
%matplotlib inline
from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

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

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

[2]:
```
from lorenz import solve_lorenz
w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
```

sigma 10.00

beta 2.67

0 1 Python 3 | Idle Saving completed Mode: Command Ln 1, Col 1 Lorenz.ipynb

Two Kinds of Notebooks

- Different notebooks for different stages of work
  - Exploratory notebooks
  - Explanatory notebooks
- Parallels a general approach to research
  - Brainstorm questions and explore answers
  - Distill initial results into something understandable and verifiable
  - Publish polished text and (interactive) visualizations

Support for Rapid Exploration

- Flexible environment
  - Edit any cell whenever you want
  - Execute whichever cells you want
- Inline views of outputs
  - No context switch
  - Easily compare and trace outputs
- Explore data in situ
  - Notebooks run in browser
  - Kernels can run remotely

Support for Clear Explanation

- Textual explanation: markdown cells
- Graphical explanation: inline figures
- Interactive explanation: widgets
- Publishing: Web pages, LaTeX, etc.
- Structure: clear, linear cell layout
- Reproducible

Exploratory Notebooks

```
[291]: import pandas as pd
       from dbanalysis import stop_tools
       data = stop_tools.random_stop_data()
```

```
[292]: data['weekend']=data['day']>4
       data['peak1']=(data['hour'] >= 7) & (data['hour'] <= 10)
       data['peak2']=(data['hour'] >= 16) & (data['hour'] <= 19)
       data['hour']=data['hour'].astype('category')
       data['day']=data['day'].astype('category')
       data['month']=data['month'].astype('category')
       data=data[data['traveltime']>0]
```

```
[293]: data=pd.get_dummies(data,columns=['hour','day','month'])
```

```
[294]: data.columns
```

```
[294]: Index(['index', 'dayofservice', 'tripid', 'plannedtime_arr_from',
              'plannedtime_dep_from', 'actualtime_arr_from', 'actualtime_dep_from',
              'plannedtime_arr_to', 'actualtime_arr_to', 'routeid', 'fromstop',
              'tostop', 'traveltime', 'dwelltime', 'distance', 'speed', 'dt', 'date',
              'year', 'dewpt', 'msl', 'rain', 'rhum', 'temp', 'vappr', 'wetb',
              'weekend', 'peak1', 'peak2', 'hour_5', 'hour_6', 'hour_7', 'hour_8',
              'hour_9', 'hour_10', 'hour_11', 'hour_12', 'hour_13', 'hour_14',
              'hour_15', 'hour_16', 'hour_17', 'hour_18', 'hour_19', 'hour_20',
              'hour_21', 'hour_22', 'hour_23', 'day_0', 'day_1', 'day_2', 'day_3',
              'day_4', 'day_5', 'day_6', 'month_1', 'month_2', 'month_3', 'month_4',
              'month_5', 'month_6'],
             dtype='object')
```

```
[295]: data['weekend']=data['day']>4
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~/anaconda3/lib/python3.6/site-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3062             try:
-> 3063                 return self._engine.get_loc(key)
   3064             except KeyError:
```

Explanatory Notebooks

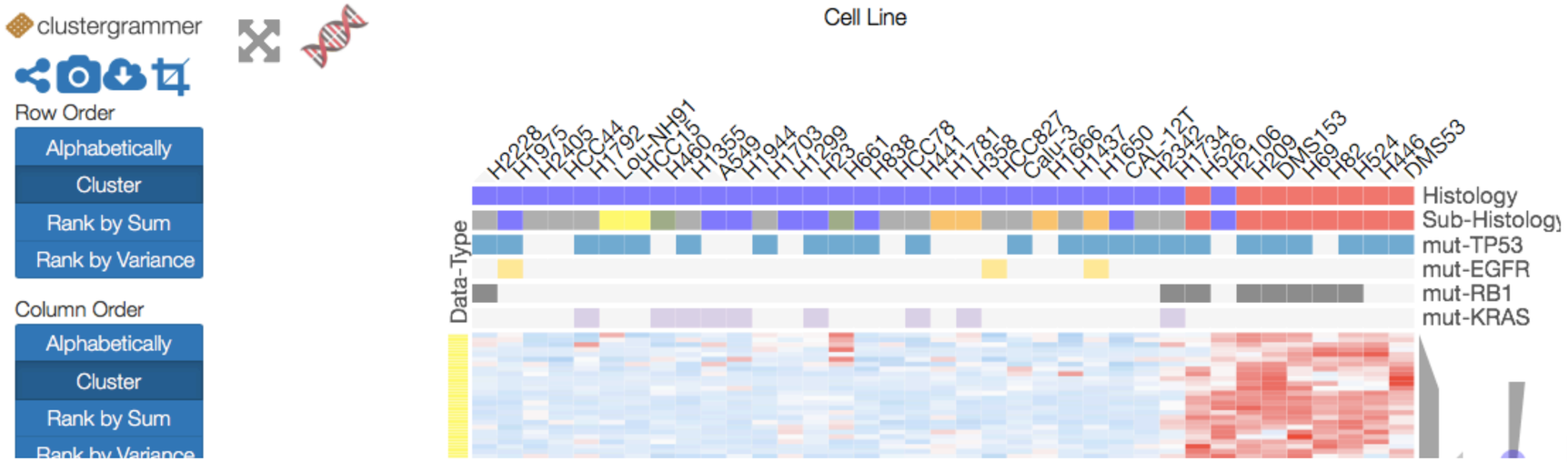
Gene Expression Data

We obtained gene expression data from the Cancer Cell Line Encyclopedia (CCLE) for 37 lung cancer cell lines assayed by our collaborators at CST. This independent dataset can be used to find novel correlations between differentially expressed genes and PTMs as well as determine whether lung cancer cell lines behave similarly in gene-expression-space and PTM-space. The gene expression data was processed in the CST_Data_Processing.ipynb notebook that: kept the top 1000 genes with the greatest variance across the cell lines, and Z-score normalized the genes across the cell lines to highlight differential expression across the lung cancer cell lines.

In [4]:
```
net.load_file('../lung_cellline_3_1_16/lung_cl_all_ptm/precalc_processed/CST_CCLE_exp.txt')
print('Expression data shape: ' + str(net.dat['mat'].shape))
```

```
Expression data shape: (1000, 37)
```

In [5]:
```
net.set_cat_color('row', 1, 'Data-Type: Exp', 'yellow')
net.cluster(views=[])
net.widget()
```



[N. Fernandez et al.]

These nice explanations often take a lot of work!

Reproducibility of Notebooks

- Environment: have installed correct versions of libraries/packages
- Data: have access to all of the necessary data
- **Execution: able to rerun the cells as intended**

Exploration Potholes

- If you redefine or **mutate** a variable, another cell may break
- The **order** of cell execution matters
- If you forget to run an edited cell, the result may **not match** the code

Dataflow Notebooks

```
In [a0a358]: raw_df = pd.read_csv("fifa17-top20-women.txt",sep="-",header=None)
```

raw_df:

	0	1	2
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aaa3c6]: column_names = {0: "Name", 1: "Country", 2: "Rating"}
```

```
column_names: {0: 'Name', 1: 'Country', 2: 'Rating'}
```

```
In [a249ea]: named_df = raw_df.rename(columns=column_names)
```

named_df:

	Name	Country	Rating
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aab079]: named_df.groupby("Country").size().sort_values(ascending=False)
```

```
Out[aab079]: Country
             USA          6
             Canada       3
             Brazil       3
             ...        ...
```

[dataflownb.github.io]

Dataflow Notebooks

- Persistent Identifiers

```
In [a0a358]: raw_df = pd.read_csv("fifa17-top20-women.txt",sep="-",header=None)
```

raw_df:

	0	1	2
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aaa3c6]: column_names = {0: "Name", 1: "Country", 2: "Rating"}
```

```
column_names: {0: 'Name', 1: 'Country', 2: 'Rating'}
```

```
In [a249ea]: named_df = raw_df.rename(columns=column_names)
```

named_df:

	Name	Country	Rating
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aab079]: named_df.groupby("Country").size().sort_values(ascending=False)
```

```
Out[aab079]: Country
             USA          6
             Canada       3
             Brazil       3
             ...        ...
```

[dataflownb.github.io]

Dataflow Notebooks

- Persistent Identifiers
- Named Outputs

```
In [a0a358]: raw_df = pd.read_csv("fifa17-top20-women.txt",sep="-",header=None)
```

raw_df:

	0	1	2
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aaa3c6]: column_names = {0: "Name", 1: "Country", 2: "Rating"}
```

column_names:

```
{0: 'Name', 1: 'Country', 2: 'Rating'}
```

```
In [a249ea]: named_df = raw_df.rename(columns=column_names)
```

named_df:

	Name	Country	Rating
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aab079]: named_df.groupby("Country").size().sort_values(ascending=False)
Out[aab079]: Country
             USA          6
             Canada       3
             Brazil       3
             ...        ...
```

[dataflownb.github.io]

Dataflow Notebooks

- Persistent Identifiers
- Named Outputs
- Unnamed Outputs

```
In [a0a358]: raw_df = pd.read_csv("fifa17-top20-women.txt",sep="-",header=None)
```

raw_df:

	0	1	2
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aaa3c6]: column_names = {0: "Name", 1: "Country", 2: "Rating"}
```

```
column_names: {0: 'Name', 1: 'Country', 2: 'Rating'}
```

```
In [a249ea]: named_df = raw_df.rename(columns=column_names)
```

named_df:

	Name	Country	Rating
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aab079]: named_df.groupby("Country").size().sort_values(ascending=False)
```

```
Out[aab079]: Country
USA          6
Canada       3
Brazil       3
...        ...
```

[dataflownb.github.io]

Dataflow Notebooks

- Persistent Identifiers
- Named Outputs
- Unnamed Outputs
- Connection by Variable Reference

```
In [a0a358]: raw_df = pd.read_csv("fifa17-top20-women.txt",sep="-",header=None)
```

raw_df:

	0	1	2
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aaa3c6]: column_names = {0: "Name", 1: "Country", 2: "Rating"}
column_names: {0: 'Name', 1: 'Country', 2: 'Rating'}
```

```
In [a249ea]: named_df = raw_df.rename(columns=column_names)
```

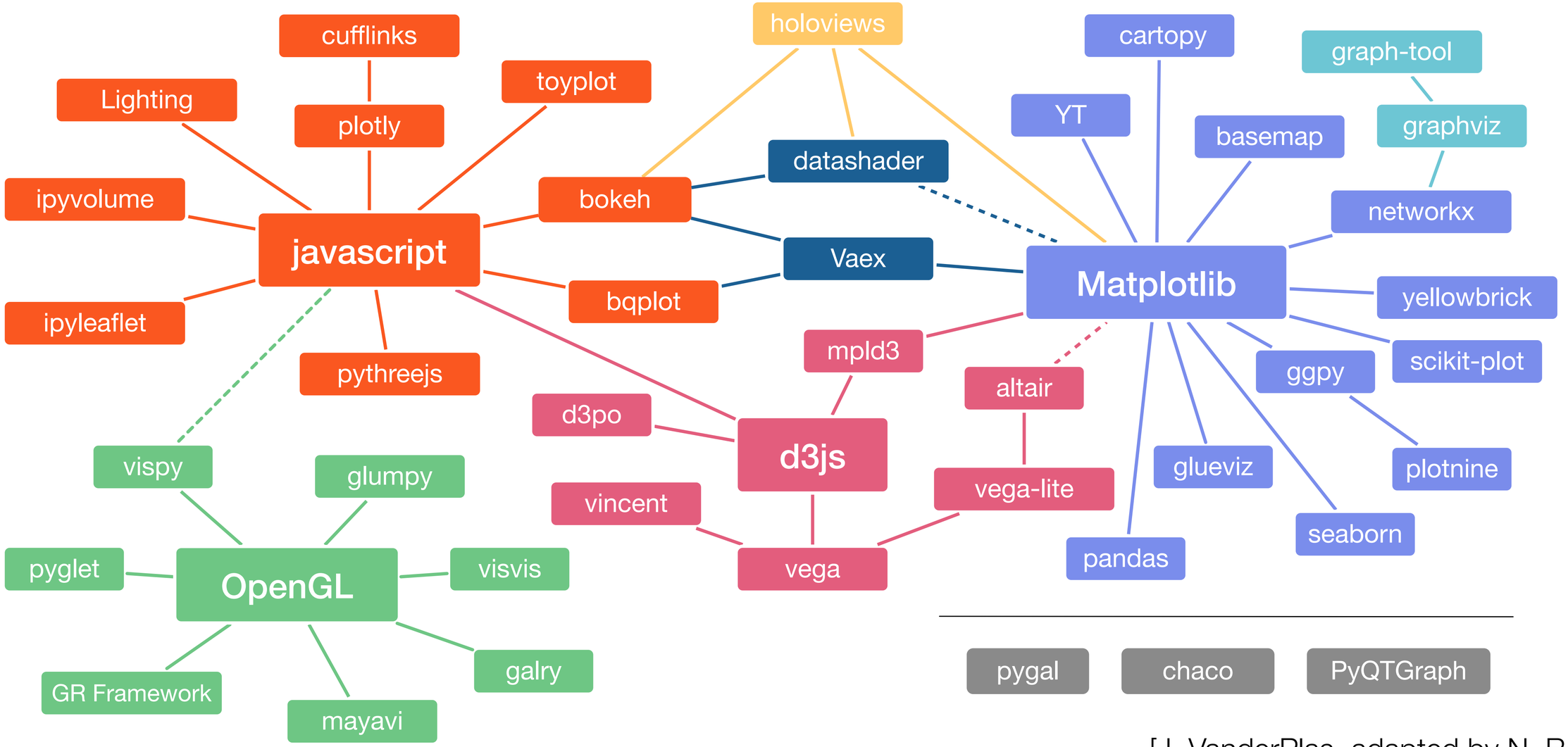
named_df:

	Name	Country	Rating
0	Caroline Seger	Sweden	85
1	Wendie Renard	France	85
2	Steph Houghton	England	85
...	...	...	...

```
In [aab079]: named_df.groupby("Country").size().sort_values(ascending=False)
Out[aab079]: Country
             USA          6
             Canada       3
             Brazil       3
             ...        ...
```

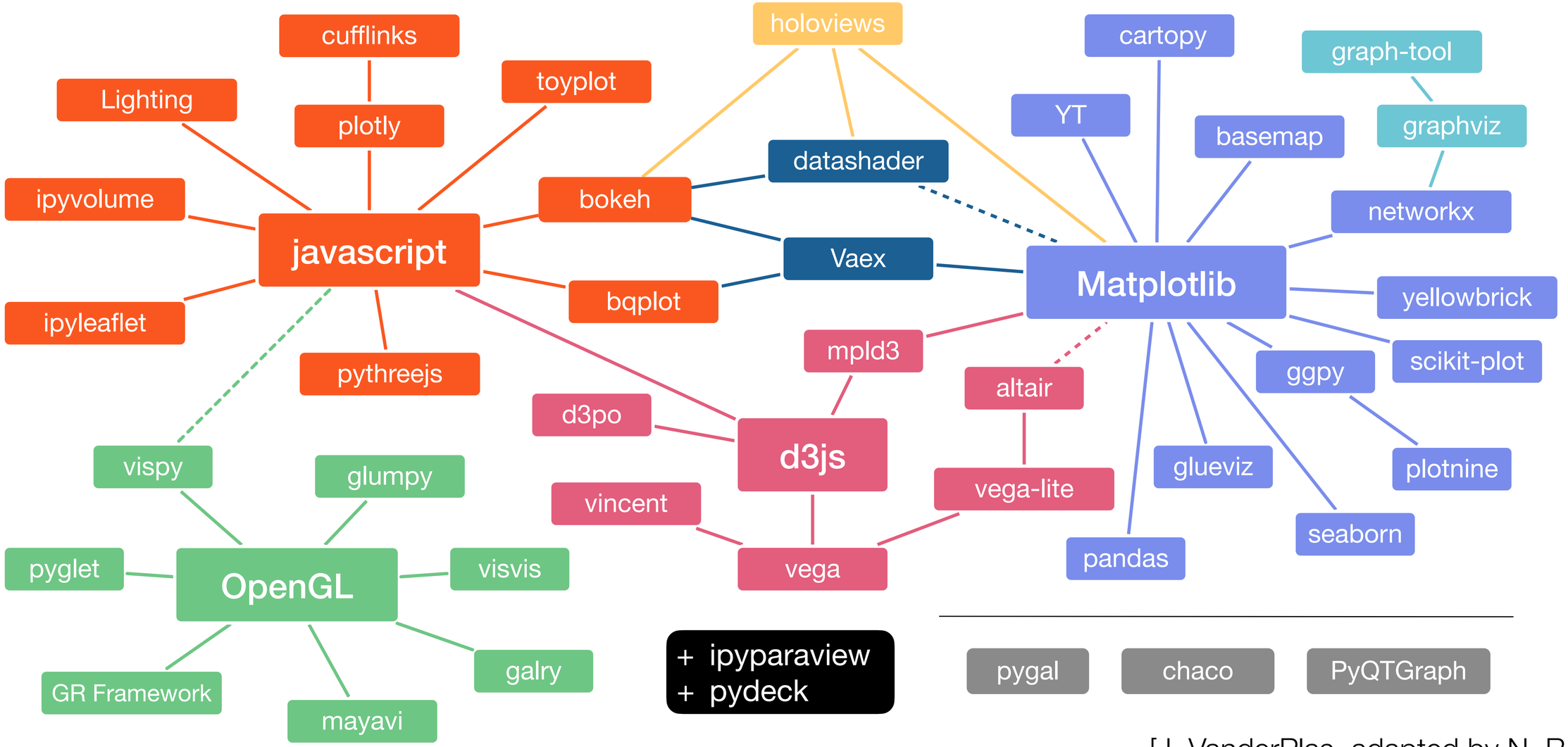
[dataflownb.github.io]

The Python Visualization Landscape



[J. VanderPlas, adapted by N. Rougier]

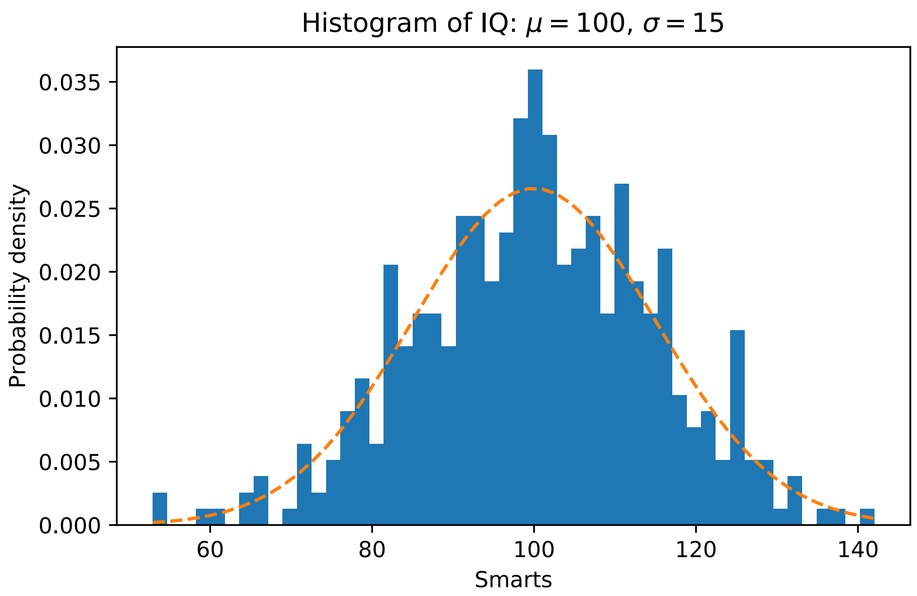
The Python Visualization Landscape



[J. VanderPlas, adapted by N. Rougier]

matplotlib

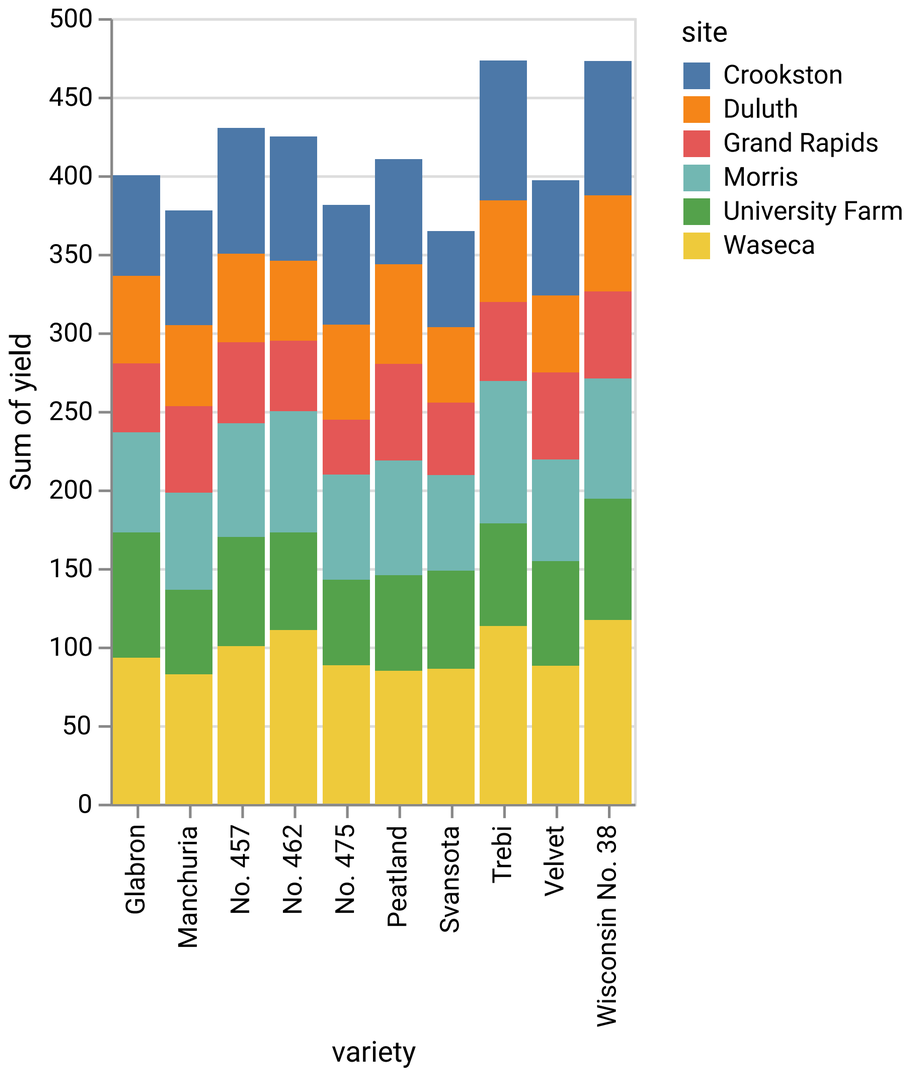
- Strengths:
  - Designed like Matlab
  - Many rendering backends
  - Can reproduce almost any plot
  - Proven, well-tested
- Weaknesses:
  - API is imperative
  - Not originally designed for the web
  - Dated styles



[J. VanderPlas]

Altair

- Declarative Visualization
  - Specify **what** instead of how
  - Separate specification from execution
- Based on VegaLite which is browser-based



Examples

- Examine airfoil data on Cooley
- Login to:
  - jupyter.alcf.anl.gov
- Click on "Login Cooley"
- Use upload button to upload the two airfoil notebooks
  - airfoil-flow.ipynb
  - airfoil-line-plots.ipynb
- Once uploaded, click on the flow notebook to start