

D3.js

What is D3js?

- “It's a library for making interactive graphs and stuff”
- ... not really (I thought so too)
- All the visualization stuff is just SVG (or even HTML if you want)
- D3 is (mostly) a library to associate **data** with **DOM elements**

D3 as DOM Manipulator

- Say you want to make a bar graph
- You have pieces of data to plot, maybe something like [{name: 'Chilis', rating: 2.5}, {name: 'Applebees', rating: 2.4}, ...]
- Each element of the data array should be associated with a bar (what's the viz term for that?)
- D3 lets us link each data element to a DOM element, (maybe an SVG rect or even just an html p element), and then set its attributes based on its value
- For a bar, we'd set it's width based on the data value, and maybe add some text based on the name property.

Other tools

- Dom ↔ Data linking is the core of D3, but it provides a bunch of stuff useful for making visualization
 - scaling functions to transform, for example temperature to pixels
 - UI elements like axes
 - Tools for working with geometric data
 - Tools for doing simulation (for force based graph layouts, etc)
 - Interpolation tools (`d3.interpolateLab("steelblue", "brightred")`
`(0.5);`)
- They are helpful tools that you'll reach for

Basics:

```
d3.select('#parentElement')  
  .selectAll('rect') //each data point will get a rect  
  .data(myArrayOfData) //associate the with data  
  .append('rect') //sort of weird repetition  
  .enter() //next methods operate on rects newly created  
  .attr('width', (d) => d.someProperty) // CSS attribute  
  //calls the lambda, passing the data
```

What's happening

- D3 operates on “selections” which are groups of elements
- We start with a `selectAll` which grabs all children of a given element type inside a parent element
- Usually, when we start, there are none, so this is sort of weird...
- the `.data()` method associates each of those with an element in our array
- The `.append()` call creates one for each data element which does not yet have a DOM element it's associated with
- The `.enter()` method runs operations on newly created DOM elements
- There are also `update()` and `exit()` methods that run when we change or remove data and the corresponding DOM elements need to be updated/removed

Combo: join

- If you want to do different stuff for new, existing, and deleted entries, use `join` instead of `enter`:

```
selectAll('p')  
  .data(whatever)  
  .join( enter ⇒ enter.append(p).text(whatever),  
        update ⇒ update.doSomethingWithExistentElements(),  
        exit ⇒ exit.remove()) //remove delete elements
```

- Each of the 3 args to `join` is a lambda that takes and returns a selection of elements
- When we `append`, we're returning the newly created elements in as our selection

Lets make something

- We'll build up the bar graph of bad restaurants I mentioned above
- We'll start with the basic D3-DOM manipulation and add usage of D3 tools as we go to make it nicer
- Next class we'll work on a more interesting, real data set