

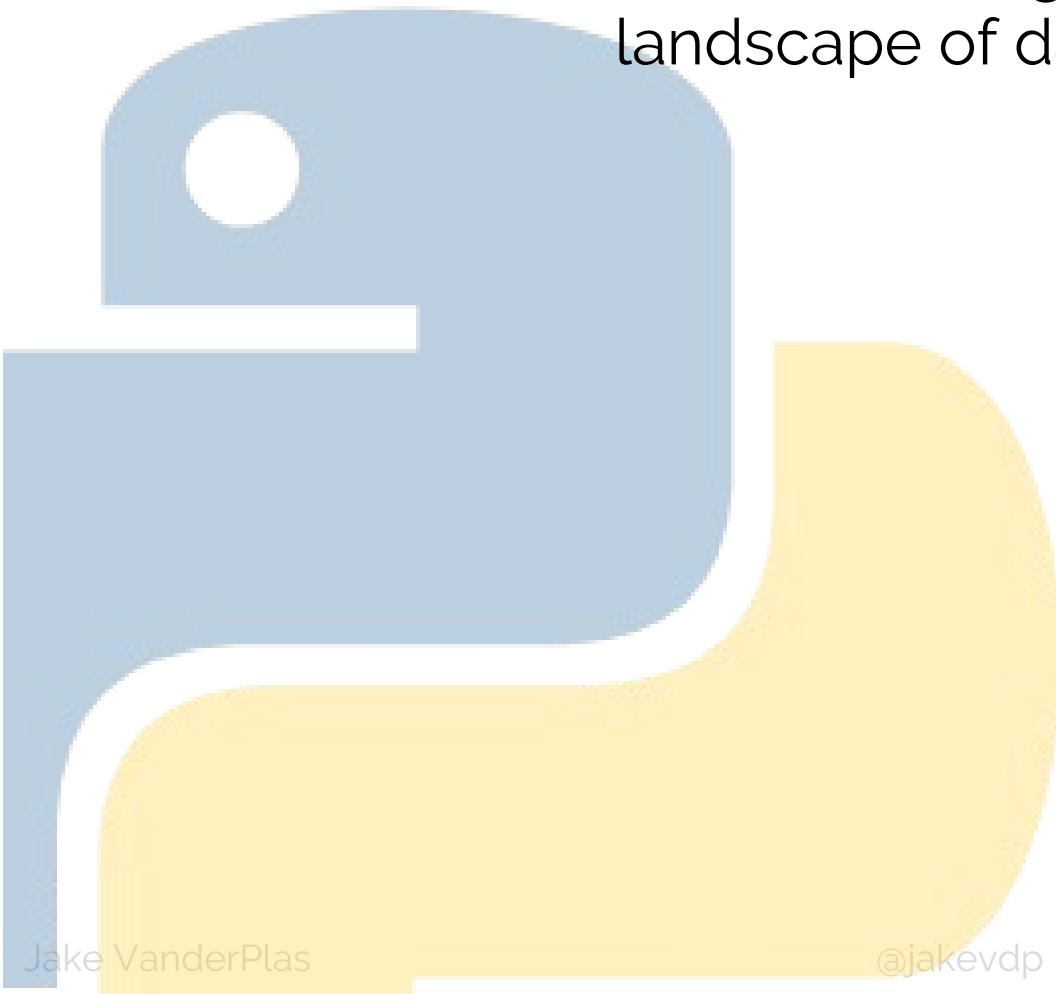
Python's Visualization Landscape

Jake VanderPlas
@jakevdp #PyCon2017

[Python's Visualization Landscape]

From the abstract:

“In this talk I’ll give an overview of the landscape of dataviz tools in Python . . .”



@jakevdp

[Python's Visualization Landscape]

From the abstract:

"In this talk I'll give an overview of the landscape of dataviz tools in Python . . ."



Jake VanderPlas @jakevdp · Apr 10

My [@pycon](#) talk is a survey of Python viz tools. Any others I should check out?

- matplotlib
- seaborn
- bokeh
- bqplot
- ggpy
- altair
- chaco

53

30

146



Visualization Landscape]

From the abstract:

In this talk I'll give an overview of the landscape of dataviz tools in Python . . ."

@jakevdp · Apr 10

survey of Python viz tools. Any others I should check out?

53

30

146



UNIVERSITY of WASHINGTON

eScience Institute

ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

Visualization Landscape]

be abstract:
alk I'll give an overview of the
pe of dataviz tools in Python . . ."

Apr 10

Python viz tools. Any others I should check out?

A screenshot of a social media feed, likely from Twitter, showing a thread of tweets about visualization tools. The feed includes profile pictures, names, timestamps (e.g., Apr 10), and tweet content. The content is as follows:

- vlad @dovgalec · Apr 10: Replying to plotly? ↗ 2
- Peter Wang @pwang · Apr 10: Replying to @jakevdp @pycon Holoviews? Datasader? :-)
- Ryan Abernathy @rabernat · Apr 10: big +1 to holoviews and datashader. Changing the way I work.
- Rebecca @rebeccabrown · Apr 10: Replying to Yellowbrick learn estimators ↗ 2
- Jake VanderPlas · Apr 10: Thanks! I ↗ 1
- Rebecca @rebeccabrown · Apr 10: Will be speaking ↗ 1
- Manuel Gómez · Apr 10: Replying to ggplot, cm ↗ 1
- Jake VanderPlas · Apr 10: It's called ↗ 1
- Manuel Gómez · Apr 10: oh snap, o ↗ 1
- Molly Peacock · Apr 10: Replying to @yt_astro ↗ 1
- Thomas R · Apr 10: Replying to If you want great if yo ↗ 1
- Peter Wang @pwang · Apr 10: Replying to @jakevdp @pycon Holoviews? Datasader? :-)
- Ryan Abernathy @rabernat · Apr 10: big +1 to holoviews and datashader. Changing the way I work.
- Rebecca @rebeccabrown · Apr 10: Replying to Yellowbrick learn estimators ↗ 1
- Jake VanderPlas · Apr 10: Thanks! I ↗ 1
- Rebecca @rebeccabrown · Apr 10: Will be speaking ↗ 1
- Manuel Gómez · Apr 10: Replying to ggplot, cm ↗ 1
- Jake VanderPlas · Apr 10: It's called ↗ 1
- Manuel Gómez · Apr 10: oh snap, o ↗ 1
- Molly Peacock · Apr 10: Replying to @yt_astro ↗ 1
- Thomas R · Apr 10: Replying to If you want great if yo ↗ 1
- Jason Sundram @jsundram · Apr 10: Replying to @jakevdp @pycon Are you going to make recommendations? I recently remade the same plot with several of the above. It was ... instructive.
- Filipe Fernandes @ocefpaf · Apr 10: Replying to @jakevdp @pycon vispy maybe?
- Sylvain Corlay @SylvainCorlay · Apr 10: Replying to @jakevdp @pycon ipyvolume, pythreejs, ipyleaflet.
 ⓘ Translate from Czech
- Sylvain Corlay @SylvainCorlay · Apr 10: With the new binary serialization work coming up in ipywidgets 7.0, these will become much much faster.
- Alejandro Vidal @doblepensador · Apr 10: Where can I read about this serialization?
- Jeff Mayse @jeff_mayse · 14h: Replying to @jakevdp @pycon Datasaders is worth a look if you're not already folding it into Bokeh
- Kelsey Jordahl @kajord · Apr 10: Replying to @jakevdp @pycon Mayavi

At the bottom of the feed, there are navigation controls: 53, 146, and a search icon.

ation Landscape]

react:
give an overview of the
dataviz tools in Python . . ."

viz tools. Any others I should check out?

vlad @dovgalec · Apr 10
Replying to @plotly
Replies 2

Peter Wang @pwing · Apr 10
Replying to @jakevdp @nvcnn
Replies 1

Ariel Rokem @arokem · Apr 10
And VTK more generally
Replies 1

Ryan Ab... big +1 to work.
Replies 1

Ian @ianHopkinson_ · Apr 10
Replies to @jakevdp @YhatHQ @pycon
Have a look at datashader - does good aggregations of millions of plots for display
datashader.readthedocs.io/en/latest/

Rebecca...
Replies 2

Jason S...
Replies 3

Chris Holdgraf @choldgraf · Apr 10
Replies to @jakevdp @pycon
I think it's worth mentioning the up-and-coming html 3d viz libraries, e.g. ipyvolume + pythreejs.

Filipe Ferreira...
Replies 1

1 more reply

François Dion @f_dion · Apr 10
Replies to @jakevdp @pycon
Might as well plug mine:
stemgraphic.org
works with dask too. Demoed at PyDataCarolinias 2016 & tutorial at IEEE SouthEastCon 2017

Home
Welcome to Stemgraphic.
Stemgraphic is a command line tool
that can be used to analyse distribut...
stemgraphic.org

Sylvain...
Replies 1

Sylvain...
With the 7.0, these
Replies 1

Alejandro...
Where c
Replies 1

Molly Pe...
Replies 1

Jeff May...
Replies 4

Ryan Rosario @DataJunkie · Apr 10
Replies to @jakevdp @pycon
ggplot2 (if it's still being actively developed for Python)?

Jeroen Janssens @jeroenjhanssens · Apr 10
Replies to @jakevdp @pycon
What a coincidence. Today I taught my students how to use #rstats ggplot2 in Jupyter Notebook via rpy2.

Landscape]

An overview of the tools in Python . . ."

Any others I should check out?

vlad @dovgalec · Apr 10
Replying to plotly?
↳ 2

Peter Wang @pwang · Apr 10
Replying to @jakevdp @pycon
Holoviews
↳ 1

Ariel Rokem
And VTK now
↳ 2

Ryan Ab
big +1 to work.
↳ 1

Ian (@ianHop)
Replying to @jakevdp
Have a look millions of plots in datashader.
↳ 1

Rebecca
Replying to Yellowbrick
learn estimators
↳ 2

Jake Van
Thanks! I
↳ 1

Rebecca
Will be speaking
↳ 1

Manuel G
Replying to ggplot, cm
↳ 1

Jake Van
It's called
↳ 1

Manuel G
oh snap, o
↳ 1

Molly Pee
Replying to @yt_astro
↳ 1

Thomas F
Replying to If you want great if yo
↳ 1

Kelsey J
Replying to Mayavi
↳ 1

Maarten A. Breddels @maartenbreddels · Apr 10
Replying to @jakevdp @pycon
I'd also say ipyvolume, but as author that makes me slightly biased 😊
↳ 2
1 more reply

Zane Selvans @ZaneSelvans · 22h
Replying to @jakevdp @teoliphant @pycon
Plotly?
↳ 2
1 more reply

Gautham Narayan @gsnarayan · Apr 10
Replying to @jakevdp @pycon
glueviz
↳ 2
1 more reply

Tom Brander @dartdog · Apr 10
Replying to @jakevdp @pycon
altair?
altair-viz/altair
altair - Declarative statistical visualization library for Python
github.com
↳ 2
1 more reply

Francois Fleuret (@francoisf)
Replying to @jakevdp
Might as well stemgraphic works with a tutorial at IEP
stemgraphic
↳ 1

Sylvain Corlay (@jakevdp)
Replying to @jakevdp
With the 7.0, these
↳ 1

Alejandra (@alejandromm)
Where can I find
↳ 1

Jeff Maynard (@pumpkin_bob)
Replying to @jakevdp
There's a real Handbook - take a look.
↳ 1

Kelsey J
Replying to Mayavi
↳ 1

Ryan Rosar
Replying to ggplot2 (if it's still around)
↳ 1

Suman (@suman12029) · Apr 10
Replying to @jakevdp @pycon
Matplotlib from the ground up would be enough!! 😂😂
↳ 1

Jeroen Janssens (@jeroenhjanssens) · Apr 10
Replying to @jakevdp @pycon
What a coincidence. Today I taught my students how to use #rstats ggplot2 in Jupyter Notebook via rpy2.
↳ 1

dscape]

view of the
in Python . . ."

I should check out?

vlad @dovgalec · Apr 10
Replying to plotly?
↳ 2

Peter Wang @pwang · Apr 10
Replying to @jakevdp @pycon
Holoviews
↳ 1

Ariel Rokem And VTK mo
Ryan Ab big +1 to work.
↳ 1

Ian @ianhop Replying to plotly? Have a look millions of plots in datashader.
↳ 1
1 more reply

Jason S Replying to plotly? Are you the same?
↳ 1

Chris Holdgr Replying to plotly? I think it's well libraries, e.g.
↳ 1
1 more reply

Filipe Ferreira Replying to plotly? vizpy makes things easier.
↳ 1

Sylvain Lefebvre @jakevdp Replying to plotly? Might as well stemgraphic works with a tutorial at IEP
↳ 1

Sylvain Lefebvre With the 7.0, these are great.
↳ 1

Alejandro Sanchez Where can I find the documentation?
↳ 1

Jeff May Replying to plotly? Datasheets, Bokeh, and ggplot2.
↳ 1

pumpkin_bob Replying to plotly? There's a real Handbook - take a look.
↳ 1

Alejandro Sanchez Replying to plotly? Pyqtgraph is great.
↳ 1

Ryan Rosar Replying to plotly? ggplot2 (if it's available)
↳ 1

Kelsey J Replying to plotly? Mayavi is great.
↳ 1

Jeroen Janssens @jeroenhjanssen Replying to @jakevdp @pycon What a coincidence. Today I taught my students how to use #rststats ggplot2 in Jupyter Notebook via rpy2.

Maarten A. Breddels @maartenbreddels · Apr 10
Replying to @jakevdp @pycon
I'd also say ipyvolume, but as author that makes me slightly biased
↳ 1

Zane Selk Replying to plotly? Plotly?
↳ 1

Gautham Replying to plotly? glueviz
↳ 1

Tom Brander @dardog · Apr 10
Replying to @jakevdp @pycon
duh can't read.
↳ 1

أسماء العابد @OAlotalk · Apr 10
Replying to @jakevdp @YhatHQ @pycon
Folium? Not exactly a viz tool; but has some viz aspects to it.
↳ 1

Andrew Heusser @andyheusser · 8h
Replying to @jakevdp @pycon
hypertools - viz for high dim data:
readthedocs.org/projects/hyper/
↳ 1

Majid alDosari @msdtechcode · 8h
Replying to @jakevdp @pycon
... holoviews vs altair
↳ 1

PyData Bratislava @PyDataBA · 17h
Replying to @jakevdp @teoliphant @pycon Try also physt (if you need just histograms it's the best library so far).
↳ 1

Gowrishankarnath @g_s_nath · 19h
Replying to @jakevdp @teoliphant @pycon
may be plotly should also be added
↳ 1

Melodie Wilson @moor1204 · Apr 11
Replying to @jakevdp @pycon
Plotly is great! As is their cufflinks library
↳ 1

dscape]

the
on . . .

check out?

vlad @dovgalec · Apr 10
Replying to plotly?
↳ 2

Peter Wang @pwang · Apr 10
Replying to @jakevdp @pycon
Holoviews
↳ 1

Ariel Rokem And VTK mo
Ryan Ab big +1 to work.
↳ 1

Ian @lanHop Replying to I Have a look millions of p datashader.r
1 more reply

Jason S Replying Are you the same
Chris Holdg Replying to I think it's w libraries, e.g.
1 more reply

Filipe Fe Replying vispy ma
1 more reply

Sylvain @jakevdp ⚡ Transl Might as we stemgraphic works with a tutorial at IEB

Sylvain With the 7.0, thes
Alejandro Where c

Molly Pee Replying to @yt_astro

Thomas R Replying to If you war great if yo

Kelsey J Replying to Mayavi
1 more reply

Jeroen Janssens @jeroenhjanss Replying to @jakevdp @pycon What a coincidence. Today I taught my students how to use #rstats ggplot2 in Jupyter Notebook via rpy2.

Maarten A. Breddels @maartenbreddels · Apr 10
Replying to @jakevdp @pycon I'd also say ipyvolume, but as author that makes me slightly biased

Zane Se Replying Plotly?
Gautham Replying glueviz

Tom Bra Replying altair?

Andrew Heuss Replying to @ja hypertools - viz readthedocs.or

Maarten Replying I'd choos might be

Hannah Replying @tdhopp

pumpkin_b Replying to There's a real Handbook - a look.

Alejandro Replying Pyqtgraph

Ryan Rosar Replying to ggplot2 (if it's

Suman C Replying Matplotlib

Melodie Wilson Replying to @ja Plotly is great! A

Faro @faroit · Apr 10
Replying to @jakevdp gr.gr-framework.org

Eric Fischer @fish_in_mn · Apr 10
Replying to @jakevdp @pycon plotly

Don't Blink @sparkyblix · Apr 10
Replying to @jakevdp @pycon NetworkX and graph-tool both interface with graphviz which has python bindings

Paula R Alves @LadyData · Apr 10
Replying to @jakevdp @pycon Toyplot! This project needs some love and more contributors.

NatureBoyShow @natureboyshow · Apr 10
Replying to @jakevdp @pycon Jupyter notebook

Kevin Dungs @kdungs · Apr 10
Replying to @jakevdp @pycon ROOT? 😂😂😂

Rick Galbo @RickGalbo · Apr 10
Replying to @jakevdp @pycon custom js in python -> for network visualization

Peter Kurpinski @londoncatblue · Apr 10
Replying to @jakevdp @pycon ggplot maybe? just out of ecumenistic attitudes :)

Miloš Miljković @mlishke · Apr 10
Replying to @jakevdp @pycon If it's 3D inclusive, ipyvolume.

Jernej Zupančič @JernejZupancic · Apr 10
Replying to @jakevdp @pycon Maybe Holoviews and Datashader.

vlad @dovgalec · Apr 10



Replying to [plotly?](#)

↳ 2

Jake Van
Thanks! O
:)

Rebecca
Replying to
Yellowbrick
learn estri
↳ 2

Jake Van
Thanks! I
↳ 1

Rebecca
Will be sp
↳ 2

Manuel G
Replying to
ggplot, cm
↳ 1

Jake Van
It's called
↳ 1

Manuel G
oh snap, o
↳ 1

Molly Pee
Replying to
@yt_astro
↳ 1

Thomas F
Replying to
If you war
great if yo
↳ 1

Peter Wang @pwang · Apr 10
Replying to [@jakevdp @pycon](#)
Holoviews

Ariel Rokem
And VTK mo
↳ 1

Ryan Ab
big +1 to
work.

Ian @ianHop
Replying to [@jakevdp](#)
Have a look
millions of p
dataloader.r

1 more re

Jason S
Replying to
Are you
the same

Chris Holdg
Replying to [@jakevdp](#)
I think it's w
libraries, e.g.

1 more re

Filipe Fe
Replying to
vispy ma

Sylvain @jakevdp
Transl

Sylvain @jakevdp
Might as we
stemgraphic
works with d

tutorial at IE

Alejandro
Where c

Jeff May
Replying to
Datalash

Bokeh

pumpkin_b
Replying to [@jakevdp](#)
There's a rea

Handbook -
a look.

Kelsey J
Replying to
Mayavi

↳ 1

Ryan Rosar
Replying to [@jakevdp](#)
ggplot2 (if it'

Suman S
Replying to [@jakevdp](#)
Matplotlib

Jeroen Janssens @jeroenhjanssen
Replying to [@jakevdp @pycon](#)
What a coincidence. Today I taught my students how to use

#rstats ggplot2 in Jupyter Notebook via rpy2.

Maarten A. Breddels @maartenbreddels · Apr 10

Replying to [@jakevdp @pycon](#)

I'd also say ipyvolume, but as author that makes me slightly
biased 😊

Faro @faroit · Apr 10
Replying to [@jakevdp](#)
gr.gr-framework.org

Zane Se
Replying to
Plotly?

Ian @ianHop
Replying to [@jakevdp](#)
Have a look
millions of p
dataloader.r

Gauthan
Replying to
glueviz

Chris Holdg
Replying to [@jakevdp](#)
I think it's w
libraries, e.g.

Tom Bra
Replying to
altair?

Andrew Heuss
Replying to [@jakevdp](#)
hypertools - viz
readthedocs.or

Maarten
Replying to
I'd choos
might be

Majid alDosari
Replying to [@jakevdp](#)
... holoviews vs

Hannah
Replying to
@tdhopp

PyData Bratisl
Replying to [@jakevdp](#)
@elephant @p
it's the best lib

Alejandro
Replying to [@jakevdp](#)
Pyqtgraph

Gowrishankar
Replying to [@jakevdp](#)
maybe be plotly s

Melodie Wilson
Replying to [@jakevdp](#)
Plotly is great!

Miloš Miljković @
Replying to [@jakevdp](#)
ggplot maybe? jus

Peter Kurpinski @
Replying to [@jakevdp](#)
ggplot maybe? jus

Jernej Zupančič @JernejZupancic · Apr 10
Replying to [@jakevdp @pycon](#)
Maybe Holoviews and Datalader.

H Summers @GnomicOne · Apr 10

Replying to [@jakevdp @pycon](#)
Plot.ly gets a lot of chatter. Mid3 perhaps?

David Paolella @ddpaolella · Apr 10
Replying to [@jakevdp @pycon](#)
Will there be a way to watch your talk online?

Denis Akhiyarov @denfromufa · Apr 10
Replying to [@jakevdp @pycon](#)
Pandas plotting API

Translate from Estonian

Michael Zingale @Michael_Zingale · Apr 10
Replying to [@jakevdp @pycon](#)
yt?

Eric Fischer @fish_in_mn · Apr 10
Replying to [@jakevdp @pycon](#)
plotly

Don't Blink @sparkyblix · Apr 10
Replying to [@jakevdp @pycon](#)
NetworkX and graph-tool both interface with graphviz which
has python bindings

Paula R Alves @LadyData · Apr 10
Replying to [@jakevdp @pycon](#)
Toynet! This project needs some love and more contributors.

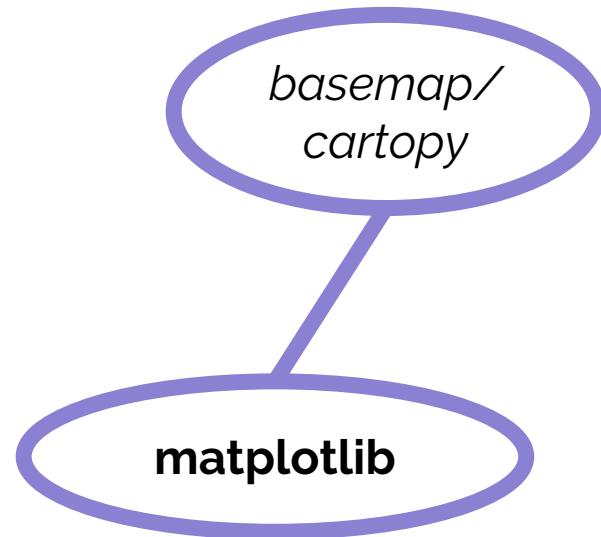
NatureBoyShow @natureboyshow · Apr 10
Replying to [@jakevdp @pycon](#)
Jupyter notebook

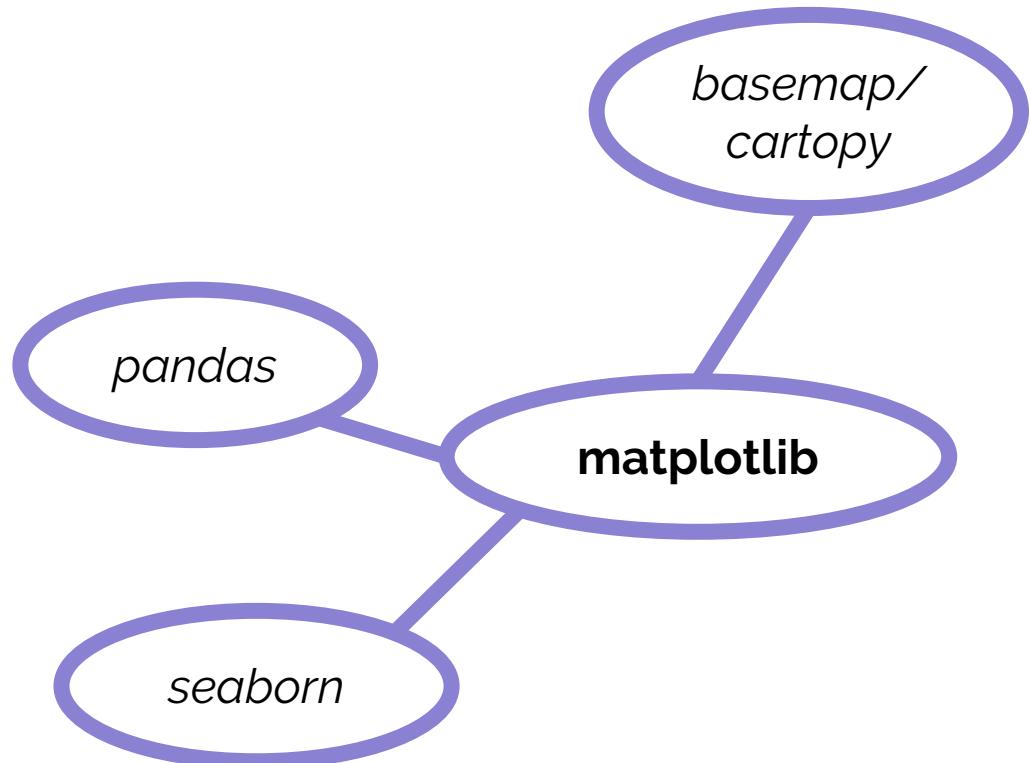
Kevin Dungs @kdungs · Apr 10
Replying to [@jakevdp @pycon](#)
ROOT? 😂😂😂

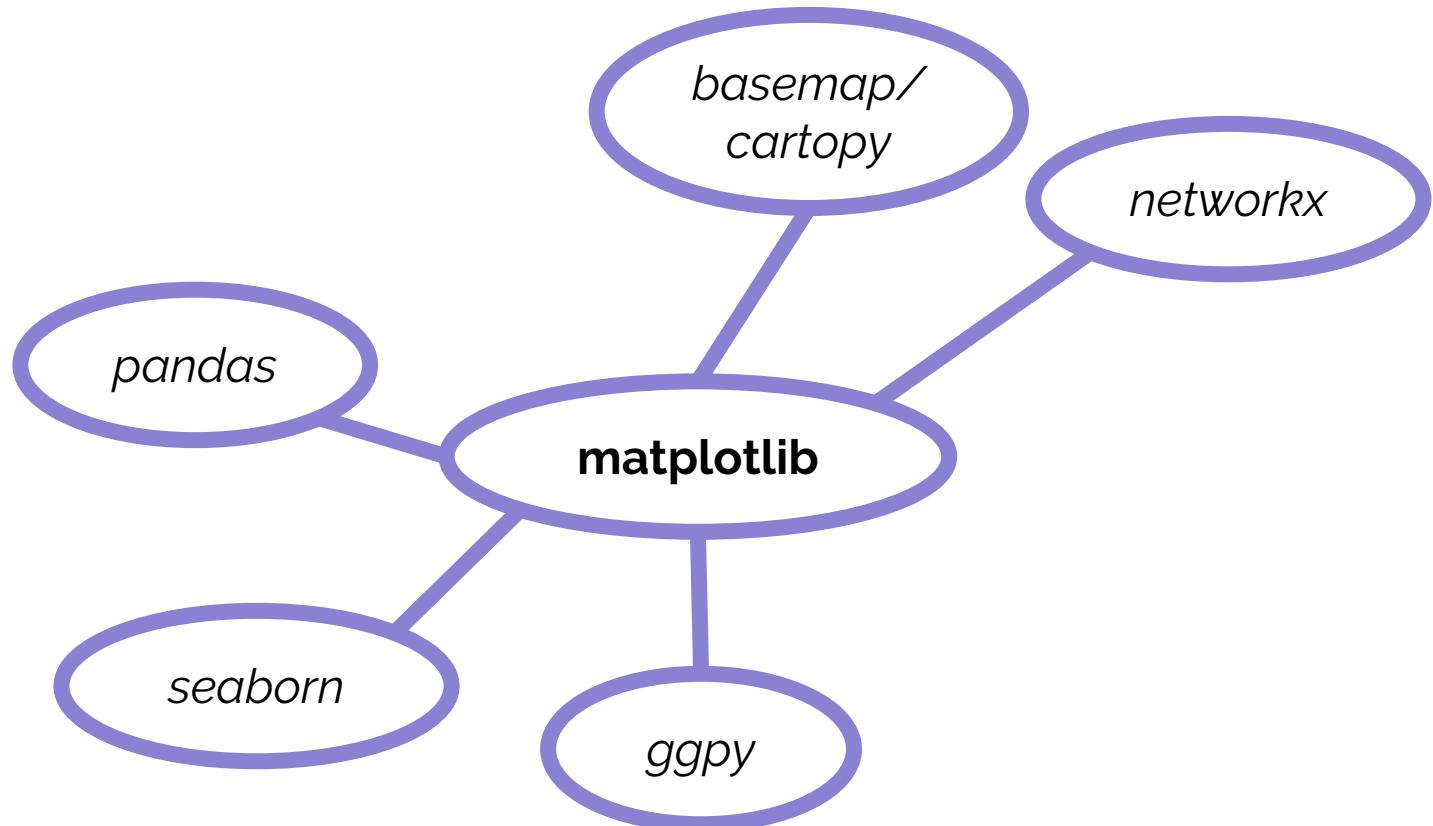
eScience Institute
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

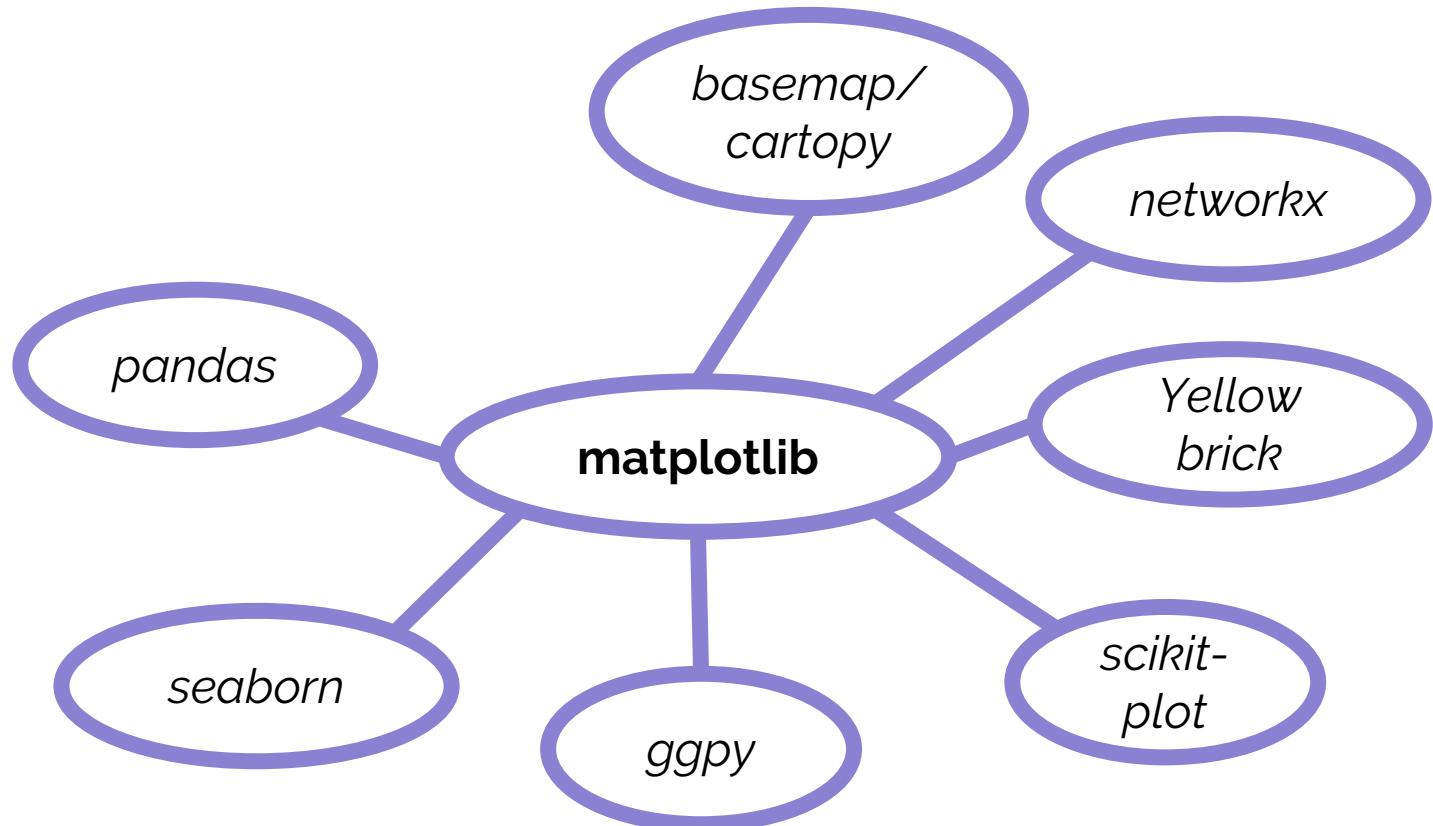
[Making Sense of the Deluge]

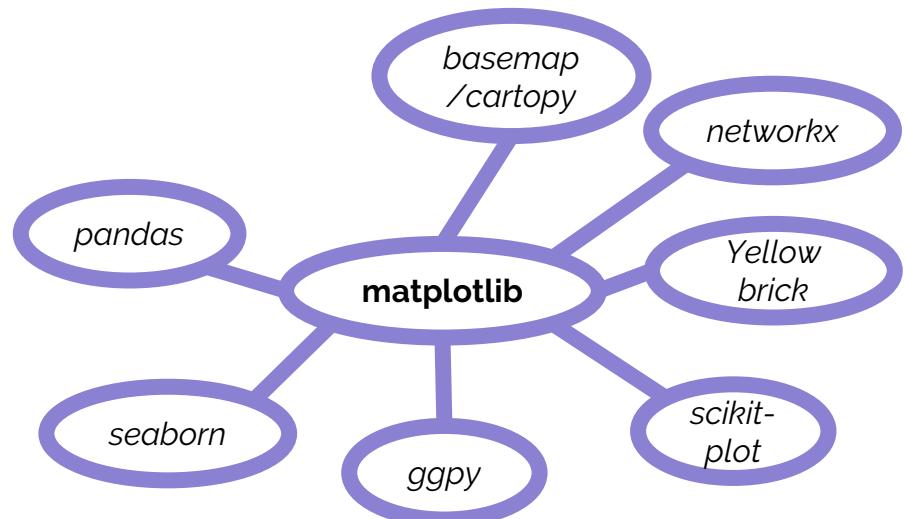
matplotlib

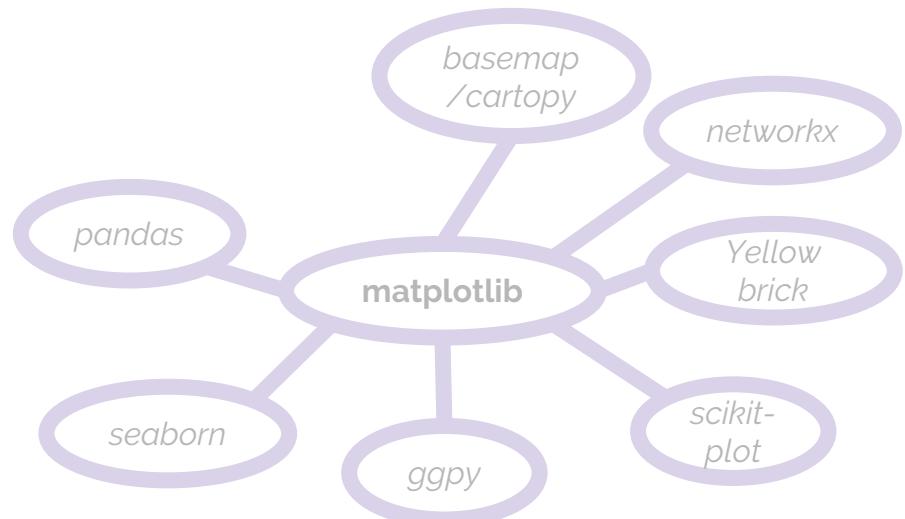




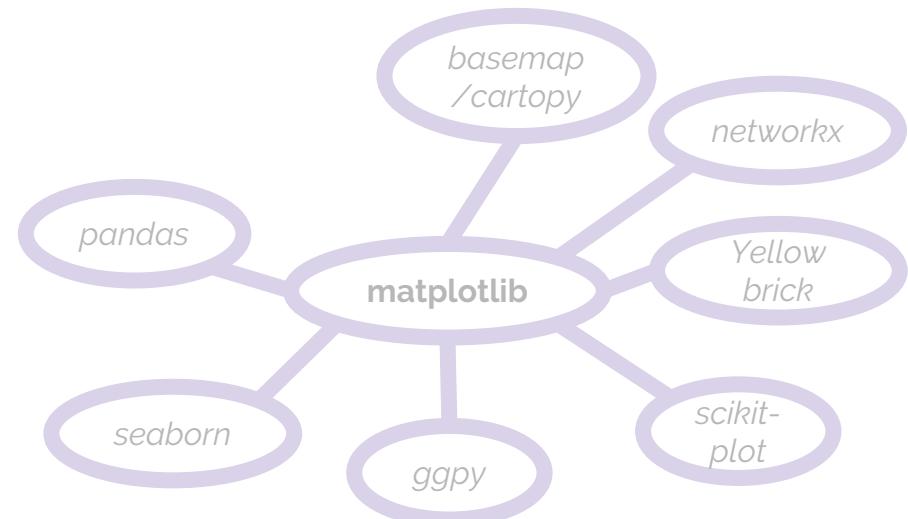


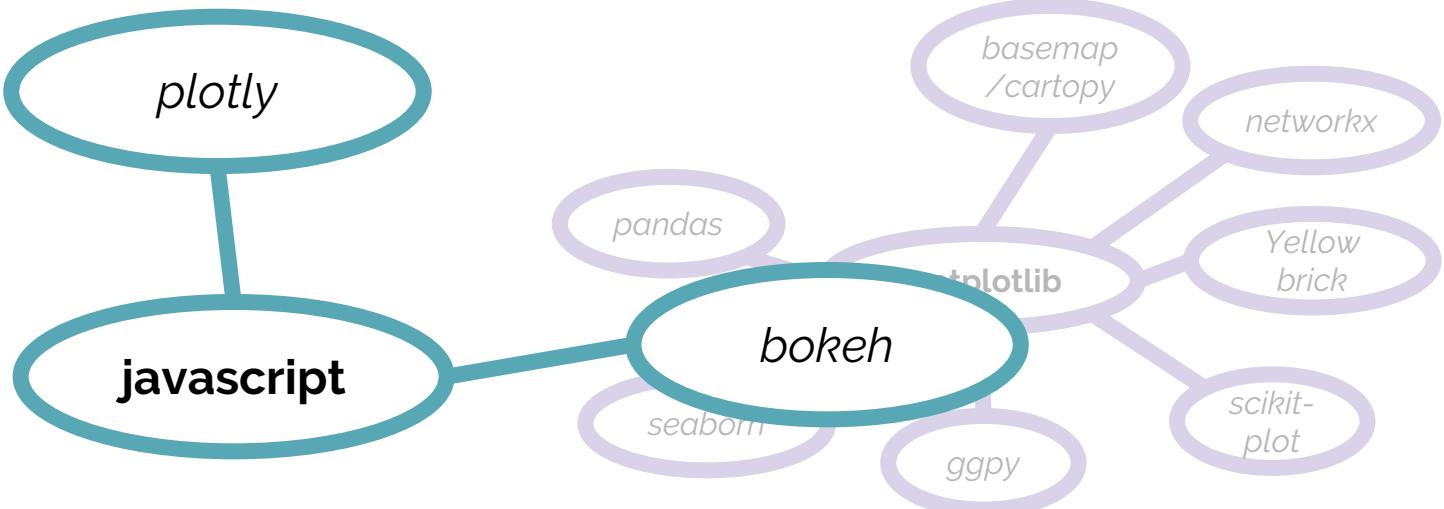


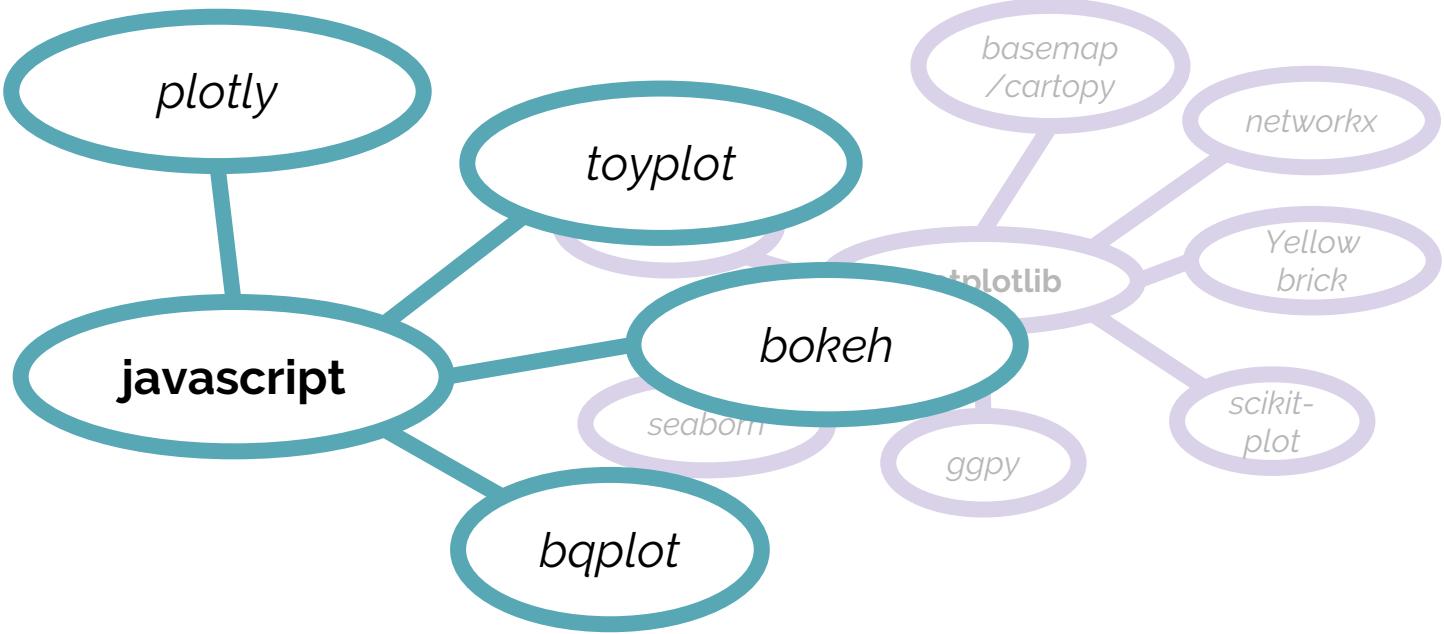


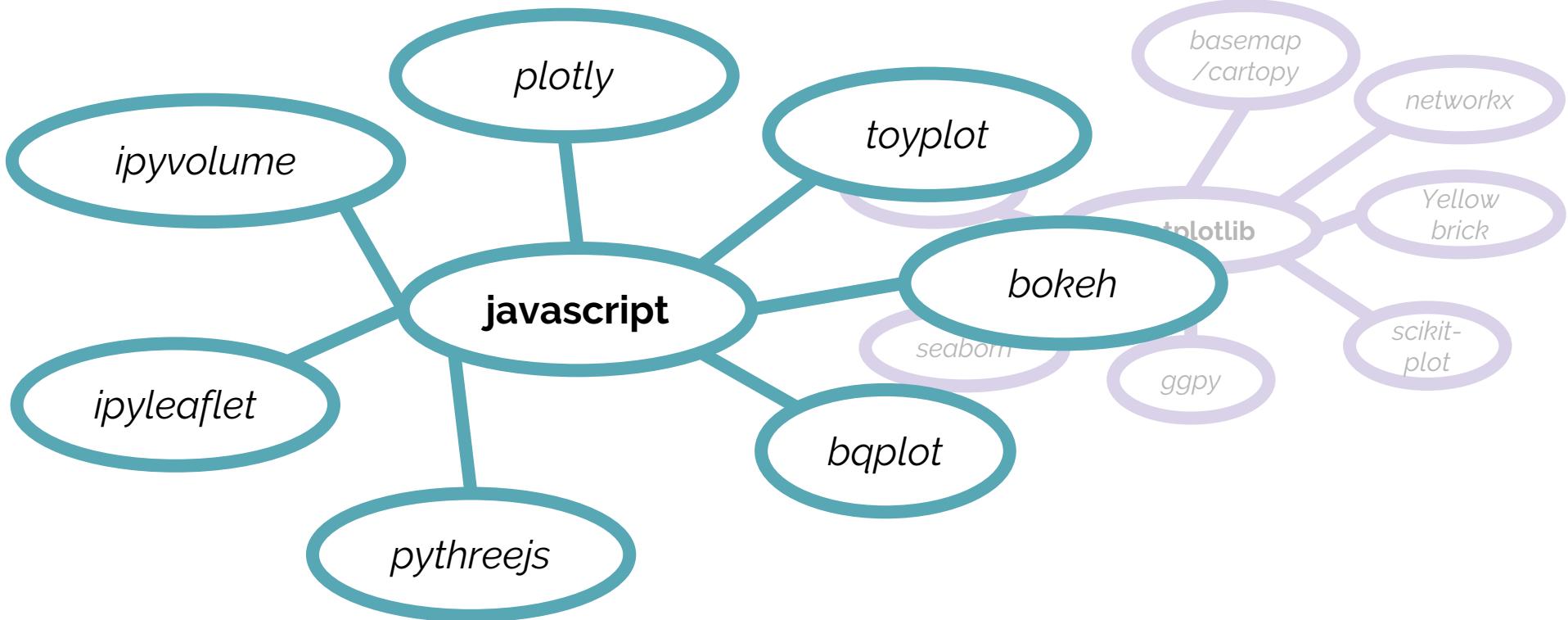


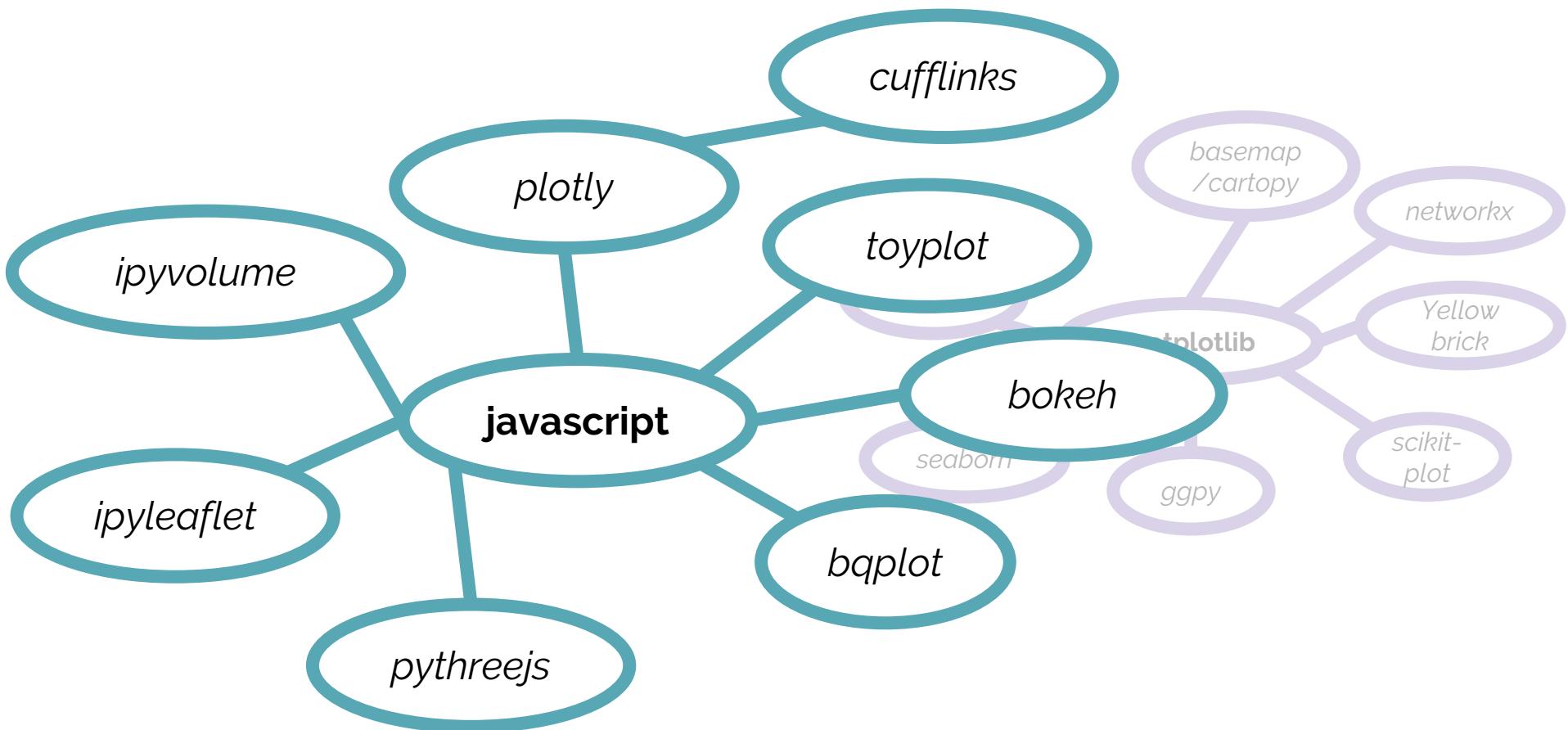
javascript

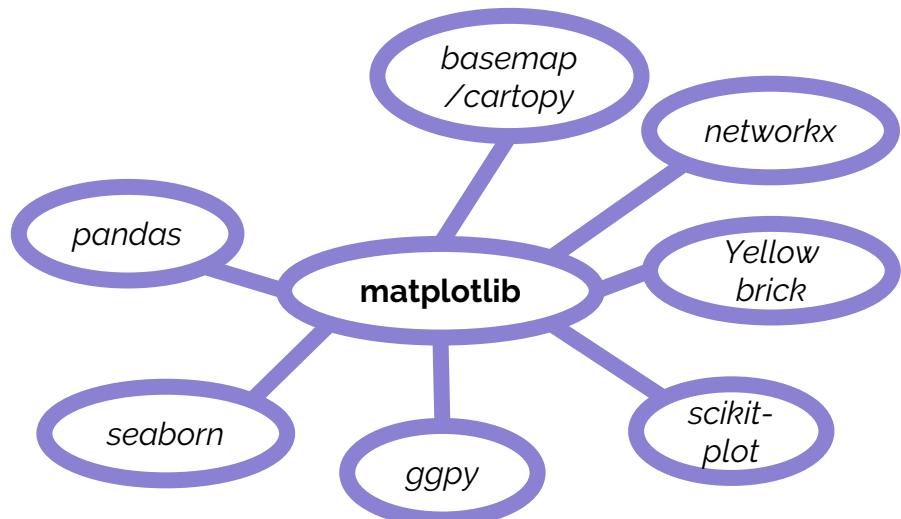
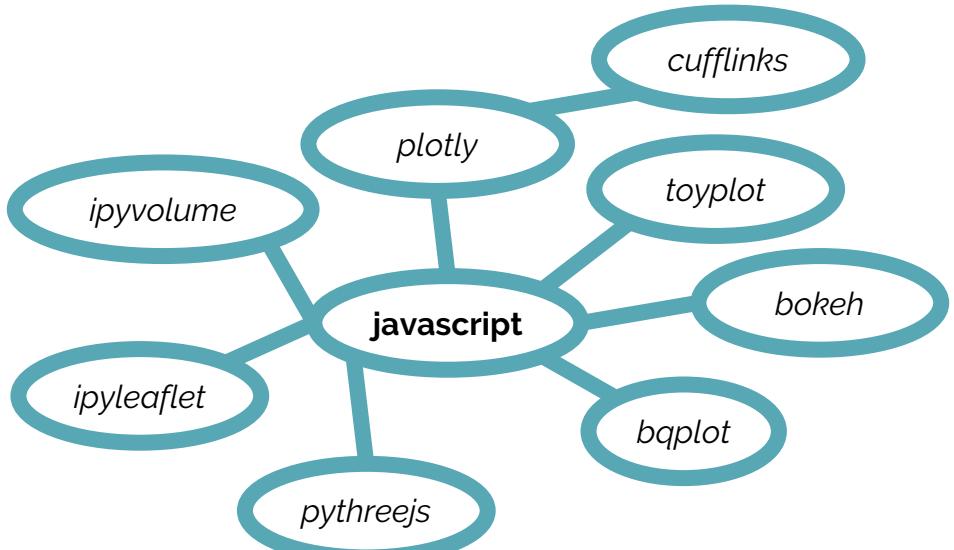


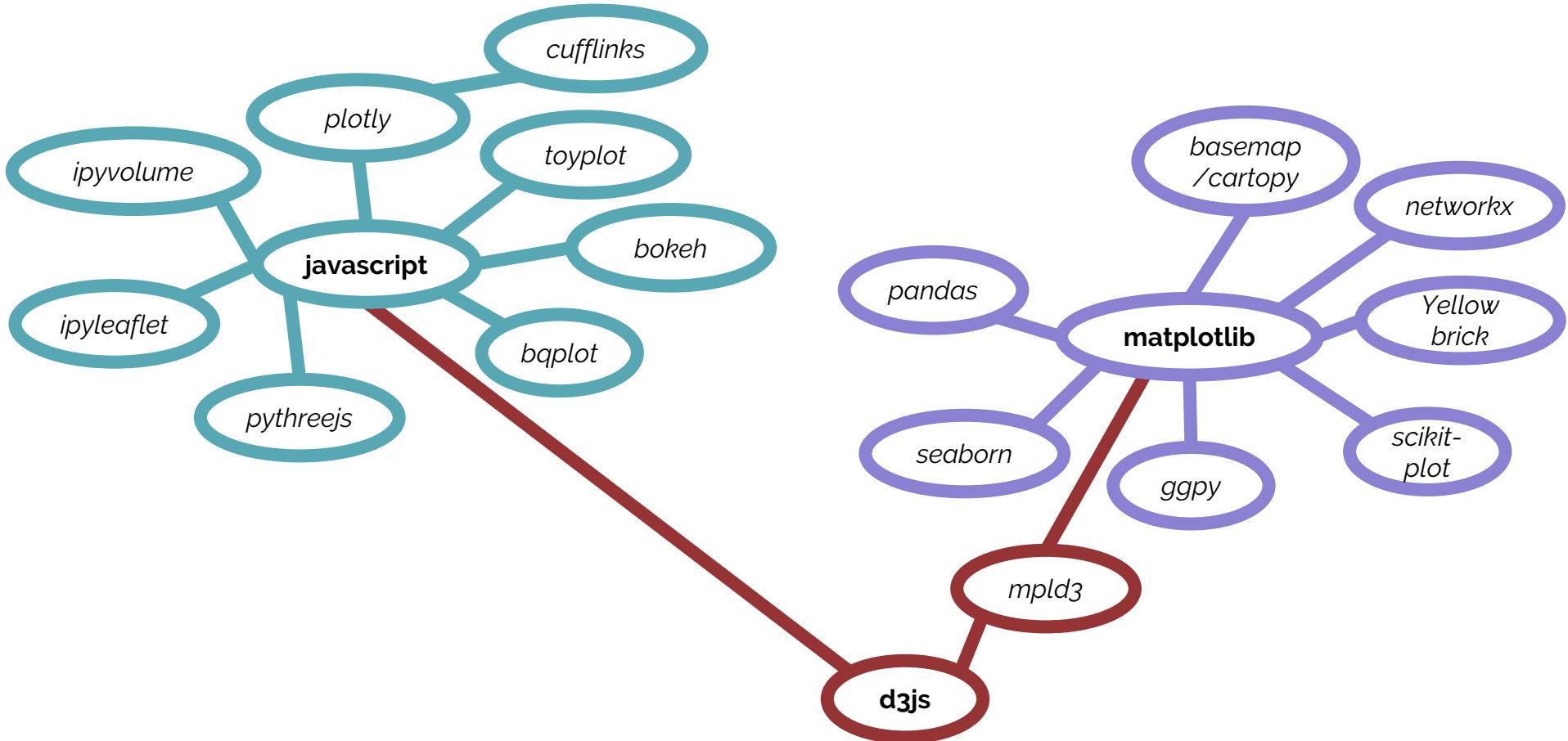


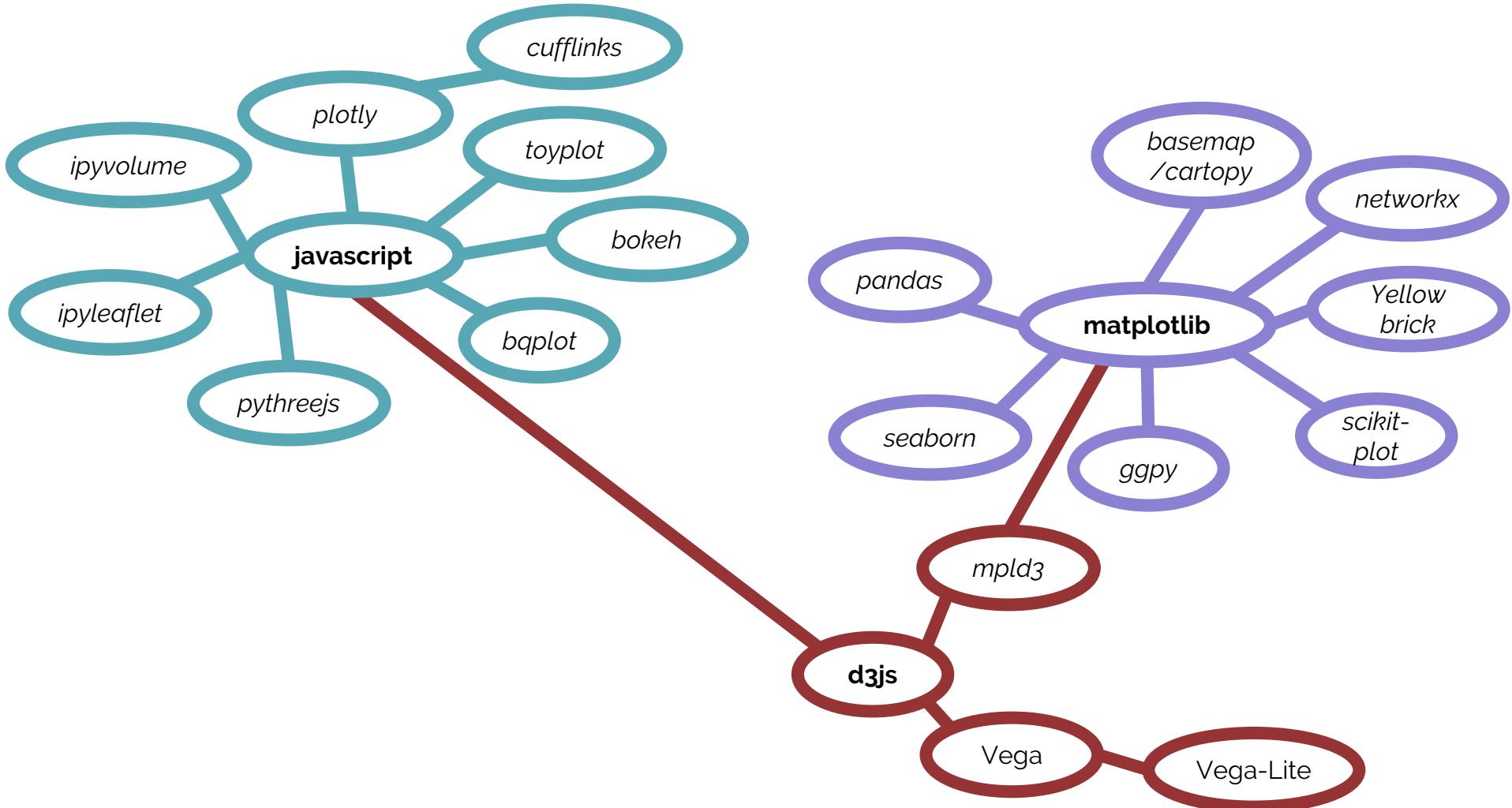


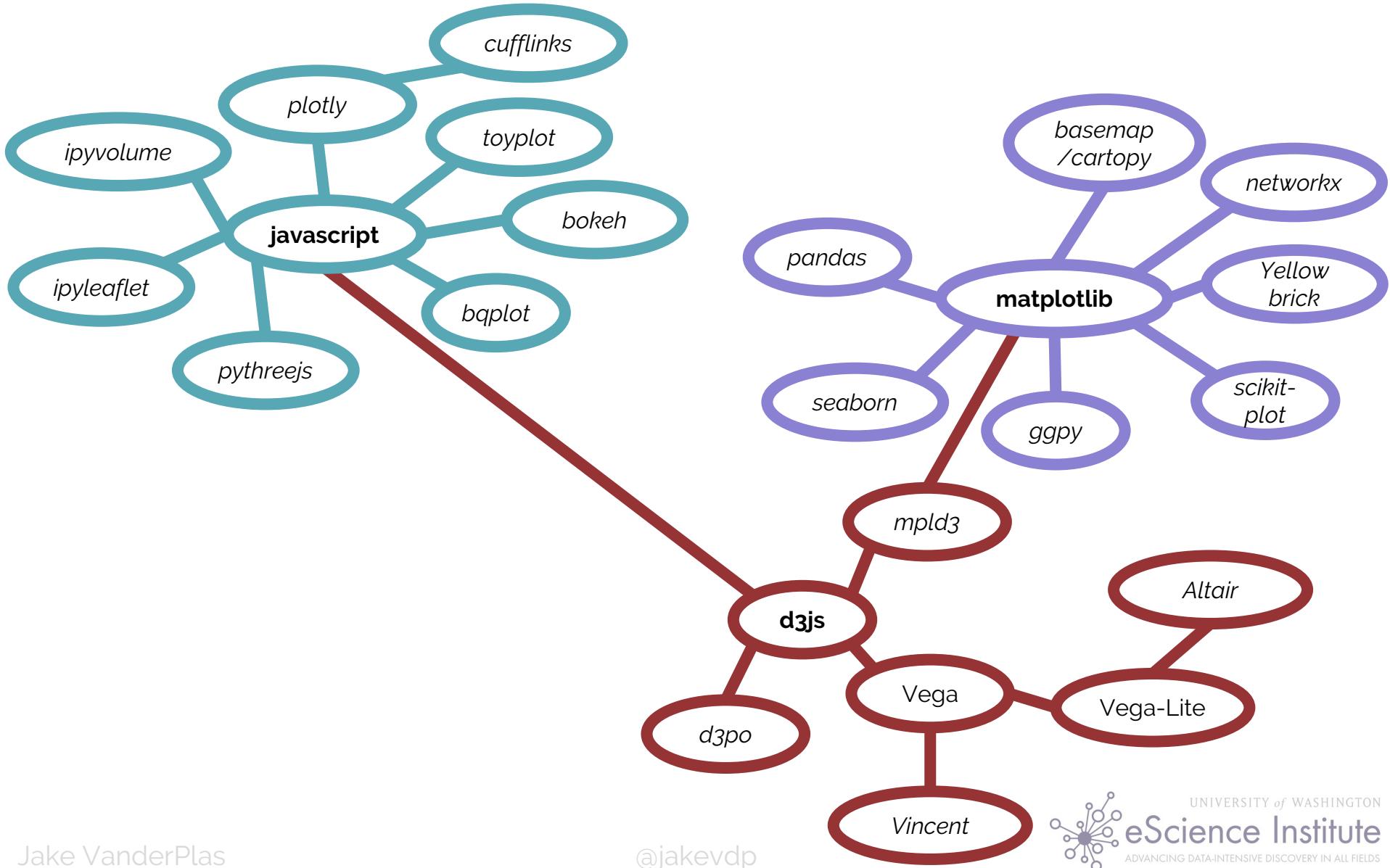


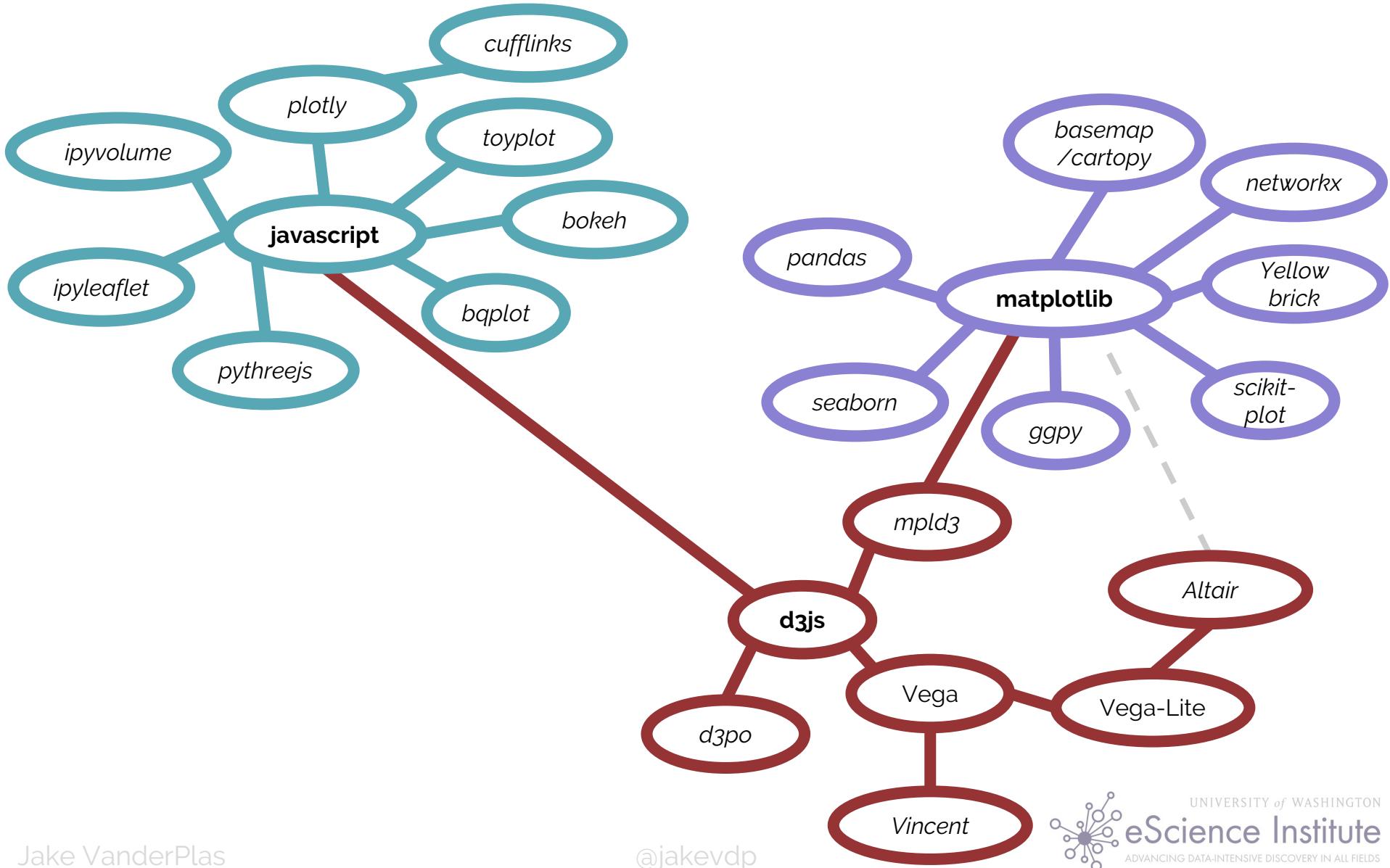


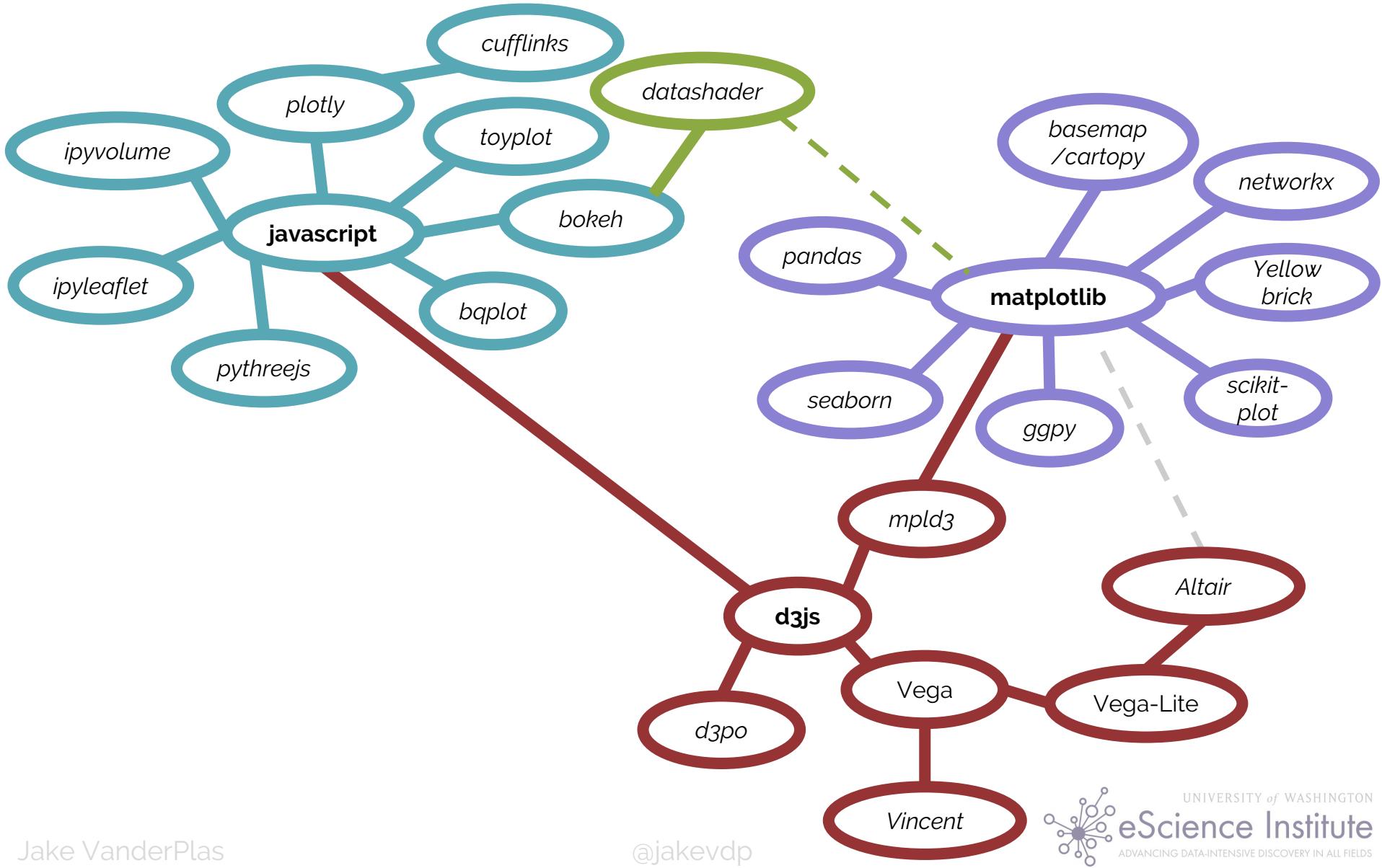


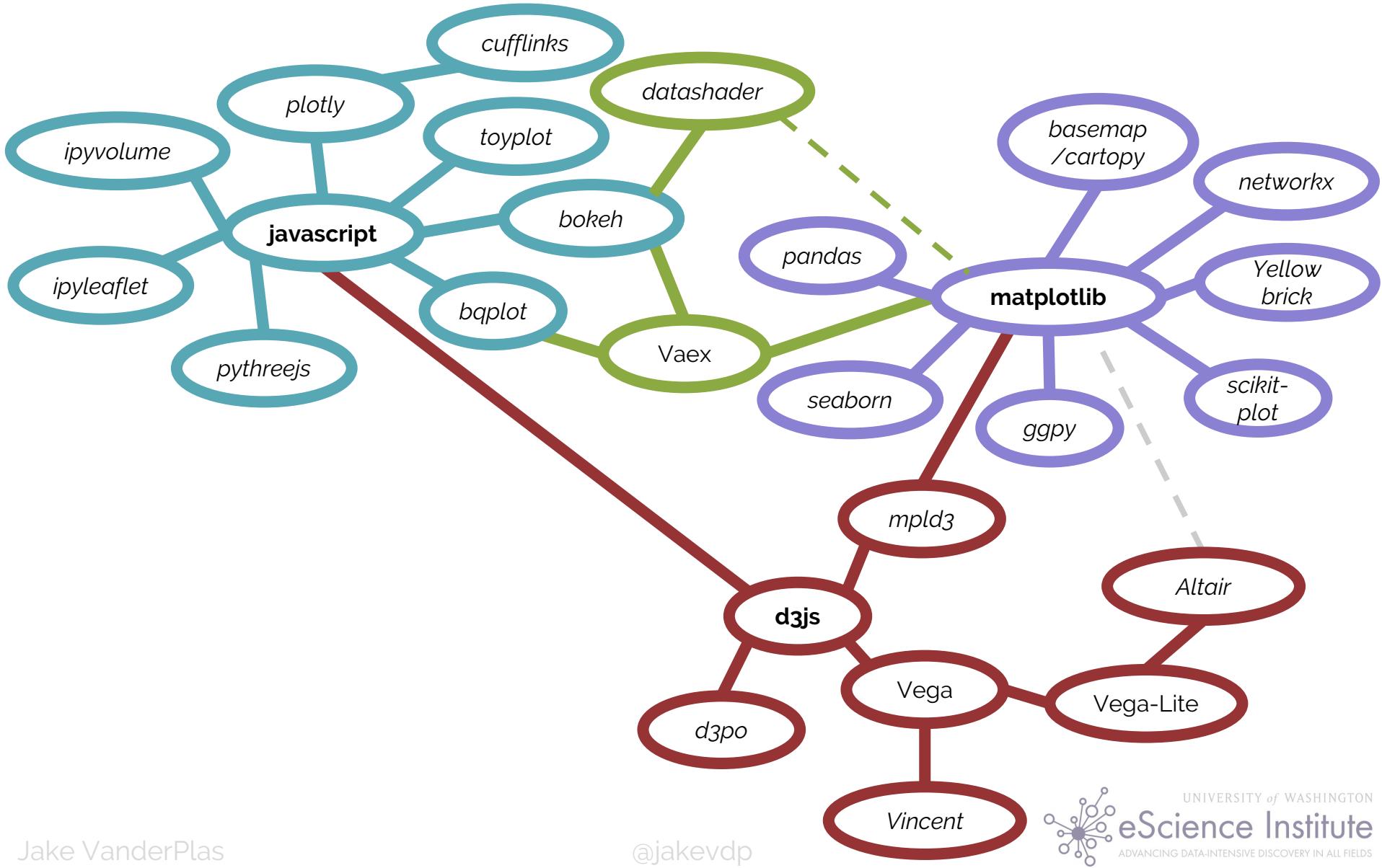


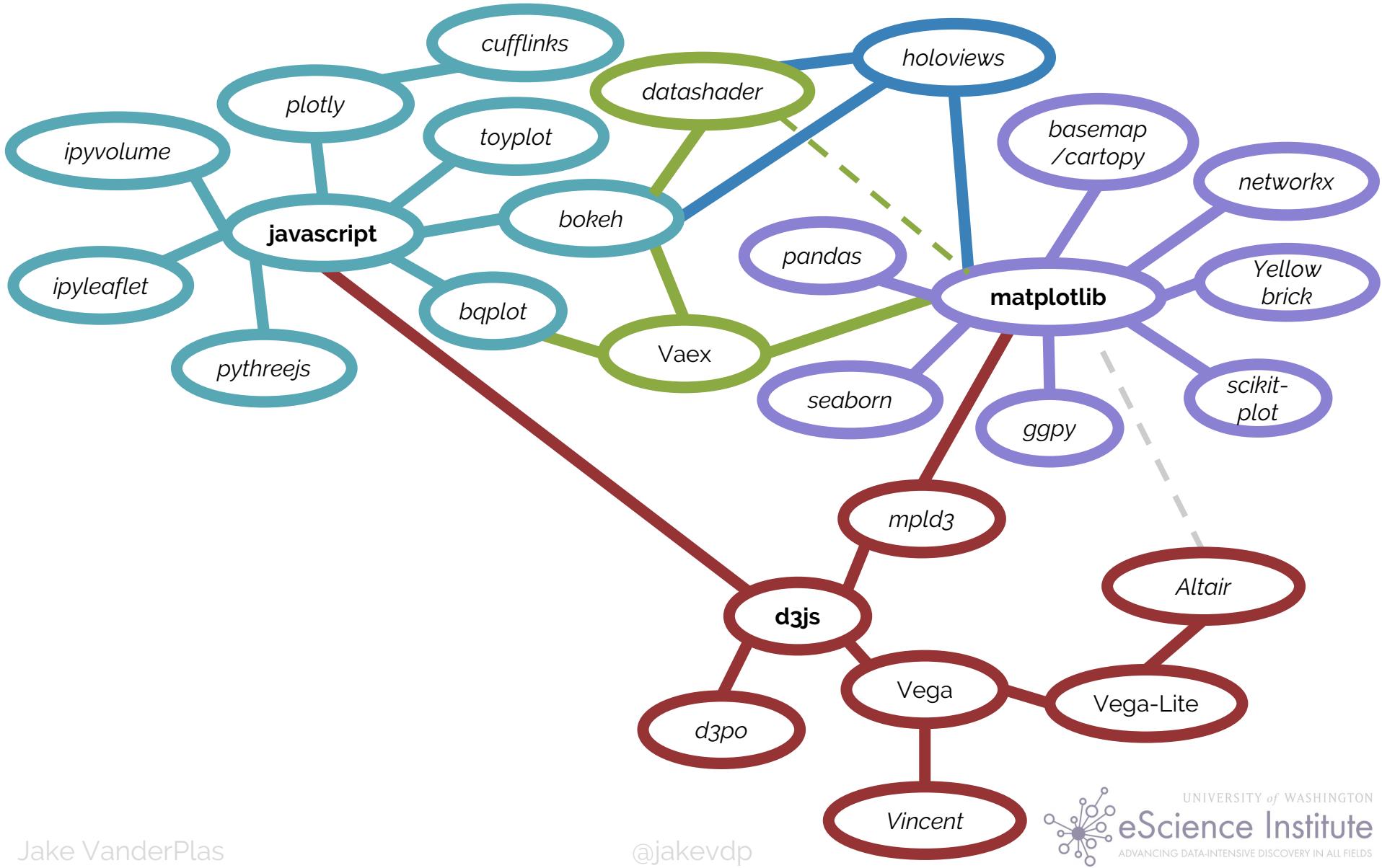


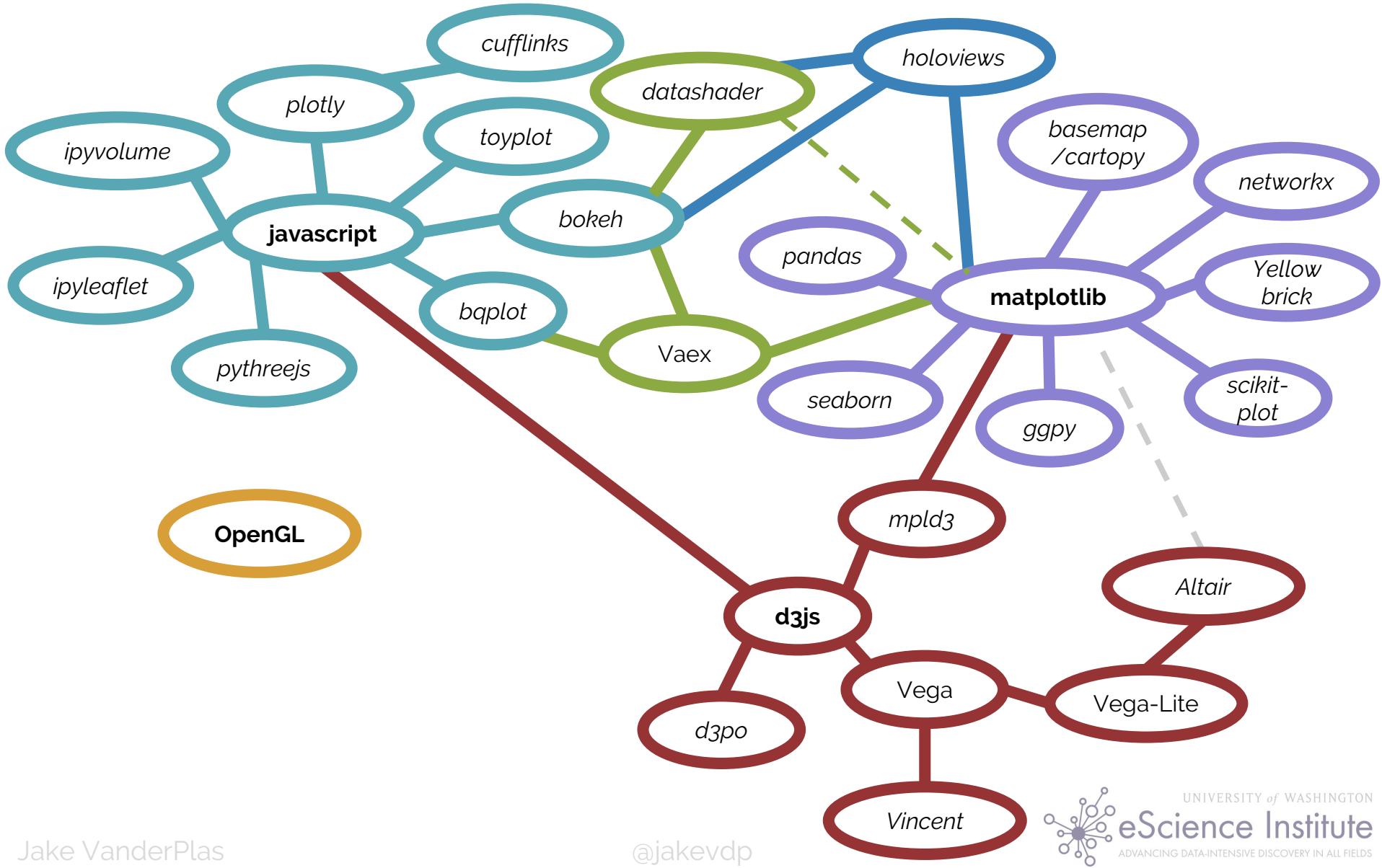


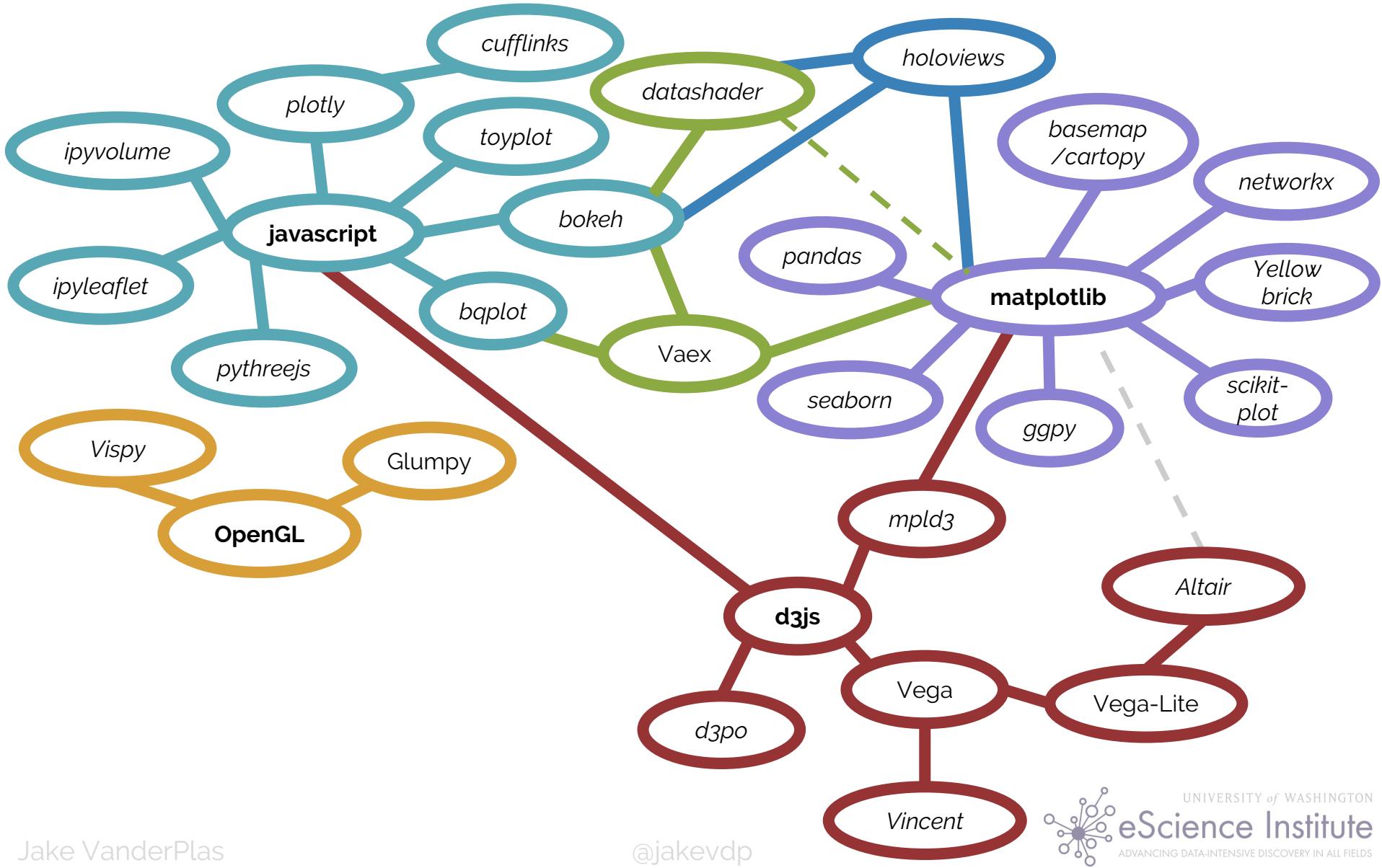


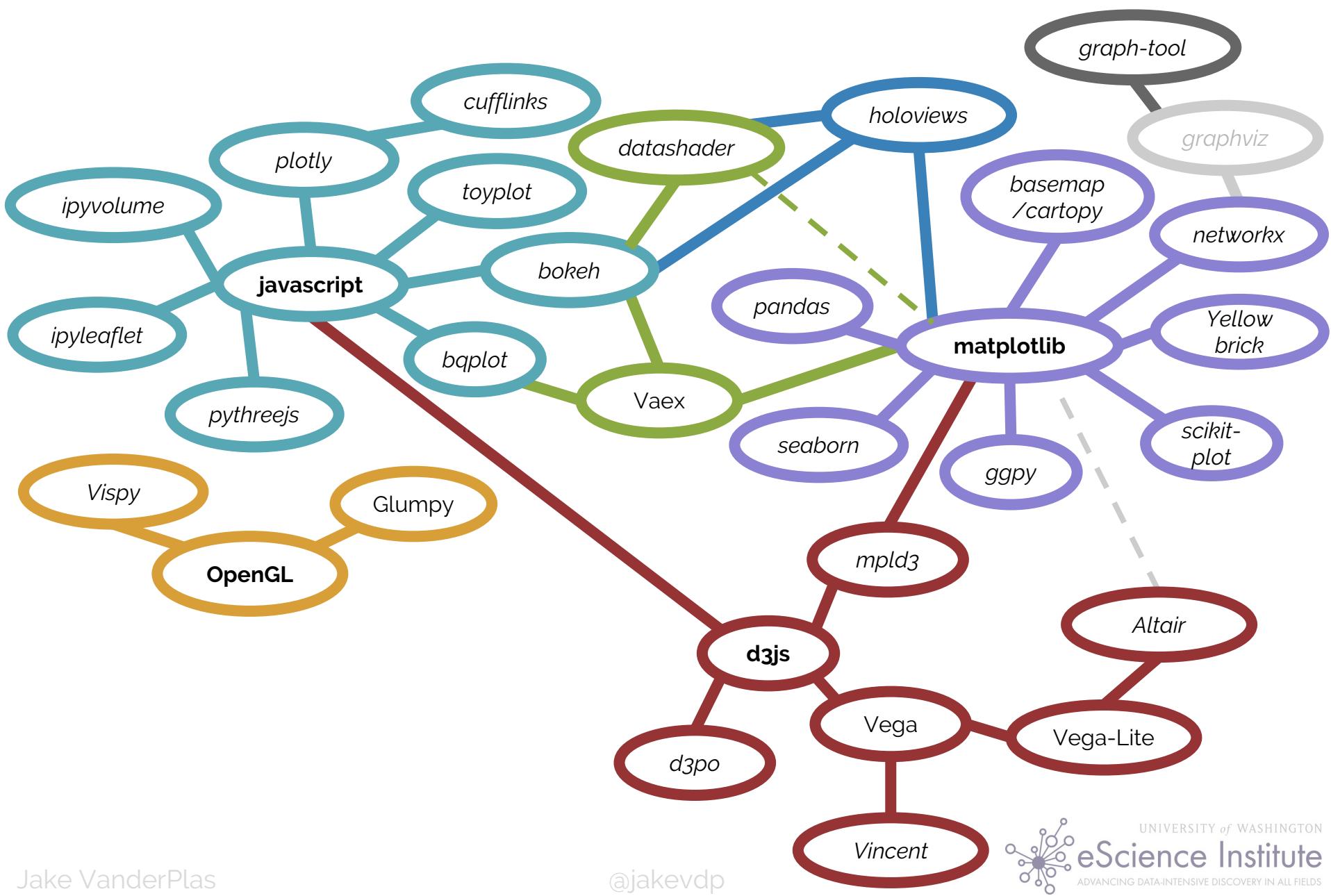


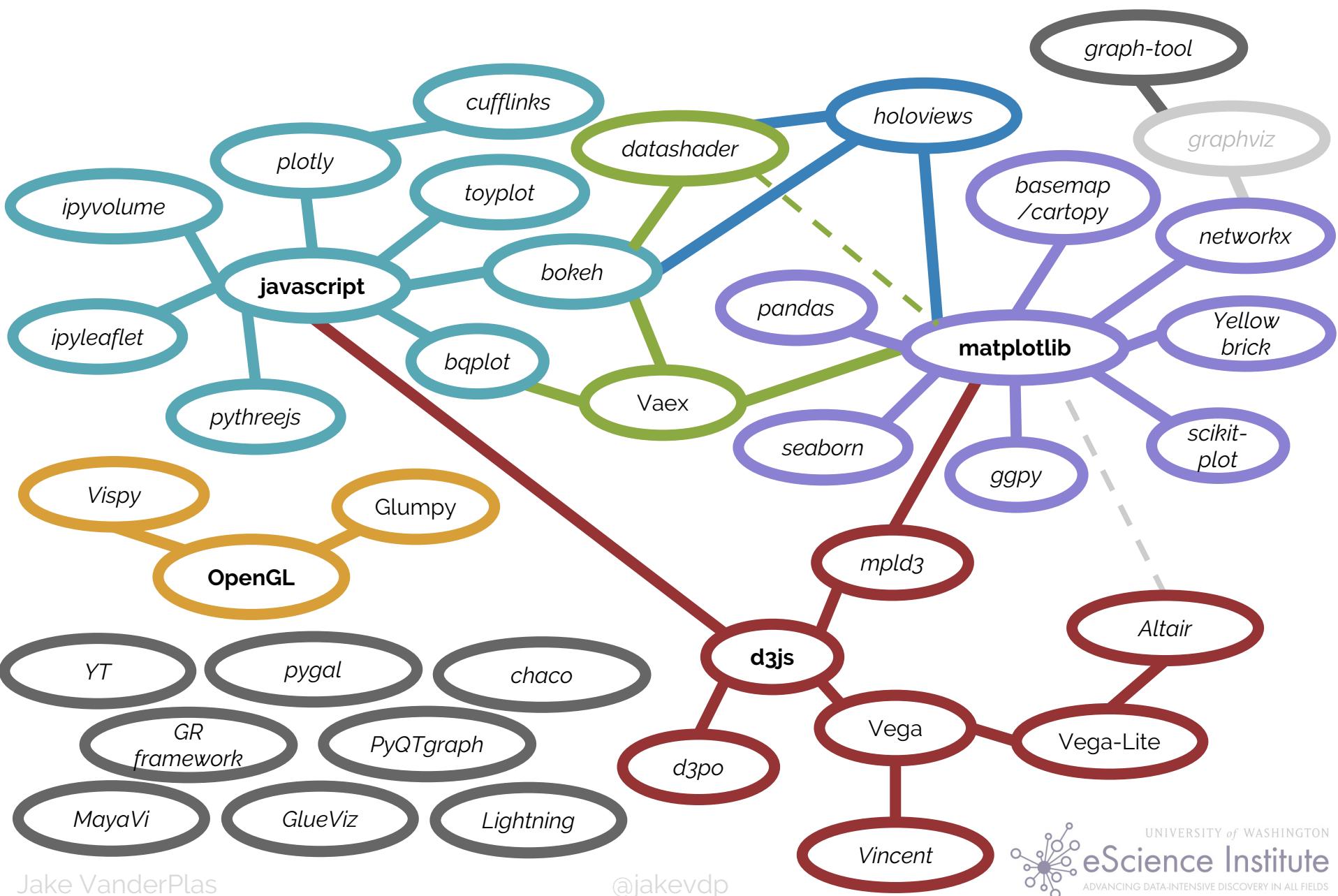




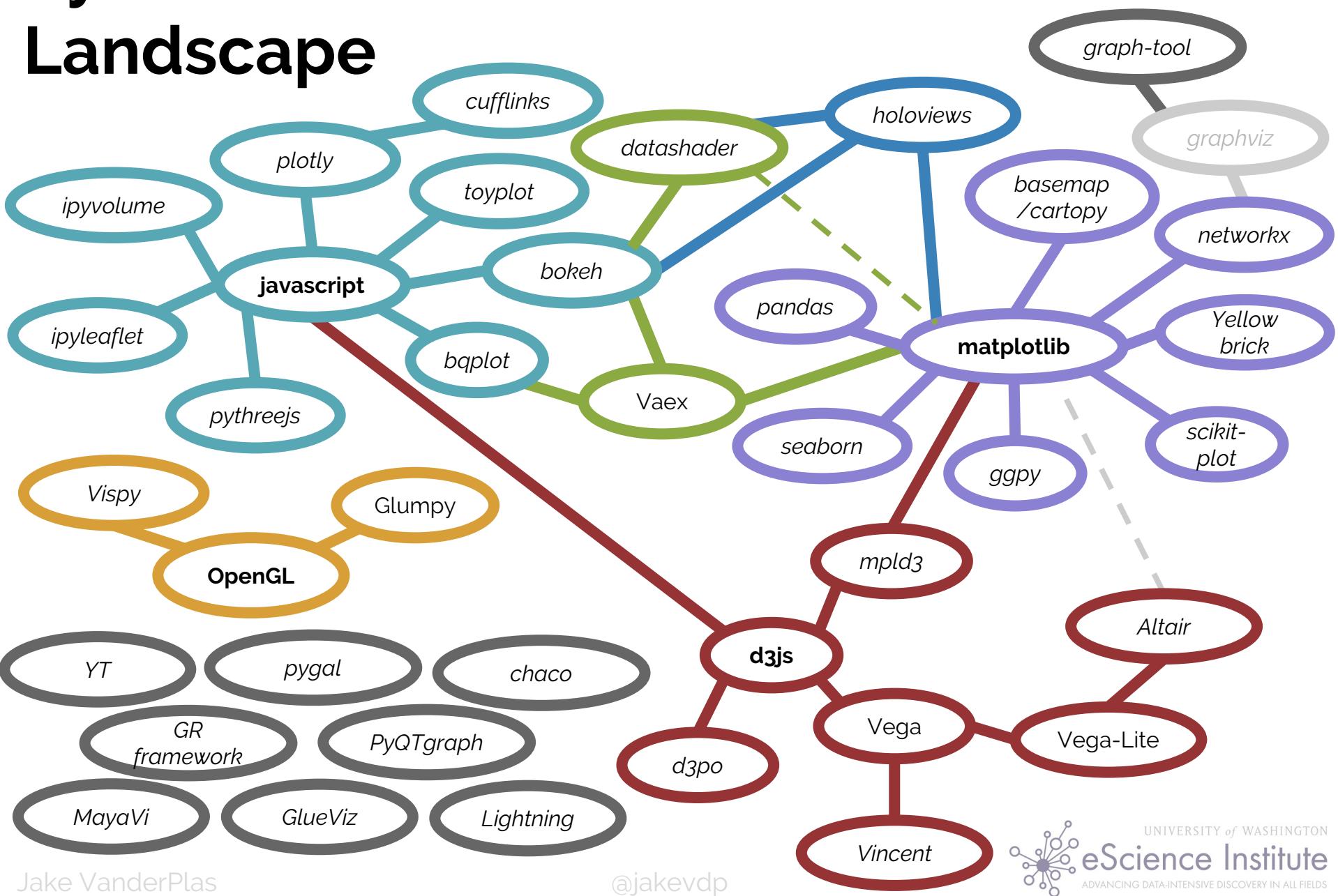








Python's Visualization Landscape





How did we get here?

In the beginning was matplotlib*

* well, actually... Python visualization existed before matplotlib, but was not very mature.

Plotting with Matplotlib

Strengths:

- Designed like MatLab: switching was easy



Plotting with Matplotlib

Strengths:

- Designed like MatLab: switching was easy
- Many rendering backends

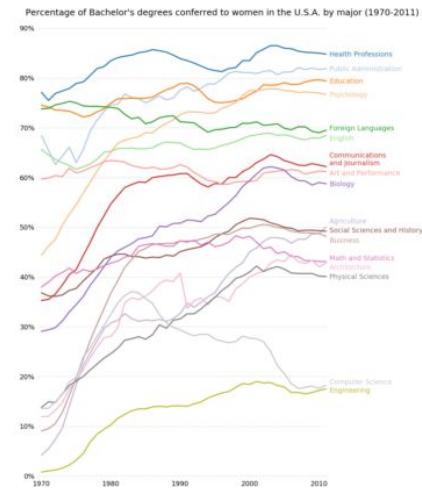
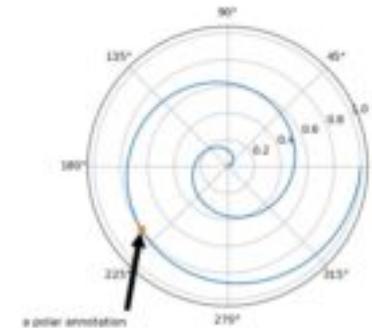
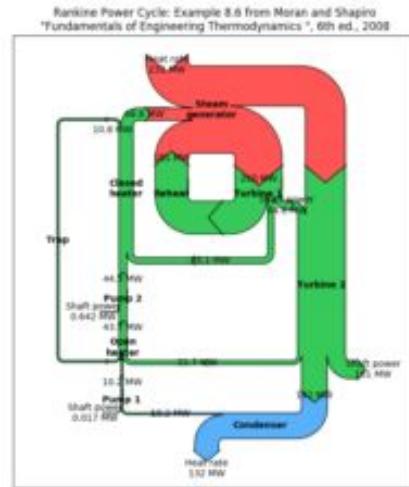
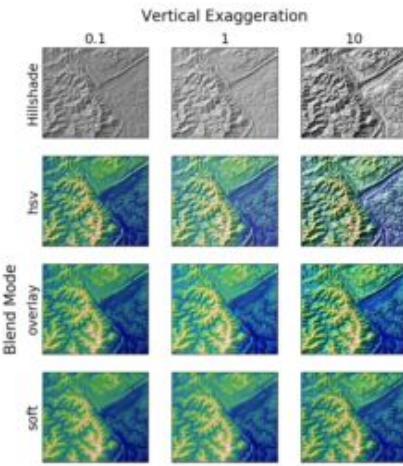
```
In [26]: from matplotlib import rcsetup
rcsetup.all_backends

Out[26]: ['GTK',
           'GTKAgg',
           'GTKCairo',
           'MacOSX',
           'Qt4Agg',
           'Qt5Agg',
           'TkAgg',
           'WX',
           'WXAgg',
           'GTK3Cairo',
           'GTK3Agg',
           'WebAgg',
           'nbAgg',
           'agg',
           'cairo',
           'gdk',
           'pdf',
           'pgf',
           'ps',
           'svg',
           'template']
```

Plotting with Matplotlib

Strengths:

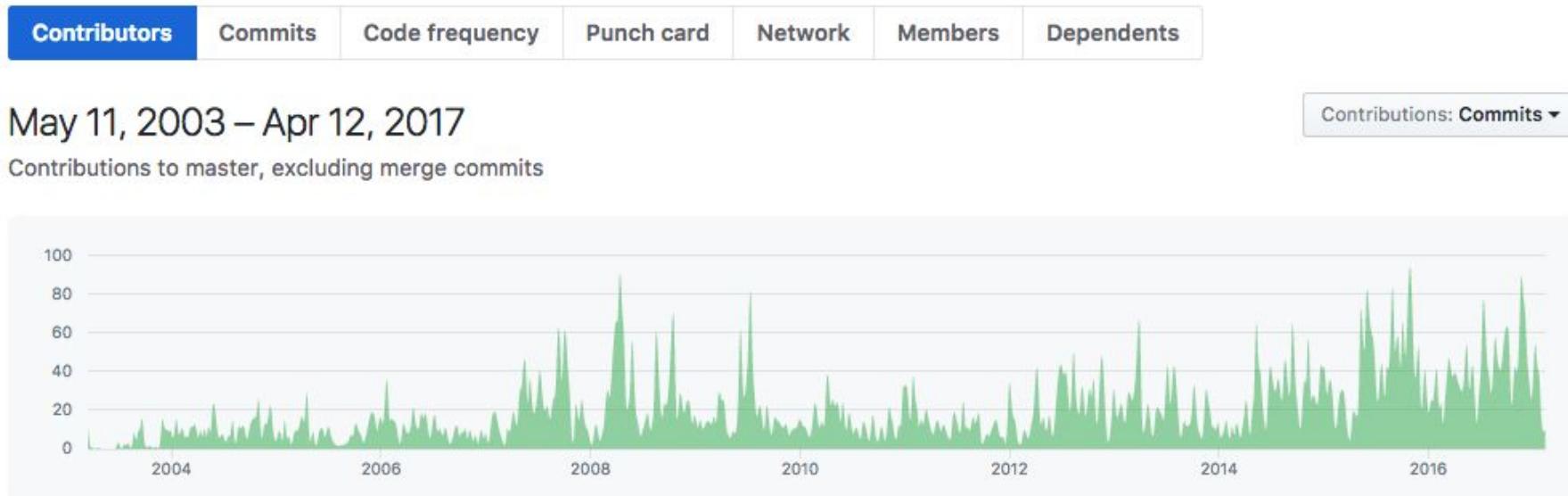
- Designed like MatLab: switching was easy
- Many rendering backends
- Can reproduce just about any plot (with a bit of effort)



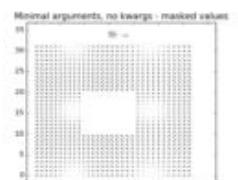
Plotting with Matplotlib

Strengths:

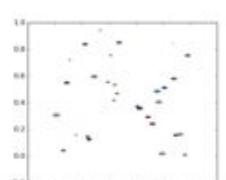
- Designed like MatLab: switching was easy
- Many rendering backends
- Can reproduce just about any plot (with a bit of effort)
- Well-tested, standard tool for over a decade



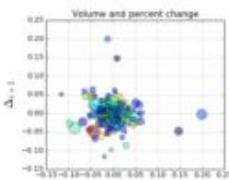
Matplotlib Gallery



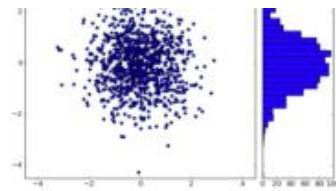
quiver_demo



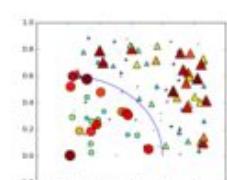
scatter_custom_symbol



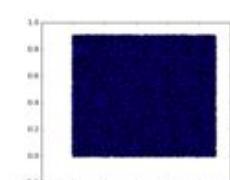
scatter_demo2



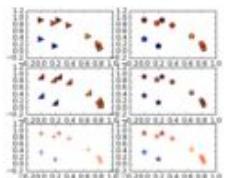
scatter_hist



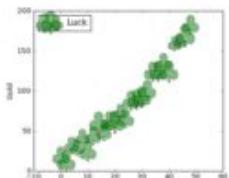
scatter_masked



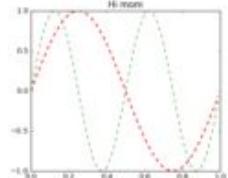
scatter_profile



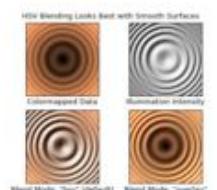
scatter_star_poly



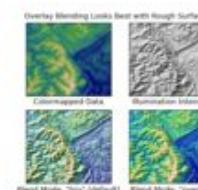
scatter_symbol



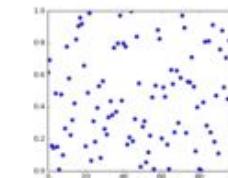
set_and_get



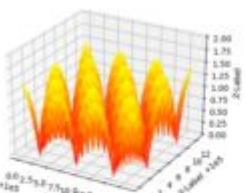
shading_example



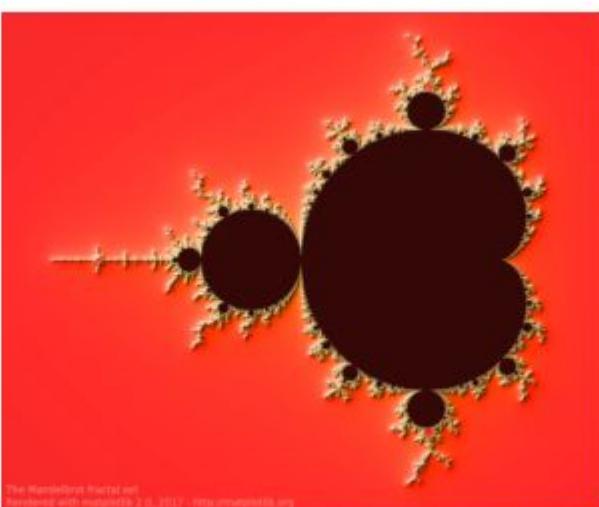
shading_example



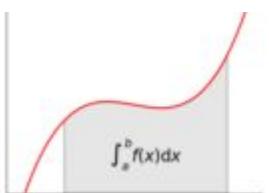
shared_axis_across_figures



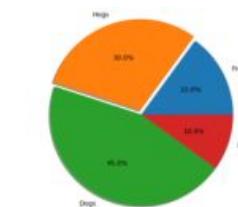
offset_demo



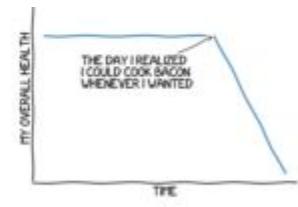
mandelbrot



integral_demo

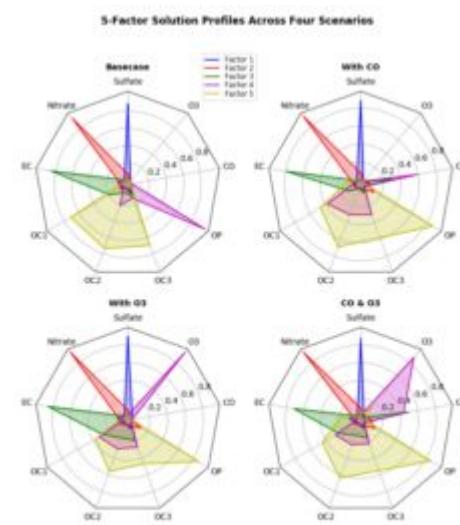


svg_filter_pie



STOVE OWNERSHIP FROM XKCD BY RANDALL MONROE

xkcd



radar_chart



UNIVERSITY of WASHINGTON
eScience Institute
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

Example: Statistical Data

```
import pandas as pd  
iris = pd.read_csv('iris.csv')  
iris.head()
```

	petalLength	petalWidth	sepalLength	sepalWidth	species
0	1.4	0.2	5.1	3.5	setosa
1	1.4	0.2	4.9	3.0	setosa
2	1.3	0.2	4.7	3.2	setosa
3	1.5	0.2	4.6	3.1	setosa
4	1.4	0.2	5.0	3.6	setosa

Tidy data: i.e. rows are samples, columns are features

Just a simple visualization . . .

*"I want to scatter petal length vs.
sepal length, and color by species"*

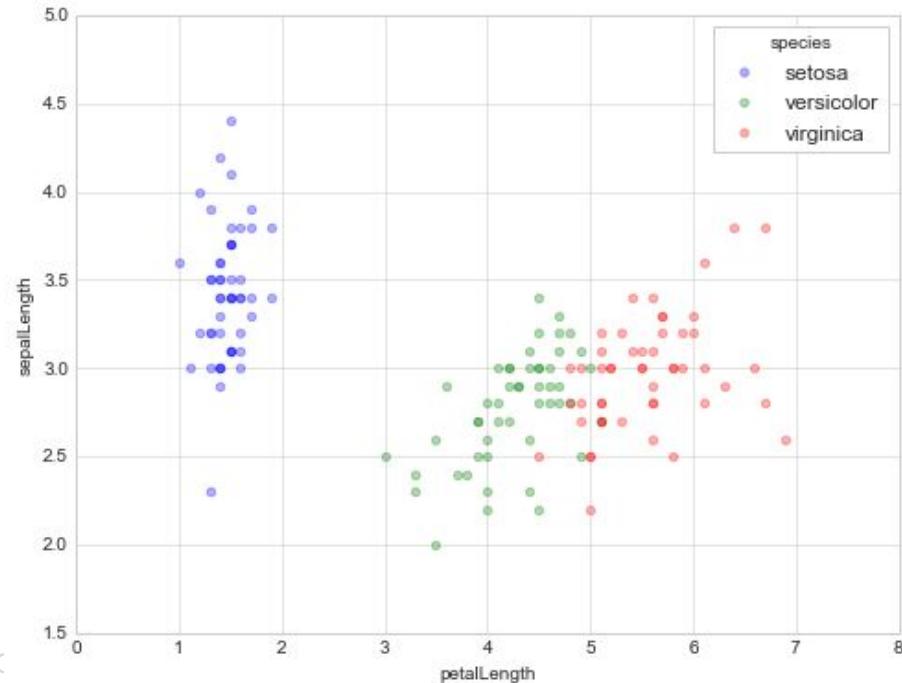
	petalLength	petalWidth	sepalLength	sepalWidth	species
0	1.4	0.2	5.1	3.5	setosa
1	1.4	0.2	4.9	3.0	setosa
2	1.3	0.2	4.7	3.2	setosa
3	1.5	0.2	4.6	3.1	setosa
4	1.4	0.2	5.0	3.6	setosa

Just a simple visualization . . .

```
color_map = dict(zip(iris.species.unique(),
                     ['blue', 'green', 'red']))

for species, group in iris.groupby('species'):
    plt.scatter(group['petalLength'], group['sepalLength'],
                color=color_map[species],
                alpha=0.3, edgecolor=None,
                label=species)

plt.legend(frameon=True, title='species')
plt.xlabel('petalLength')
plt.ylabel('sepalLength')
```



Plotting with Matplotlib

Strengths:

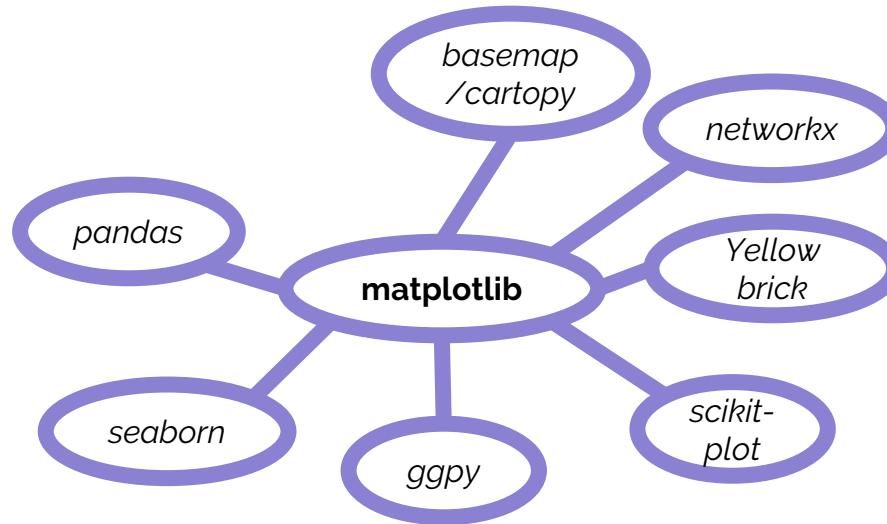
- Designed like MatLab: switching was easy
- Many rendering backends
- Can reproduce just about any plot with a bit of effort
- Well-tested, standard tool for over a decade

Weaknesses:

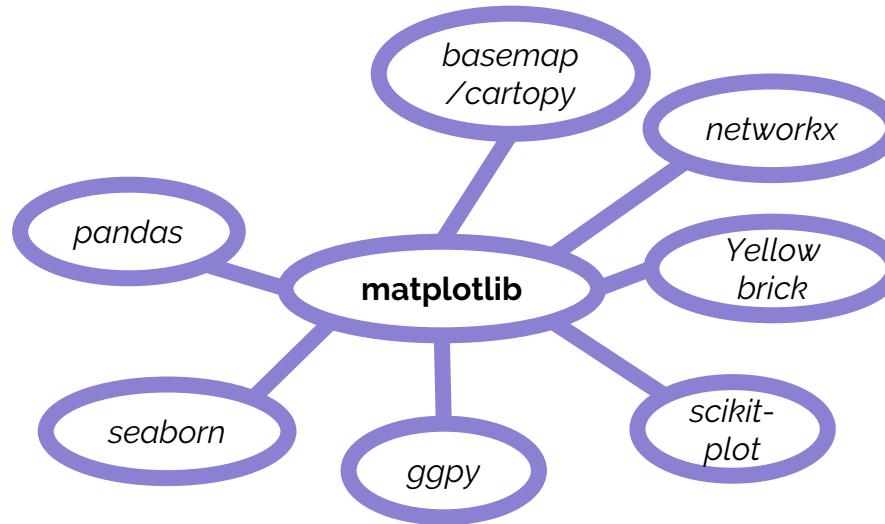
- API is imperative & often overly verbose
- Sometimes poor stylistic defaults
- Poor support for web/interactive graphs
- Often slow for large & complicated data

Everyone's Goal:
Improve on the weaknesses of matplotlib
(without sacrificing the strengths!)

Building on Matplotlib . . .

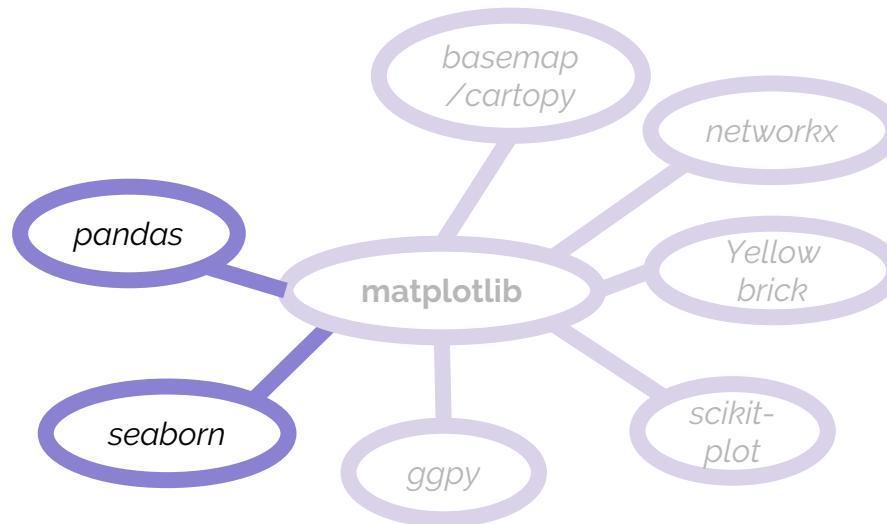


Building on Matplotlib . . .



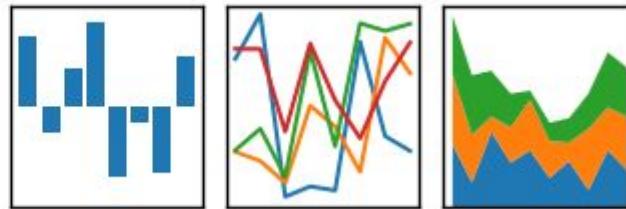
Common Idea: Keep matplotlib as a **versatile, well-tested backend**, and provide a new domain-specific API.

Building on Matplotlib . . .



pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

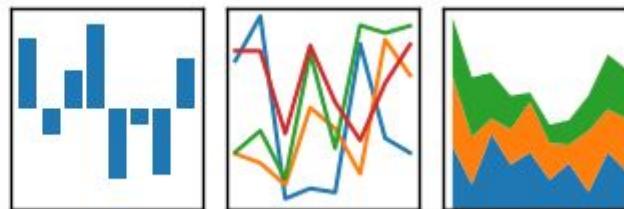


Key Features:

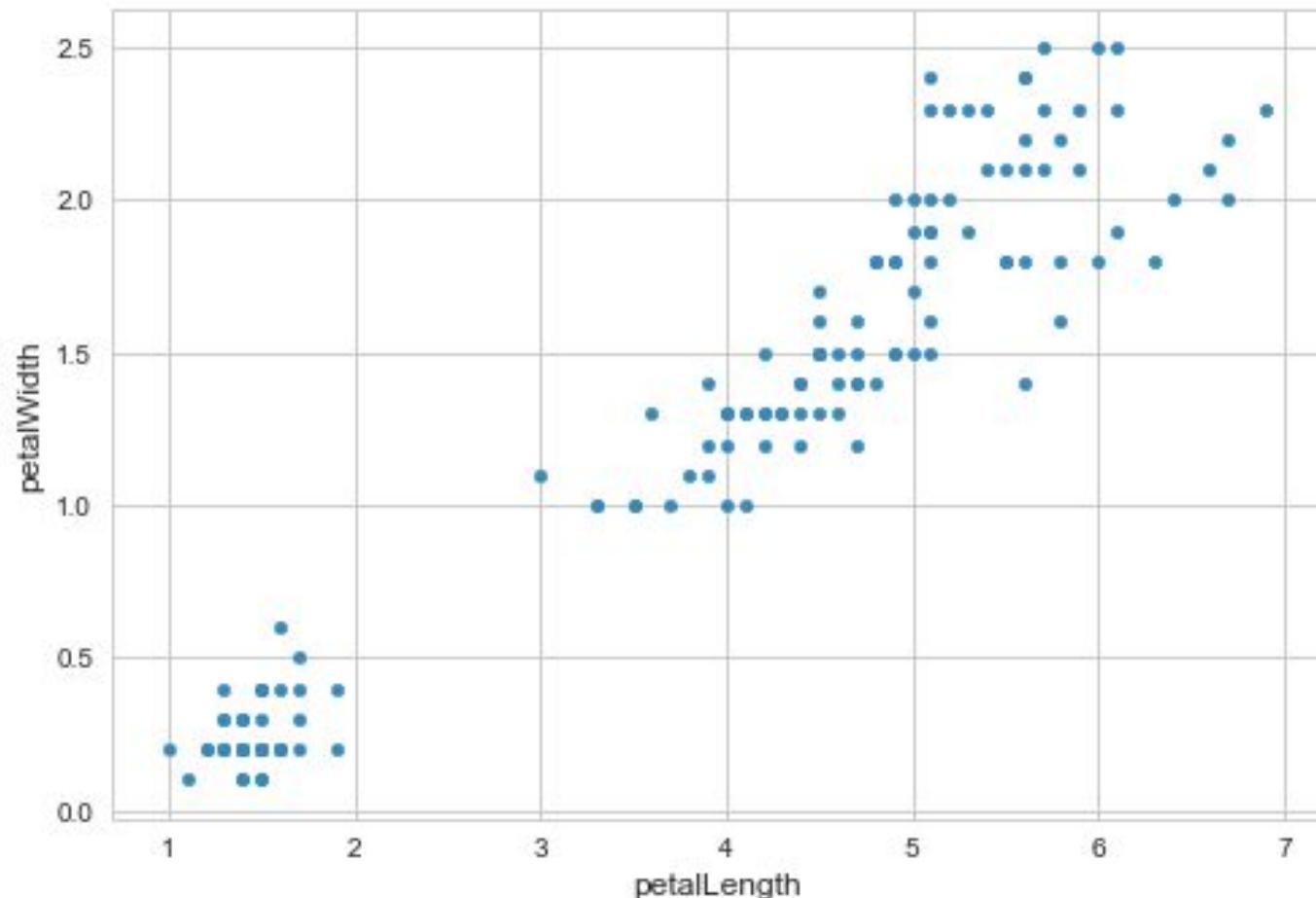
- Pandas provides a **DataFrame** object
- Also provides a simple API for plotting DataFrames

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

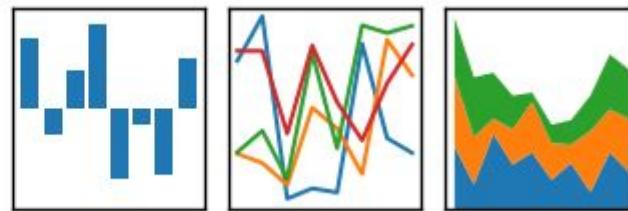


```
iris.plot.scatter('petalLength', 'petalWidth')
```



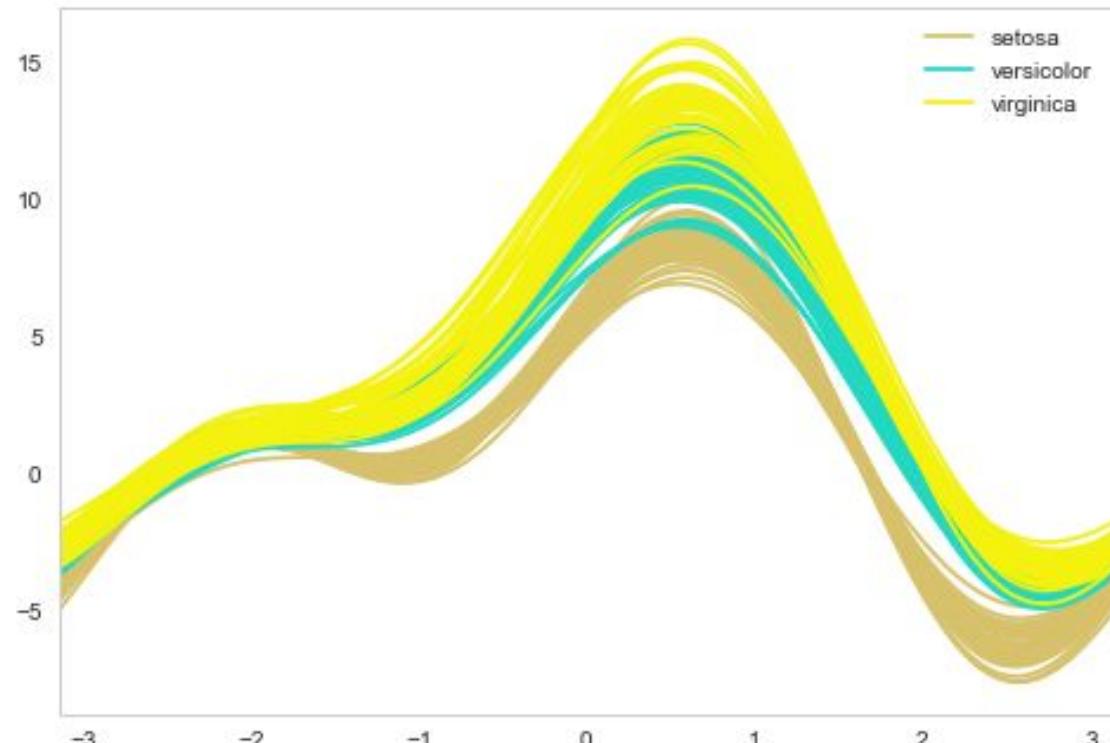
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

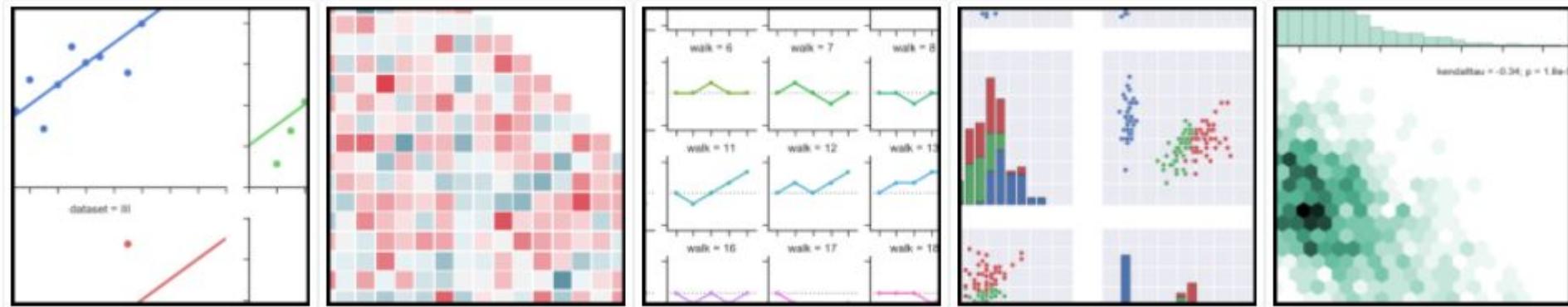


- More sophisticated statistical visualization tools have recently been added

```
from pandas.tools.plotting import andrews_curves  
andrews_curves(iris, 'species')
```



Seaborn: statistical data visualization



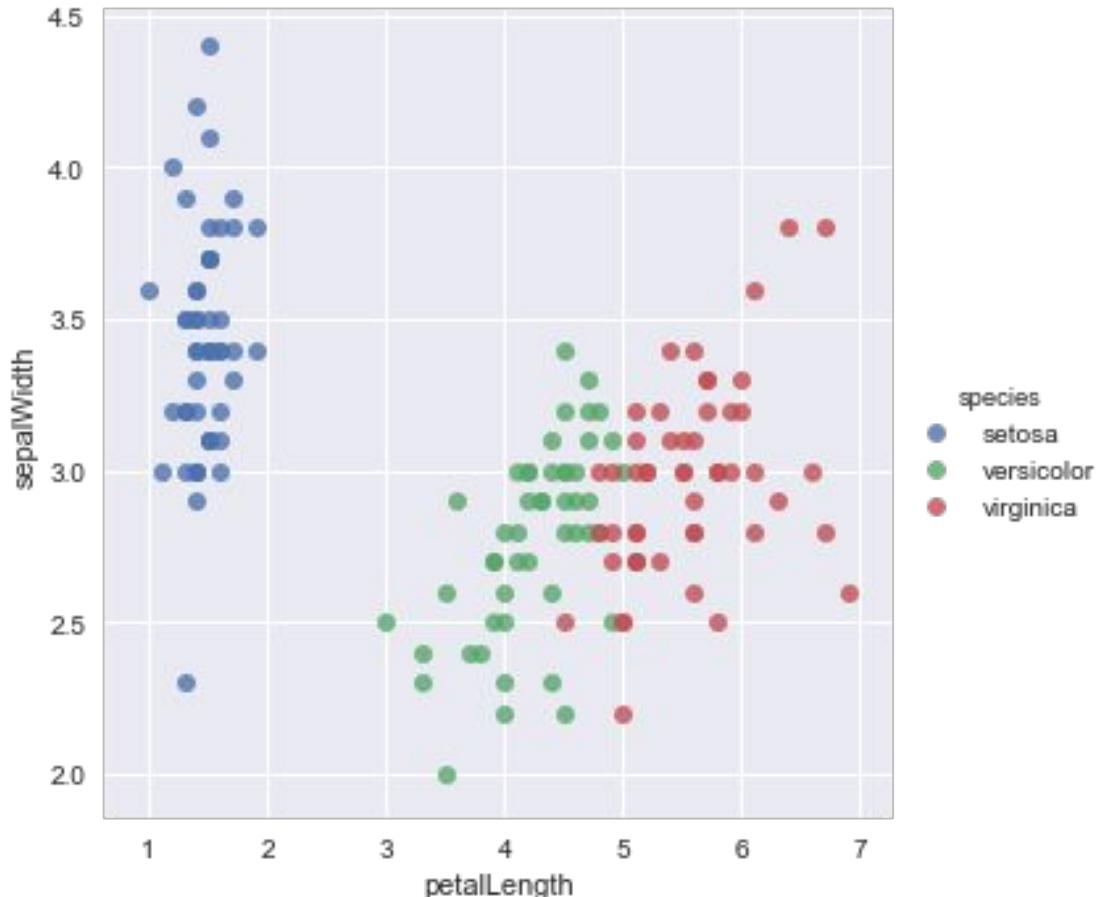
Key Features:

- Like Pandas, wraps matplotlib
- Nice set of color palettes & plot styles
- Focus on statistical visualization & modeling

<http://seaborn.pydata.org>

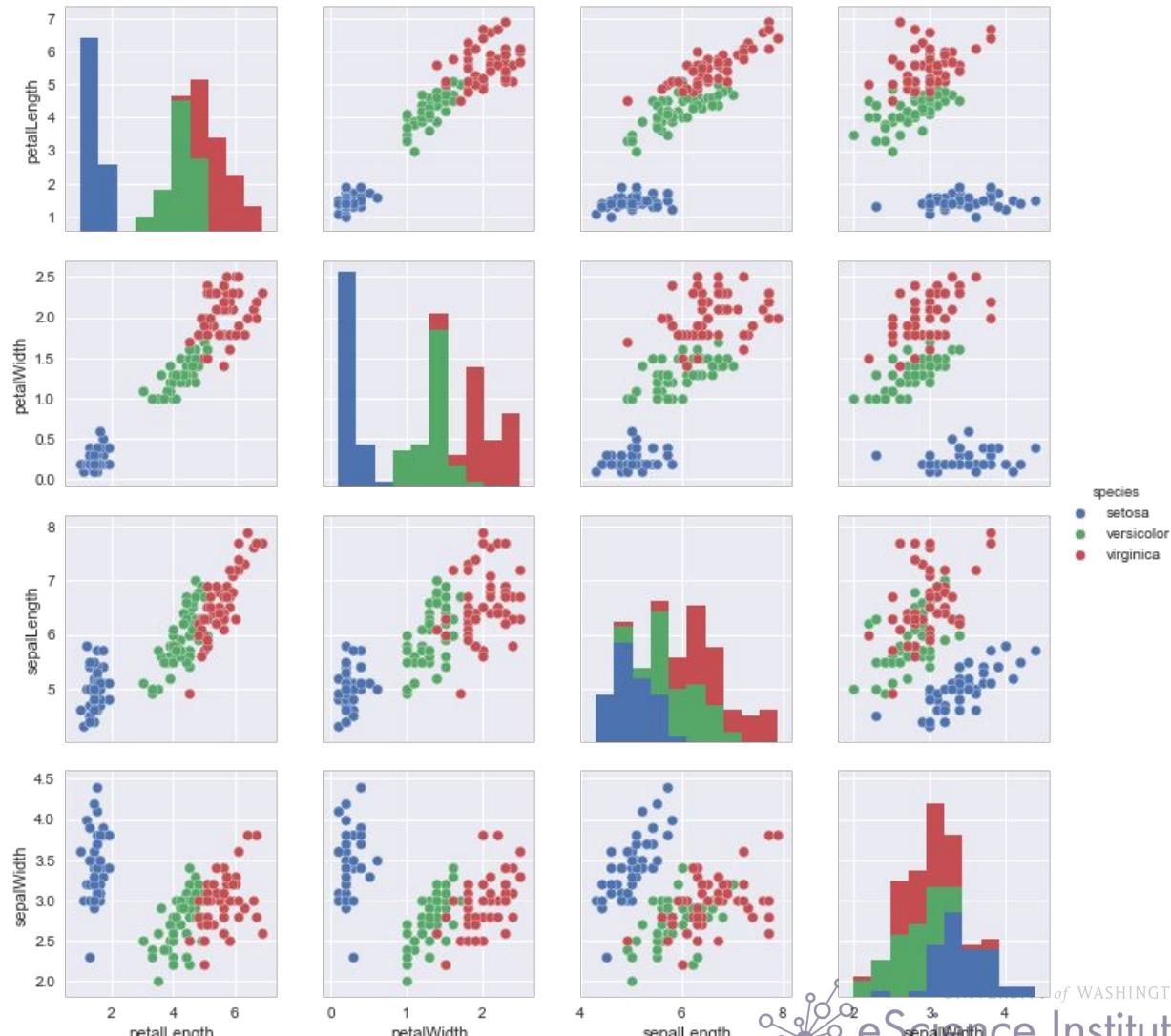
Seaborn examples

```
import seaborn as sns  
sns.lmplot('petalLength', 'sepalWidth', iris,  
            hue='species', fit_reg=False)
```

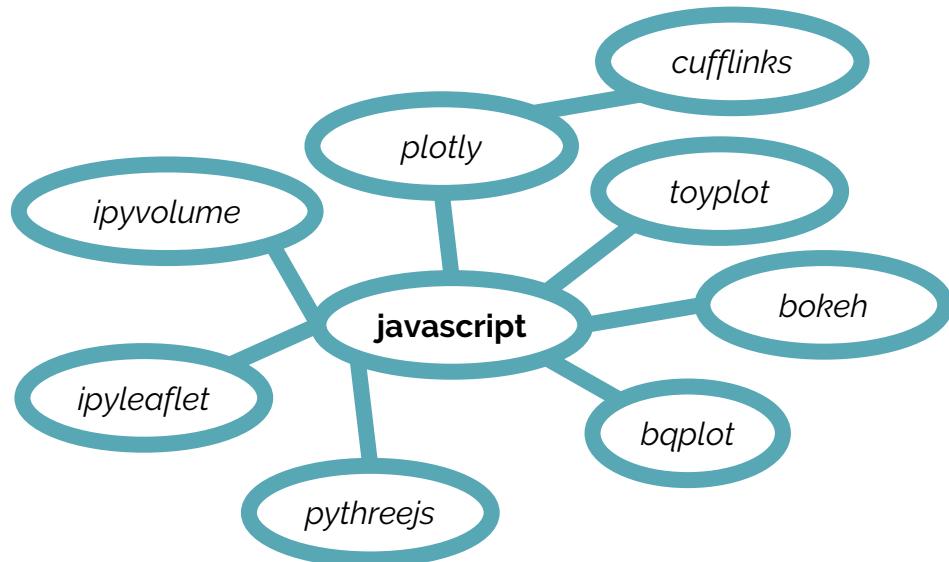


Seaborn examples

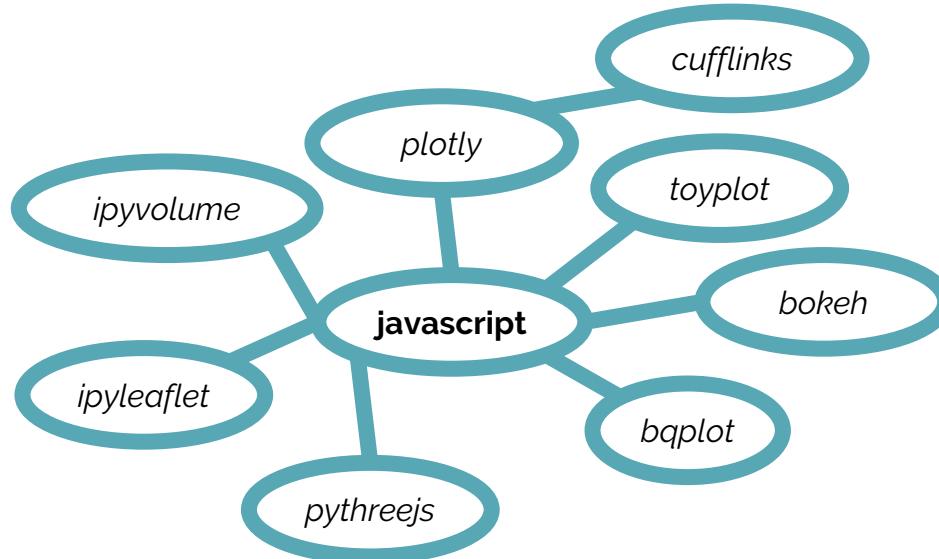
```
sns.pairplot(iris, hue='species')
```



Javascript-based Viz:

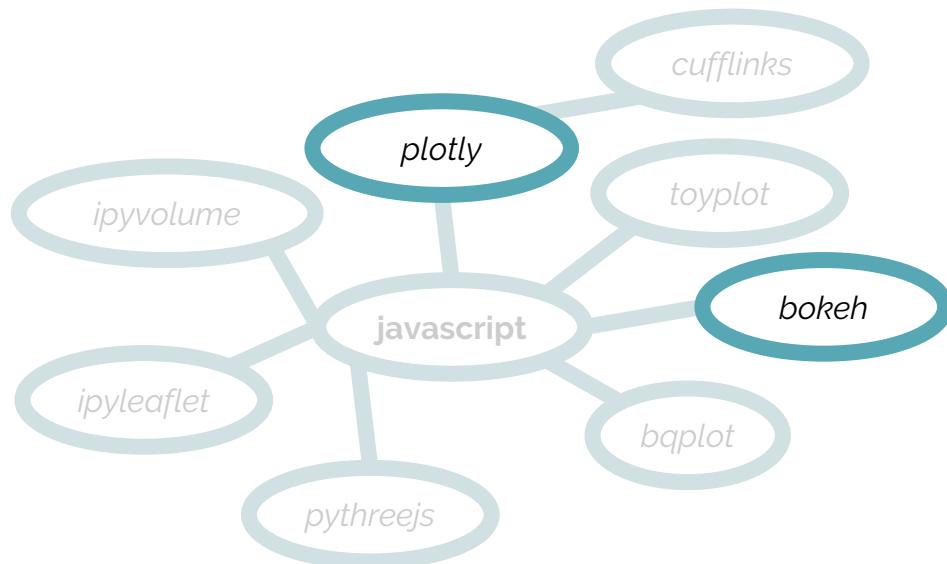


Javascript-based Viz:



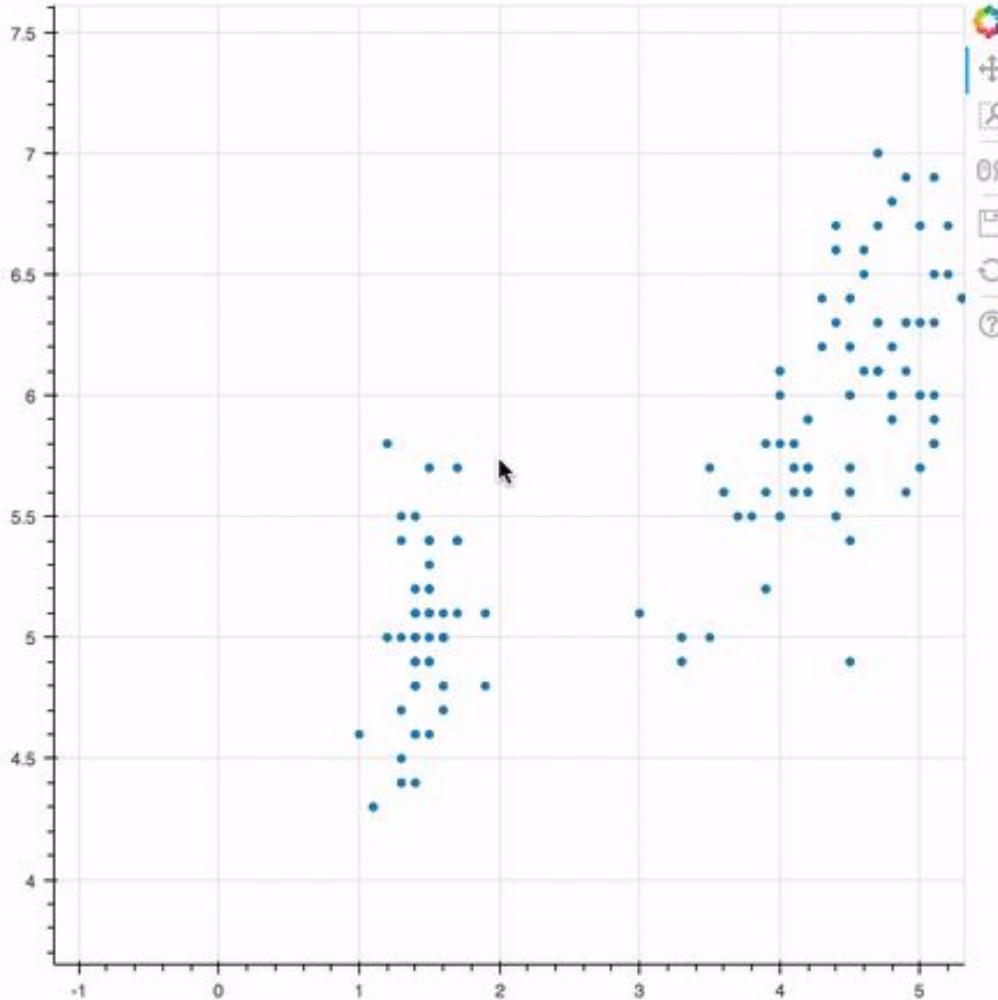
Common Idea: build a new API that produces a plot serialization (often JSON) that can be displayed in the browser (often in Jupyter notebooks)

Javascript-based Viz:

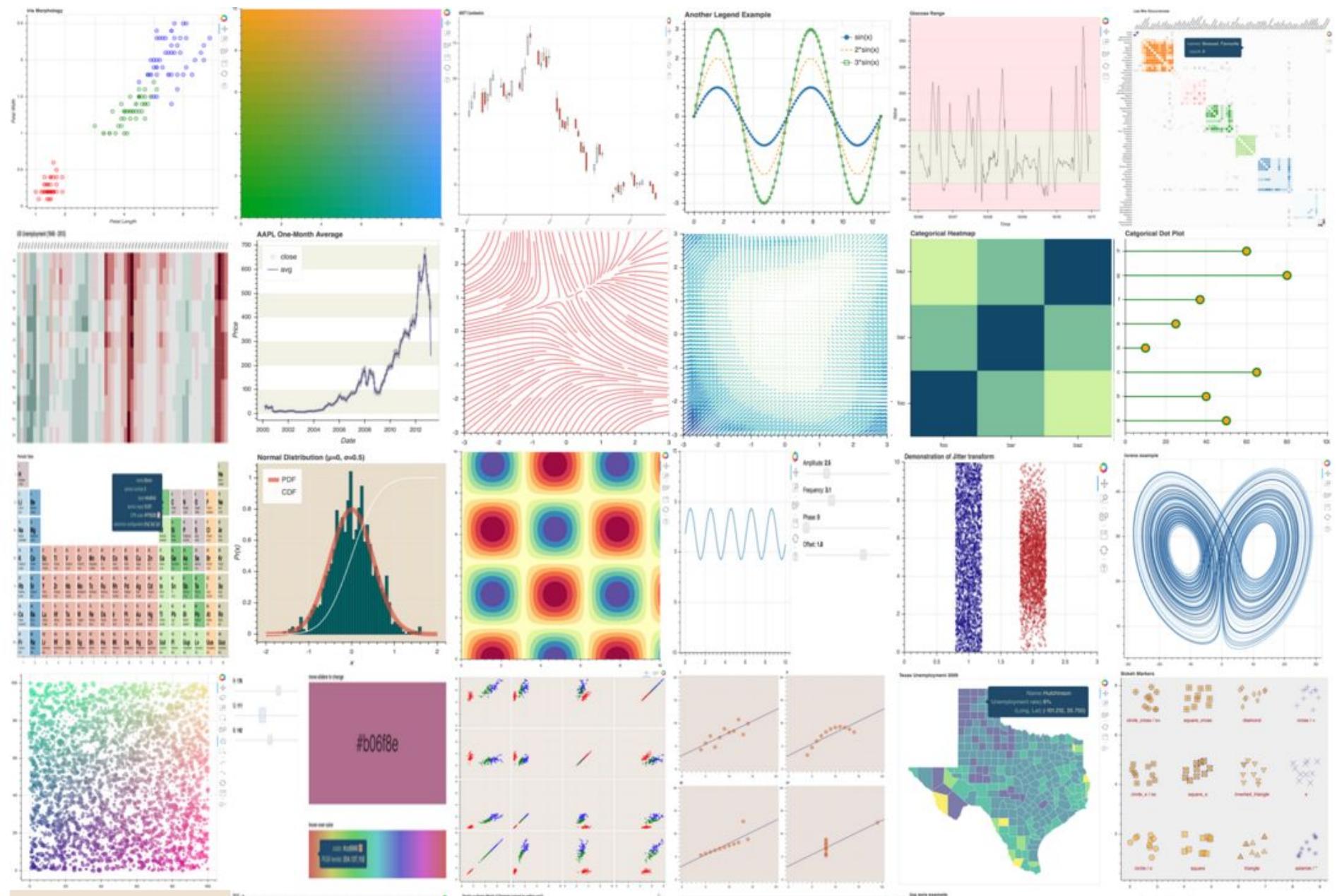


Plotting with Bokeh

```
In [10]: p = figure()  
p.circle(iris.petalLength, iris.sepalLength)  
show(p)
```



Bokeh Gallery



Plotting with Bokeh

Advantages:

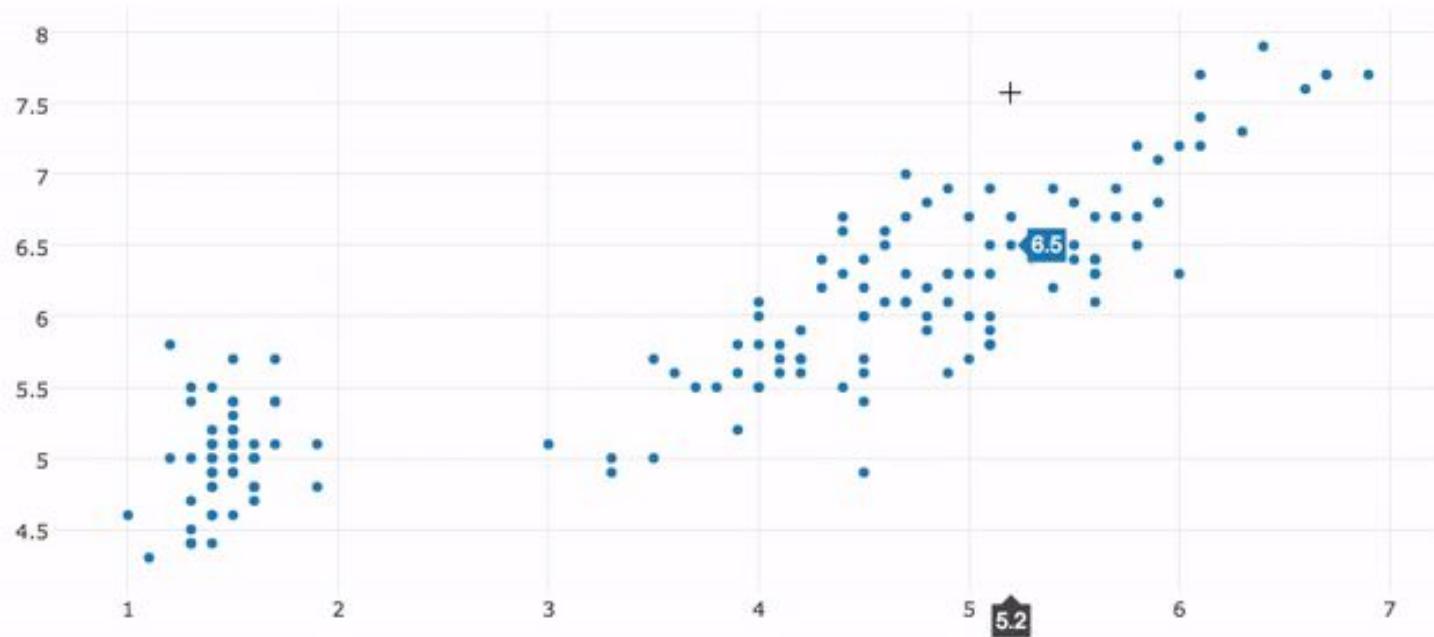
- Web view/interactivity
- Imperative and Declarative layer
- Handles large and/or streaming datasets
- Geographical visualization
- Fully open source

Disadvantages:

- No vector output (need PDF/EPS? Sorry)
- Newer tool with a smaller user-base than matplotlib

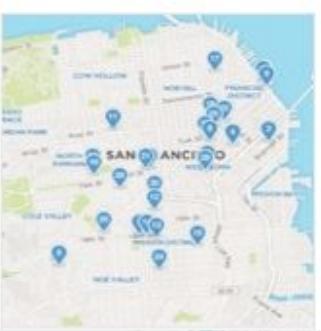
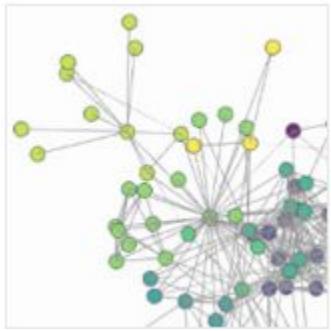
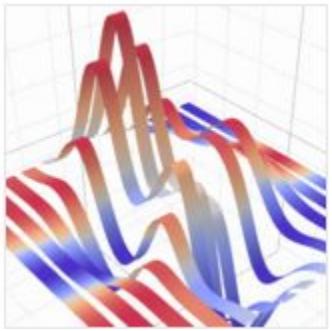
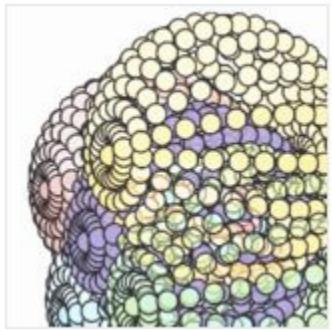
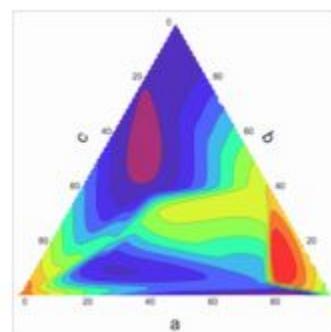
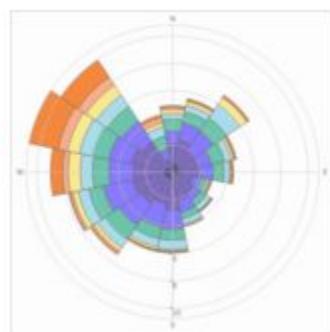
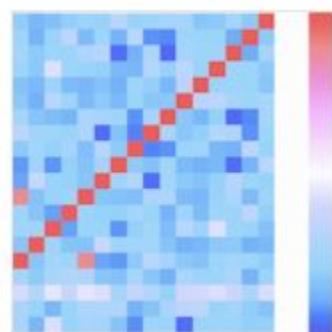
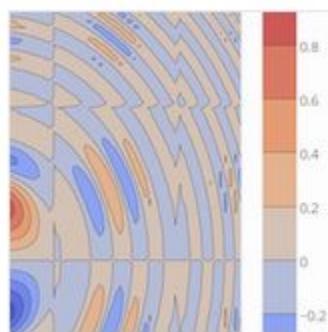
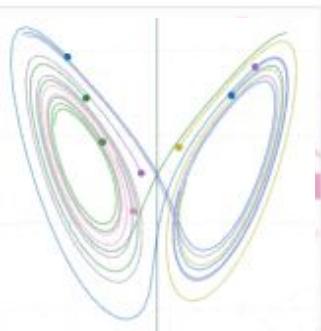
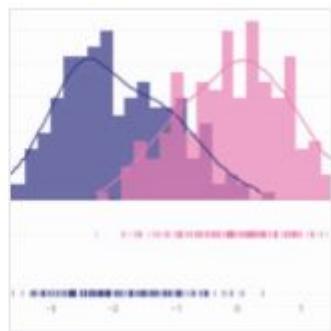
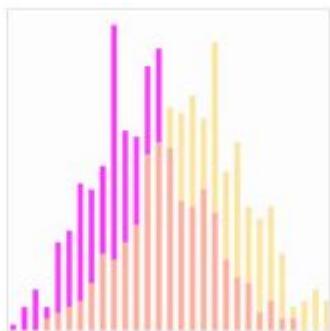
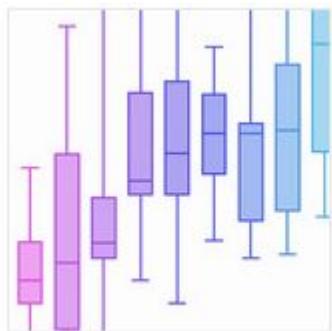
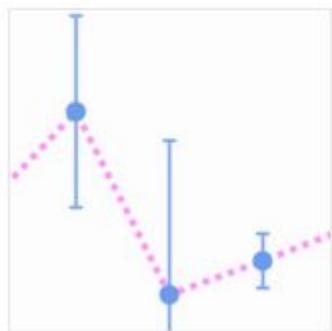
Basic Plotting with Plotly

```
In [8]: from plotly.graph_objs import Scatter  
from plotly.offline import iplot  
  
p = Scatter(x=iris.petalLength,  
            y=iris.sepalLength,  
            mode='markers')  
  
iplot([p])
```



[Export to plot.ly >](#)

Plotly Gallery



Plotting with Plotly

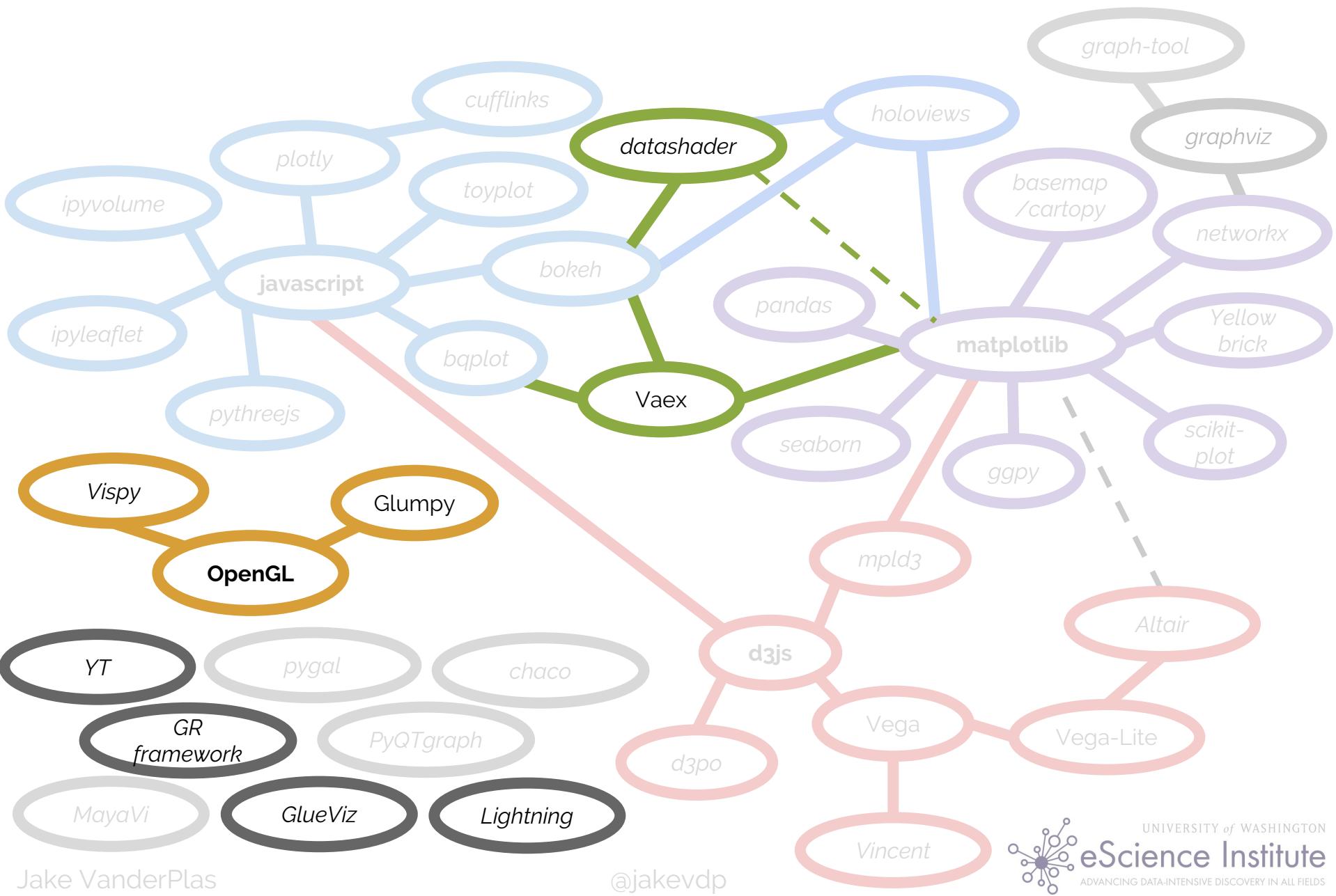
Advantages:

- Web view/interactivity
- Multi-language support
- 3D plotting capability
- Animation capability
- Geographical visualization

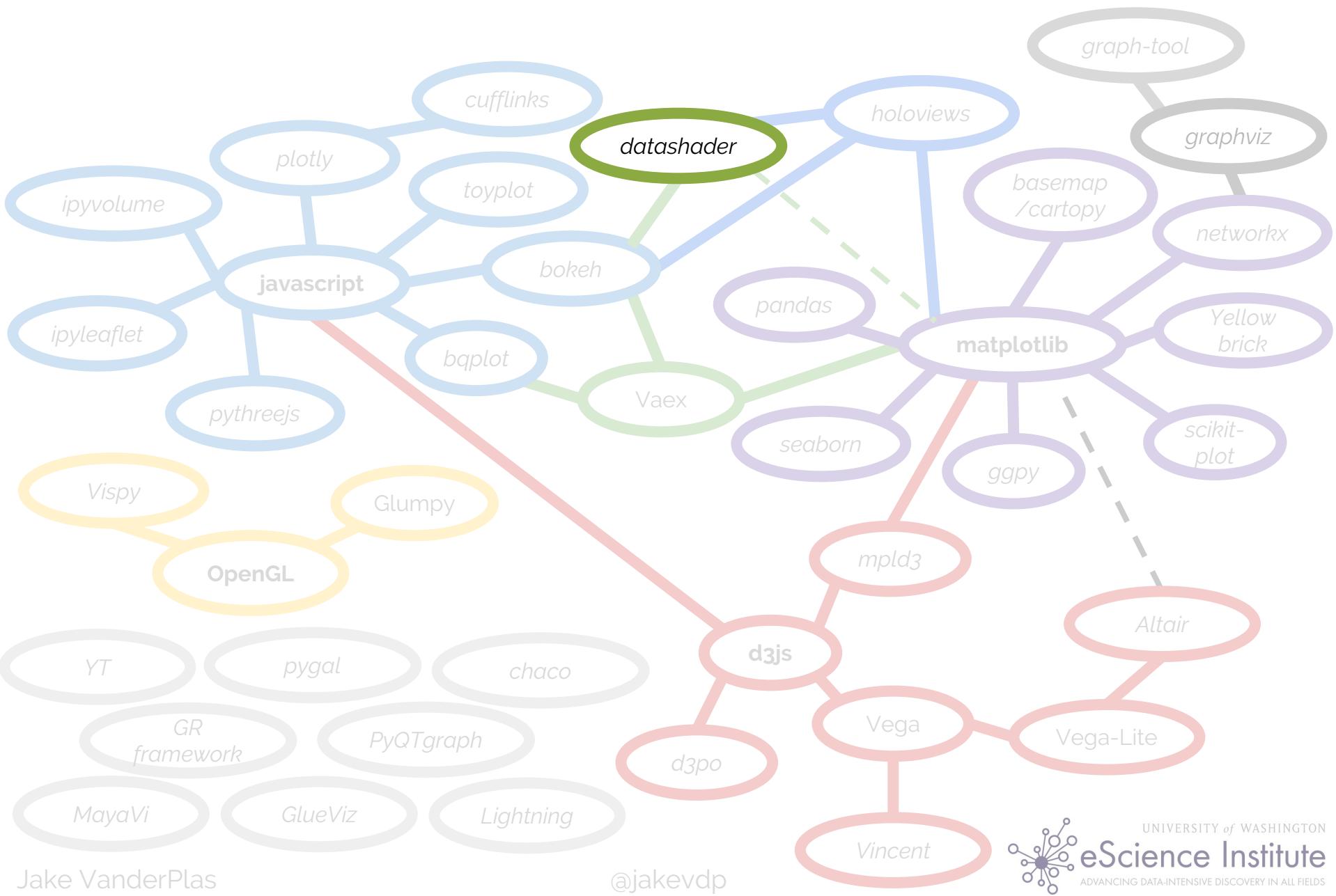
Disadvantages:

- Some features require a paid plan

Visualization for Larger Data . . .



Visualization for Larger Data . . .

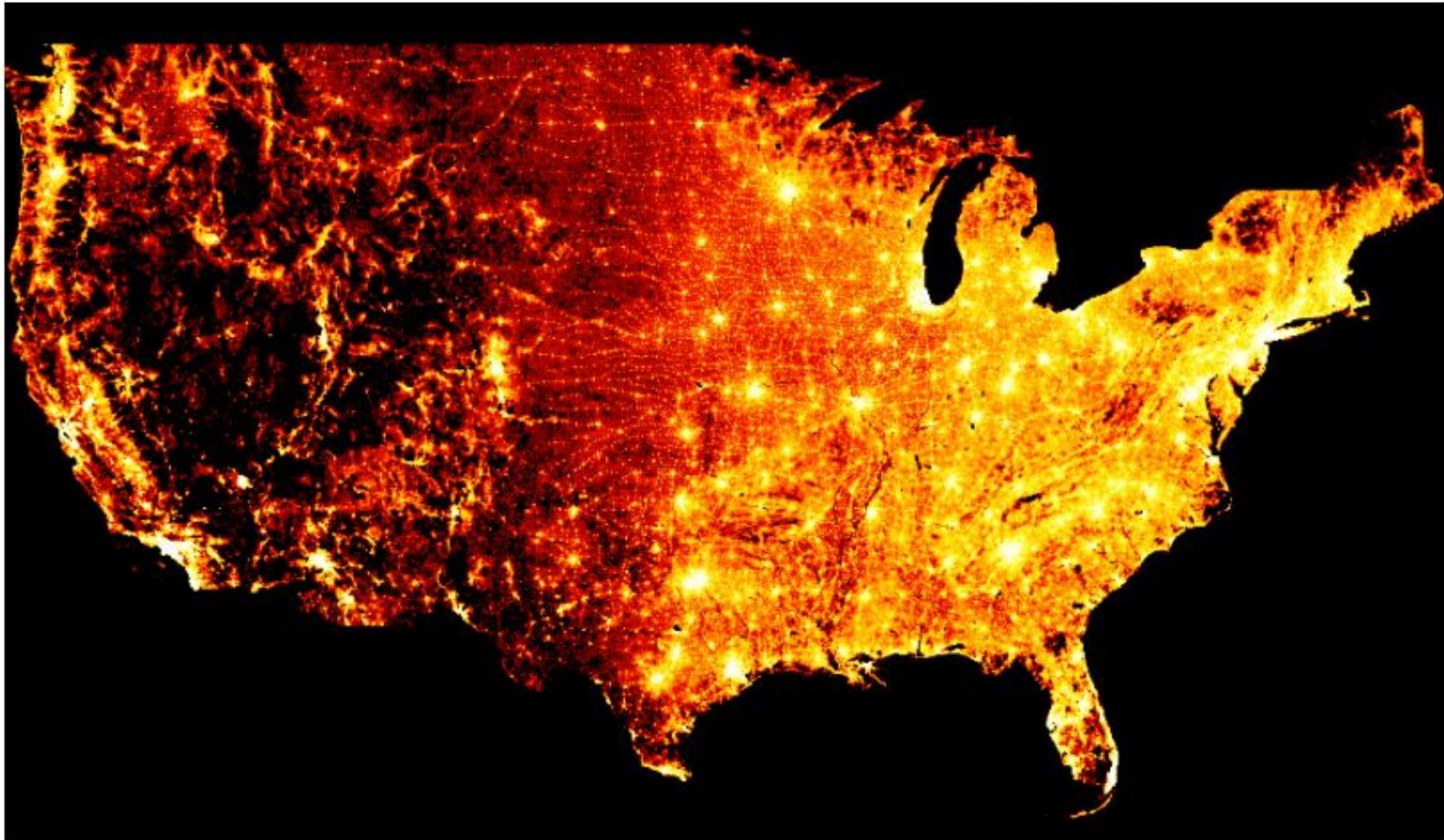


Datashader

Fast server-side engine for dynamic data aggregation

```
In [12]: from colorcet import fire  
export(tf.shade(agg, cmap = cm(fire,0.2), how='eq_hist'),"census_ds_fire_eq_hist")
```

Out[12]:



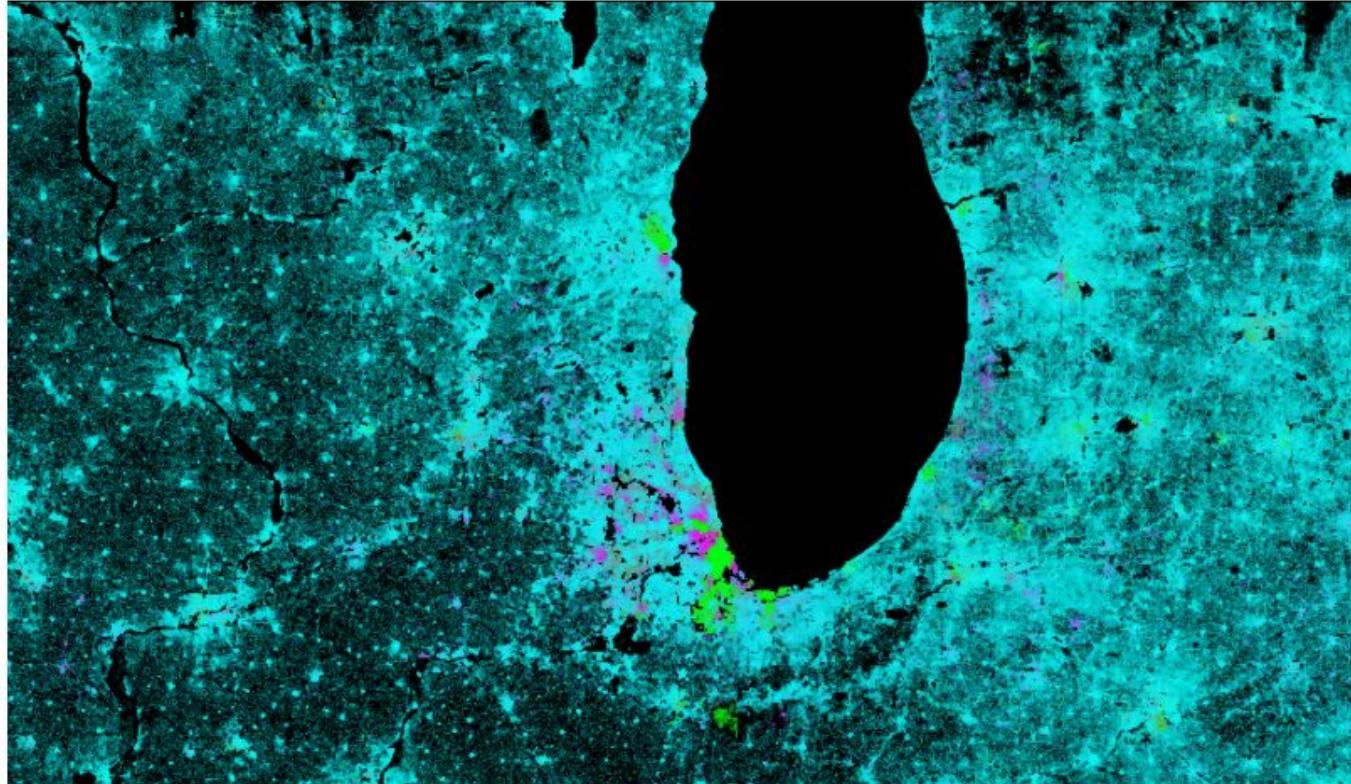
NGTON

Datashader

- Compute layer that works with Bokeh
- Rather than sending *data* to the client, it aggregates data and sends *pixels*.
- Can handle interactive visualization of billions of rows.

```
In [19]: export(create_image(*LakeMichigan), "Zoom 1 - Lake Michigan")
```

```
Out[19]:
```



ITY OF WASHINGTON

e Institute



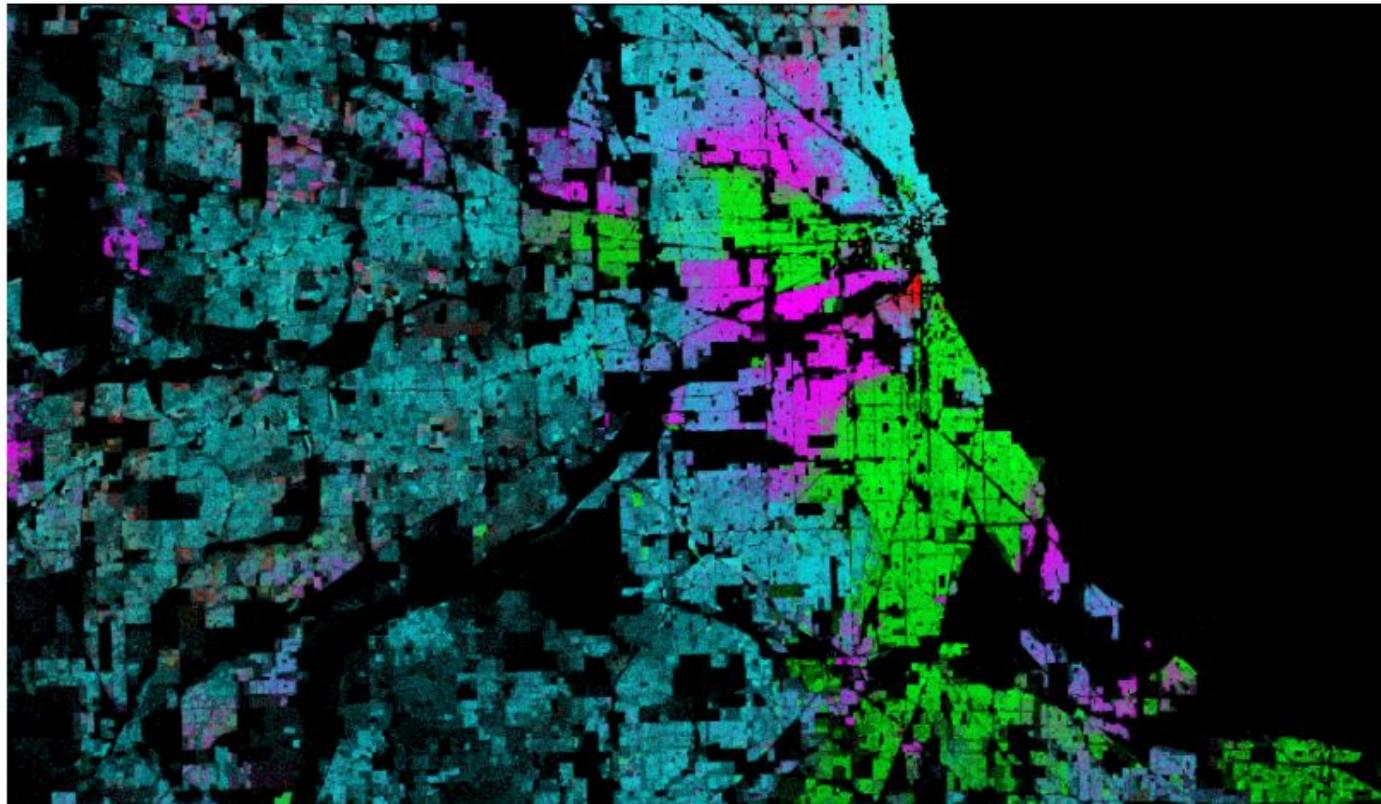
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

Datashader

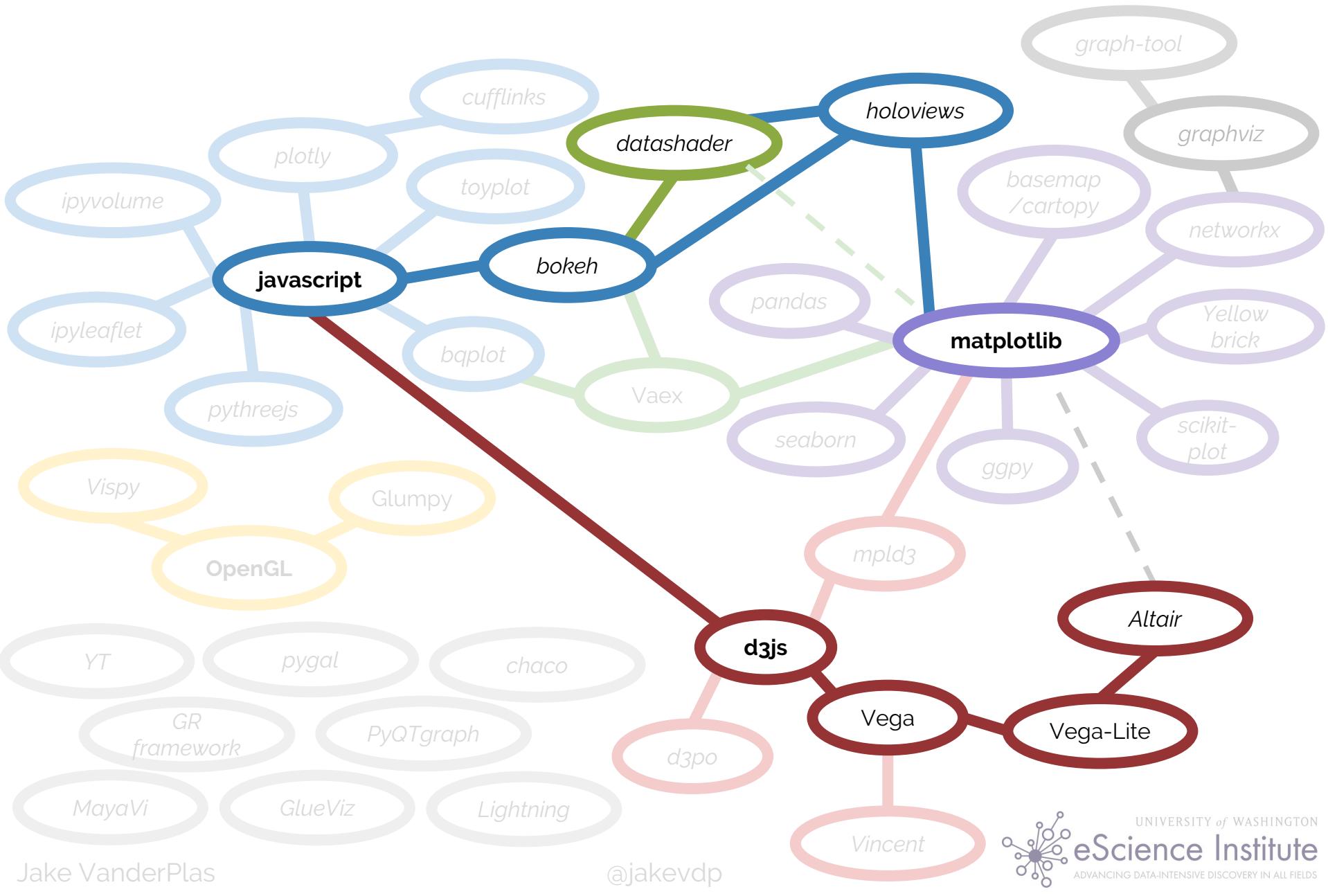
- Compute layer that works with Bokeh
- Rather than sending *data* to the client, it aggregates data and sends *pixels*.
- Can handle interactive visualization of billions of rows.

```
In [20]: export(create_image(*Chicago), "Zoom 2 - Chicago")
```

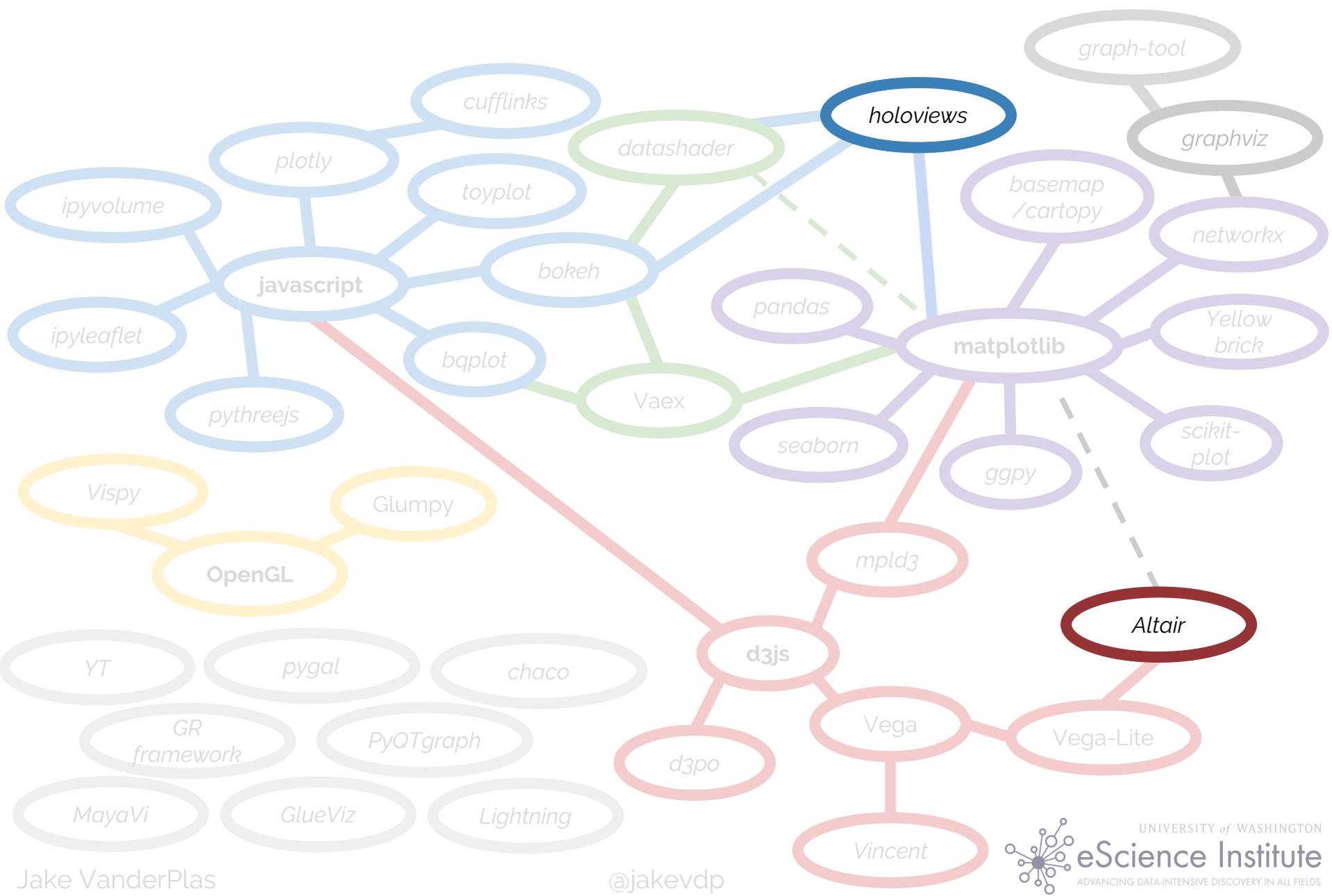
```
Out[20]:
```



Toward Declarative Visualization . . .



Toward Declarative Visualization . . .

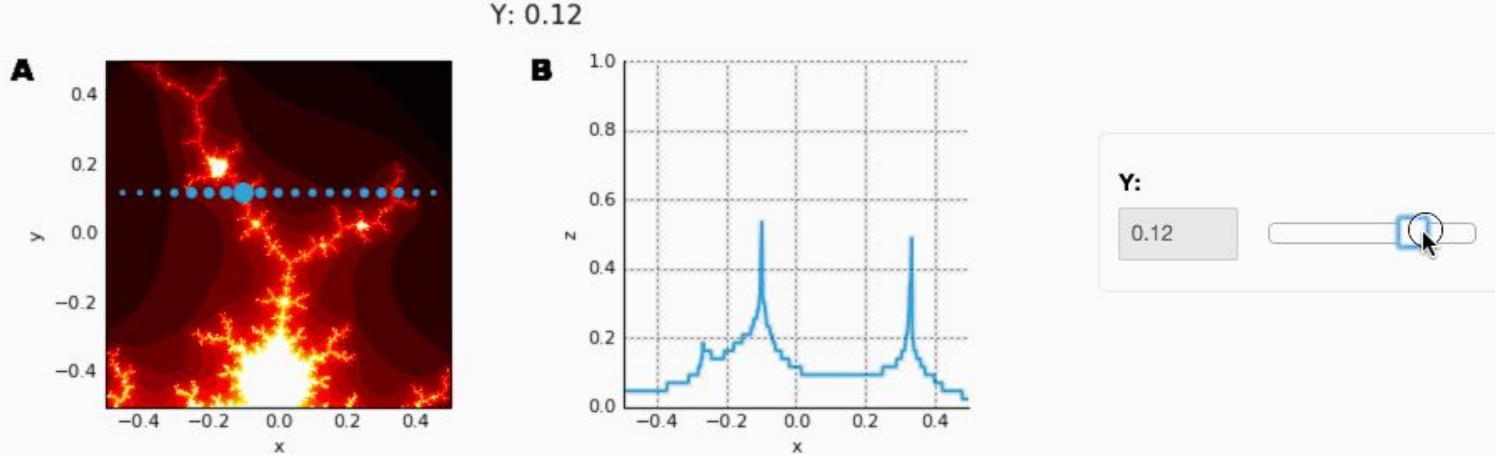


Holoviews

- Datasets themselves stored in objects that *automatically* produce intelligent visualizations
- Composition & Interactivity via operator overloading
- Renders to Bokeh, DataShader, and Matplotlib

```
In [3]: %%opts Points [scaling_factor=50] Contours (color='w')
dots = np.linspace(-0.45, 0.45, 19)
hv.HoloMap({y: (fractal * hv.Points(fractal.sample([(i,y) for i in dots])) + fractal.sample(y=y))
            for y in np.linspace(-0.3, 0.3, 21)}, kdims=['Y']).collate().cols(2)
```

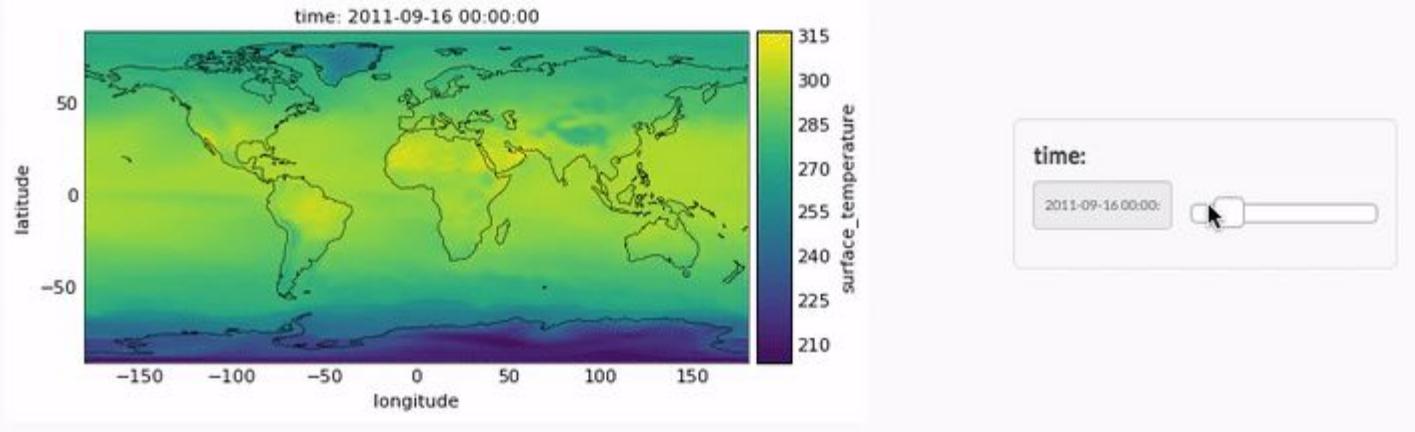
Out[3]:



Holoviews

- Also can handle geographic data & time-series

```
%opts Image [colorbar=True] (cmap='viridis') Overlay [fig_size=200]
ensemble = xr.open_dataset('./sample-data/ensemble.nc')
dataset = gv.Dataset(ensemble, crs=crs.PlateCarree())
dataset.to(gv.Image, ['longitude', 'latitude'], ['surface_temperature'], ['time']) * gf.coastline()
```



Altair

What if instead of passing
around *pixels*, we pass around
visualization specifications plus data?

Altair

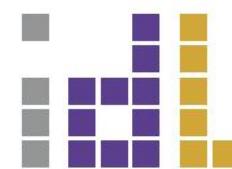
What if instead of passing
around *pixels*, we pass around
visualization specifications plus data?

“Declarative Visualization”

Altair

What if instead of passing
around *pixels*, we pass around
visualization specifications plus data?

“Declarative Visualization”



Declarative Visualization: Viz for data science

Imperative

- Specify *How* something should be done.
- Must manually specify plotting steps
- Specification & Execution intertwined.

Declarative

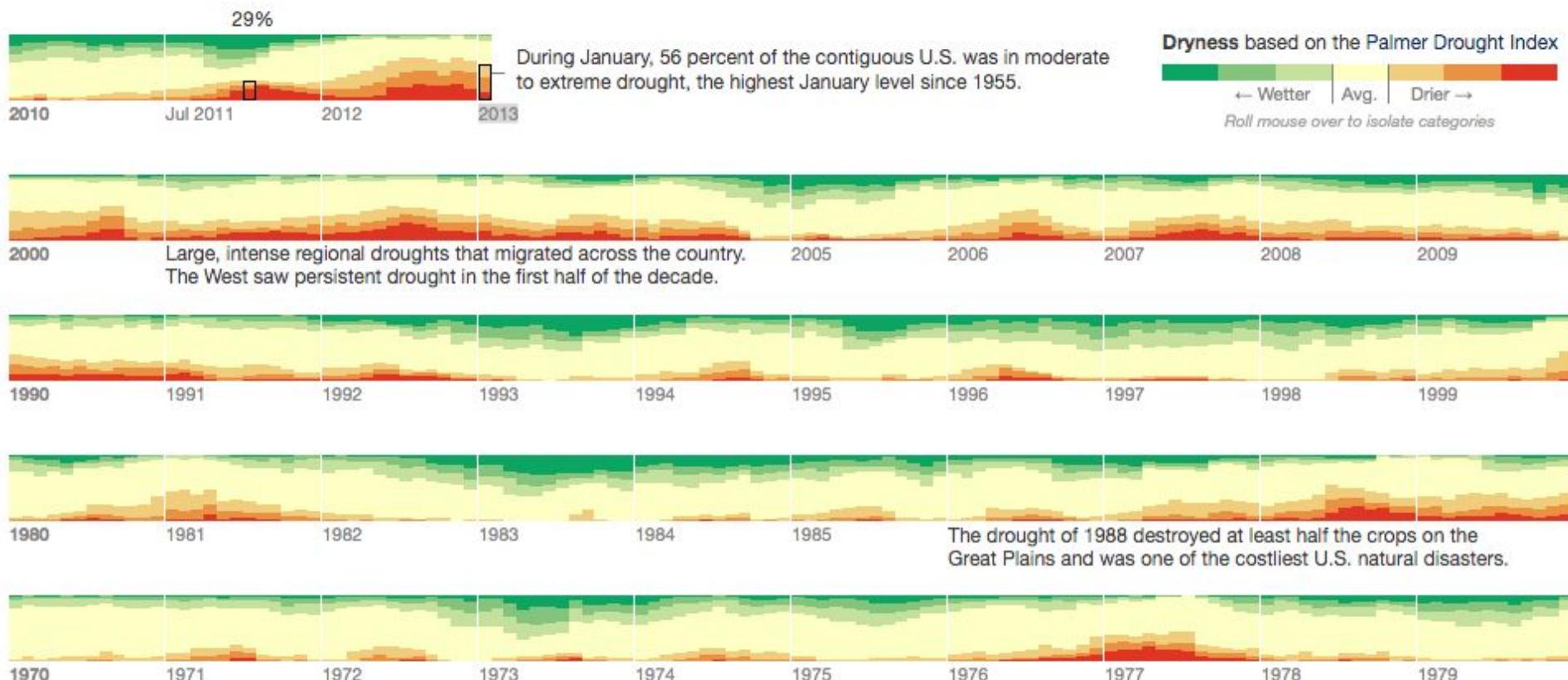
- Specify *What* should be done
- Details determined automatically
- Separates Specification from Execution

Declarative visualization lets you think about **data** and **relationships**, rather than incidental details.

From D3 to Altair . . .

Drought and Deluge in the Lower 48

Last summer's drought, one of the worst in a century, has continued through the winter. This chart shows the proportion of what is now the contiguous U.S. in various stages of drought over 118 years of record-keeping. Roll mouse over individual months to see what percentage of the lower 48 was in drought. [Related Article »](#)



[\(link to live version\)](#)

**But working in D3 can
be challenging . . .**

```

var margin = {top: 20, right: 20, bottom: 30, left: 40},
width = 960 - margin.left - margin.right,
height = 500 - margin.top - margin.bottom;

var x = d3.scale.ordinal()
.rangeRoundBands([0, width], .1);

var y = d3.scale.linear()
.range([height, 0]);

var xAxis = d3.svg.axis()
.scale(x)
.orient("bottom");

var yAxis = d3.svg.axis()
.scale(y)
.orient("left")
.ticks(10, "%");

var svg = d3.select("body").append("svg")
.attr("width", width + margin.left + margin.right)
.attr("height", height + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + " " + margin.top + ")");

d3.tsv("data.tsv", type, function(error, data) {
if (error) throw error;

x.domain(data.map(function(d) { return d.letter; }));
y.domain([0, d3.max(data, function(d) { return d.frequency; })]);

svg.append("g")
.attr("class", "x axis")
.attr("transform", "translate(0," + height + ")")
.call(xAxis);

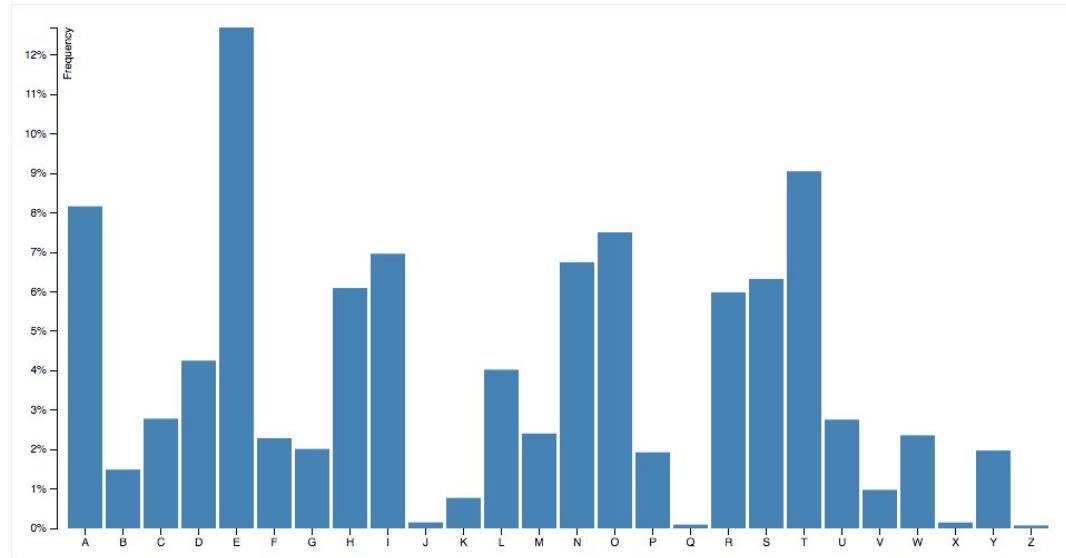
svg.append("g")
.attr("class", "y axis")
.call(yAxis)
.append("text")
.attr("transform", "rotate(-90)")
.attr("y", 6)
.attr("dy", ".71em")
.style("text-anchor", "end")
.text("Frequency");

svg.selectAll(".bar")
.data(data)
.enter().append("rect")
.attr("class", "bar")
.attr("x", function(d) { return x(d.letter); })
.attr("width", x.rangeBand())
.attr("y", function(d) { return y(d.frequency); })
.attr("height", function(d) { return height - y(d.frequency); });
});

function type(d) {
d.frequency = +d.frequency;
return d;
}

```

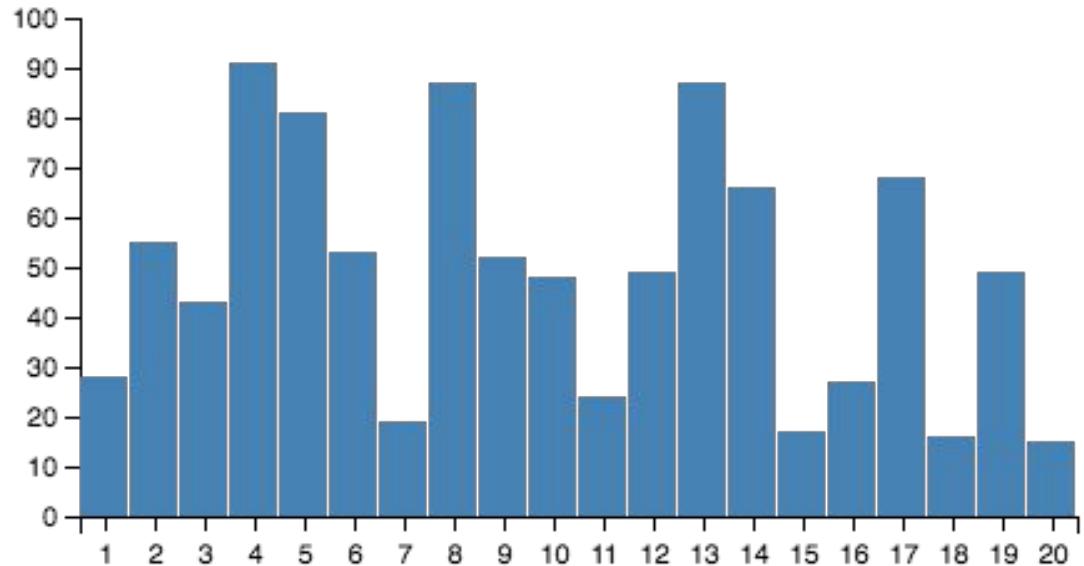
Bar Chart: d3



D3 is a Javascript package that streamlines manipulation of objects on a webpage.

```
{
  "width": 400,
  "height": 200,
  "padding": {"top": 10, "left": 30, "bottom": 30, "right": 10},
  "data": [
    {
      "name": "table",
      "values": [
        {"x": 1, "y": 28}, {"x": 2, "y": 55},
        {"x": 3, "y": 43}, {"x": 4, "y": 91},
        {"x": 5, "y": 81}, {"x": 6, "y": 53},
        {"x": 7, "y": 19}, {"x": 8, "y": 87},
        {"x": 9, "y": 52}, {"x": 10, "y": 48},
        {"x": 11, "y": 24}, {"x": 12, "y": 49},
        {"x": 13, "y": 87}, {"x": 14, "y": 66},
        {"x": 15, "y": 17}, {"x": 16, "y": 27},
        {"x": 17, "y": 68}, {"x": 18, "y": 16},
        {"x": 19, "y": 49}, {"x": 20, "y": 15}
      ]
    }
  ],
  "scales": [
    {
      "name": "x",
      "type": "ordinal",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y",
      "type": "linear",
      "range": "height",
      "domain": {"data": "table", "field": "y"},
      "nice": true
    }
  ],
  "axes": [
    {"type": "x", "scale": "x"},
    {"type": "y", "scale": "y"}
  ],
  "marks": [
    {
      "type": "rect",
      "from": {"data": "table"},
      "properties": {
        "enter": {
          "x": {"scale": "x", "field": "x"},
          "width": {"scale": "x", "band": true, "offset": -1},
          "y": {"scale": "y", "field": "y"},
          "y2": {"scale": "y", "value": 0}
        },
        "update": {
          "fill": {"value": "steelblue"}
        }
      }
    }
  ]
}
```

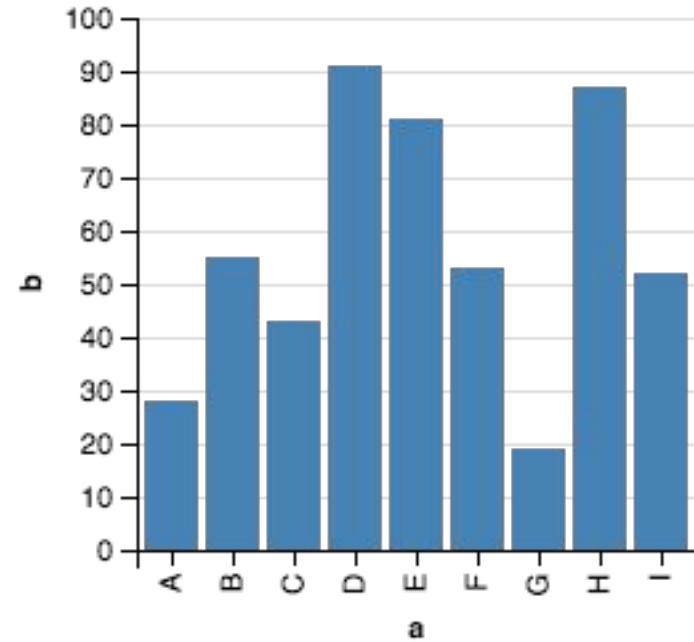
Bar Chart: Vega



Vega is a detailed declarative specification for visualizations, built on D3.

```
{  
  "description": "A simple bar chart with embedded data.",  
  "data": {  
    "values": [  
      {"a": "A", "b": 28}, {"a": "B", "b": 55}, {"a": "C", "b": 43},  
      {"a": "D", "b": 91}, {"a": "E", "b": 81}, {"a": "F", "b": 53},  
      {"a": "G", "b": 19}, {"a": "H", "b": 87}, {"a": "I", "b": 52}  
    ]  
  },  
  "mark": "bar",  
  "encoding": {  
    "x": {"field": "a", "type": "ordinal"},  
    "y": {"field": "b", "type": "quantitative"}  
  }  
}
```

Bar Chart: Vega-Lite



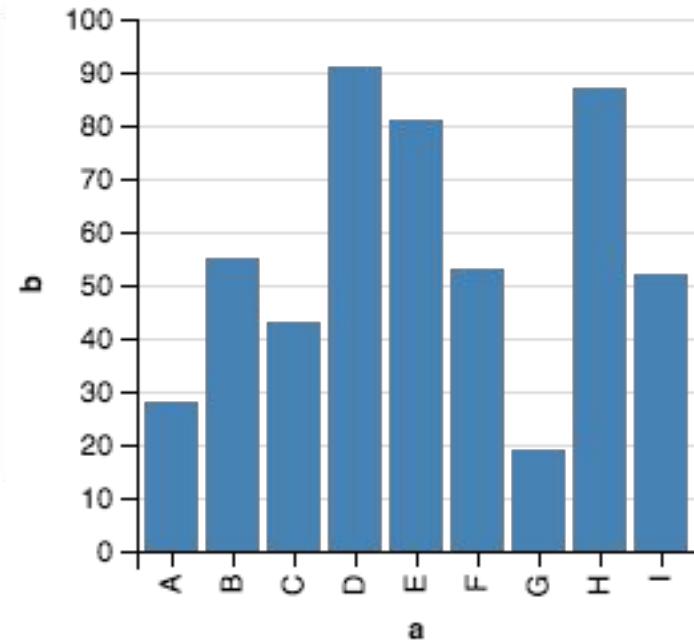
Vega-Lite is a simpler declarative specification aimed at statistical visualization.

Bar Chart: Altair

```
import pandas as pd
from altair import Chart

data = pd.DataFrame({'a': ['A', 'B', 'C', 'D', 'E',
                           'F', 'G', 'H', 'I'],
                      'b': [28, 55, 43, 91, 81,
                            53, 19, 87, 52]})

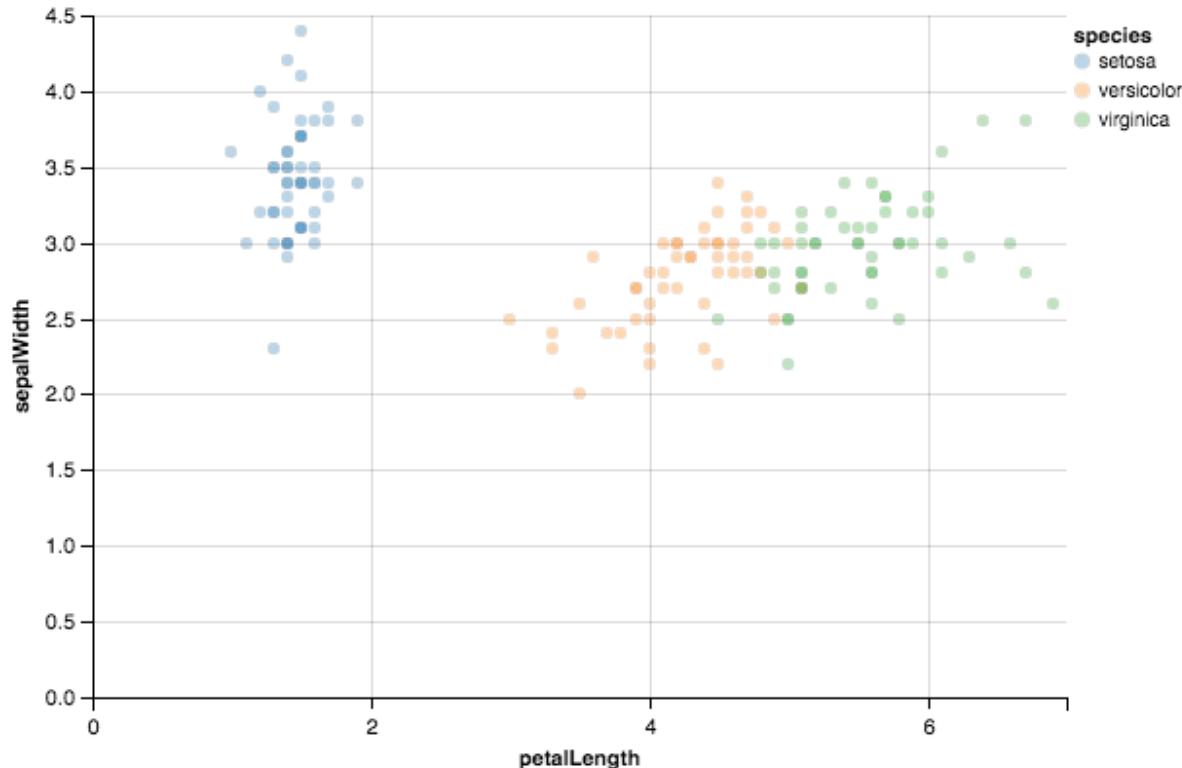
Chart(data).mark_bar().encode(
    x='a',
    y='b',
)
```



Altair is a Python API for creating Vega-Lite specifications.

From Declarative API to declarative Grammar

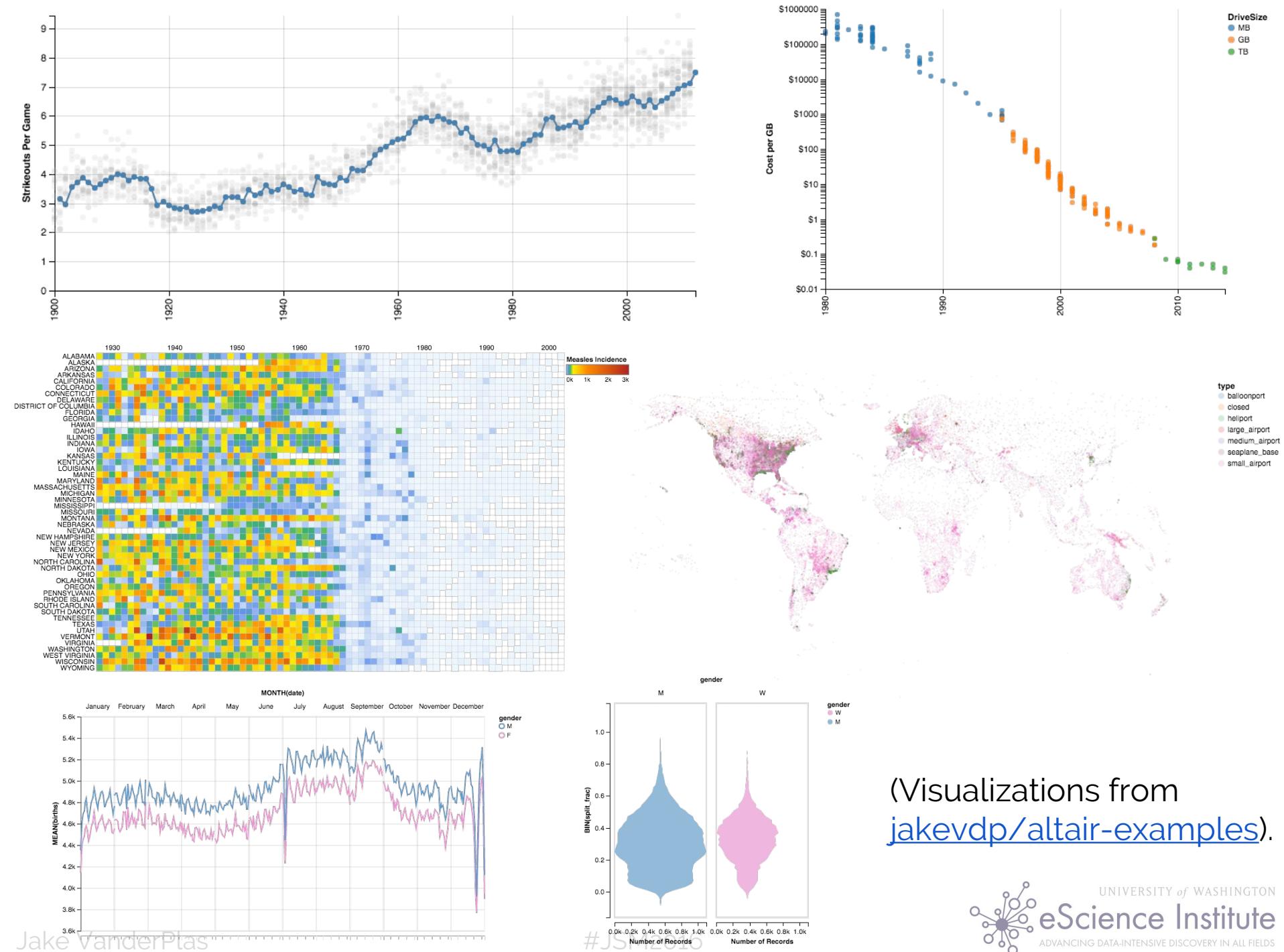
```
chart = Chart(data).mark_circle(  
    opacity=0.3  
) .encode(  
    x='petalLength:Q',  
    y='sepalWidth:Q',  
    color='species:N',  
)  
chart.display()
```



From Declarative API to declarative *Grammar*

```
>>> chart.to_dict()

{'config': {'mark': {'opacity': 0.3}},
 'data':
  {'url': 'https://vega.github.io/vega-datasets/data/iris.json'},
 'encoding': {'color': {'field': 'species', 'type': 'nominal'},
              'x': {'field': 'petalLength', 'type': 'quantitative'},
              'y': {'field': 'sepalWidth', 'type': 'quantitative'}},
 'mark': 'circle'}
```



(Visualizations from
[jakevdp/altair-examples](https://jakevdp.github.io/altair-examples/))

Coming Very Soon: Altair 2.0

- Includes a Grammar of Interaction



Try Altair:

```
$ conda install altair --channel conda-forge
```

or

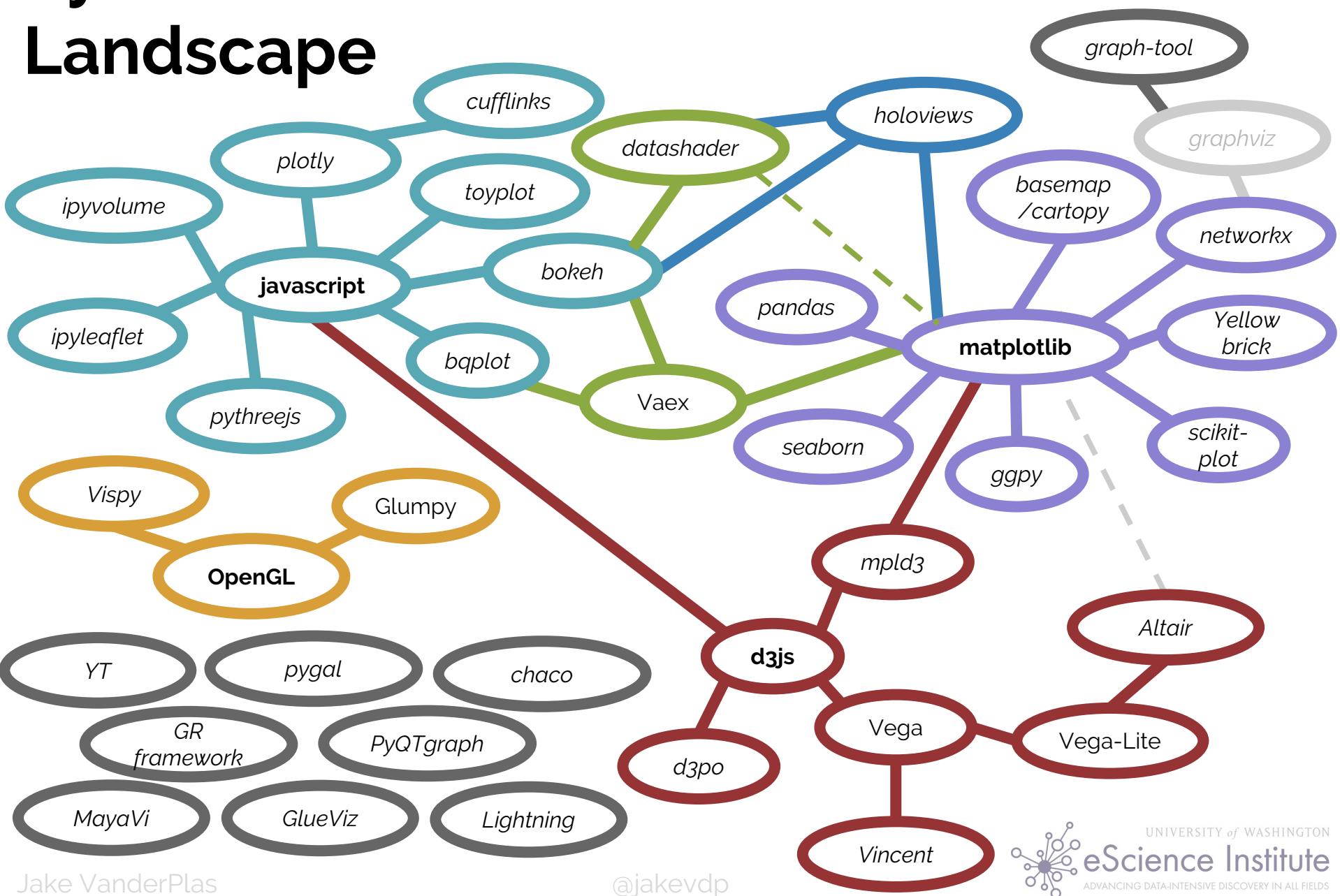
```
$ pip install altair
$ jupyter nbextension install --sys-prefix --py vega
```

For a Jupyter notebook tutorial, type

```
import altair
altair.tutorial()
```

<http://github.com/ellisonbg/altair/>

Python's Visualization Landscape



Thank You!



Email: jakevdp@uw.edu



Twitter: [@jakevdp](https://twitter.com/jakevdp)



Github: [jakevdp](https://github.com/jakevdp)



Web: <http://vanderplas.com/>



Blog: <http://jakevdp.github.io/>