CIS 467/602-01: Data Visualization

Introduction to D3

Dr. David Koop
Recap (Marks & Channels)

• How do we encode data visually?
  - **Marks** are the basic graphical elements in a visualization
  - **Channels** are ways to control the appearance of the marks

• Marks:
  - Points
  - Lines
  - Areas
Recap (Channels)

- **Position**
  - Horizontal
  - Vertical
  - Both

- **Color**

- **Shape**
  - △  ★  /  L

- **Tilt**
  - \[///\]

- **Size**
  - Length
  - Area
  - Volume

[Munzner (ill. Maguire), 2014]
Recap (Effectiveness)

**Channels:** Expressiveness Types and Effectiveness Ranks

**Magnitude Channels:** Ordered Attributes

- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)

**Identity Channels:** Categorical Attributes

- Spatial region
- Color hue
- Motion
- Shape

[Munzner (ill. Maguire), 2014]
Relative vs. Absolute Judgments

- Weber’s Law:
  - We judge based on relative not absolute differences
  - The amount of perceived difference depends is relative to the object’s magnitude!

[Unframed Unaligned]
[Unframed Aligned]
[Framed Unaligned]

[Munzner (ill. Maguire), 2014]
Luminance Perception

Edward H. Adelson

[E. H. Adelson, 1995]
Luminance Perception

[Edward H. Adelson, 1995]
More JavaScript Background

• Building Blocks: HTML, CSS, SVG, and JavaScript

• More Ideas:
  - JavaScript Libraries
    • `<script src="http://d3js.org/d3.v3.js" charset="utf-8"></script>`
  - Minification: smaller code, no functional change
    • `<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>`
    • Can make debugging more difficult
  - Content Delivery Networks
    • Faster delivery of Web content, also works for js
      • `https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.5/d3.min.js`
JavaScript Functions

• Object-oriented programming: classes have definitions with fields and methods

• JavaScript Objects:
  - var obj;
  - var obj = {};
  - var obj = {x: 30, y: 50};
  - console.log(obj.x)

• JavaScript Functions:
  - function test() { console.log(“Hello World”); }  
  - var f = test;
  - var g = function() { console.log(“Hello Universe”); }
JavaScript Closures

- Functions can return functions
  - function x() { return function y(a) { return a+1; } }

- **Closures**: functions that remember their environment
  - Nested functions keep track of variables outside of their definition (independent variables)
  - function x(a) { return function y(b) { return a+b; } } 
    var z = x(3);
    console.log(z(5));
Method Chaining

• Methods can also return the objects passed in or derivative objects to allow you to call another function on the result

• You often end up following specific patterns where an object being manipulated requires multiple calls:
  - rect.attr("width", 200).attr("height", 100);

• Or it is clear that the method returns a specific object that you wish to make changes to:
  - svg.select("myrect").style("fill", "blue");

• Of course, you may store the returned object as a variable and make each call separately

• Coding style: Indentation, often put each call on a new line
Selections

- Remember CSS selections?
  - id and class attributes
  - document.getElementById()
  - document.querySelector(), element.querySelector()
Selections

- The **element types** themselves (the HTML tag)
  - strong

- **Classes** of elements (ties to HTML `class` attribute)
  - .cool

- A **specific** element (ties to HTML `id` attribute)
  - #main-section

- Relationships
  - Descendant: `p em`
  - Child: `p > em`
Data-Driven Documents (D3)

• [http://d3js.org/](http://d3js.org/)
• Original Authors: Mike Bostock, Vadim Ogievestky, and Jeff Heer
• Open Source
• Focus on Web standards, customization, and usability
• Grew from work on Protovis: more standard, more interactive
• By nature, a **low-level** library; you have control over all elements and styles if you wish
• A top project on GitHub (35,373 stars as of 2/17/2015)
• Lots of impressive examples
  - Bostock now a New York Times Graphics Editor
  - [http://bost.ocks.org/mike/](http://bost.ocks.org/mike/)
D3 Key Features

• Supports data as a core piece of Web elements
  - Loading data
  - Dealing with changing data (joins, enter/update/exit)
  - Correspondence between data and DOM elements
• Selections (similar to CSS) that allow greater manipulation
• Method Chaining
• Integrated layout algorithms, axes calculations, etc.
• Focus on interaction support
  - Straightforward support for transitions
  - Event handling support for user-initiated changes
D3 Introduction

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

• http://vadim.ogievetsky.com/IntroD3/

• Other references:
  - Murrary’s book on Interactive Data Visualization for the Web
  - The D3 website: d3js.org
Next Class

- More on D3
- Assignment 2