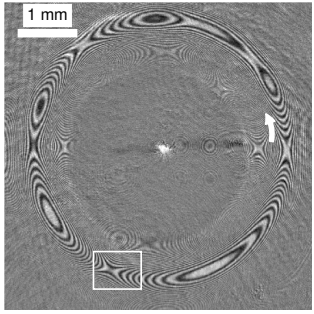
Matplotlib and Scientific Visualization

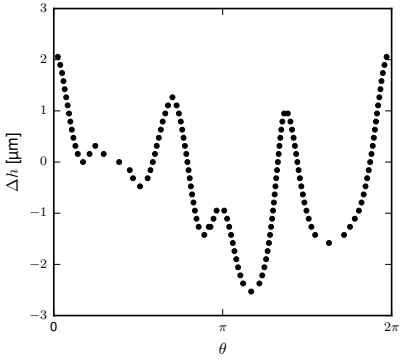
Thomas A Caswell

2021-03-17

Who am I?

- Trained as a physicist
  - Jamming + dynamics of Leidenfrost drops with Nagel and Gardel at UChicago





Who am I?

- Trained as a physicist
  - Jamming + dynamics of Leidenfrost drops with Nagel and Gardel at UChicago
- Currently in Data Science and System Integration program at NSLS-II
- Current Project Lead of Matplotlib

Acknowledgments

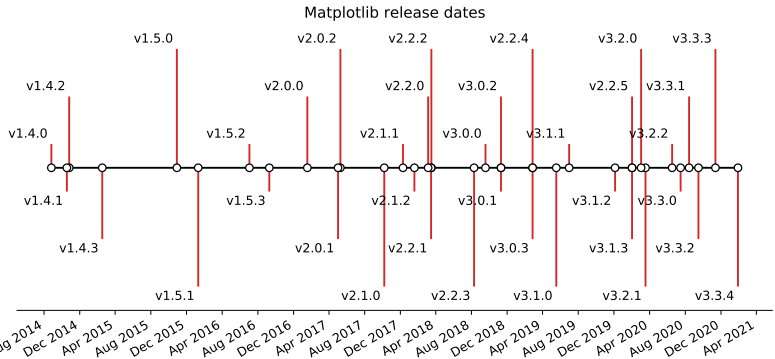
- John Hunter (1968-2012)
- Michael Droettboom
- The whole Matplotlib development team
  - Over 1,250+ have contributed code, many more in bug reports, feature requests, and user support
- Dora Caswell

Recent funding from Chan Zuckerberg Initiative (2020-present)

Matplotlib

. . . is a comprehensive library for creating static, animated, and interactive visualizations in Python.

- Widely used through out science
  - over 10% of arXiv has at least one Matplotlib figure (as of 2018)
  - estimated over 1M users
- Continuously developed for past 19 years
  - first commit in 2003, initial work in 2001-2002

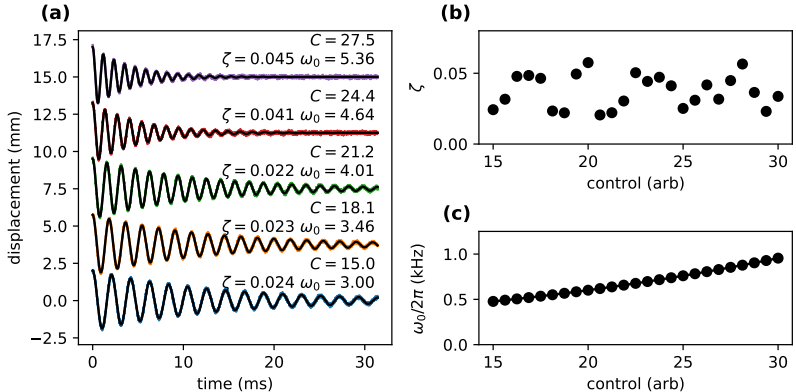


What is visualization for?

1. Exploratory data analysis
   - just get the data on the screen in a way *you* can understand as fast as you can
   - `matplotlib.pyplot`
   - `seaborn`
   - plotting methods on data structures (e.g. `obj.plot(...)`)
2. Paper figures
   - need to be *just right*
3. Part of a standard (interactive) workflow
   - repeatedly visualize data with same data-structure

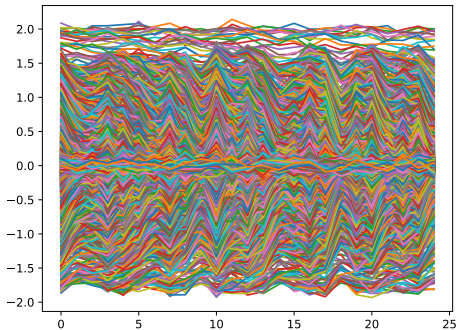
Case Study: Paper Figure

- Assume:
  - you have fabricated 25 cantilevers
  - Varied something (called 'control') in fabrication
- Experiment:
  - displace away from equilibrium position by some amount
  - release at t=0 and watch vibrations ring down



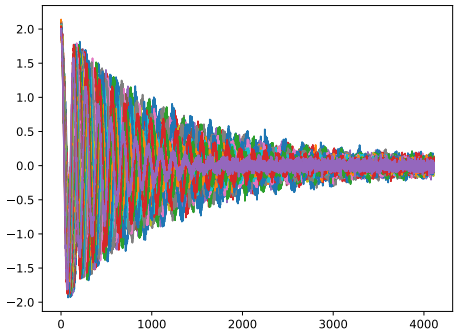
Step 0.0

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d)
```



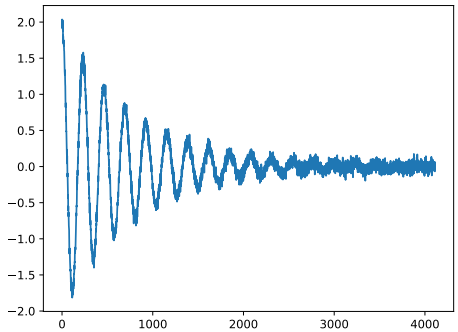
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



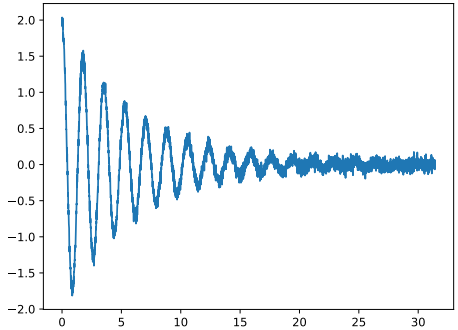
Step 1.0

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  m = d[6]
6  plt.plot(m)
```



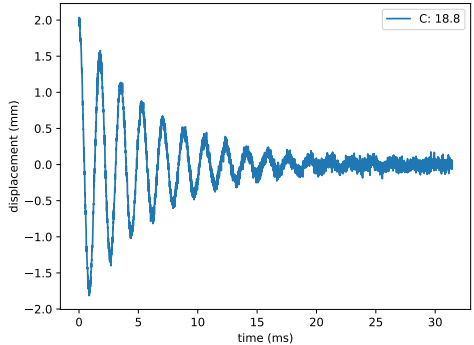
Step 1.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  m = d[6]
6
7  fig, ax = plt.subplots()
8  ax.plot(m.time, m)
```



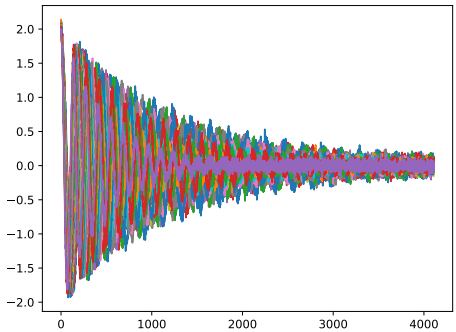
Step 1.2

```
1  ...
2
3  d = get_data()
4  m = d[6]
5
6  fig, ax = plt.subplots()
7
8  ax.plot(m.time, m, label=f"C: {float(m.
9  ax.legend()
10 ax.set_xlabel("time (ms)")
11 ax.set_ylabel("displacement (mm)")
```



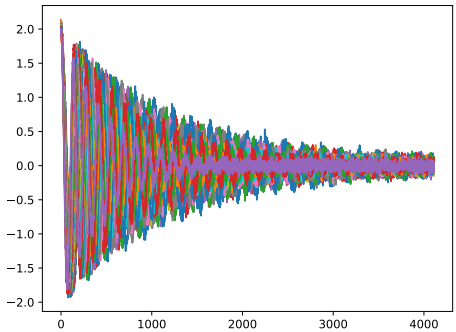
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



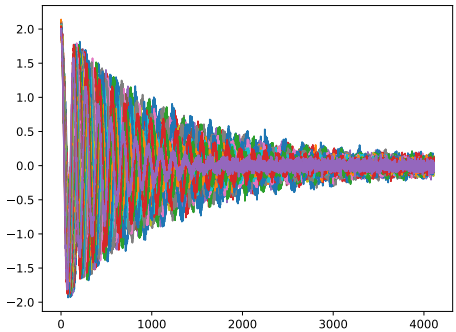
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



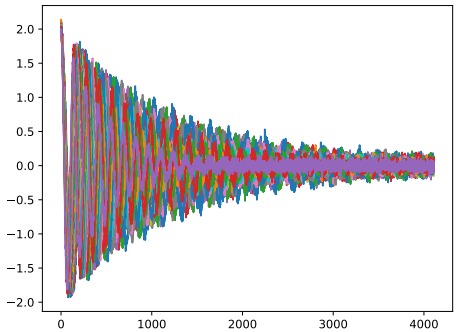
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



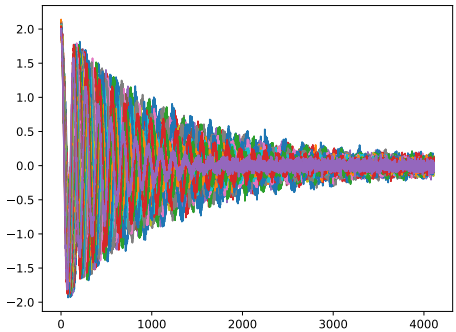
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



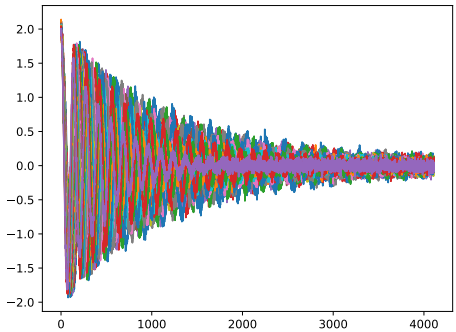
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



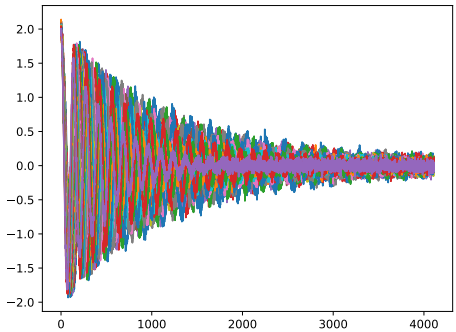
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



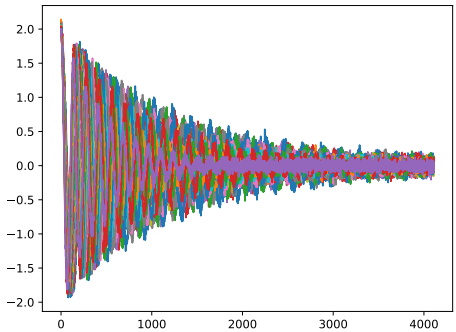
Step 0.1

```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```

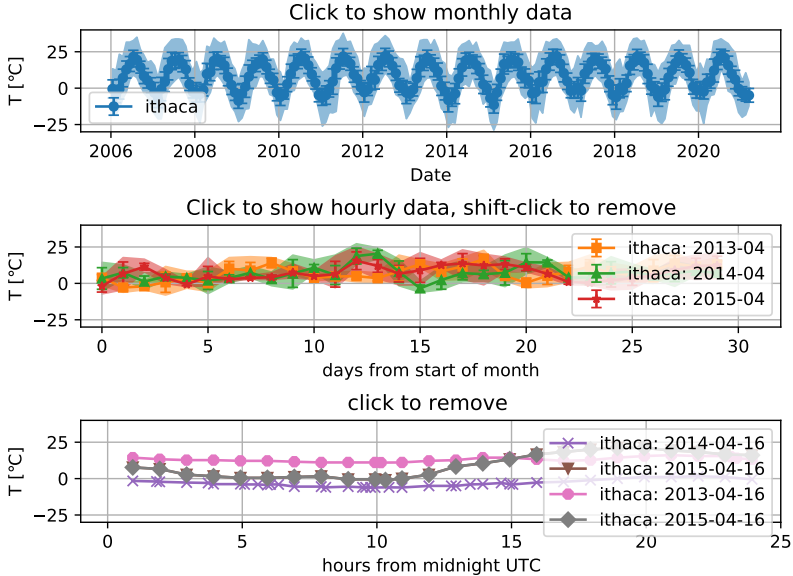


Step 0.1

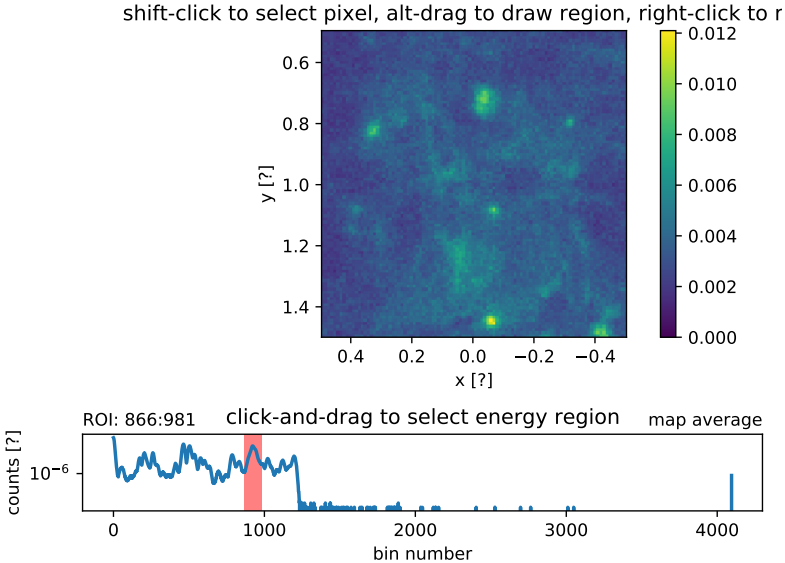
```
1  from gen_data import get_data
2  import matplotlib.pyplot as plt
3
4  d = get_data()
5  plt.plot(d.T)
```



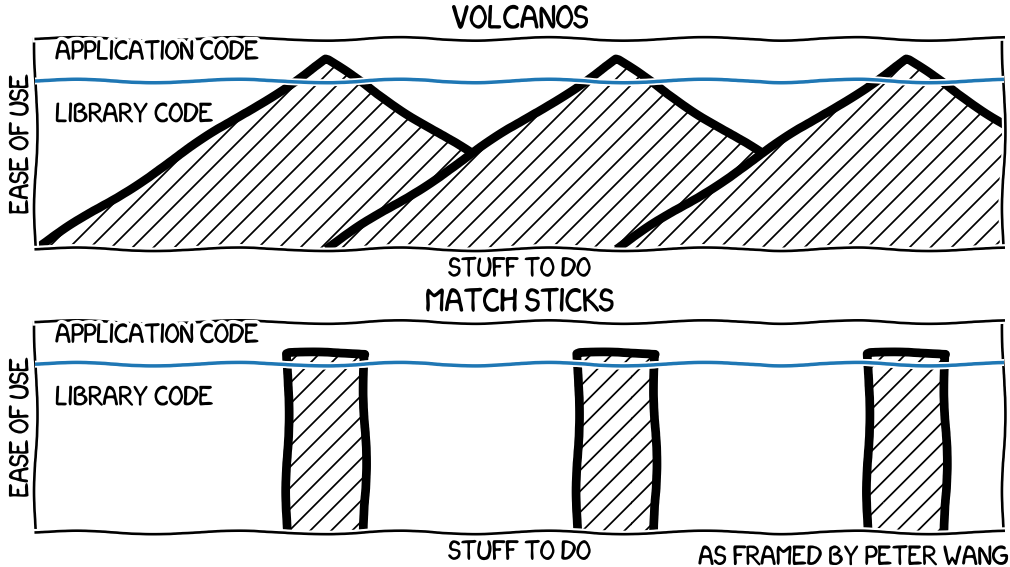
Interactive application (temperature)



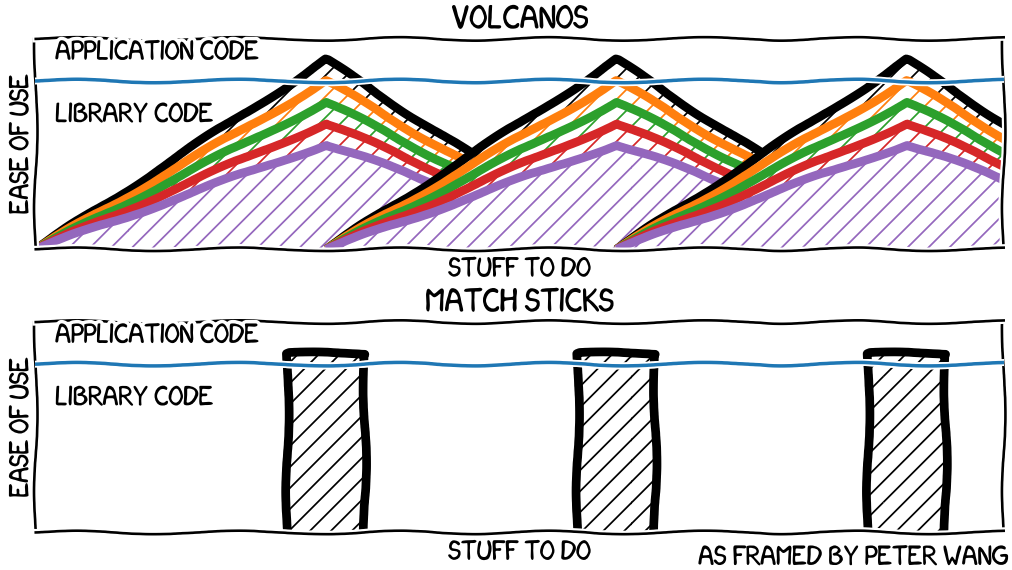
Interactive applications (x-ray fluorescence map)



Iterative software development



Iterative software development



Future Work

- On going incremental improvements, bug fixes, and maintenance
- Improvements to Figure and Axes layout tooling (Jody Klymak)
- Re-designing Matplotlib's internal data model (Hannah Aizenman)

Resources

This material: https://github.com/tacaswell/2021-03_APS

- docs: https://matplotlib.org/stable
- cheatsheets: https://github.com/matplotlib/cheatsheets
- chat: https://gitter.im/matplotlib
- forum: https://discourse.matplotlib.org
- tutorials: https://github.com/matplotlib/interactive_tutorial, https://github.com/matplotlib/AnatomyOfMatplotlib https://github.com/matplotlib/GettingStarted
- Interactive Applications Using Matplotlib, Benjamin V. Root (2015)
- domain-specific libraries
- Building a maintainable plotting library (PyData NYC 2019) https://youtu.be/NV4Y75ZUDJA
- Seperation Of Scales (PyData Gobal 2020) https://youtu.be/P85UIuMovnI
- explain flouresence better
- look at ligo notebooks ?!

ligo or ETH