Data Visualization (CIS/DSC 468)

D3 and Marks & Channels

Dr. David Koop
D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3’s emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.
D3 Data Joins

- Two groups: data and visual elements
- Three parts of the join between them: enter, update, and exit
- enter: `s.enter()`, update: `s`, exit: `s.exit()`
Assignment 2

- Use D3
  1. Repeat Part 3b of A1 using D3
  2. Extend Part 1 to create a **stacked** bar chart
  3. Create a line chart that shows a region's numbers that is linked to a dropdown menu allowing you to select the region. Use transitions!
D3 Introduction

• Ogievetsky has put together a nice set of interactive examples that show off the major features of D3

• http://dakoop.github.io/IntroD3/
  - (Updated from original for D3 v4)

• Other references:
  - Murrary’s book on Interactive Data Visualization for the Web
  - The D3 website: d3js.org
  - Ros's Slides on v4: https://iros.github.io/d3-v4-whats-new/
D3 v4 vs. v3

- v4 breaks a lot of v3 code...
- v4 is more modular, can build libraries that include only the parts you care about
  - Why worry about this?
- Result is that there is a flat namespace now
  - d3.scale.linear => d3.scaleLinear
- More important change: selections are immutable now
  - Used to be that enter() modified the selection to include any appended items
  - Use merge to explicitly merge the enter and update selections
    - s.enter().append("rect")
      .merge(s)
    ...

D3 v3 Selections

var circleBinding = svg.selectAll("circle").data(data);

circleBinding.style("fill", "blue"); // UPDATE

circleBinding.enter()
  .append("circle") // ENTER; modifies UPDATE!
    .style("fill", "green");

circleBinding // ENTER + UPDATE
  .style("stroke", "black");
D3 v4 Selections

```javascript
var circleBinding = svg.selectAll("circle").data(data);

circleBinding.style("fill", "blue"); // UPDATE

circleBinding.enter()
  .append("circle") // ENTER; modifies UPDATE!
    .style("fill", "green");

  .merge(circleBinding) // ENTER + UPDATE
    .style("stroke", "black");
```

D. Koop, CIS 468, Spring 2017
Merge

• Merge creates a new selection that includes the items from both selections
• If you want to update all elements (including those just added via enter), use merge!