D3: Data Driven Documents

David Leonard

City College of New York

October 5, 2014

What is D3?

D3 is a JavaScript library which allows you to bind arbitrary datasets to the Document Object Model (DOM) and exposes methods for performing data-driven operations to the document.

Loading Data

D3 provides numerous ways to introduce data to your objects.

- Array of values

Loading Data

D3 provides numerous ways to introduce data to your objects.

- Array of values
- Tab-seperated values

Loading Data

D3 provides numerous ways to introduce data to your objects.

- Array of values
- Tab-seperated values
- Comma-seperated values

Loading Data

D3 provides numerous ways to introduce data to your objects.

- Array of values
- Tab-seperated values
- Comma-seperated values
- JSON

Selectors

Using JavaScripts native API to manipulate the DOM can be tiresome. Libraries such as jQuery provide abstractions to make this easier, and D3 exposes its own methods for working with the DOM.

Manipulating paragraphs with JS

```
var paragraphs = document.getElementsByTagName("p");
for (var i = 0; i < paragraphs.length; i++) {
    var paragraph = paragraphs.item(i);
    paragraph.style.setProperty("color", "white", null);
}
```

Manipulating paragraphs with D3

Much simpler...

```
d3.selectAll("p").style("color", "white");
```

Appending an element

What if we wanted to create an element in our DOM?

```
svg.append("circle")
    .attr("cx", d.x)
    .attr("cy", d.y)
    .attr("r", 2.5);
```

Appending multiple elements

```
var svgContainer = d3.select("body").append("svg")
    .attr("width", 200)
    .attr("height", 200);

svg.selectAll("circle")
    .data(data)
.enter().append("circle")
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; })
    .attr("r", 2.5);
```

Understanding data joins

- svg.selectAll('circle') returns an empty selection, since the svg container was empty (the parent node).

Understanding data joins

- svg.selectAll('circle') returns an empty selection, since the svg container was empty (the parent node).
- Selection is joined to each datum in our data. The enter selection holds these placeholders for our data.

Understanding data joins

- svg.selectAll('circle') returns an empty selection, since the svg container was empty (the parent node).
- Selection is joined to each datum in our data. The enter selection holds these placeholders for our data.
- The missing elements (circle) are added to the SVG container by calling selection.append on the enter selection, which appends a new circle for each data point to the SVG container.

Using transitions

Transitions are simply methods to animate changes to the DOM.
For instance:

```
d3.selectAll('.chart')
    .selectAll('div')
        .data(data)
    .enter().append('div')
        .transition()
            .style('width', function(d) { return x(d) +
                'px'; })
        .transition()
        //.style("color", "red")
```

Adding interactivity to your graphs

Many D3 visualizations come with built-in interactivity. One way to achieve your own custom interactions is to use event handlers.

```
d3.selectAll('.chart')
    .selectAll('div')
        .data(data)
    .enter().append('div')
         .on('mouseover', function(d){
            console.log(d);
        })
```