Infovis Model Revisited + D3
Visualization Defined

The use of *computer-supported, interactive, visual representations of abstract data* to amplify cognition
--Card, Mackinlay and Schneiderman 1999

Transformation of the symbolic into *the geometric.*
-- McCormick et al., 1987
Infovis Reference Model

![Diagram showing the Infovis Reference Model with stages: Source Data, Data Tables, Visual Abstraction, Views, and Task. Data Transformations, Visual Mappings, and View Transformations are also shown.]

Image from: www.prefuse.org

--Card, Mackinlay and Schneiderman 1999
Visual Abstraction

• Mark: visual that represents the objects of interest

Images from protovis.org
Infovis Reference Model

--Card, Mackinlay and Schneiderman 1999

Image from: www.prefuse.org
Visual Abstraction

- Visual properties of marks:
  - Position, x, y, z
  - Size, length, area, volume
  - Orientation, angle, slope
  - Color, gray scale, texture
  - Shape
  - Animation, blink, motion

1. Map: data items $\rightarrow$ visual marks
2. Map: data attributes $\rightarrow$ visual properties of marks
Example: Spotfire

- Film database
- Film -> dot
  - Year → x
  - Length → y
  - Popularity → size
  - Subject → color
  - Award? → shape
Infovis Reference Model

--Card, Mackinlay and Schneiderman 1999
Views

- Presentation of the Data
  - Brushing & linking
  - Overview + Detail
  - Zooming and panning
  - Focus + Context
Enter D3...

- Javascript Library
- Facilitates
  - Binding data to the DOM
  - Creating visual Marks in the DOM
  - Setting attributes of those marks, based on the bound data
  - Updating attributes of those marks
D3: Selections

• Primary tool of D3
• `.Select` or `.SelectAll (domElement)`
• Returns a selection
  – An array of elements
  – Augmented with a bunch of useful methods

```javascript
d3.select("body").selectAll("div")
```
Aside: Protoypal Inheritance

- Javascript is class-free
- Any object can inherit from any other object
- A new object can inherit properties from an old object
- We can augment previously defined types, even the primitive types with additional functionality
- This is how D3 augments the selector arrays!
D3: DOM & Data Binding with

- `.data( dataVals)`
  - tell the DOM about the data
- `selection.enter( element)`
  - `.enter` creates placeholder for the data values

```javascript
.data(dataset)
.enter()
```
• Bind data to element

```javascript
.append("div")
```

• Set the attributes

```javascript
.attr("class", "bar")
.style("height", function(d) {
    var barHeight = d * 5;
    return barHeight + "px";
})
```
• Using div as a mark is limiting!
• SVG is much more flexible
• SVG
  – Scalable Vector Graphics
  – Textual descriptions of graphics so it scales better than pixels!
  – supported by most modern browsers
SVG Basics

SVG Element: width, height in pixels

0,0

width

height

100, 35

300, 125
• rect
• circle
• ellipse
• line
• text
• ...and more
• All are added to the DOM as elements and can be manipulated by D3