Intro to D3 and Simple Statistics

plus JavaScript Basics, Debugging, and SVG

Class 4

Class Outline

- Questions?
  - Assignment 2: Project Proposal
- Today: **Our first "Hands On" session**
  - Wrapping up last class: *Web Development Crash Course Part 2*
  - Some useful JavaScript tips
  - Basics of SVG
  - Introduction to D3 and Simple Statistics
- Coming Soon
  - **Assignment 2 due next week**
  - **Get ready to form your project teams:**
    - Proposals due Tuesday @ 9:30am
    - Distributed to everyone by 5pm
    - Preferences due Wednesday @ 11am
    - Form teams Thursday in class

See "Team Formation Process" link on class schedule for more details.

Wrapping Up the Last Lecture

Web Development Crash Course Part 2

- The Chrome Debugger
  - Scripts
  - Breakpoints
  - Stepping in, over, and out
  - Watch expressions

- Example file:
  - http://ils.unc.edu/~gotz/courses/visual_analytics/debug.html

Hands On Session

Philosophy: We'll Code Together

- **I will be writing code on the fly**
  - Online manuals are filled with already-written code
  - In contrast:
    - I'll make errors!
    - I'll forget syntax or names for functions!

- **You should try to follow along and do the same work**
  - Learning by trial and error
  - The debugger
  - Use examples, documentation, and search engines
  - Help me get it right!

- **Risks and rewards**
  - Should provide you with a much richer experience than reviewing online documentation
  - We might "get stuck"
    - **I will abandon examples if they prove too time consuming…**

Piazza

- **Today's examples (and future hands on exercises) will be posted to Piazza!**

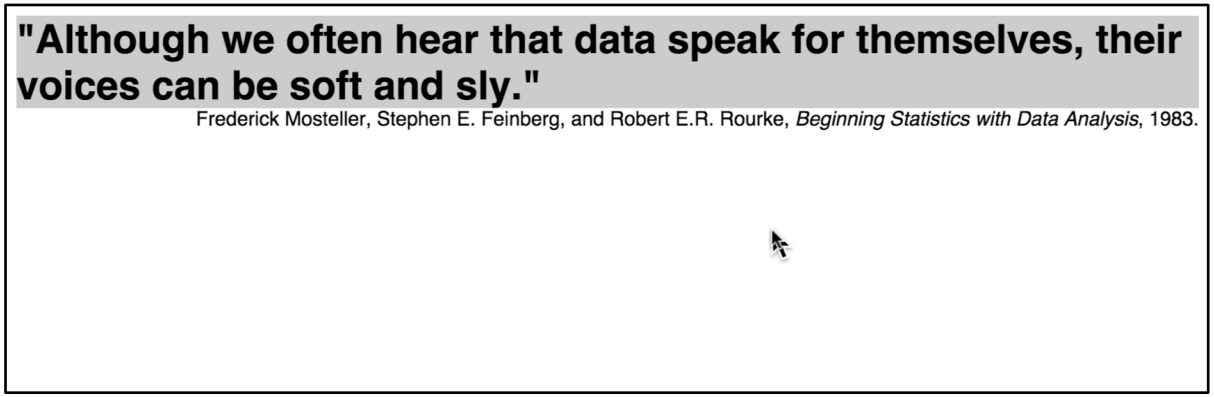
- Be sure you are registered

Getting Started

- **You should have the "handson" files. If not…**
  - Download handson1.zip from Sakai
  - It is posted to the "Resources/Hands On Materials" folder
  - Unzip the file in a convenient location

Test Your Environment

- **Open handson1.html in Chrome**
- It should look like this:



- The browser console should have a message:

```
handson.js was included in this web page.
```

Hands On: Useful JavaScript

- Functions
- JavaScript objects
- Arrays and iteration
- Array “filter()” and “map()”

Introduction to SVG

- Scalable Vector Graphics
  - Web standard for vector graphics
  - Most common way to use D3 for visualization

- Lines
- Circles
- Squares
- Polygons
- Transforms

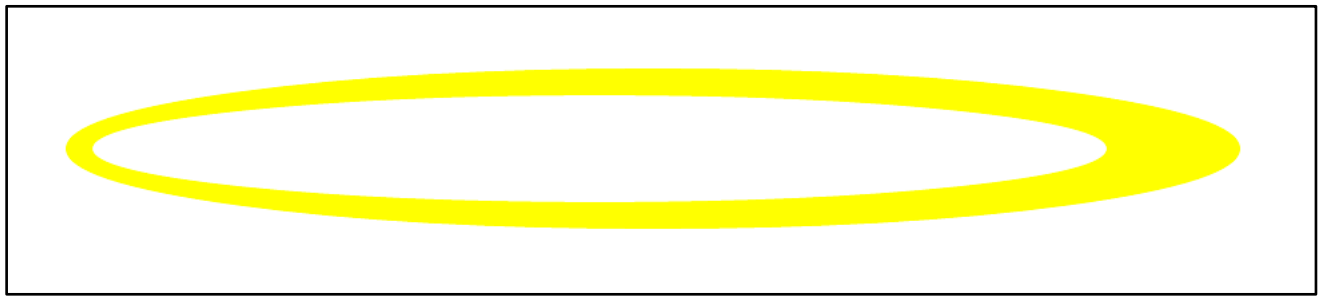
Simple SVG Scene

- Looks a lot like other HTML elements:

```
<html>
<body>

<svg height="100" width="500">
  <ellipse cx="240" cy="50" rx="220" ry="30" style="fill:yellow" />
  <ellipse cx="220" cy="50" rx="190" ry="20" style="fill:white" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```



- Let's look at some examples…
  - https://www.w3schools.com/graphics/svg_examples.asp

SVG Elements and Shapes

- <svg> … </svg>
  - The element on which the shapes will be drawn.
- **Shapes**
  - <rect>
  - <circle>
  - <ellipse>
  - <line>
  - <polygon>
  - <polyline>
  - <path>
  - <text>
- **More advanced:**
  - Groups (<g>) are containers. A bit like “divs” for regular HTML.
  - Transforms can shift/rotate SVG elements (e.g., turn a <rect> into a diamond)
    - Not an element itself, but an ***attribute*** of an element
    - <g transform=“rotate(45)”>
    - For more info: http://www.w3.org/TR/SVG/coords.html#TransformAttribute

SVG and the DOM

- SVG adds new element types to the DOM
  - Still elements with attributes
  - CSS can be used to “style” svg elements
    - fill
    - stroke
    - stroke-width
- Drawing is done by composing an SVG “scene”
  - Represented as a DOM subtree on a web page
  - Shapes are drawn in the order they appear in DOM

Hands On: SVG

- Scene Definition
- Interaction

D3 and the DOM

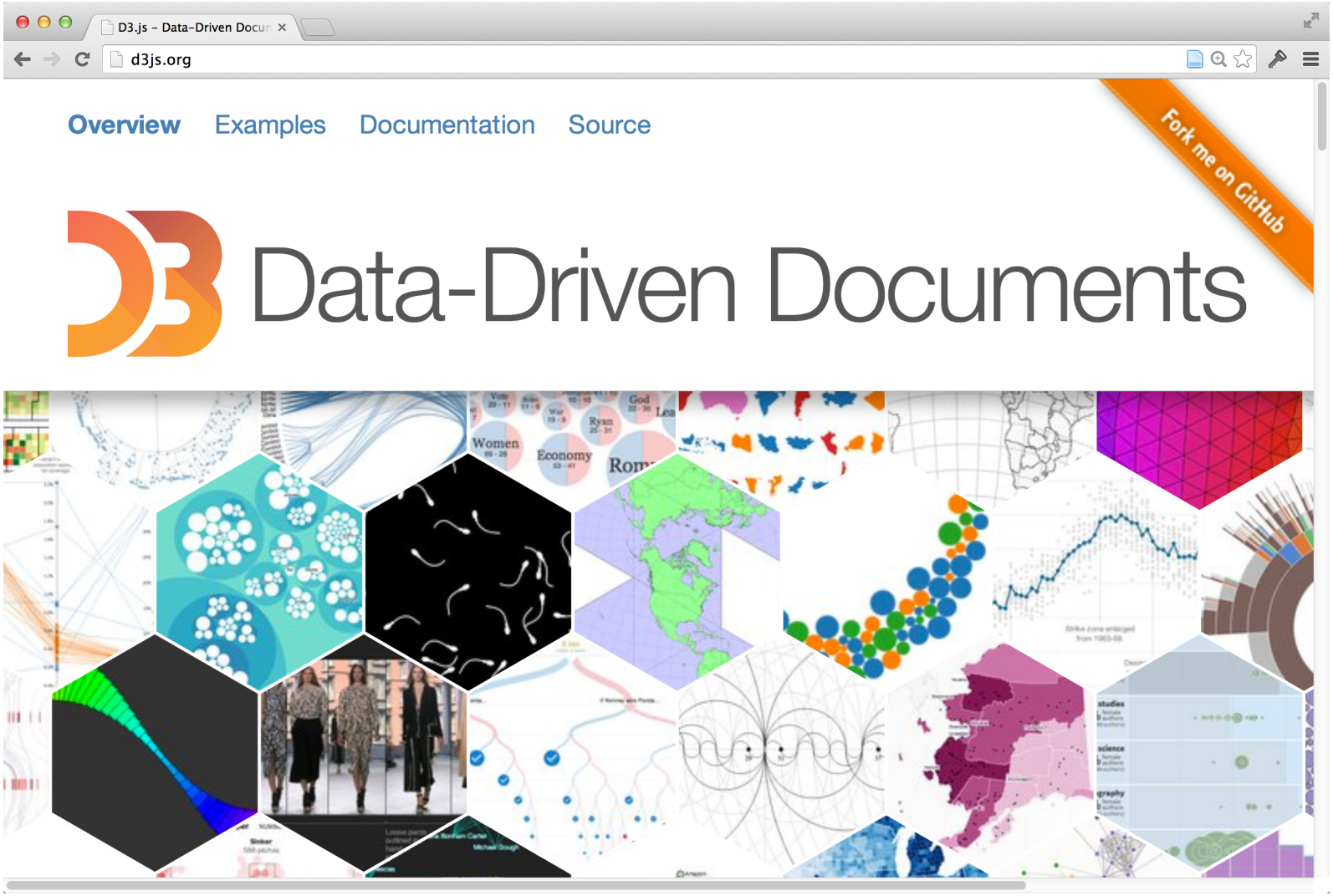
- D3 is largely a DOM manipulation language
  - Add nodes
  - Remove nodes
  - Update nodes

- D3 == **D**ata **D**riven **D**ocuments

D3 and the DOM

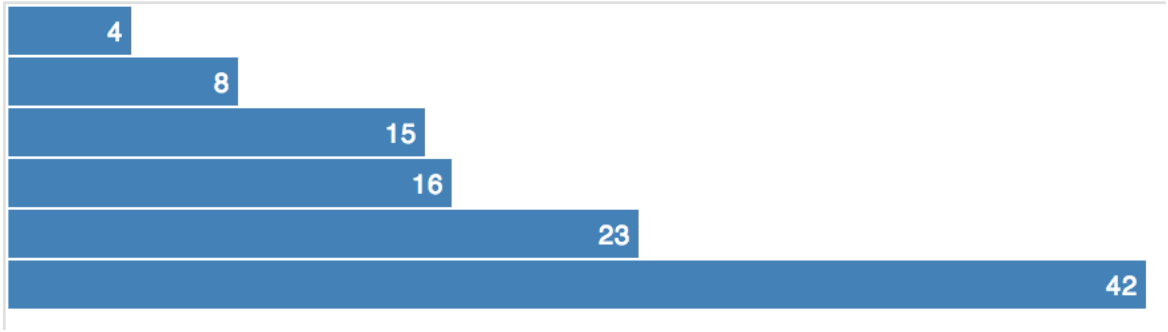
- Manipulation is driven by data
  - **Mappings** (defined as JavaScript functions) define correspondence between **Data** and **DOM Elements**
- When the DOM elements are SVG elements
  - We have a graphical **visualization**
  - Need not be SVG!
    - Canvas, “plain HTML” including color-coded DIVs.

D3js.org


D3.js – Data-Driven Docu
d3js.org
Overview
Examples
Documentation
Source
Fork me on GitHub
Data-Driven Documents

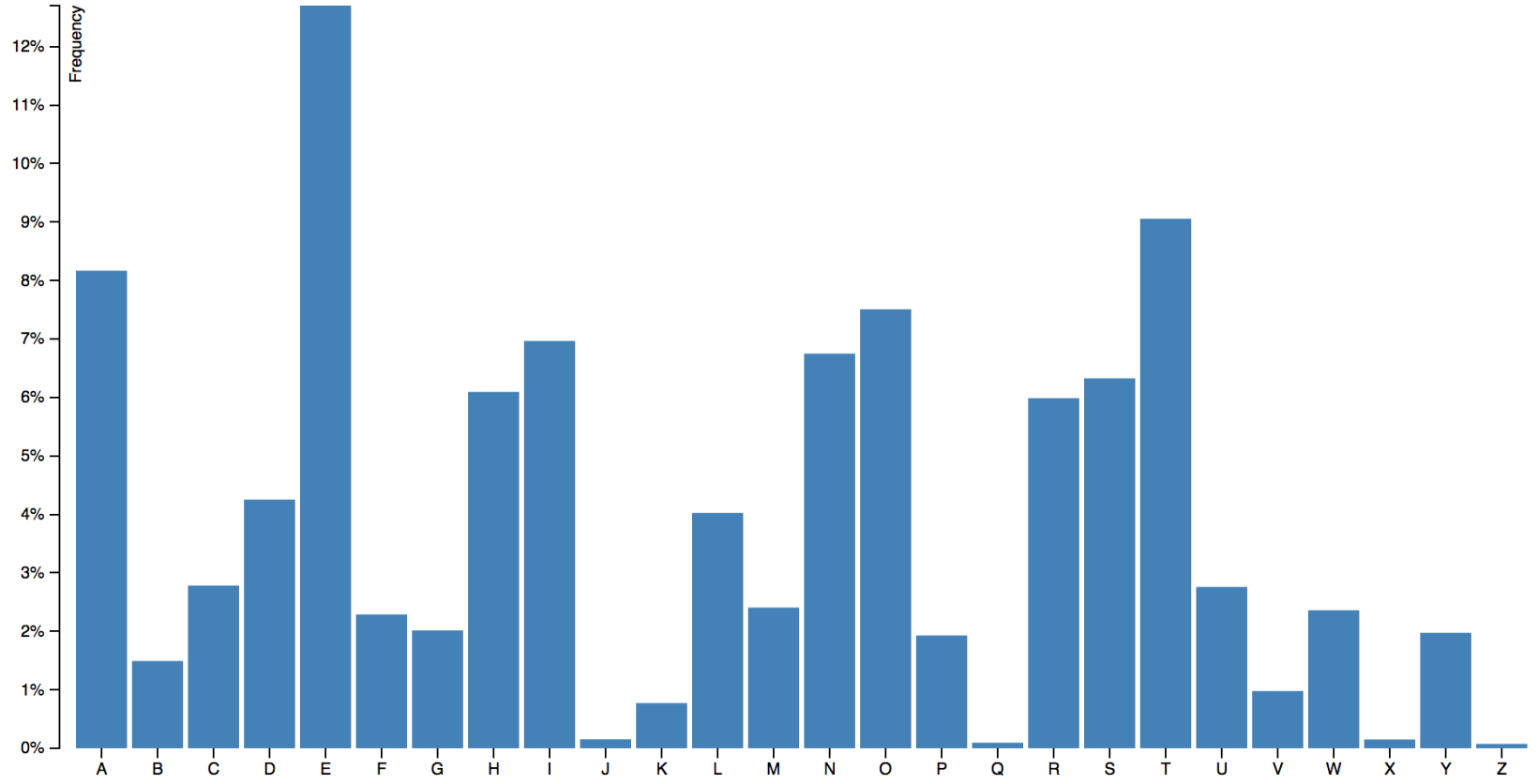
An HTML-based Example

- https://bl.ocks.org/davegotz/8c0baa7a34a137af33fd28498e269a75



An SVG-based Example

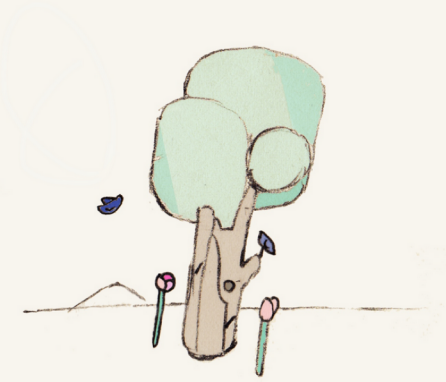
- https://bl.ocks.org/davegotz/93eae24cb3b0cf8f209334d0626bd62f



Simple Statistics

http://simplestatistics.org/

SIMPLE STATISTICS



Statistical methods in readable JavaScript for browsers, servers, and people.

GITHUB NPM DOCS

Star 862 Tweet 7

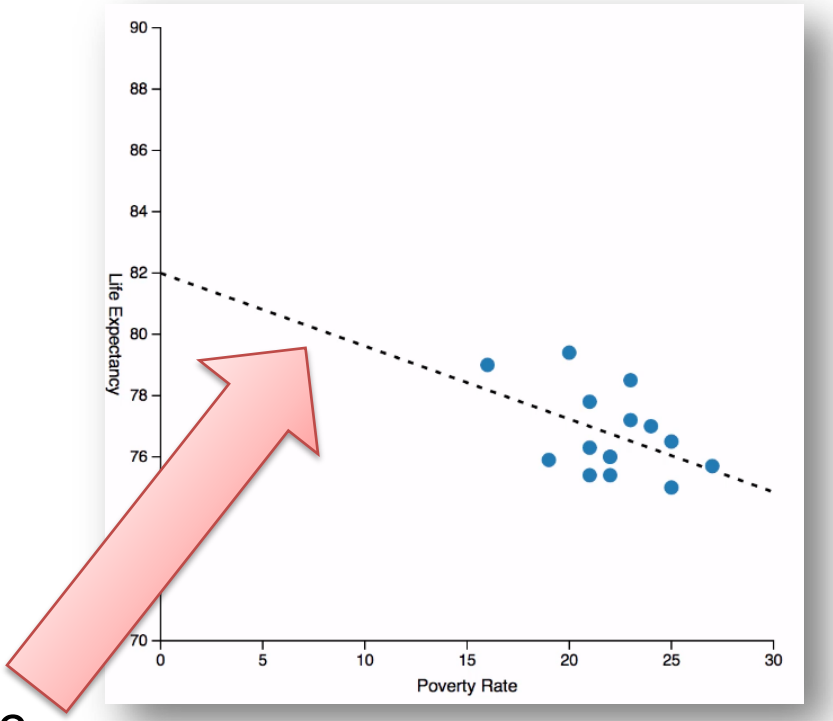
FEATURES

Simple Statistics

- Does NOT manipulate the DOM
- Basic “library” design
  - Functions
  - You call them with parameters
  - They return results
- Examples
  - Correlation
  - Regression
  - Classification

Simple Statistics

- **We'll use Simple Statistics to generate new data**
  - Used to guide data transformation
    - e.g., filter to a given class
  - Used to feed D3-based visualization
    - e.g., draw a regression line



Using D3
and Simple Statistics

- Include in your web page
  - Using the <script> tag
- Make function calls

We'll talk more about this in future classes…