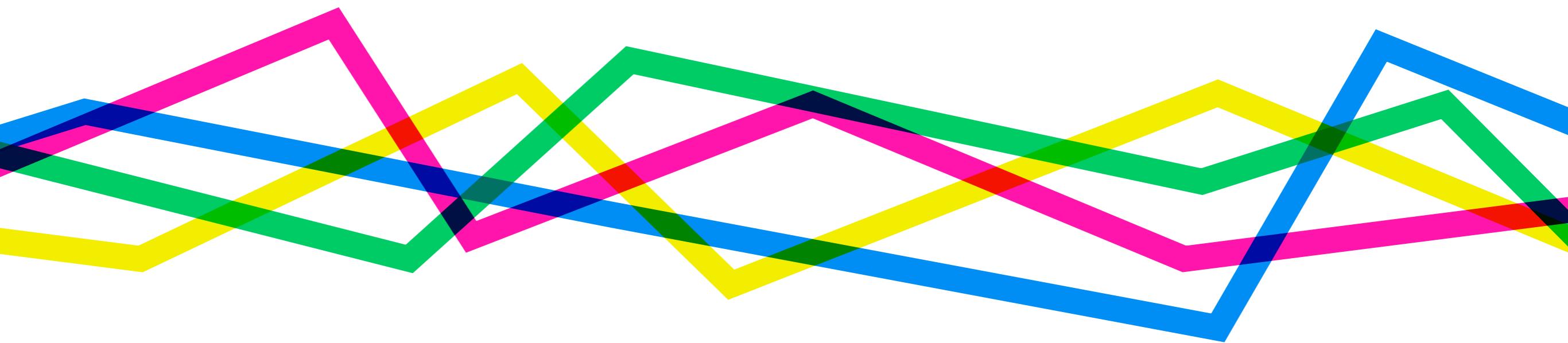


While you wait, go to Github Repository:  
[bit.ly/CodeHerD3](https://bit.ly/CodeHerD3)

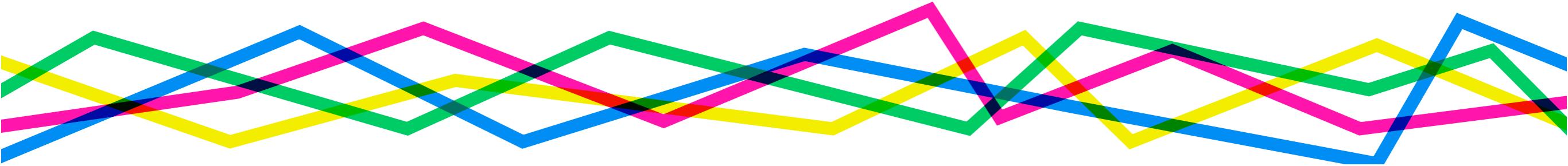


# Intro to Data Viz with d3.js

*When to use it, plus creating some fresh plots!*

•

# Hi, I'm Mollie



- Data Scientist at Datascope Analytics
- Data Viz Instructor at Metis
- Background in mathematics and geology
- Trekkie
- Avid lindy hop (swing) dancer

# Agenda

- What is D3.js and what is it for?
- D3 examples
- Check out today's workshop plot
- D3 Fundamentals
- Make Dat Fresh Plot!

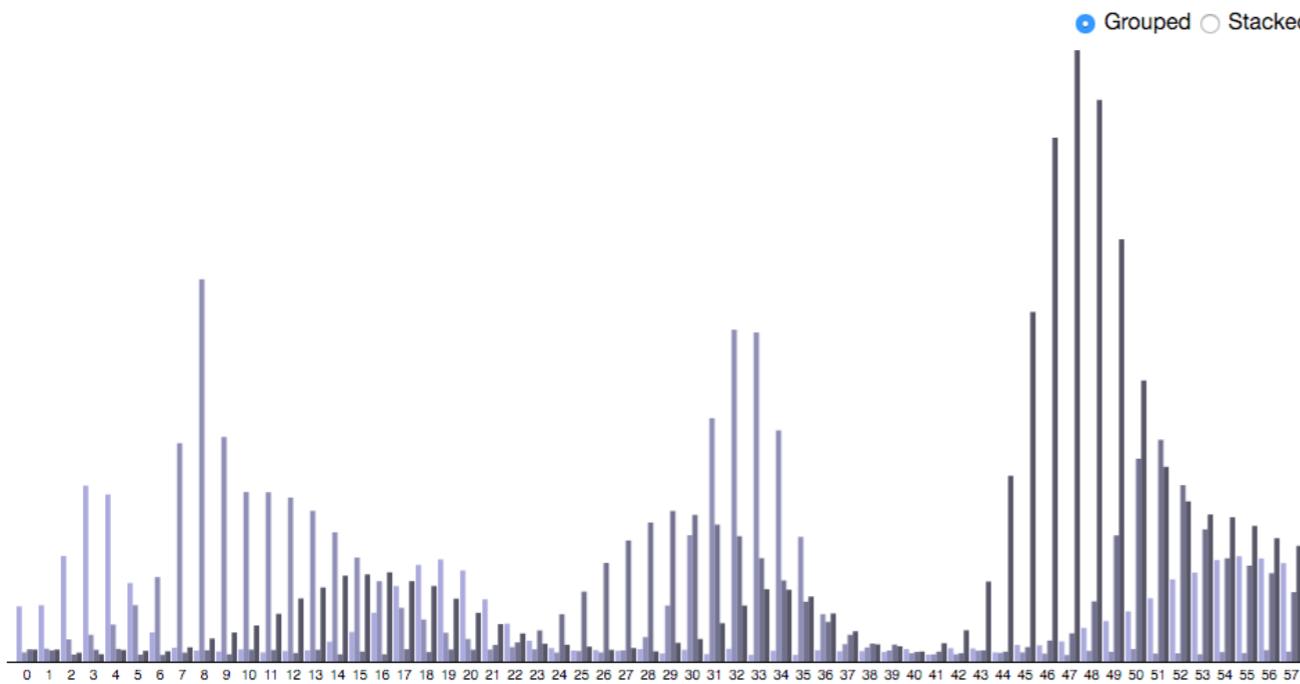


let's get to know each other

**What is D3 for?**







**Three Years Later**

Three years after the I.P.O., two-thirds of companies had negative returns, including nearly all companies that went public during the dot-com bubble of 1999 and 2000. Around 60 percent of companies with offerings since 2010 have negative returns so far.

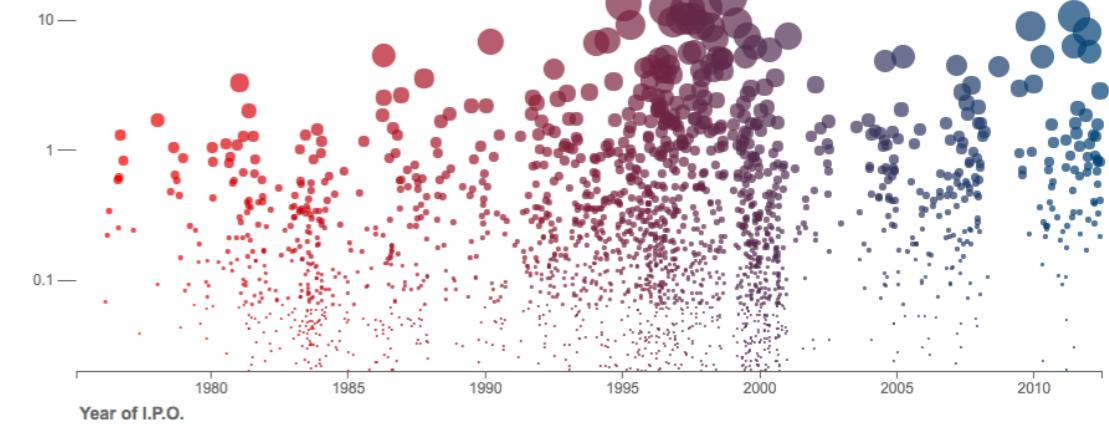
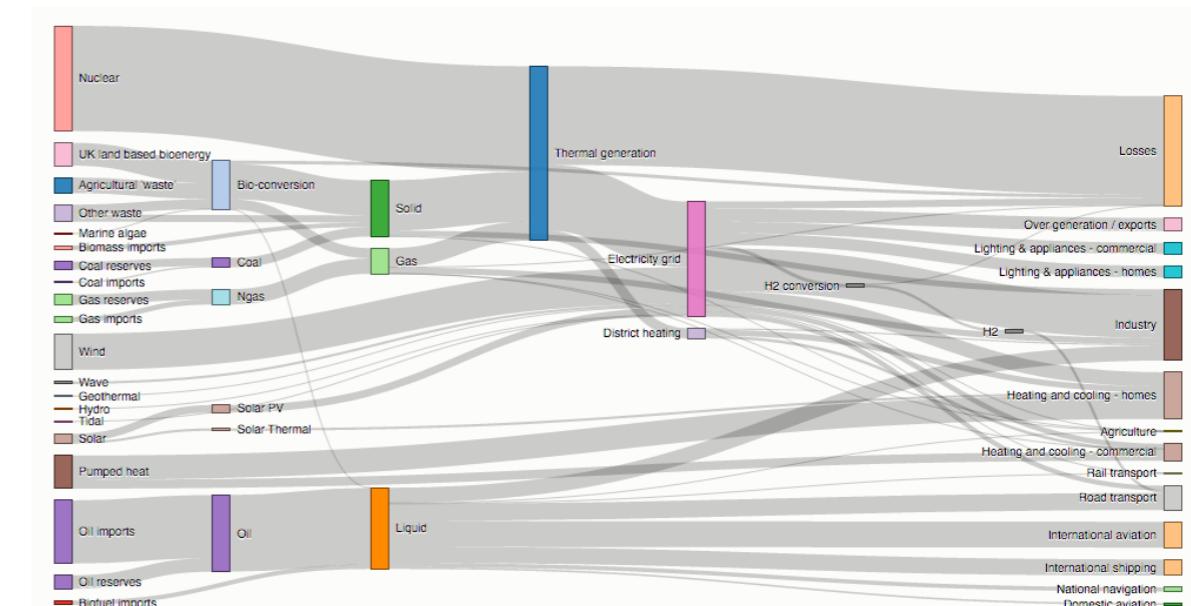
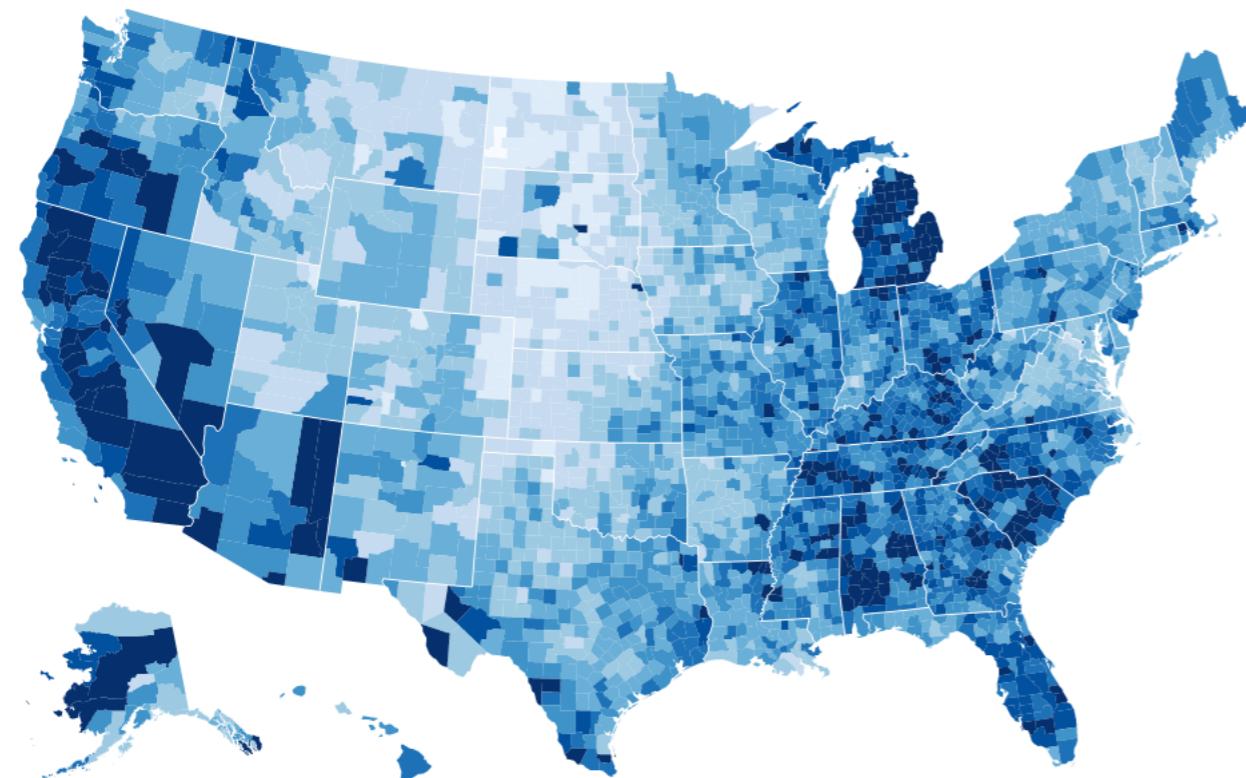


Chart shows value after three years for shares outstanding after the I.P.O.. Returns through Wednesday are shown for companies with I.P.O.'s since 2009.





D3 makes interactive  
visualizations in the  
browser easier

# Things D3 *might* not be the best tool for

- Quick, exploratory visualizations
- Plots that:
  - Won't need to be updated
  - Don't need to be interactive
  - Static and with a conventional style (line, bar, scatter plots)
  - Absolutely need to work in old, old browsers (IE8 and older)

# Alternatives



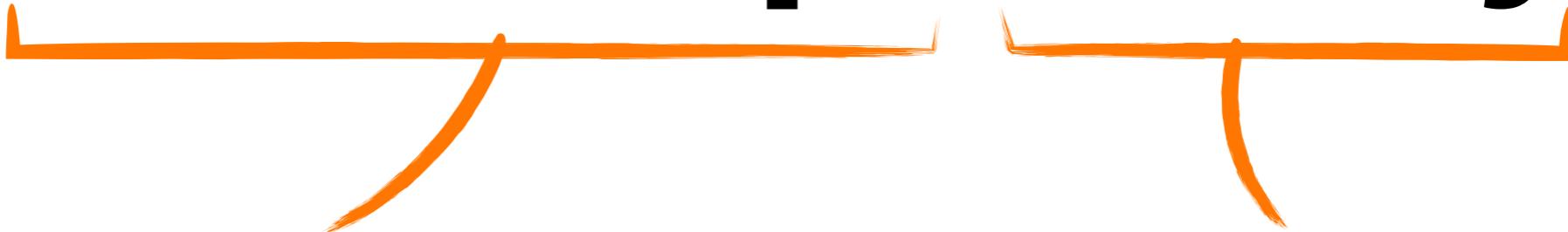
- Highcharts (Javascript)
- Tools/Services built on D3
  - Vega/Vega-Lite
  - plot.ly
- ggplot2 (R)
- matplotlib + seaborn + pandas (Python)
- Tableau

# Common workflow:

- Use plotting tool included with whatever we're using to do data exploration
- Sketch some ideas for effective visualizations
- If most compelling sketches can't be achieved using something simpler, use D3

OK, what *is* D3

# a **JavaScript library**



programming language  
that runs in web browsers

reusable pieces of code  
usable while programming

Data-  
Driven  
Documents

**D**ata- Use data

**D**riven to make stuff

**D**ocuments using web-based documents

# “web based documents”



Here is some cat text

[link to more cats](#)

HTML - the skeleton  
boxes, controls, text, images

# “web based documents”



Here is some cat text

[link to more cats](#)

random cat button

HTML - the skeleton  
boxes, controls, text, images

```
<button name="random">  
    random cat button  
</button>
```

# “web based documents”



Here is some cat text

[link to more cats](#)

random cat button

HTML - the skeleton  
boxes, controls, text, images

```
<div id="controls">  
  <button name="random">  
    random cat button  
  </button>  
</div>
```

# “web based documents”



Here is some cat text

[link to more cats](#)

HTML - the skeleton  
boxes, controls, text, images

CSS - styles for the skeleton  
definitions that change  
the looks of the page

# “web based documents”



HTML - the skeleton

CSS - styles for the skeleton

SVG - language for drawing

- kinda like HTML, but can be embedded inside it
- can be styled with CSS
- no worries about resolution with vectors

# “web based documents”



HTML - the skeleton  
CSS - styles for the skeleton  
SVG - language for drawing

```
<circle r="10">  
</circle>
```

A JavaScript library  
that makes it easier  
to use HTML, SVG, and CSS  
to make interactive visualizations  
in the browser

# in summary - why use D3.js?

- efficient manipulation of documents based on data
- utilizes full capabilities of web standards such as HTML, SVG, and CSS
- extreme flexibility - lets you build the data visualization framework that you want
- D3 is written in javascript and uses a functional style, which means you can reuse code and add specific functions to your heart's content
- minimal overhead
- extremely fast
- supports large datasets
- allows for dynamic behaviors, interactions, and animations

Let's Look at some Examples!

Go to Repository:

[bit.ly/CodeHerD3](https://bit.ly/CodeHerD3)

What we'll be  
making...

# **Let's try it out!**

*D3 Fundamentals*

example:  
appending text to the DOM

# chaining

```
var body = d3.select('body') // Select the part of the page where you want to put  
your data
```

```
var h1 = body.append('h1') // specify tag
```

```
var text = h1.text("lets make a scatterplot") //specify data we'd like to add as text  
text.style('font', '48px adelle-sans'); //add styling to text
```

**More efficiently, D3 lets you chain your code. Handy!!**

```
d3.select("body") // Select body
```

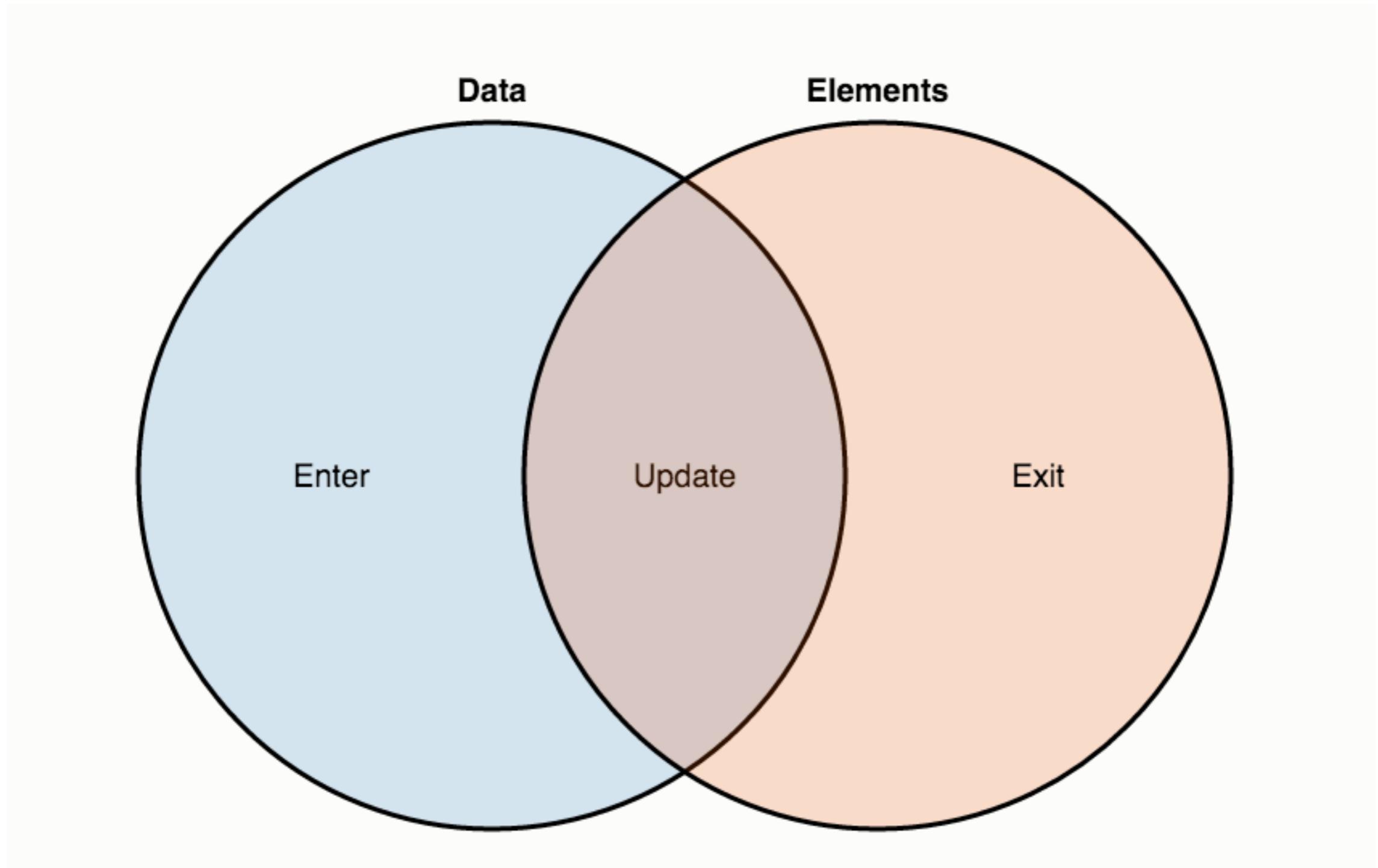
```
.append("h1") // specify h1 tag
```

```
.text("lets make a scatterplot") //specify text
```

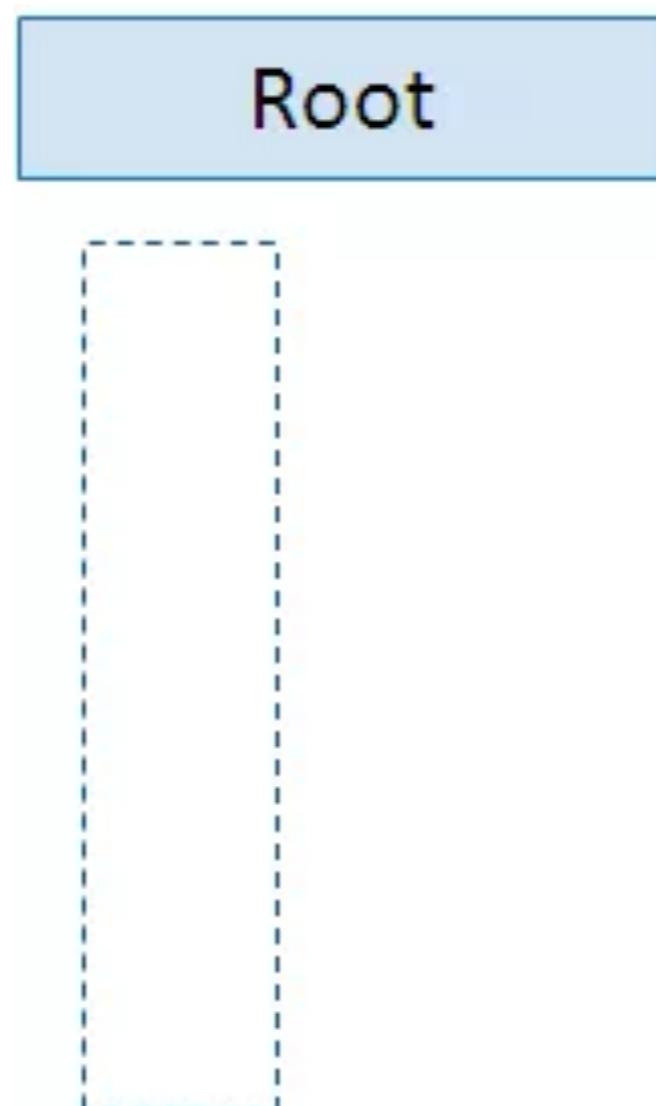
```
.style("font", "48px adelle-sans"); //add styling to text
```

example:  
your first data join

# “the data join”



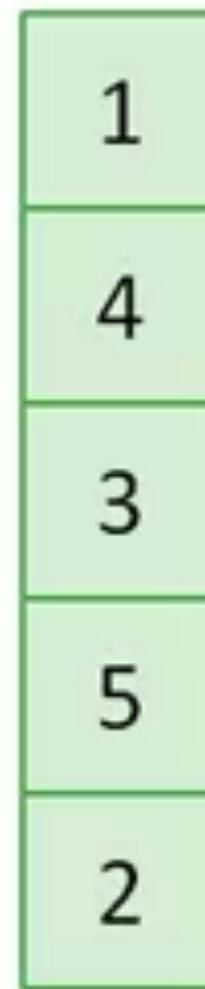
# .selectAll()



credit to Jerome Cukier

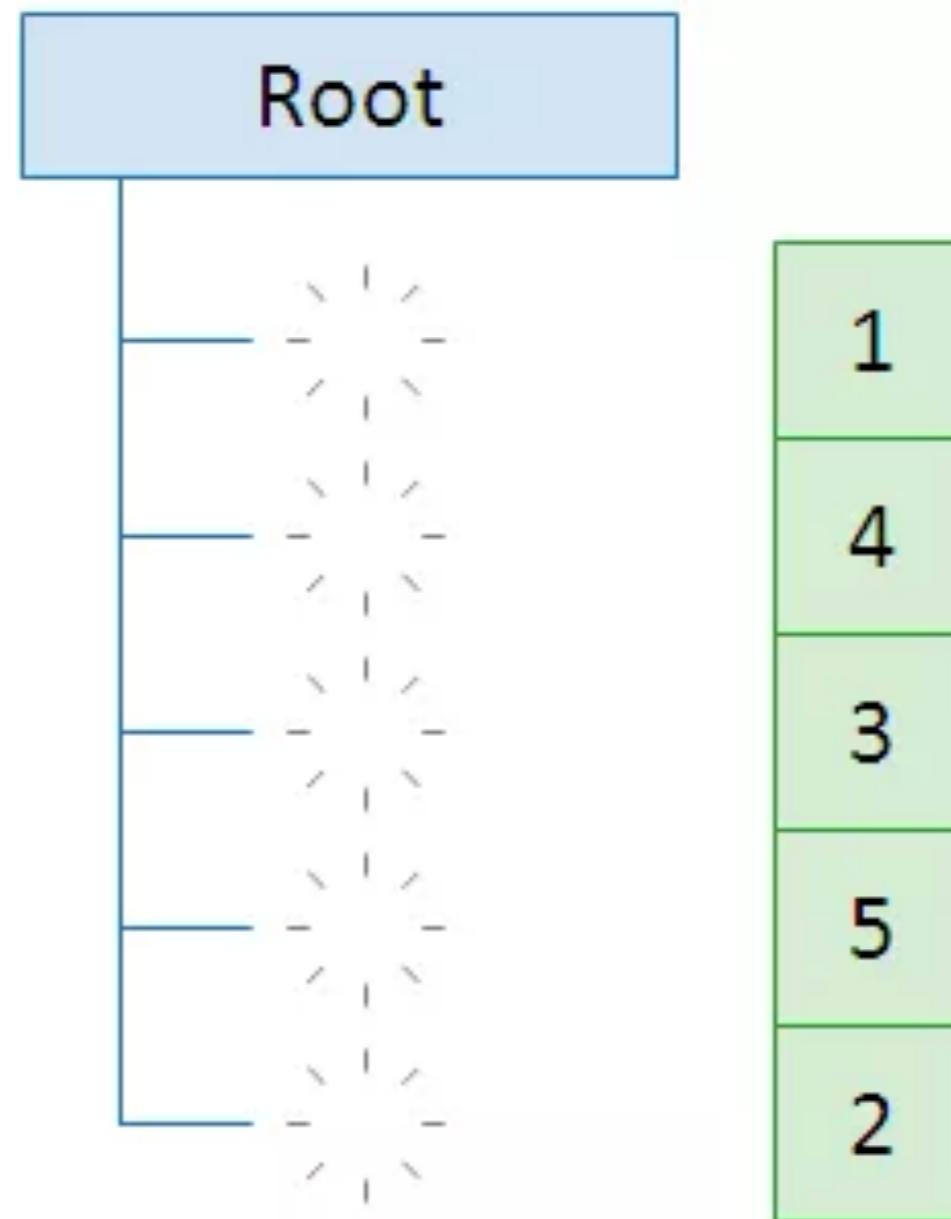
# .data()

Root



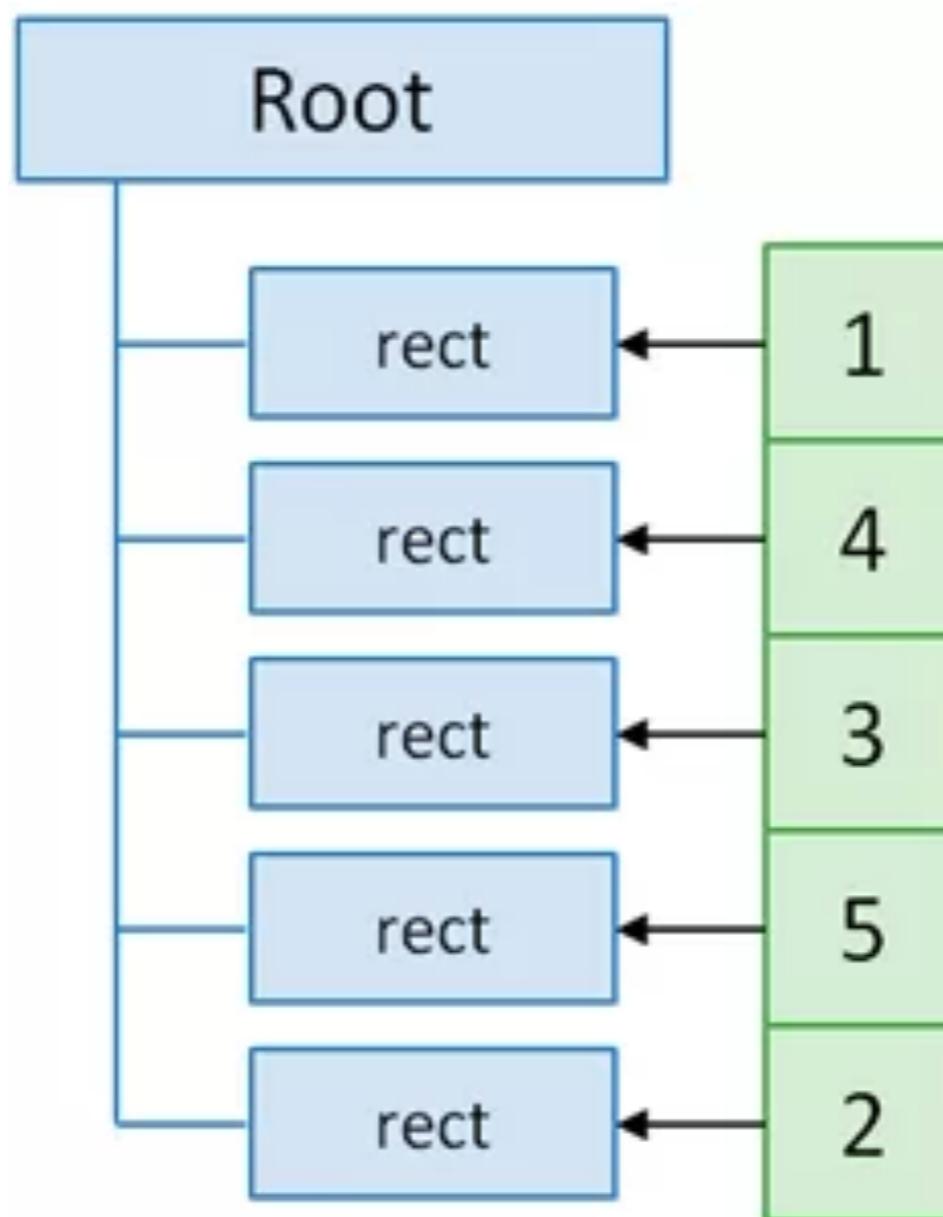
credit to Jerome Cukier

# .enter()



credit to Jerome Cukier

# .append()



credit to Jerome Cukier

# data joins

```
var parts = ["This is", "my first", "data join!"]
```

**d3.select("body").selectAll("p")** // returns a new empty selection since there are not yet any p elements in the body.

**.data(parts)** //This selection is then joined to an array of data. Additionally, three new selections that represent the three possible states: enter, update, and exit. Since the selection was empty, the update and exit selections are empty, while the enter selection contains a placeholder for each new datum.

**.enter()**

**.append("p")** //enter().append() creates as many p elements as the number of elements in the enter selection.

// The argument of append specifies the type of element to be created

**.text(function(d) { return d; })**; //adds the data text elements in the enter selection to the paragraph elements created by the enter().append()

Among the things we didn't  
cover yet

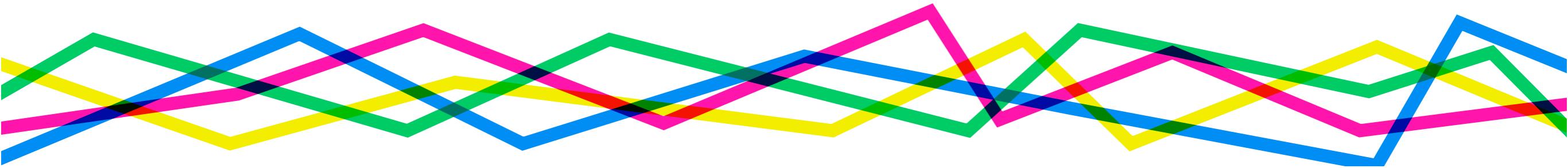
**d3.axes**      updating/removing data

**.transition() for animating changes**

**d3.time**      **d3.layouts**      **d3.color**

**adding listeners for interactivity**      **d3.geo**

# for continued learning - Metis D3.js course



- Course: Intro to Data Visualization using D3.js
- Location: Online
- Dates: November 6th - December 13th
- Duration: 6 weeks
- Course created by: Kevin Quealy
- Instructor: Paul Buffa
- Signup Link: <http://www.thisismetis.com/data-visualization-with-d3>

# Thanks!

contact me: @MollzMP

resources: <http://bit.ly/CodeHerD3>