



# INTERACTIVE PLANETARY

---

*lessons in interactive python vis  
amcasari  
pyladies seattle, 23feb2016*

*who:* @amcasari



*what:* @PyLadiesSEA

*where:* @Concur

*why:* #Pythonistas  
#PyLadiesRock

*(now we can be found)*

.....

# COORDINATES



*control systems*

*engineering +*

*robotics + legos*

*officer in USN*

*operations research*

*analyst*

*wandering dirtbag +  
conservation volunteer*

*data science via random walks*

*senior data scientist*

*@ Concur*

*consultant*

*extraordinaire*

*underwater robotics*

*SAHM*

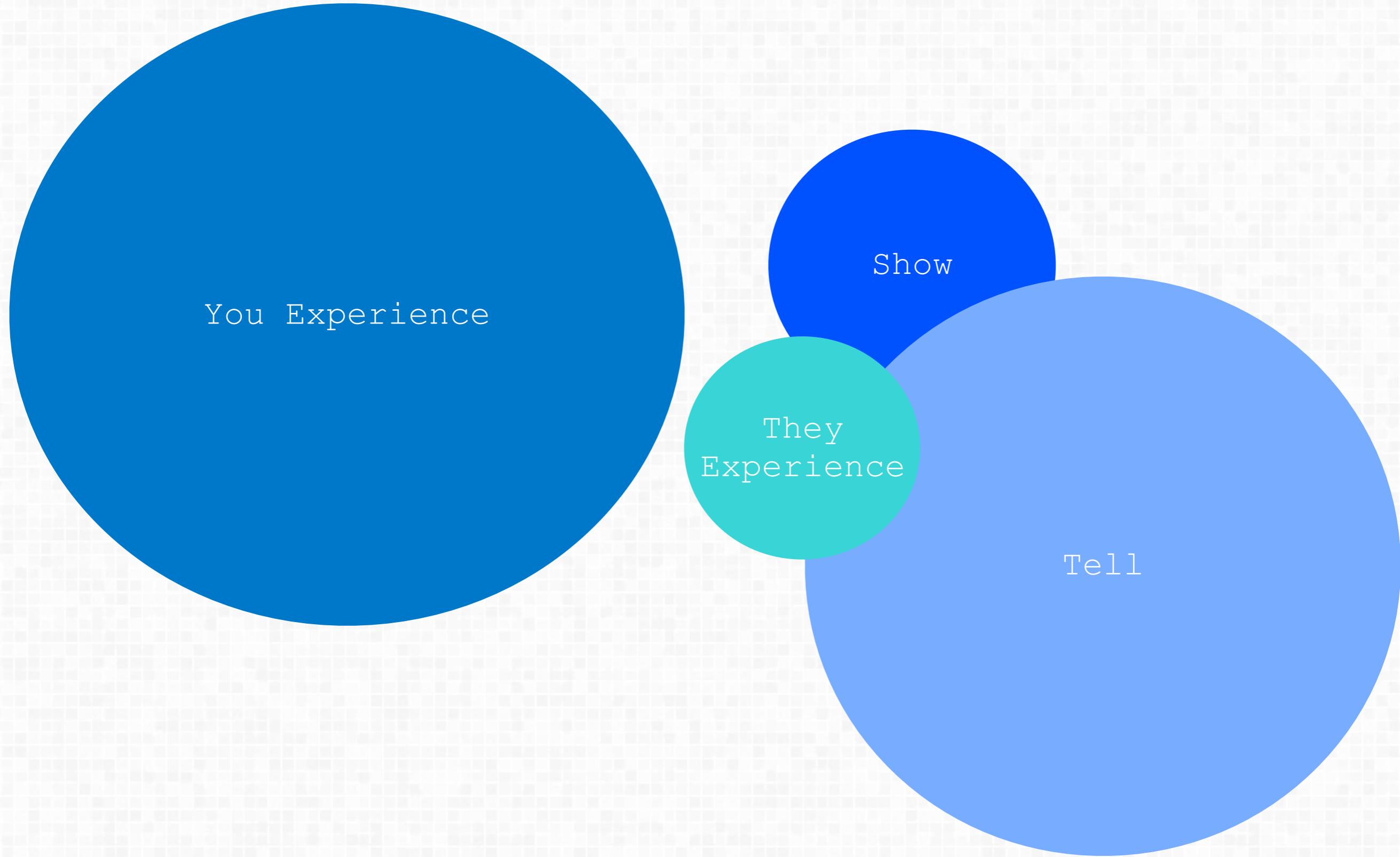
*EE +*

*applied math*

*+ complex systems*

# STORYTELLING LOOP DA LOOP

---



# TRADITIONAL ANALYSIS + STATIC VISUALIZATION: WE MOSTLY TELL

---

$$2 + 2 = 4$$

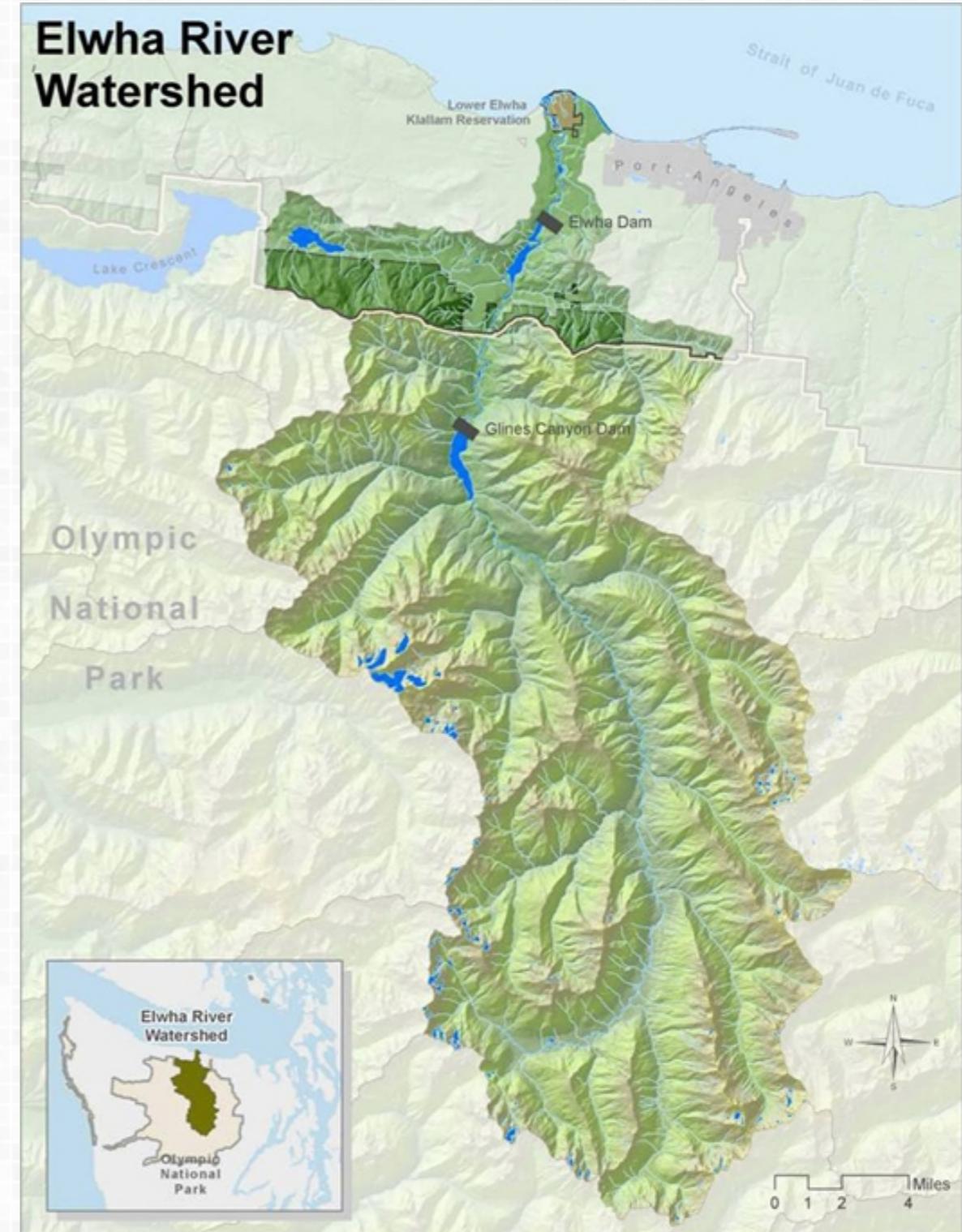
# HOW DO WE SEE THE WORLD: EXPERIENCE

---



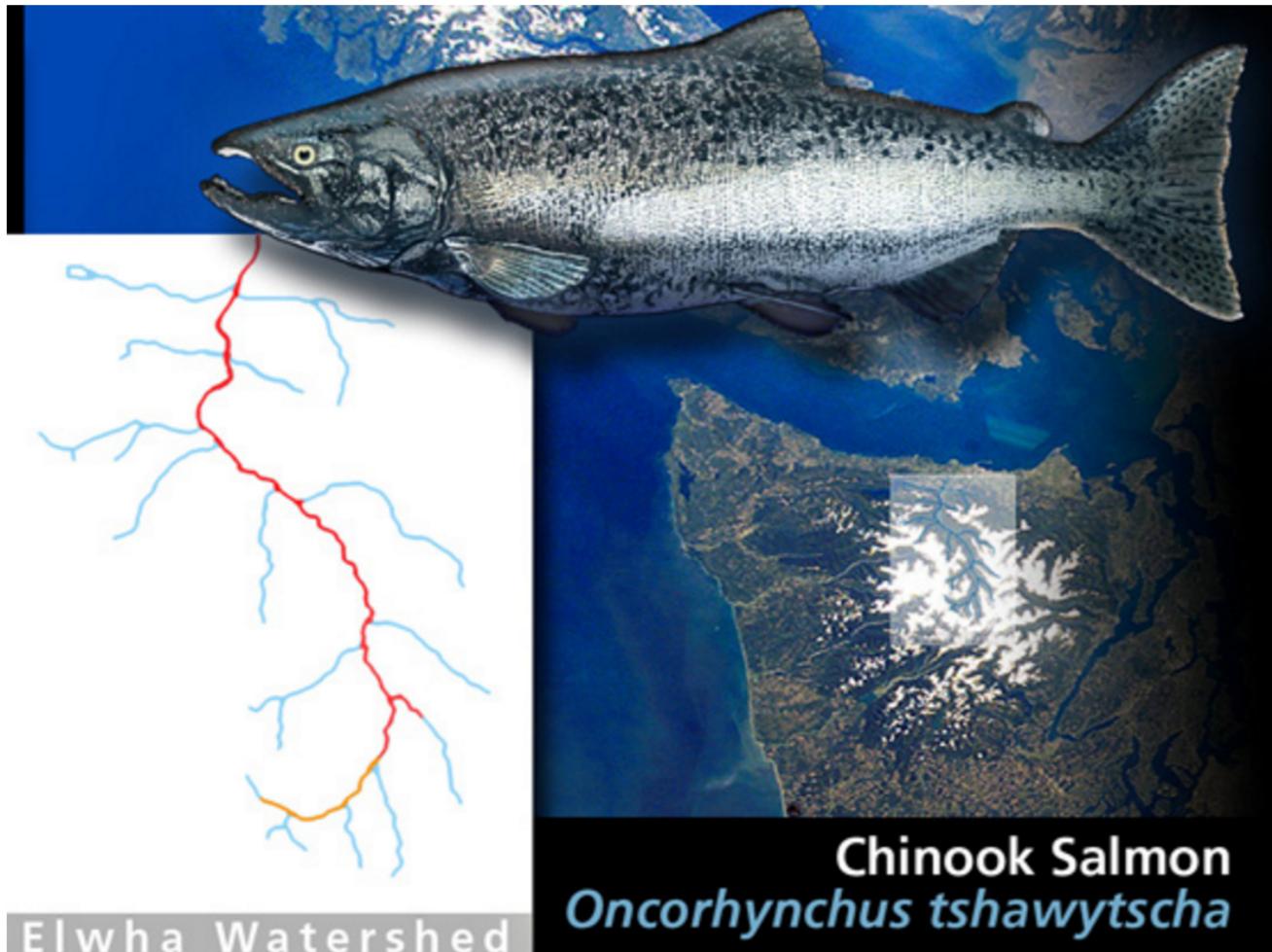
*the elwha river, olympic peninsula, wa*

# HOW DO WE SEE THE WORLD: SHOW



*the elwha river, olympic peninsula, wa*

# HOW DO WE SEE THE (DATA) WORLD: TELL

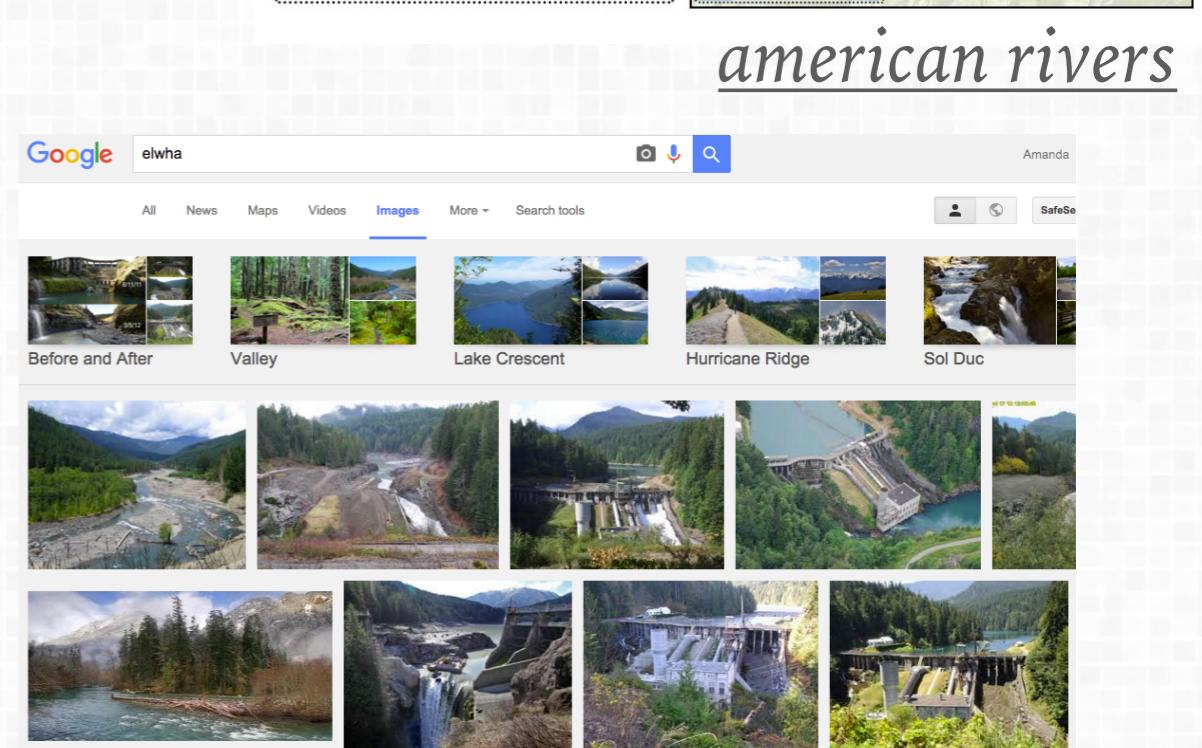
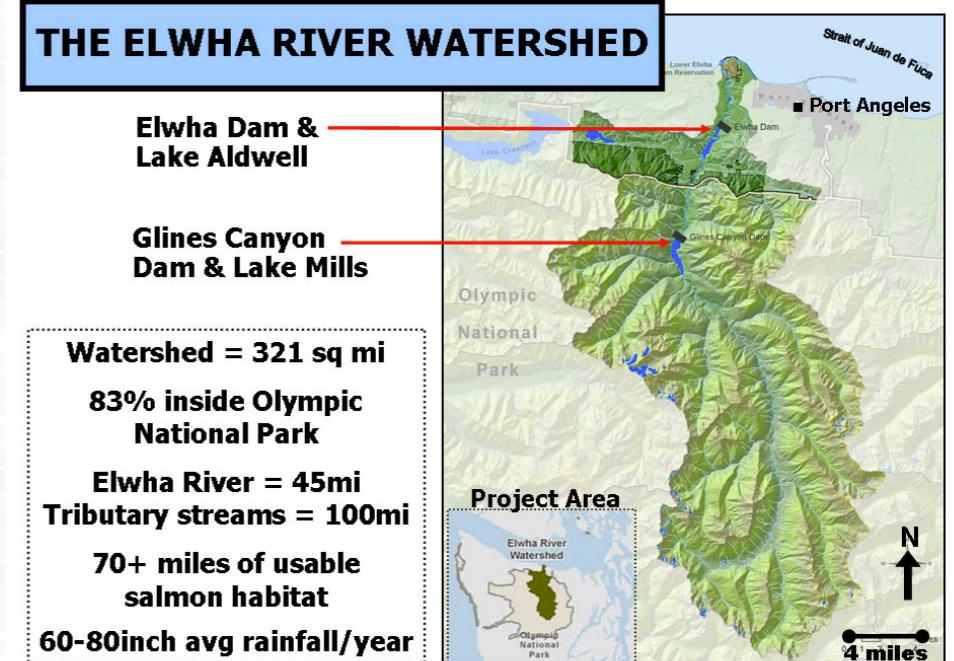


Historic range of chinook in the Elwha River.

nps

then there's this...

seattle times: the elwha river, olympic peninsula, wa



# HOW DO WE REALLY, REALLY SEE THE WORLD?

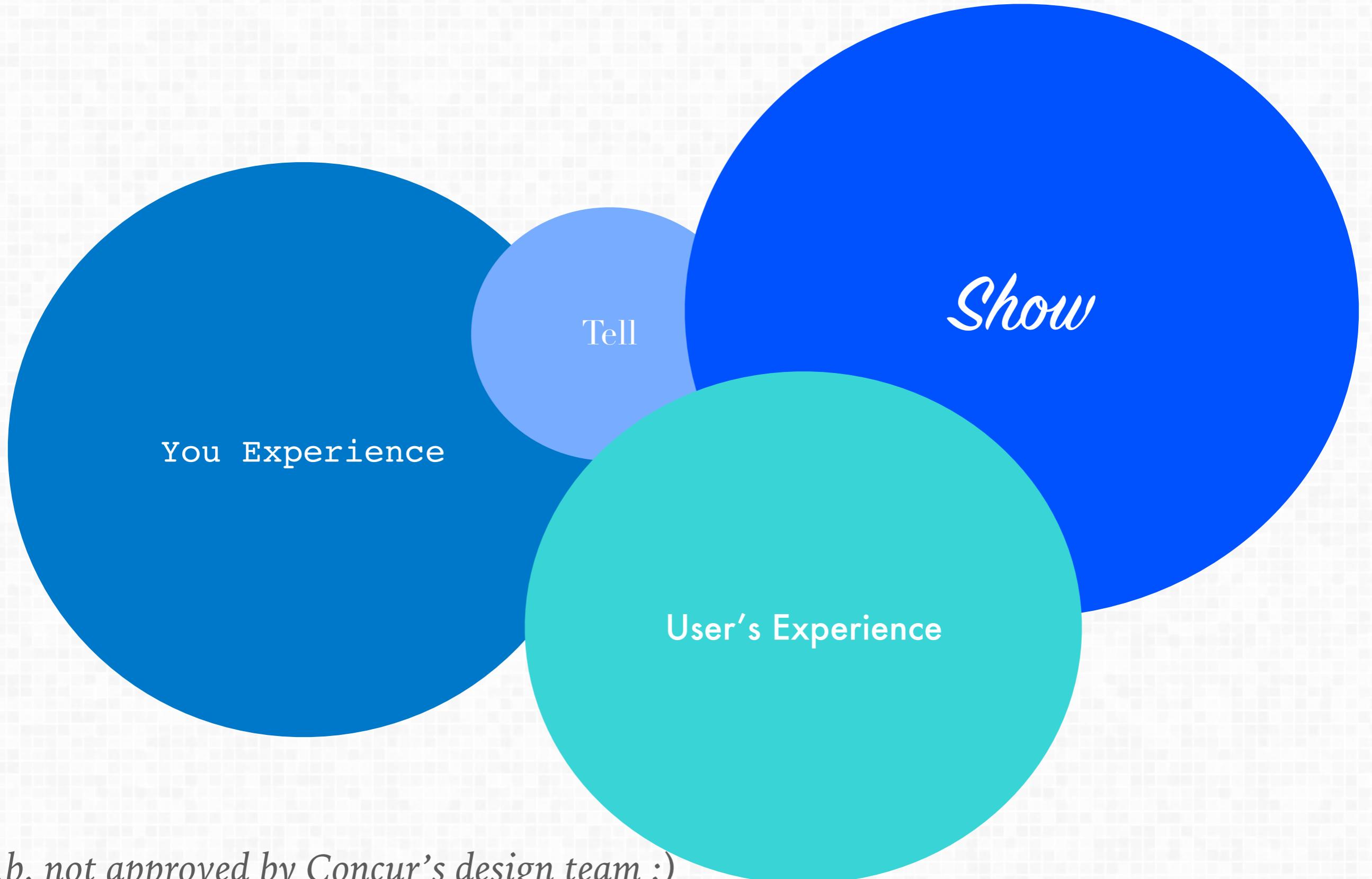
---



*field biologist*

# INTERACTIVE VISUALIZATION!

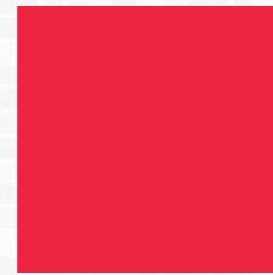
---



*n.b. not approved by Concur's design team :)*

# W/ INTERACTIVE VISUALIZATION: WE SHOW + THEY EXPERIENCE

---



“

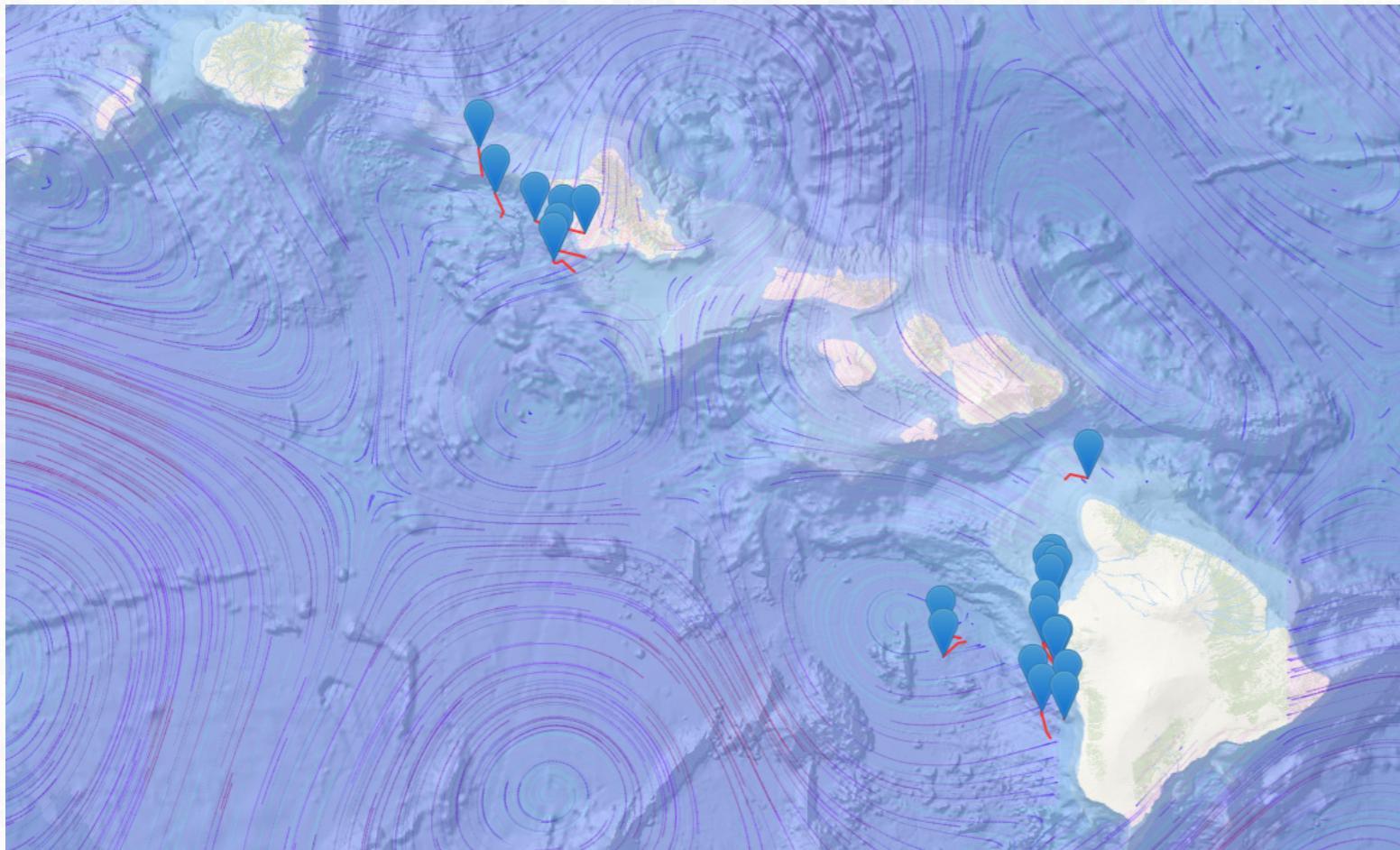
*because i can...ethics. bias.*

*data != numbers.*

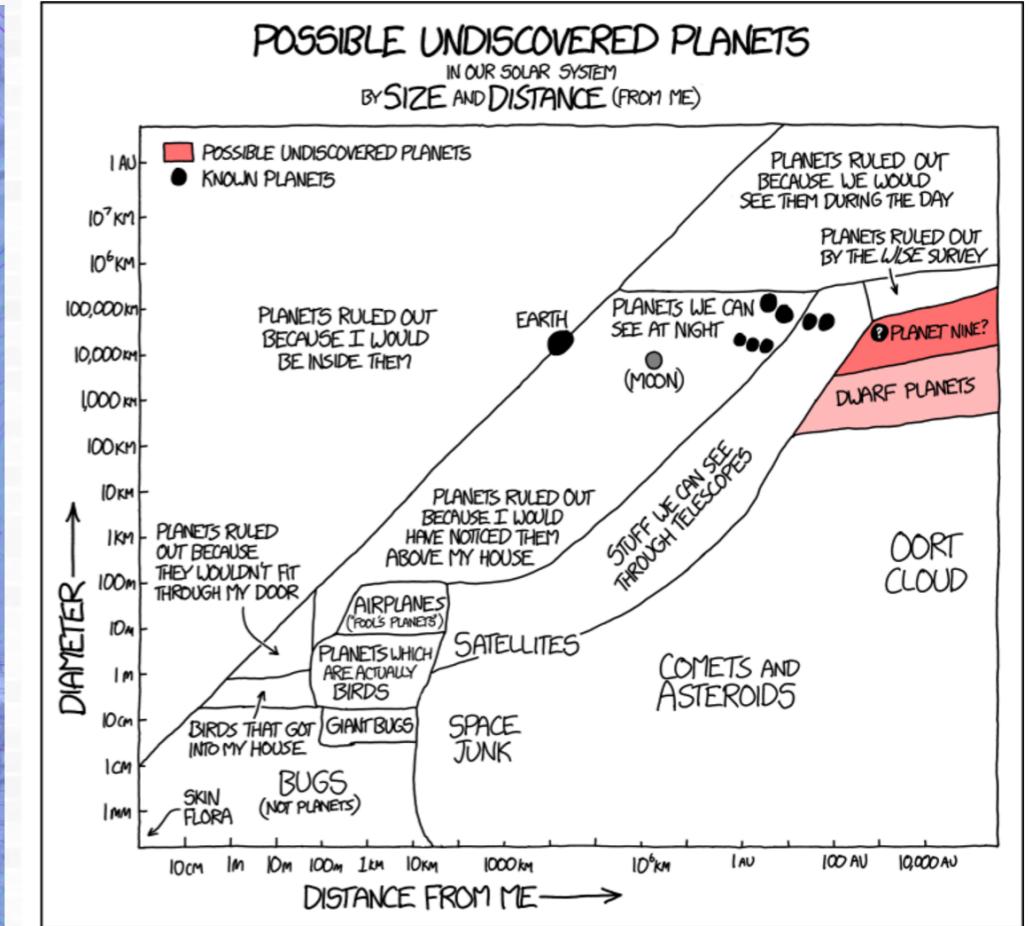
call to think critically about the politics  
+ ethics of visualization - catherine  
d'ignazio

the point of collection - mimi onuoha

# HOW DO WE (THOUGHTFULLY) SHOW OUR (DATA) WORLD?



smartmine whale tracker



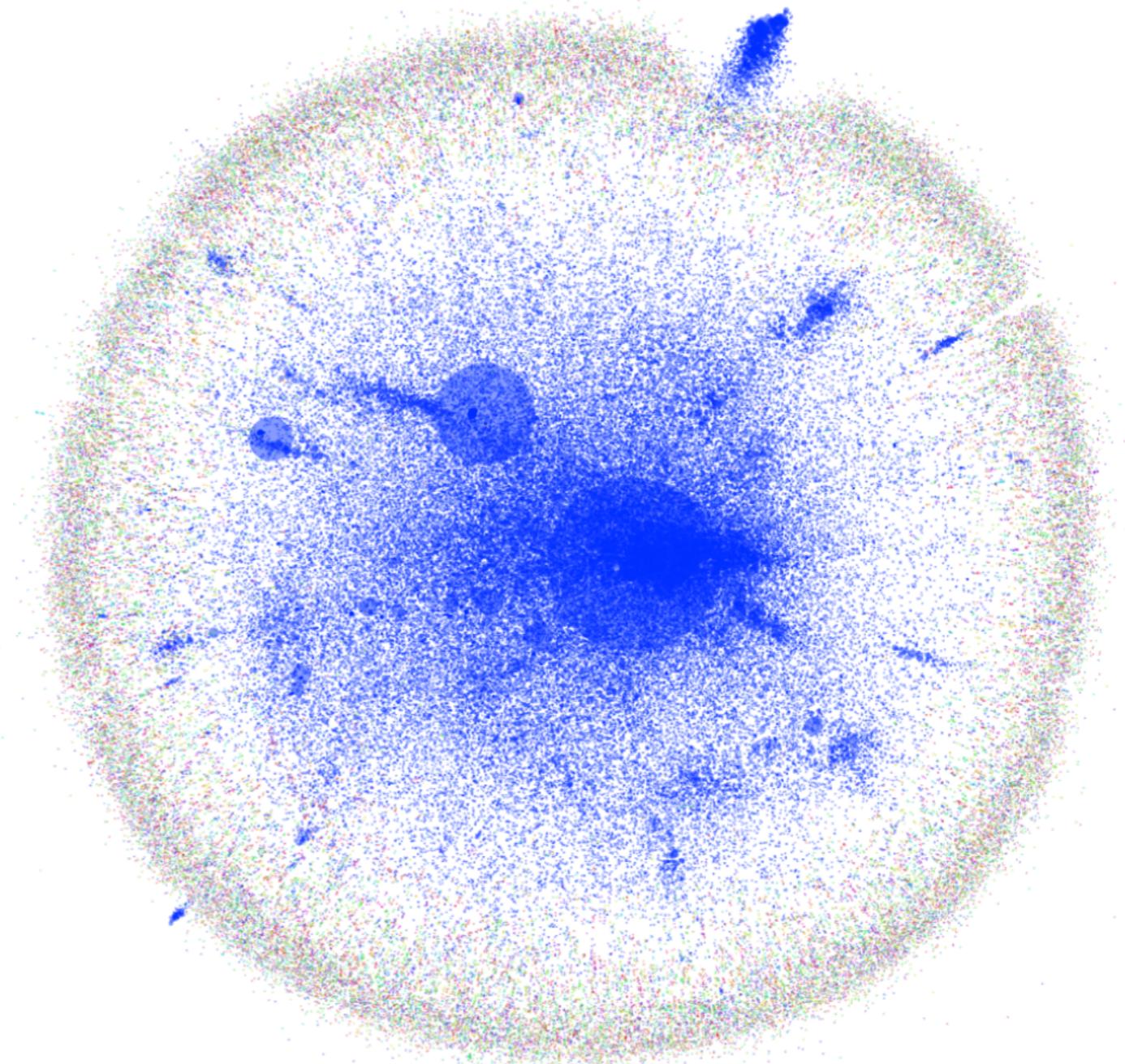
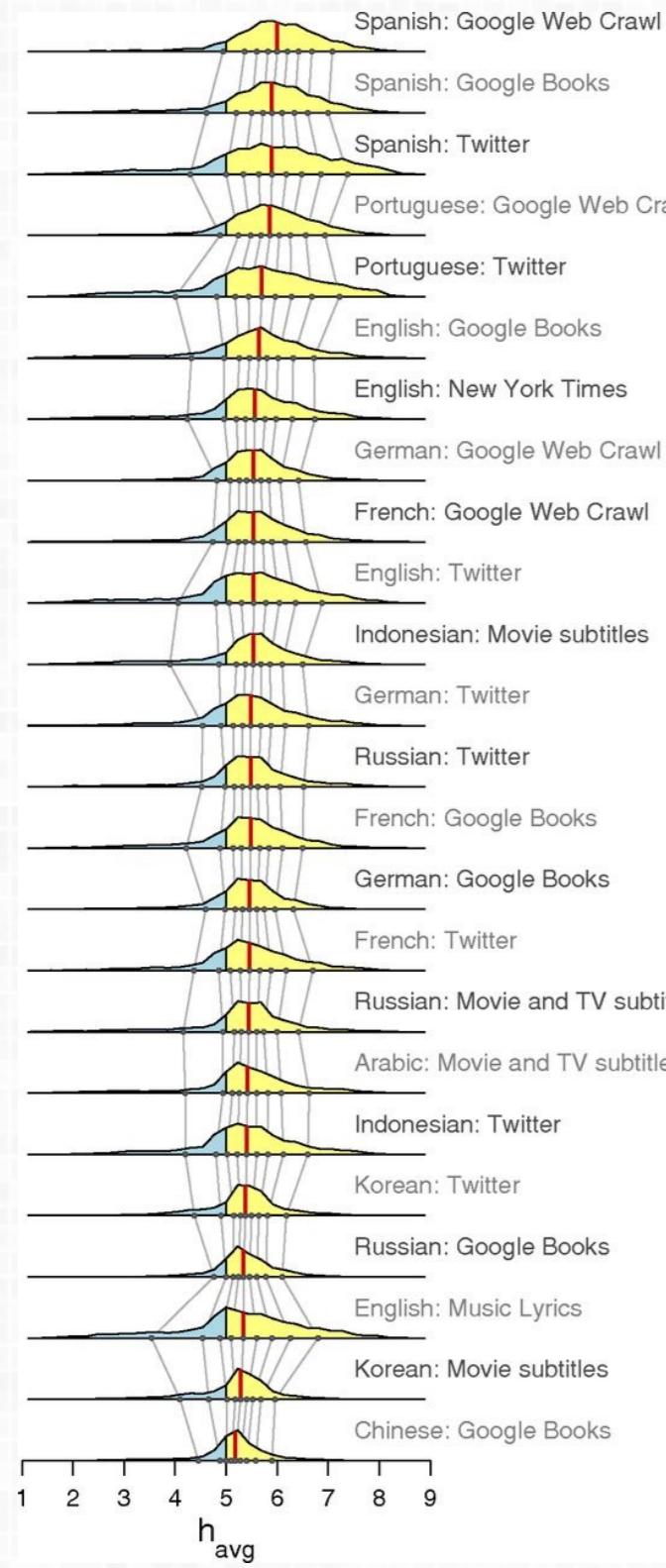
xkcd

nytimes: what does a marriage license cost you

2016 primary results + calendar

periscopic: global gender gap in phds

# HOW DO WE (THOUGHTFULLY) SHOW OUR (DATA) WORLD?



show + tell != show + experience

all figs from papers here

# HOW DO WE SHOW OUR (DATA) WORLD WITH PYTHON?



**prettyplotlib**

ggplot from **ŷhat**

**Leaflet**

Seaborn



**matplotlib**



Paper.js

NetworkX



**ipywidgets**



**Pygal**

**Sexy python charting**

Folium

# HOW DO WE SHOW OUR (DATA) WORLD WITH PYTHON?

---

matplotlib



# HOW DO WE SHOW OUR (DATA) WORLD WITH PYTHON?

```
In [13]: t = arange(0.0, 1.0, 0.01)

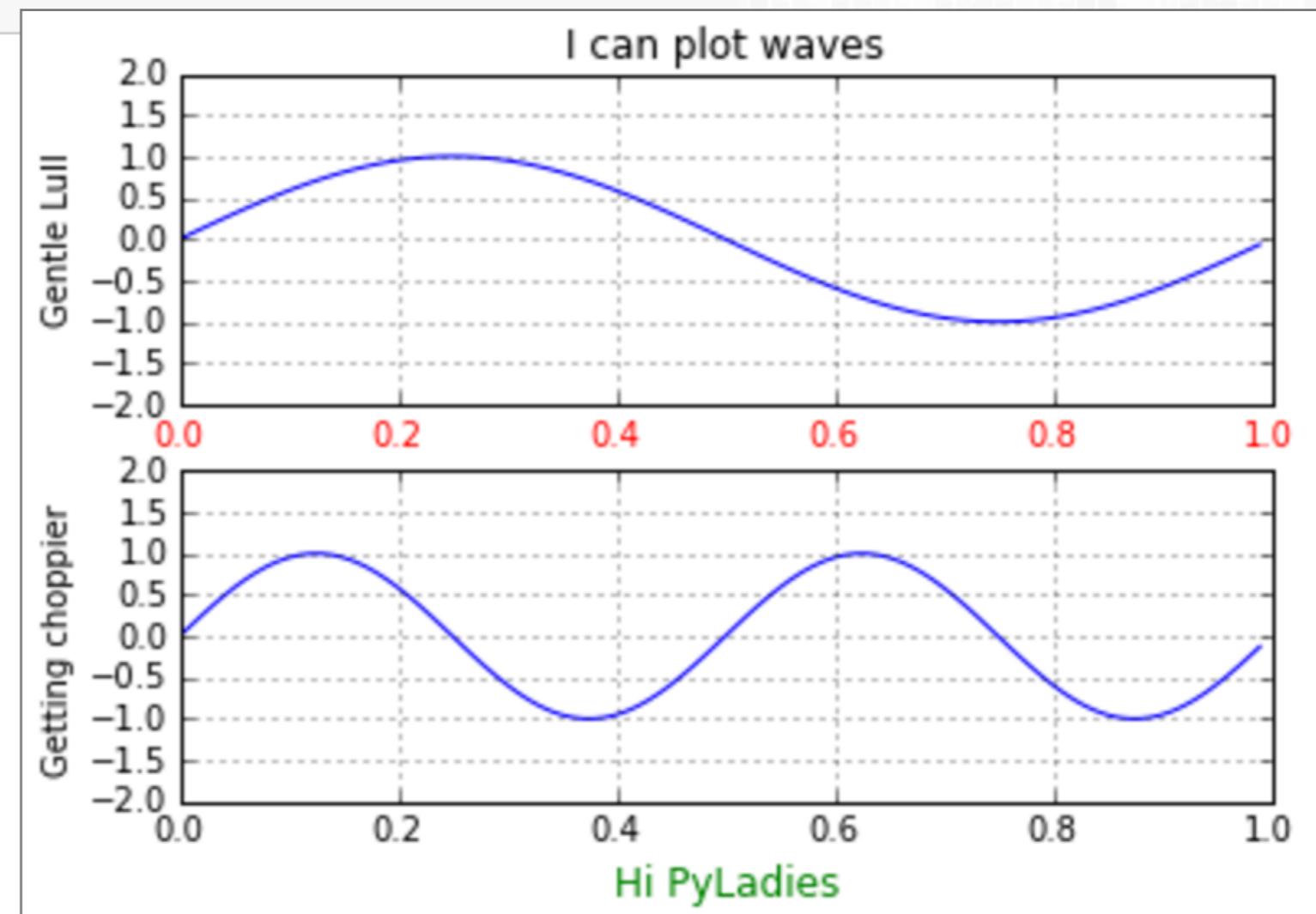
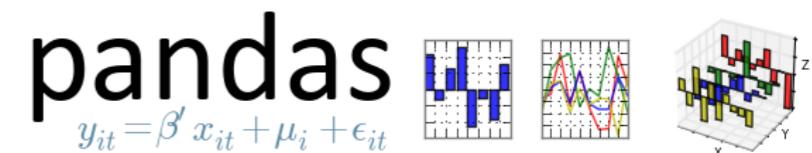
y1 = sin(2*pi*t)
y2 = sin(2*2*pi*t)

import pandas as pd

df = pd.DataFrame({'t': t, 'y1': y1, 'y2': y2})
df.head(10)
```

Out[13]:

	t	y1	y2
0	0.00	0.000000	0.000000
1	0.01	0.062791	0.125333
2	0.02	0.125333	0.248690
3	0.03	0.187381	0.368125
4	0.04	0.248690	0.481754
5	0.05	0.309017	0.587785
6	0.06	0.368125	0.684547
7	0.07	0.425779	0.770513
8	0.08	0.481754	0.844328
9	0.09	0.535827	0.904827



# HOW DO WE SHOW OUR (DATA) WORLD WITH PYTHON?

---

**matplotlib**

Seaborn

ipywidgets



**matplotlib**

**mpld3**



# matplotlib + Seaborn + ipywidgets

---

- choose your install: conda, pip
- seaborn> python viz library based on matplotlib
- ipywidgets> add interactive HTML widgets to Jupyter notebooks
  - which takes advantage of matplotlib's interactive backend connectors....
  - and allows you to layer seaborn for pretty, interactive, lightweight vis
- extensible stack for any Python visualization library!
  - Jupyter + ipywidgets + {insert your fav here}
  - deploy using Jupyter nbviewer
- easy to piece things together for reproducible analysis: reusable code + interactive plots + deployable notebooks



*# note to self: boh-kay, not boh-kah*

---

- choose your adventure: python, julia, scala, r
- choose your install: conda, pip
- targets web browsers for presentation
  - uses `tornado` to create a `bokeh` server  
`output_file()`
  - can also plot in jupyter notebook w/o `bokeh` server  
`output_notebook()`
- uses `mplexporter` to help convert `matplotlib`(!) plots in Bokeh plots
- plays well with others: `seaborn`, `ggplot.py`, `pandas`
- **excellent** documentation
- recent updates are making way for “BIG DATA” vis
- brilliant deep dive into bokeh with christine doig of continuum analytics



- choose your install: conda, pip
- uses HTML's SVG...so not really designed for “BIG DATA” vis
- still developing! noted missing features include:
  - tick locations + tick formatting
  - plt.xkcd()
  - plt.annotate()
- excellent + growing documentation, including faq’s
- extensible beyond matplotlib > client-side interface is pure javascript library (n.b. current JSON specification designed for matplotlib)

**DEMO: HELLO  
WORLD!**

# UNDERSTANDING YOUR INTERNET WORLD



*DEMO ! ? !*

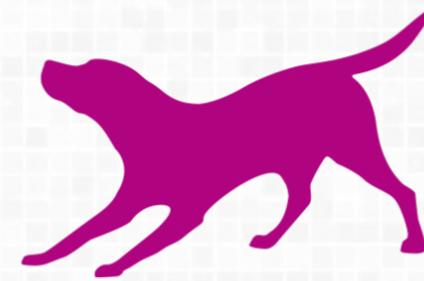
*all the vis*

# HOW DO WE SHOW OUR (DATA) WORLD WITH PYTHON?

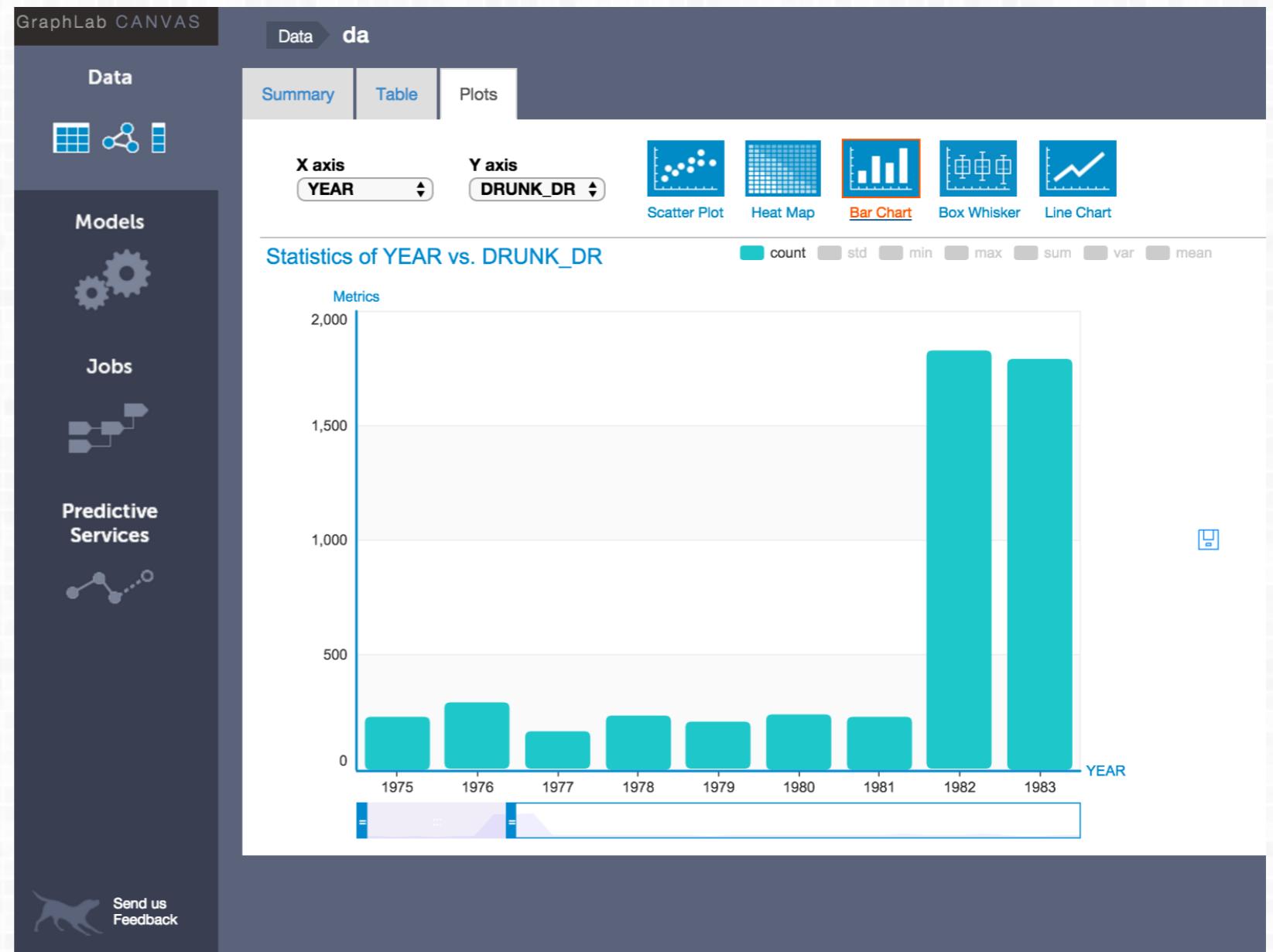
---

- Embed in your favorite Python web framework
  - Django
  - Flask
  - Tornado
  - Pyramid
- Jupyter Notebooks
  - nbviewer: standalone server for hosting notebooks
  - Jupyter + github :)
    - *n.b. github renders only static plots (non-html) on jupyter notebooks*
- Pay us to host your vis!
  - plot.ly

# NOT ALL BITS ARE OS....BUT....



- graphlab create is based on a python data science library developed + (some) os'd by dato
- graphlab canvas: interactive visualization for exploratory data analysis

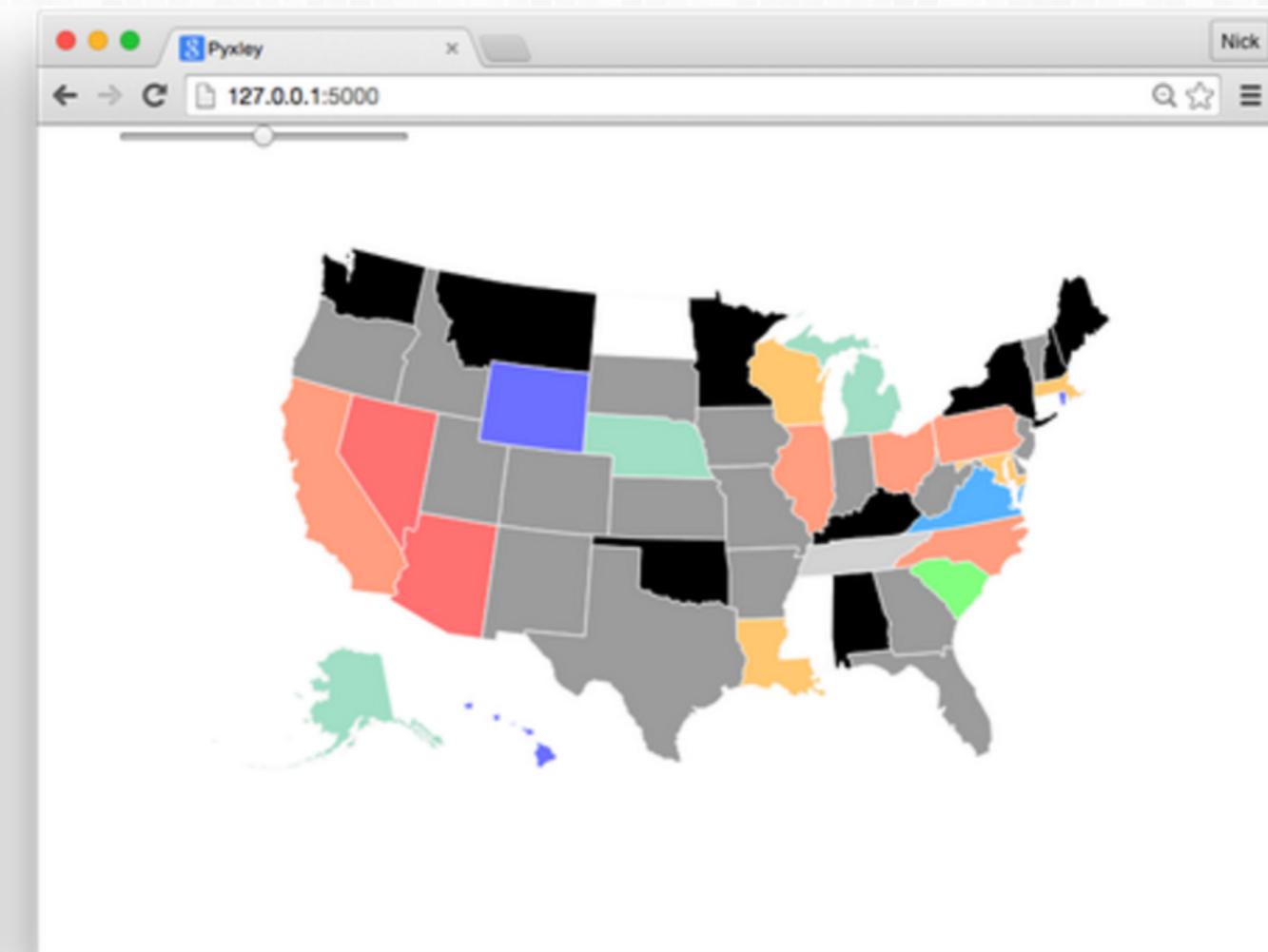
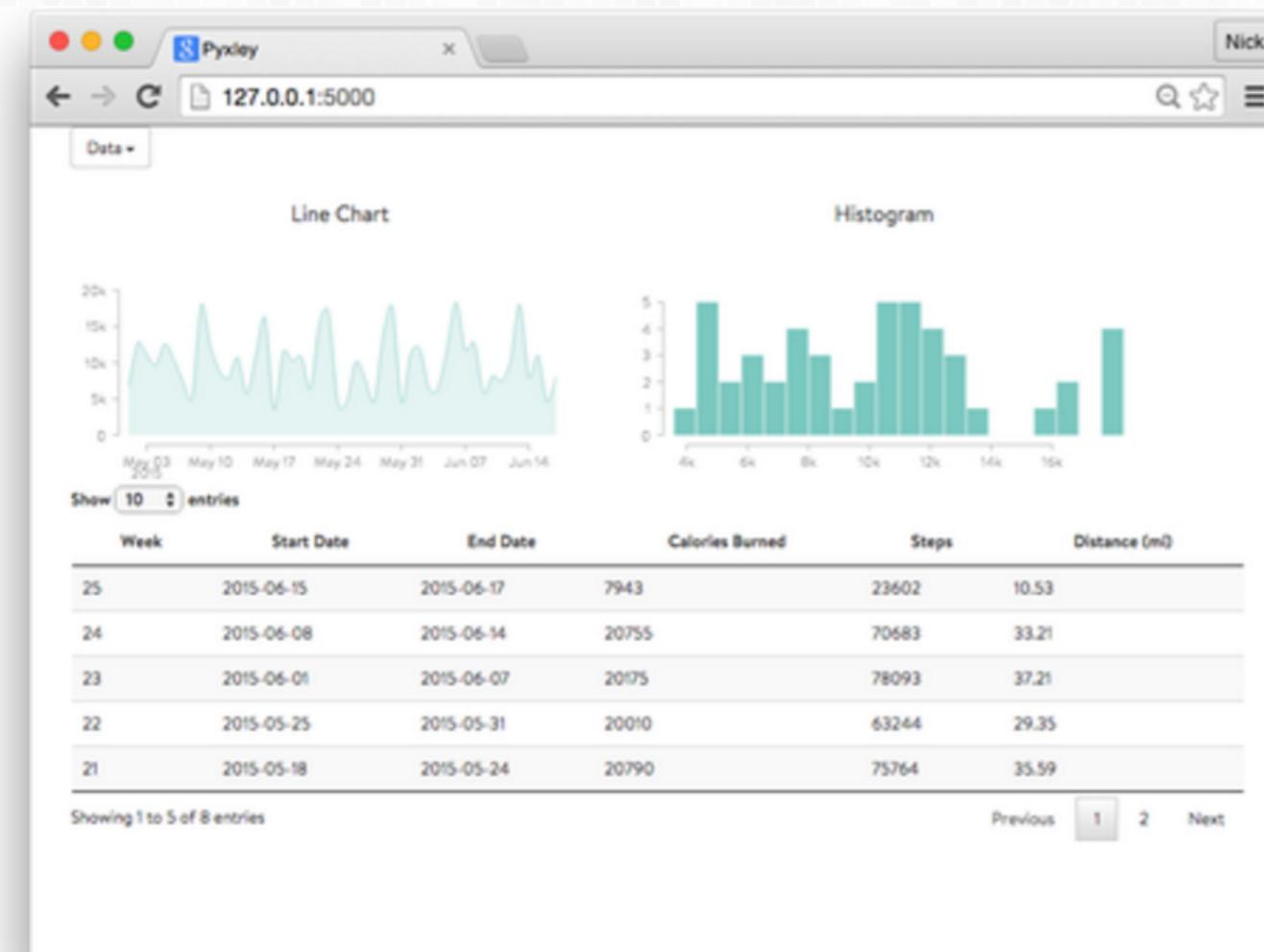


# AND EVEN SHINIER? Pyxley!

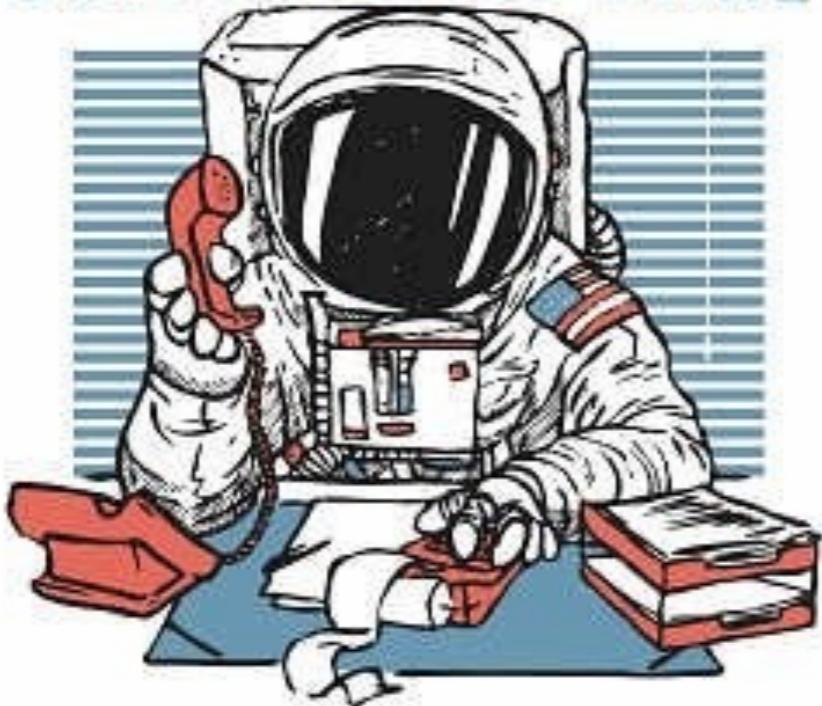
---

- python library developed + os'd by stitchfix
- pyxleyJS React to create Flask-based web apps

"Through the use of the PyReact library, we can use Jinja templating to construct and transform a single React component. The specific UI components are passed as props to the parent component. A simpler interface is provided through the use of specific wrappers for each of the component types."



\* DRESS FOR \*  
THE JOB YOU WANT



## {THANKS MUCH}

.....

- thank you to everyone in the open source community for giving me such lovely tools to talk about
- thank you @PyLadiesSEA for listening
- thank you again to @Concur for hosting, snacks + being a fantastic place to be a PyLady
- slides + git repo links will be posted to meetup page



[amanda.casari@concur.com](mailto:amanda.casari@concur.com)



@amcasari