Data Visualization on the Web with D3 and Leaflet

Shruti Mukhtyar

Web Developer

Geospatial Innovation Facility, UC Berkeley

Projects: Cal-Adapt, VTM

Twitter: *@mapchitra* *@cal_adapt*

Today's Plan

- Brief overview of Data Visualization
- Where does D3 fit in?
- Whirlwind introduction to Web Development
- Explore D3 Core Concepts and build a bar chart
- Web mapping with Leaflet
- Discussion (Visualization, Web Development, Learning, Deeper dive into some examples, etc.)

Brief Overview of Data Visualization

What is Data Visualization?

- Graphical display of abstract information for communication and sense-making

Goals

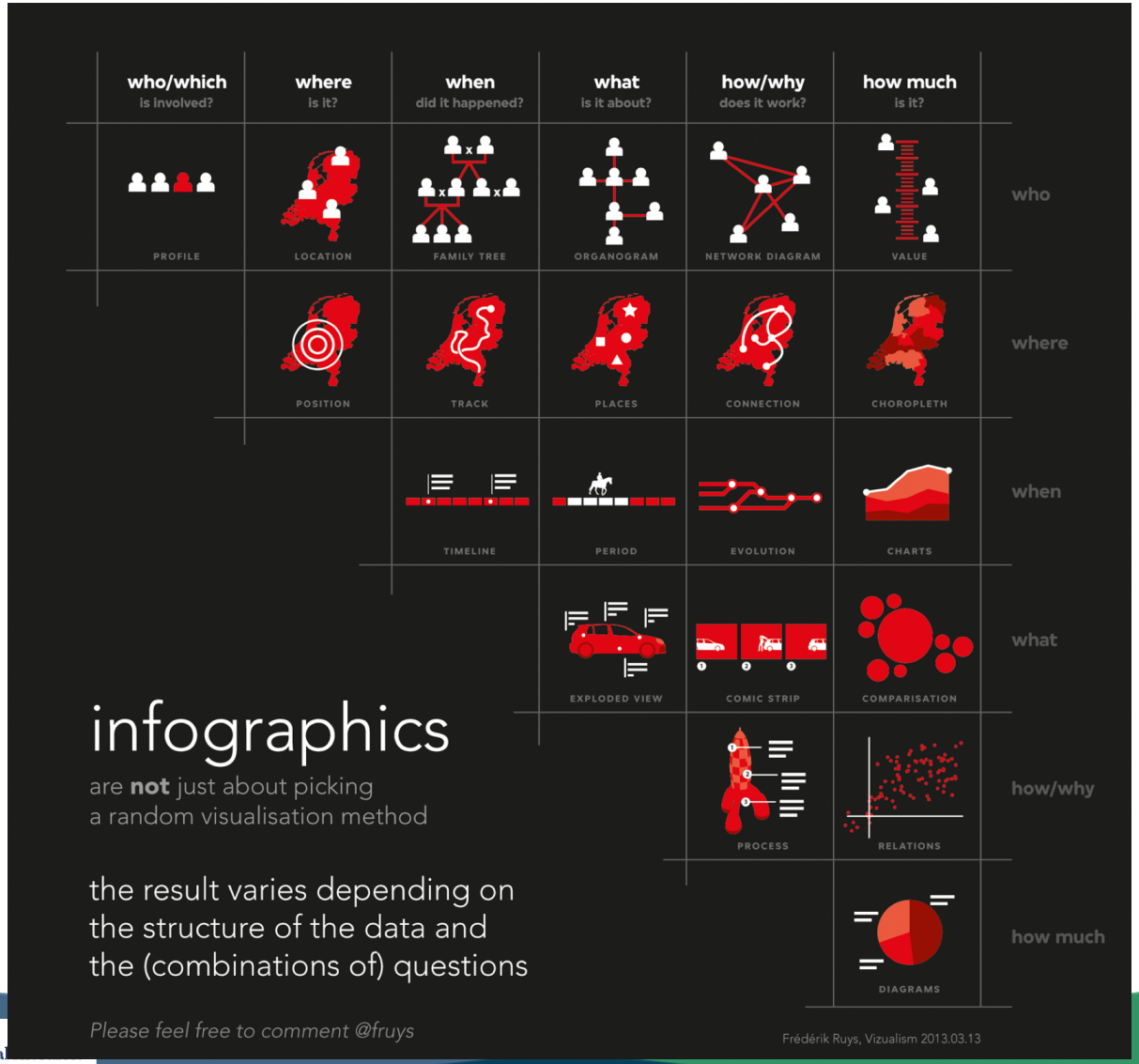
- Using graphics with or without explanatory texts for Communication
- Displaying data for exploratory Analysis
- The Changing Goals of Data Visualization

Types of Quantitative Messages (Stephen Few)

- Timeseries: A single variable is captured over a period of time
- Ranking: Categorical subdivisions are ranked in ascending or descending order
- Part-to-whole: Categorical subdivisions are measured as a ratio to the whol
- Deviation: Categorical subdivisions are compared against a reference
- Frequency Distribution: Number of observations of a particular variable for given interval
- Correlation: Comparison between observations represented by two variables
- Nominal Comparison: Comparing categorical subdivisions in no particular order
- Geographic: Comparison of a variable across a map or layout

Form follows function

Link



Today's goal

- What kind of visulizaion to build? ✗
  - List of books
  - Alberto Cairo, Cole Nussbaumer Knaflic, Andy Kirk
  - Edward Tufte, Stephen Few, Tamara Munzner,
- How do you go about building the visualization you want? ✔

Why publish on the web?

- Fastest way to reach a larger audience
- Working with web-standard technologies means that your work can be seen and experienced by anyone using a recent web browser, regardless of the operating system and device type.
- Avoiding proprietary software (e.g. GIS)

Technology Strategy

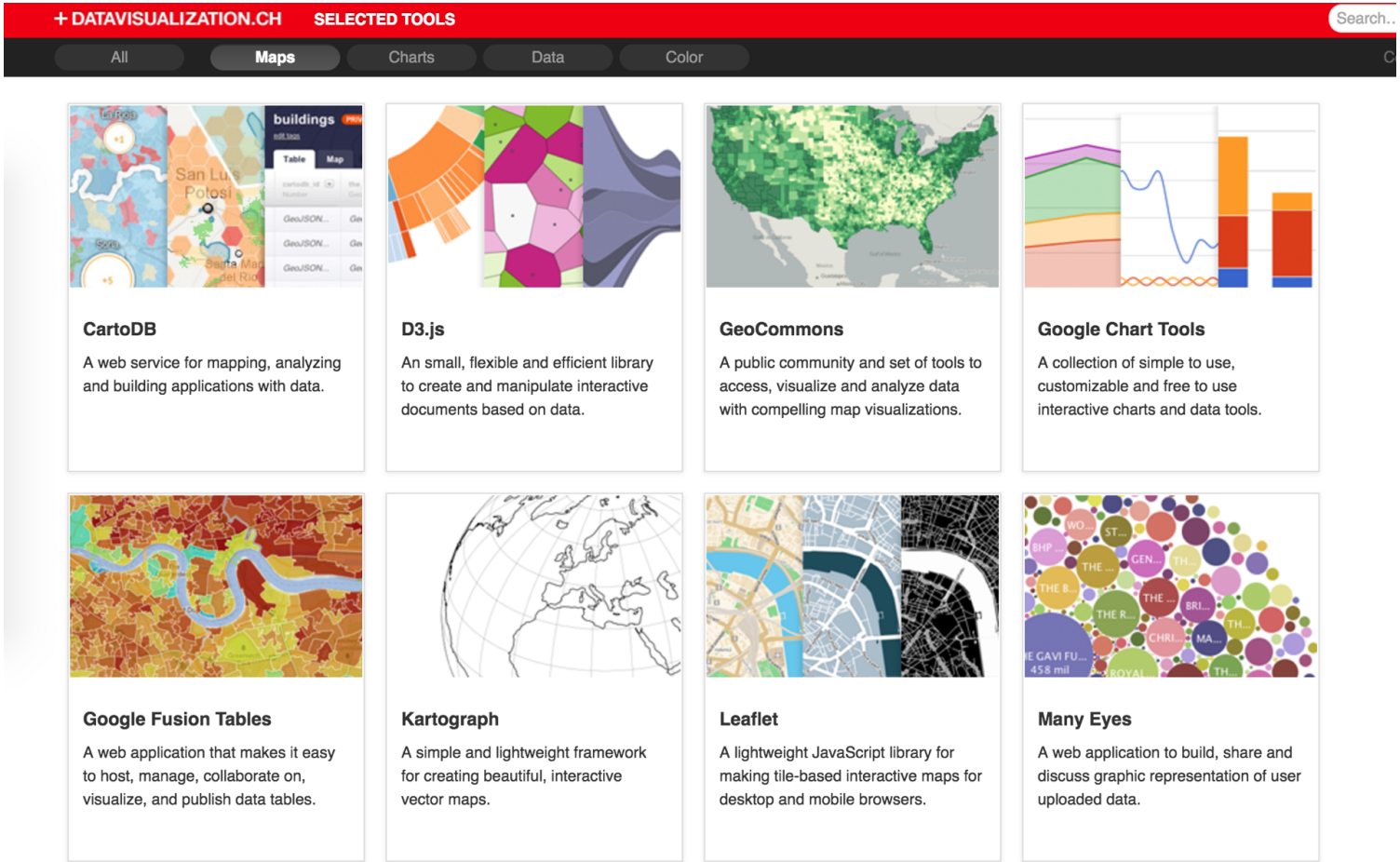
- Who's your target audience?
- How much data to you have? Where is it?
- Is the web map/dataviz going to part of an existing website?
- Do you need to support tablets and mobiles?
- Do you need to support offline access?
- Public or private?

Data Strategy

- Data has a longer life-cycle
- Web presentations of data age quickly

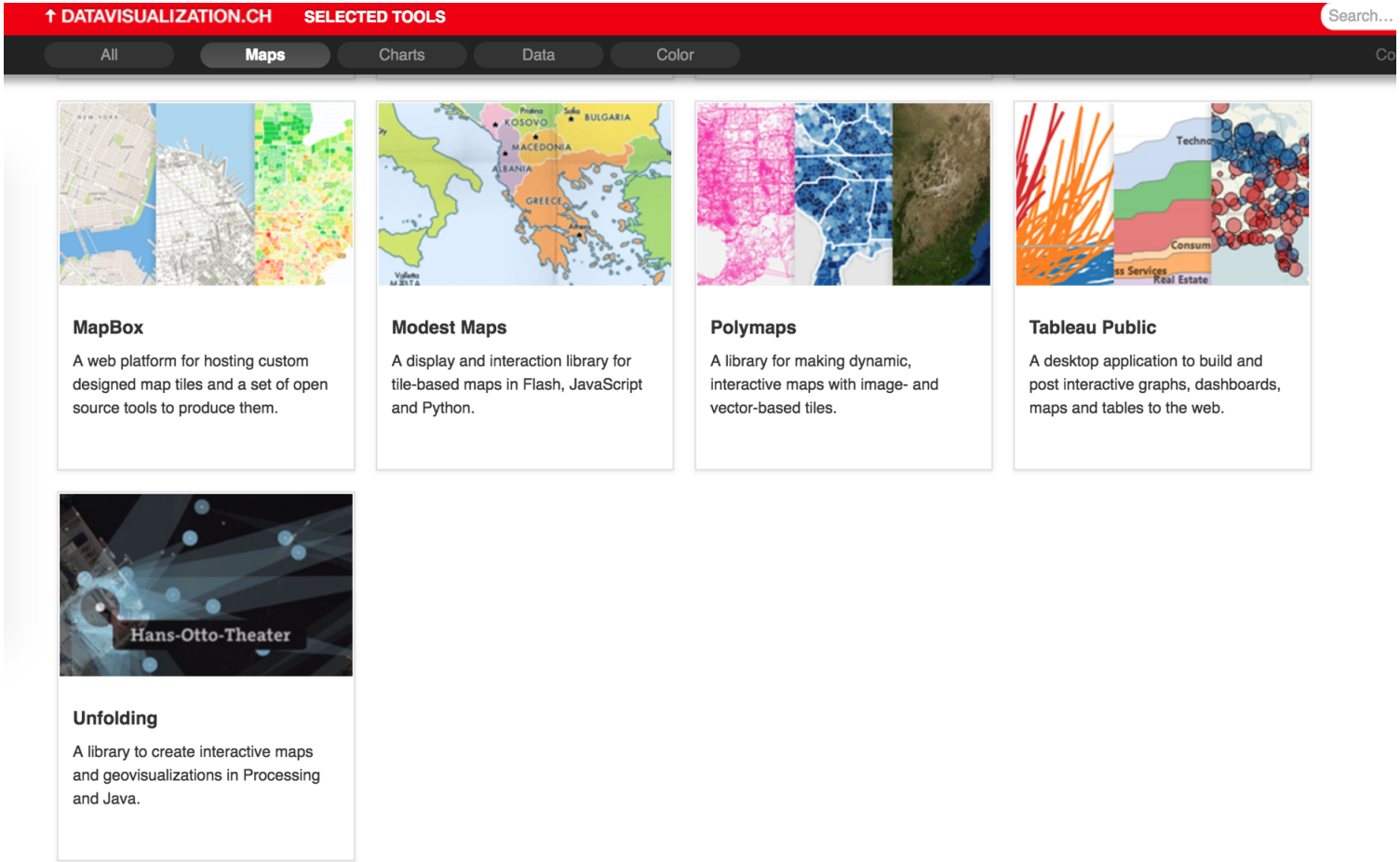
Visualization Tools

Link



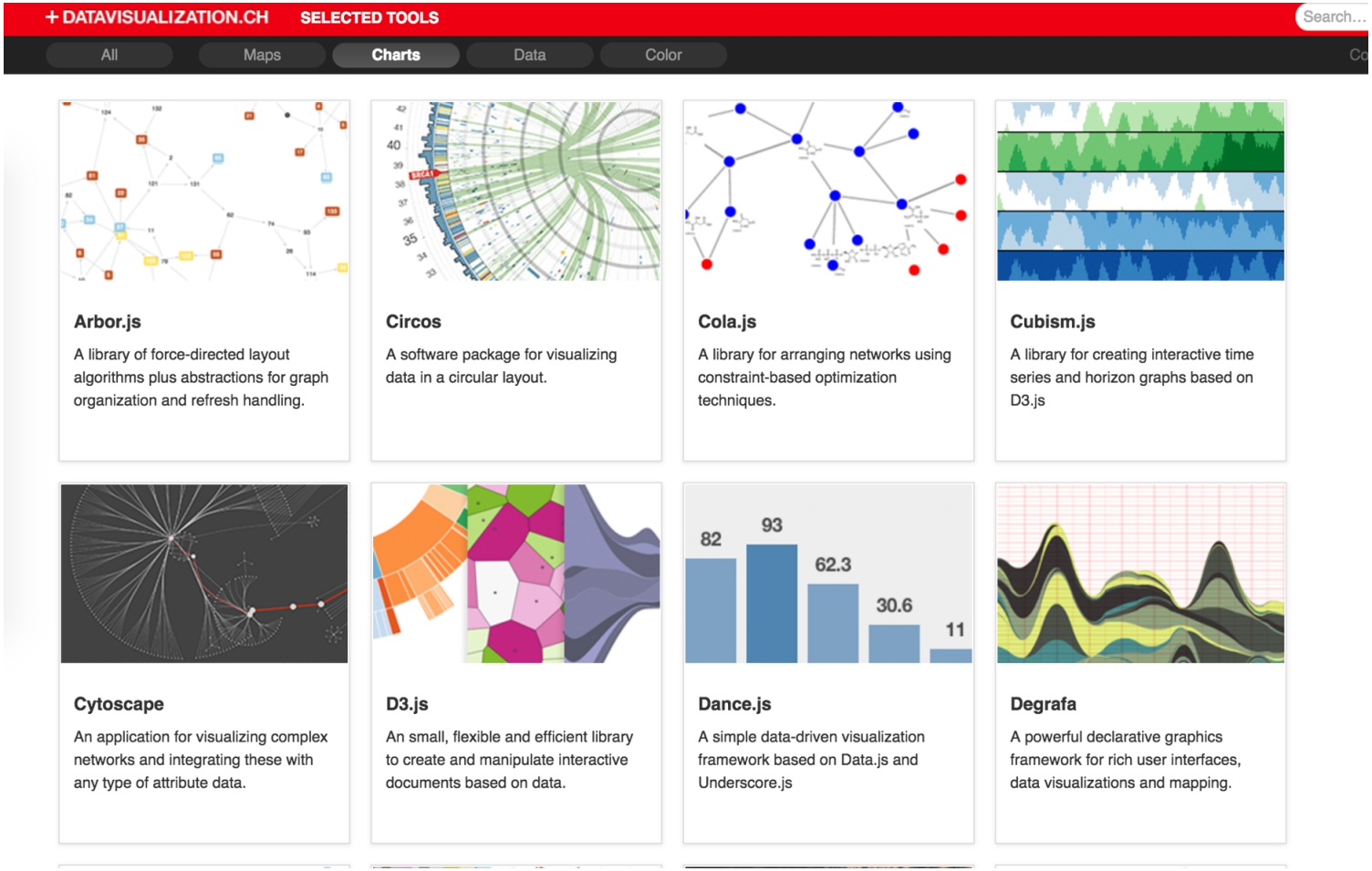
Visualization Tools

Link



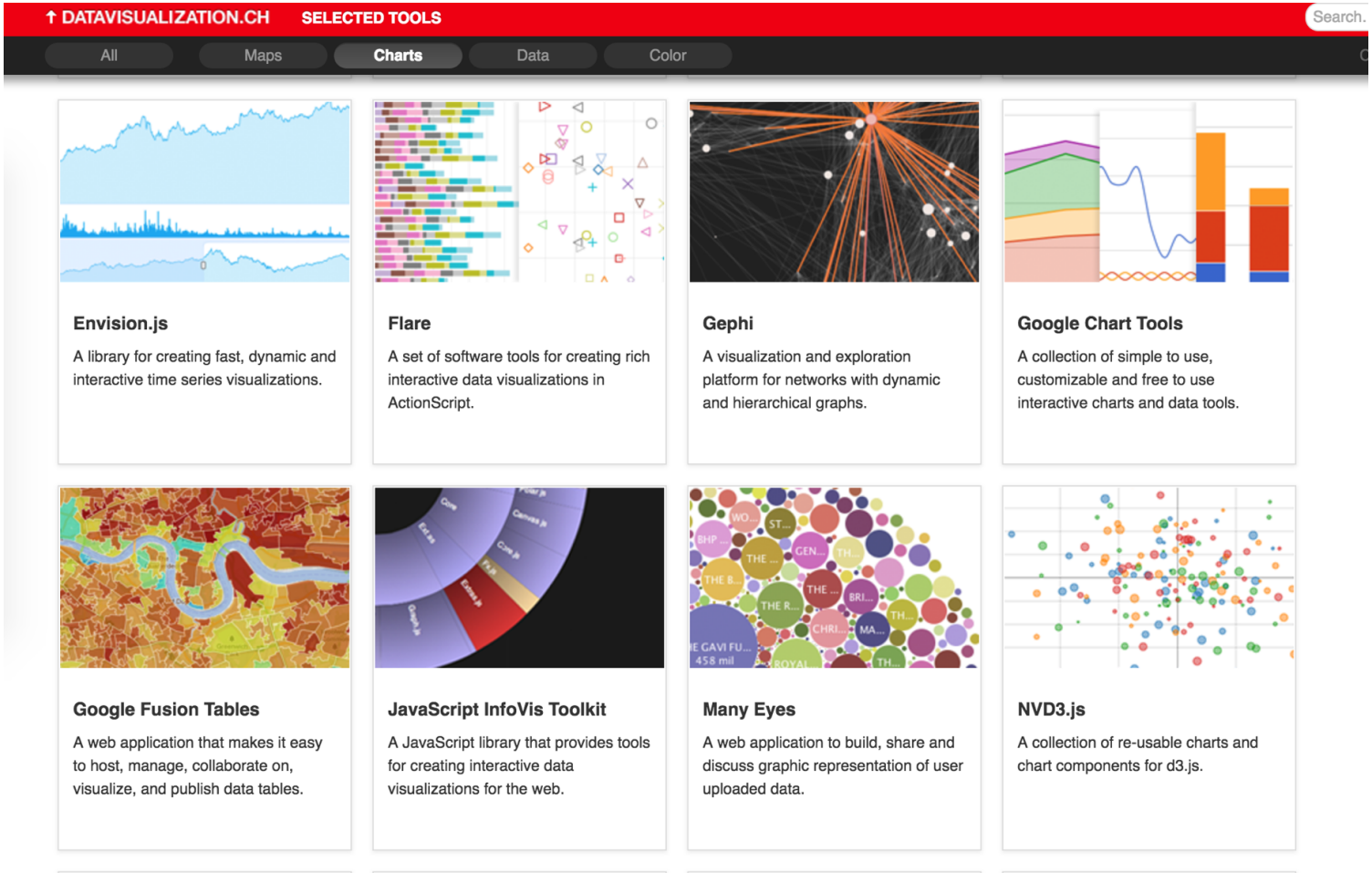
Visualization Tools

Link



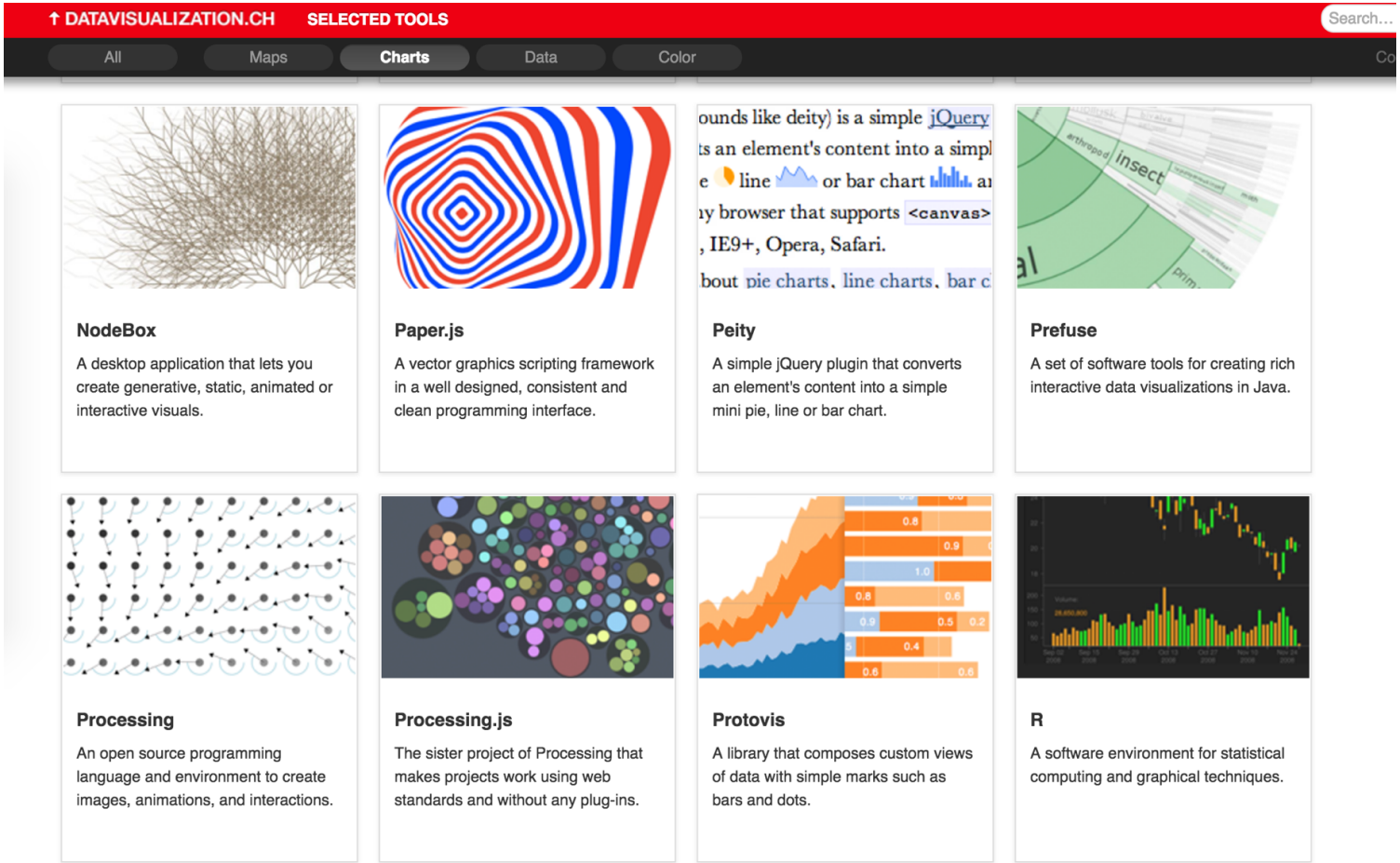
Visualization Tools

Link



Visualization Tools

Link



Where does D3 fit in?

What is D3?

- From d3js.org:
  - 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.

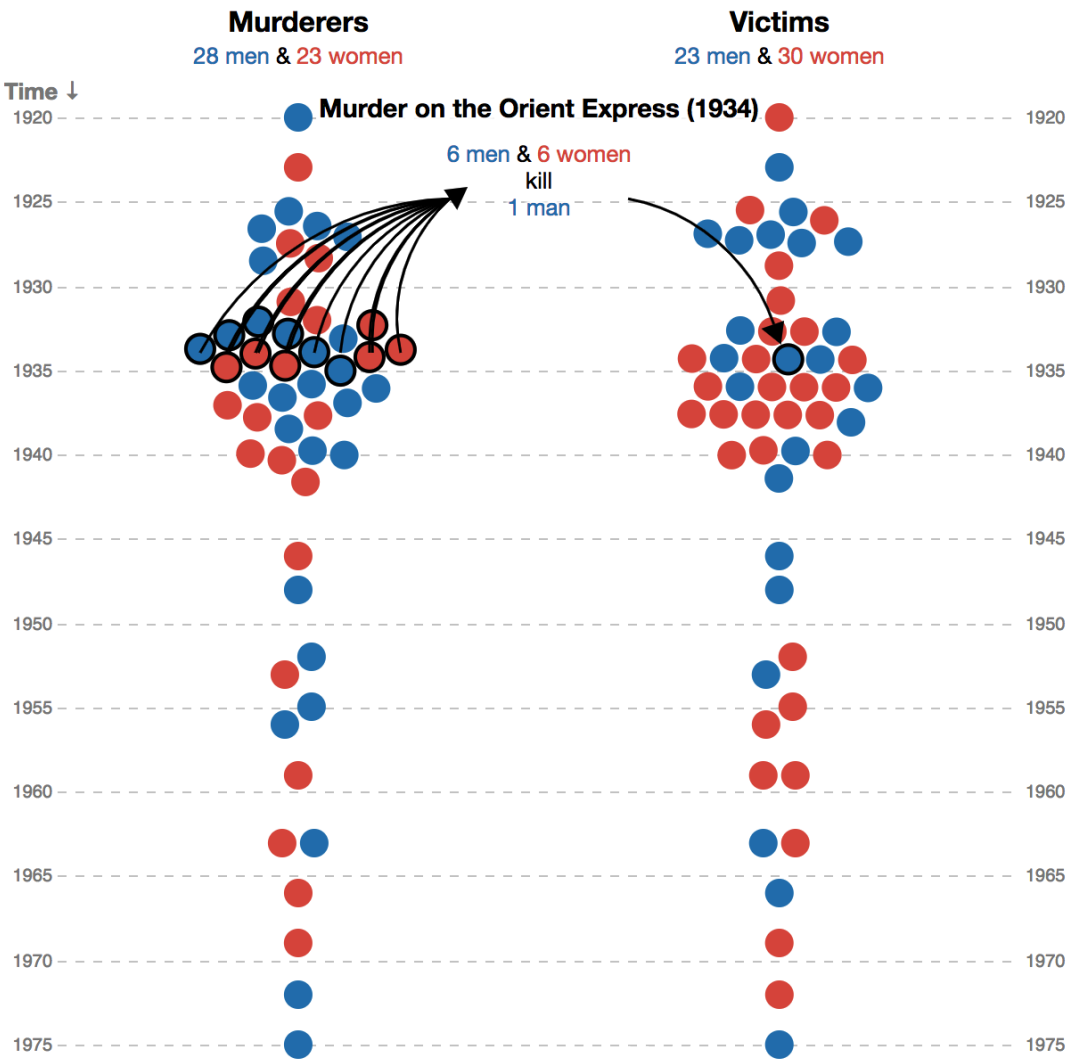
Why learn D3?

- Create highly customizable and interactive visualizations on the web
- Knowing D3 = Hirable skills
- Opportunity to learn web development
- Thousands of examples and a great open source community
- 5 reasons to learn D3

Linked Beeswarm Plot

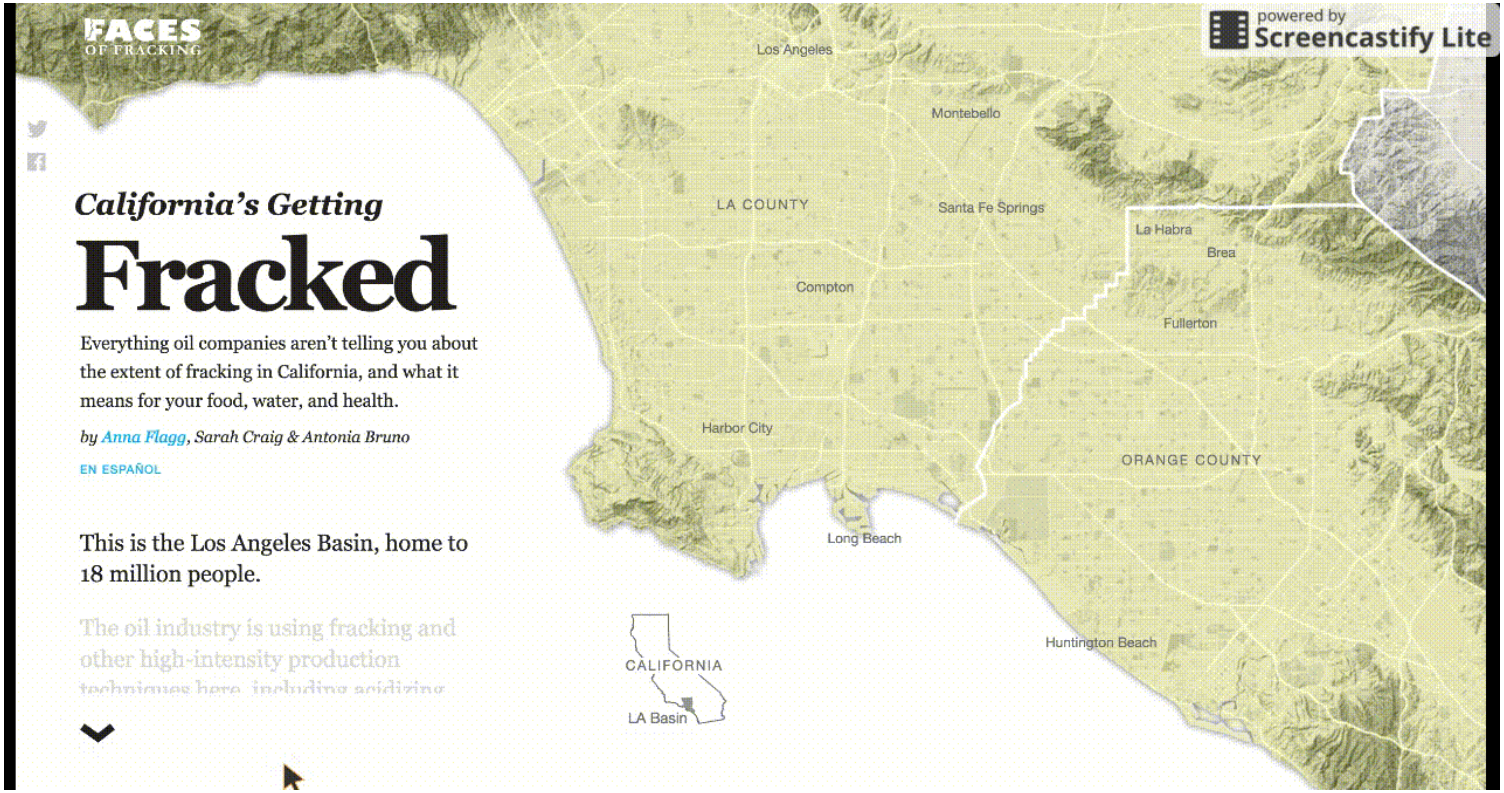
Link

Each **circle** represents a murderer or a victim. Earlier stories are on **top**; later ones are on **bottom**. **Hover or tap** on a circle for more information.



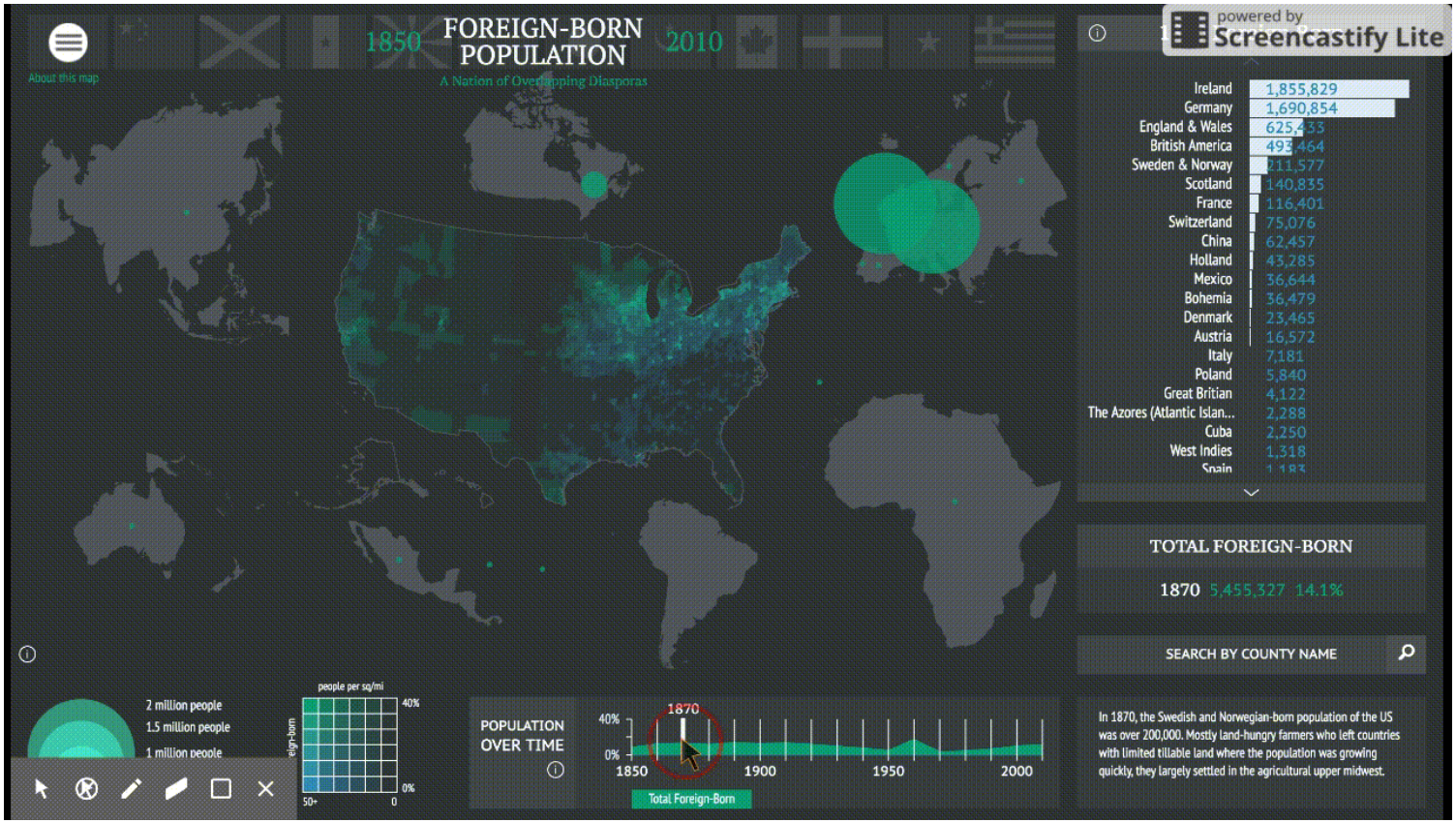
Face of Fracking

Link



American Panorama - Foreign Born Population

Link



Simpson's Paradox

Link

Proper Pooling

By "properly pooled," the investigators at Berkeley meant "broken down by department." Men more often applied to science departments, while women inclined towards humanities. Science departments require special technical skills but accept a large percentage of qualified applicants. In contrast, humanities departments only require a standard undergrad curriculum but have fewer slots.

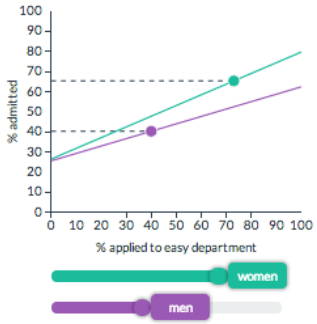
The authors concluded that any sexism occurred before Berkeley ever saw the applications:

> Women are shunted by their socialization and education toward fields of graduate study that are generally more crowded, less productive of completed degrees, and less well funded, and that frequently offer poorer professional employment prospects.
>
> — (p. 403)

To the right are data on the six largest departments, but the names have been changed to protect the innocent.

Illustration

Suppose there are two departments: one easy, one hard ('hard' as in 'hard to get into'). The sliders below set what percentage each gender applies to the easy department. Both departments prefer women, but if too many women apply to the hard one, their acceptance rate drops below the men's.



	# applied		# admitted		% admitted	
departments	men	women	men	women	men	women
"Easy"	1,076	1,340	672	1,068	62% 38%	20% 80%
"Hard"	1,615	495	413	131	26% 74%	27% 74%
Combined	2,691	1,835	1,085	1,199	40% 60%	35% 65%

Simpson's paradox? no

Generative Art with D3

Link

Optionally…

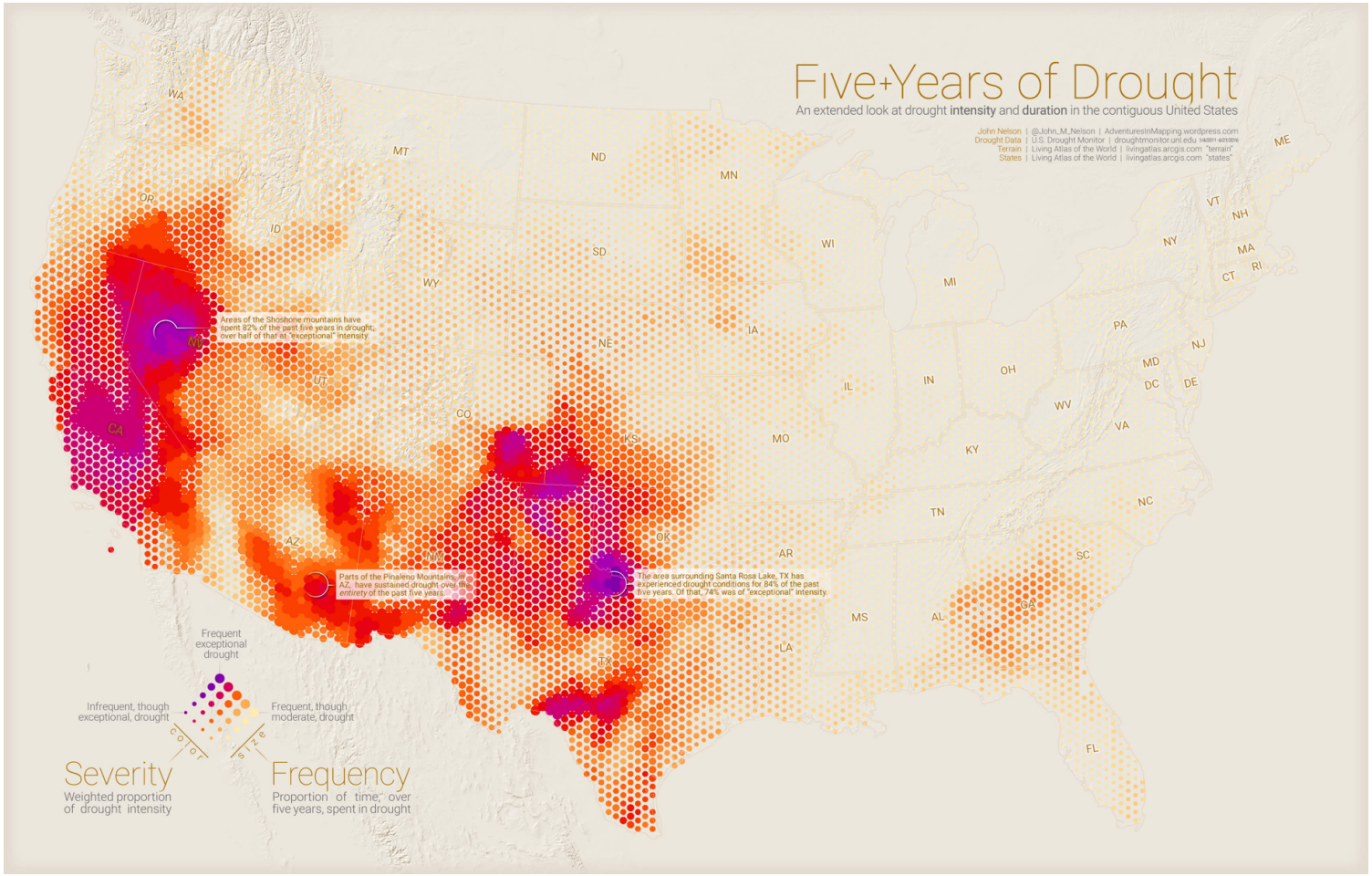
- Software
  - Tableau, Excel, GIS software
- Data visualization servies
  - Datawrapper, RAW Graphs
- R or Python
  - Associated visualization packages such as ggvis, rCharts, and Shiny
- Create interactive graphics but less customization options and/or limited pre-existing graphics types (e.g., bar charts).

Visualizing Friendships - R

Link

Five Years of Drought - ArcGIS Pro

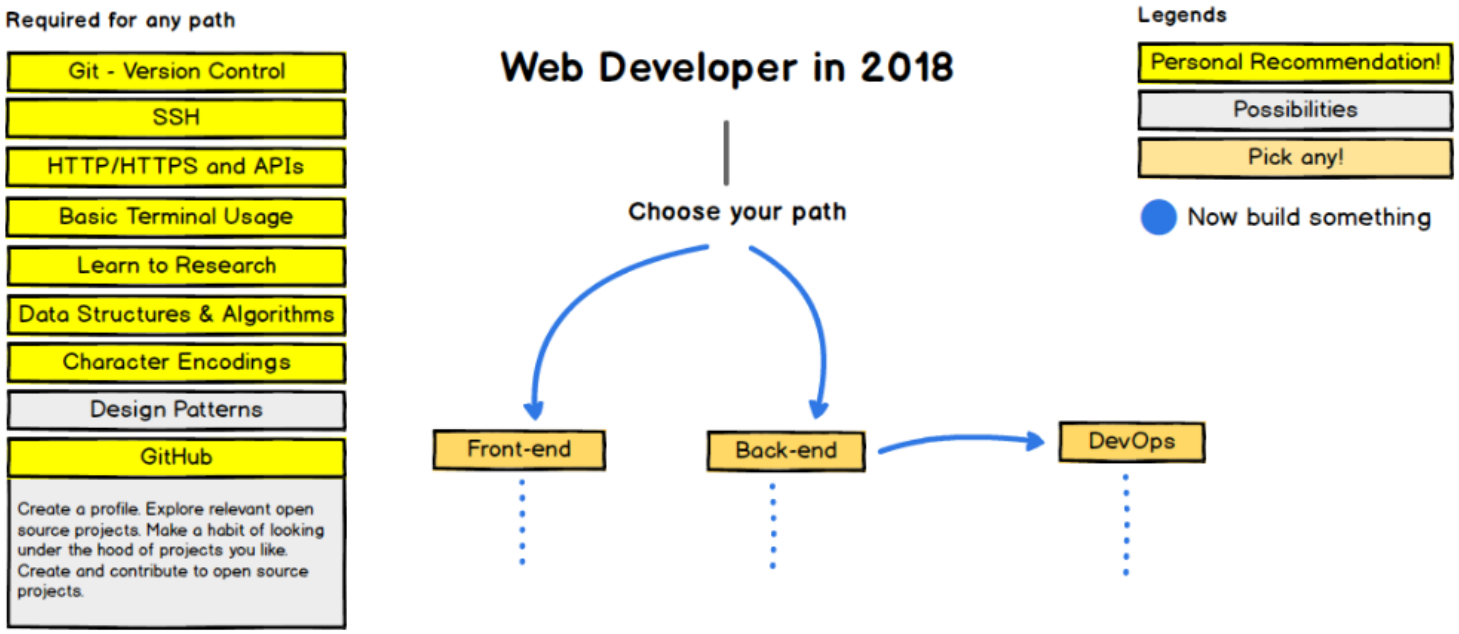
Link



Whirlwind Introduction to Web Development

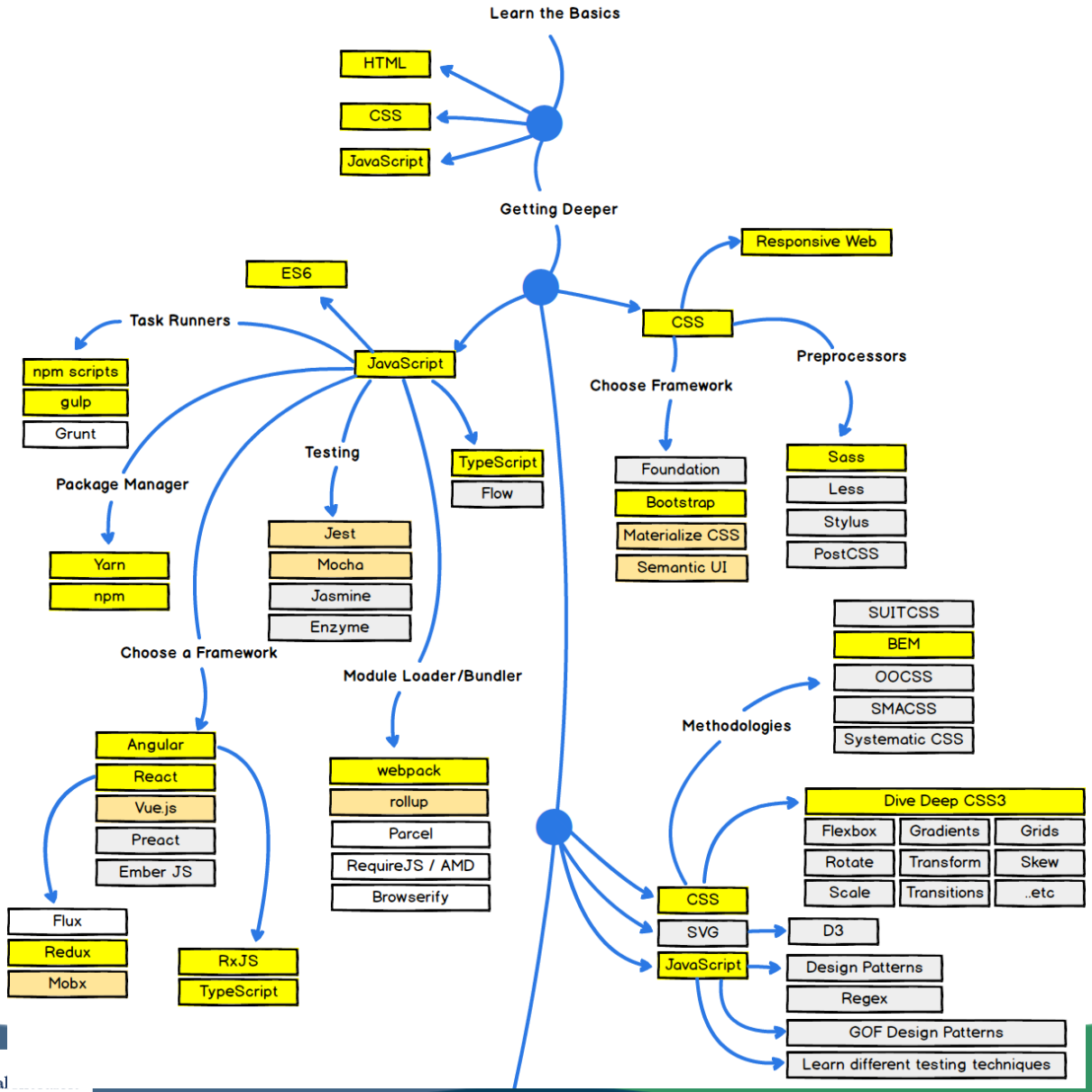
Roadmap for Learning Web Development

Link



- Frontend
  - HTML5, CSS3, JavaScript
  - Frontend frameworks
  - SVG, Canvas, WebGL
- Backend
  - Python, Java
  - Backend frameworks
  - Web Servers

Link


Learn the Basics
HTML
CSS
JavaScript
Getting Deeper
ES6
Task Runners
npm scripts
gulp
Grunt
JavaScript
Package Manager
Yarn
npm
Testing
Jest
Mocha
Jasmine
Enzyme
TypeScript
Flow
Choose a Framework
Angular
React
Vue.js
Preact
Ember JS
Flux
Redux
Mobx
RxJS
TypeScript
Module Loader/Bundler
webpack
rollup
Parcel
RequireJS / AMD
Browserify
CSS
Responsive Web
Choose Framework
Foundation
Bootstrap
Materialize CSS
Semantic UI
Preprocessors
Sass
Less
Stylus
PostCSS
Methodologies
SUITCSS
BEM
OOCSS
SMACSS
Systematic CSS
CSS
Dive Deep CSS3
Flexbox
Gradients
Grids
Rotate
Transform
Skew
Scale
Transitions
..etc
SVG
D3
JavaScript
Design Patterns
Regex
GOF Design Patterns
Learn different testing techniques

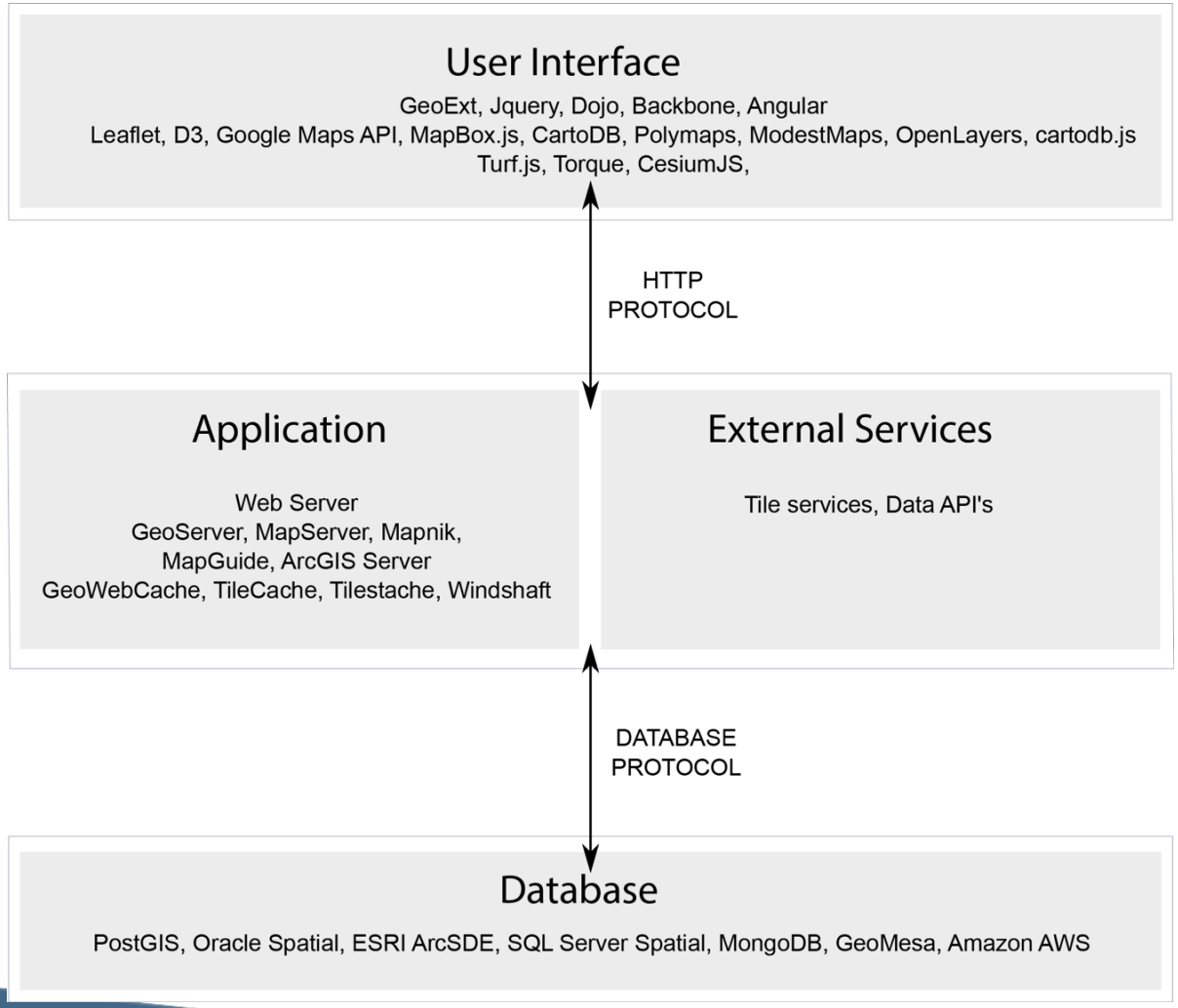
Quick Start Technology Stack

- Data, Middleware & User Interface
  - ArcGIS Online
  - CARTO
  - MapBox
  - Google
  - Tableau, RStudio, Datawrapper, RAW Graphs
- Embed widgets into your own website

Hybrid Technology Stack

- Don’t want to maintain database/mapping servers?
  - Use data from Web API's
  - Use web services from ArcGIS Online, CARTO, MapBox, Google, etc.
  - Build custom user interface with JavaScript libraries
- Don’t want to write Javascript/HTML/CSS?
  - Build your web app with Rshiny; host your data and app on a server running Shiny Server
  - Build your web app with Python packages (Folium, Bokeh), deploy on a web server
- Interactive maps and data with Google Sheets

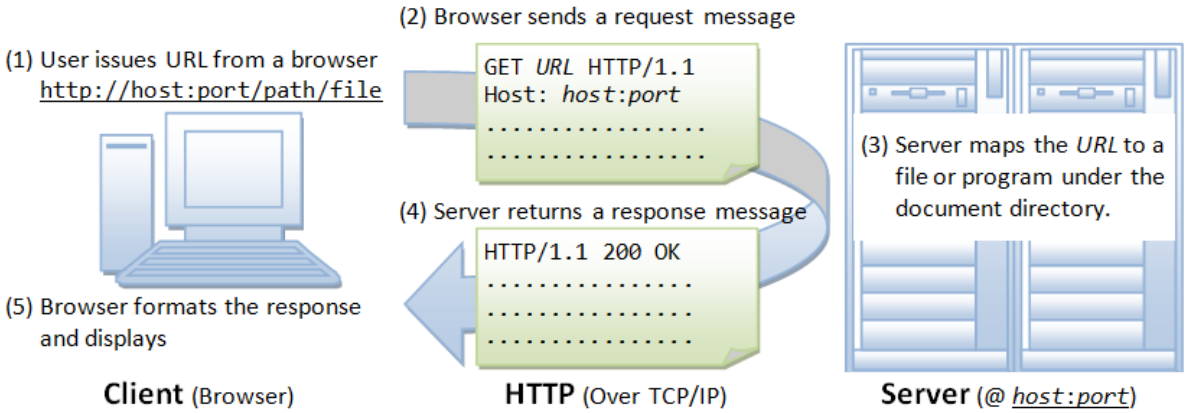
Custom Technology Stack



Quick Review

- Client & Server
- Web Developer Tools
- HTML
- CSS
- DOM
- JavaScript
- Web Graphic Formats - SVG and Canvas

Client & Server



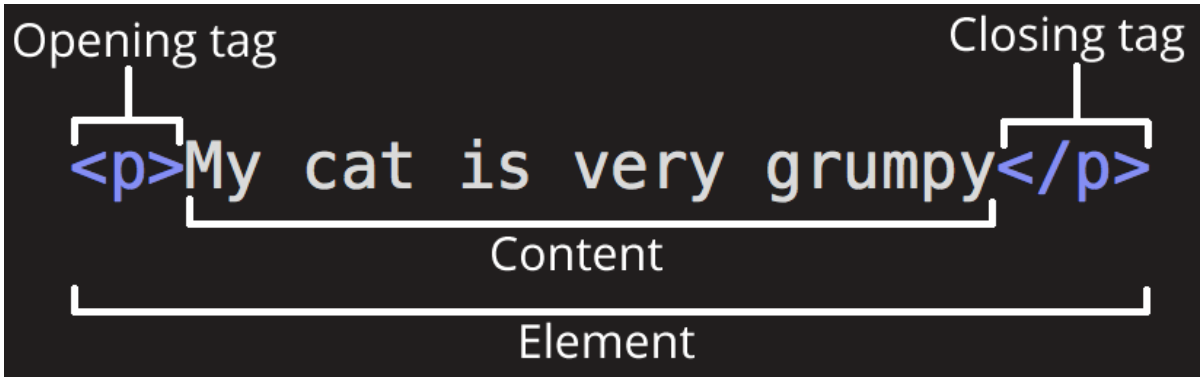
- HTTP, language of the web
- Browser sends HTTP GET Request. Server sends response for each request with content and/or status message.
- HTTP Basics

Web Developer Tools

- Code Editor
  - Sublime Text, Atom, PyCharm, Vim, Notepad++, IDLE, Gedit, TextMate
  - Syntax highlighting, indentation, autocomplete, bracket matching
  - Run interpreters, debuggers
- Browser
  - Developer Tools - press `Ctrl+Shift+I` (or `F12`) in Chrome
  - More info on Chrome Developer tools here
- Github
  - Code hosting platform for version control and collaboration

Hyper Text Markup Language (HTML)

- HTML is not a programming language; it is a markup language
- Role » Describes content
- HTML Elements consist of a opening tag, the content, and a closing tag




```
<ul>
  <li><a href="gif.berkeley.edu">Home Page</a></li>
  <li><a href="gif.berkeley.edu/people">Staff</a></li>
</ul>
<div id="map" class="center" style="height:500px;"></div>
```

- HTML5 Resources
  - Dive into HTML5
  - Mozilla Developer Network

Cascading Style Sheet (CSS)

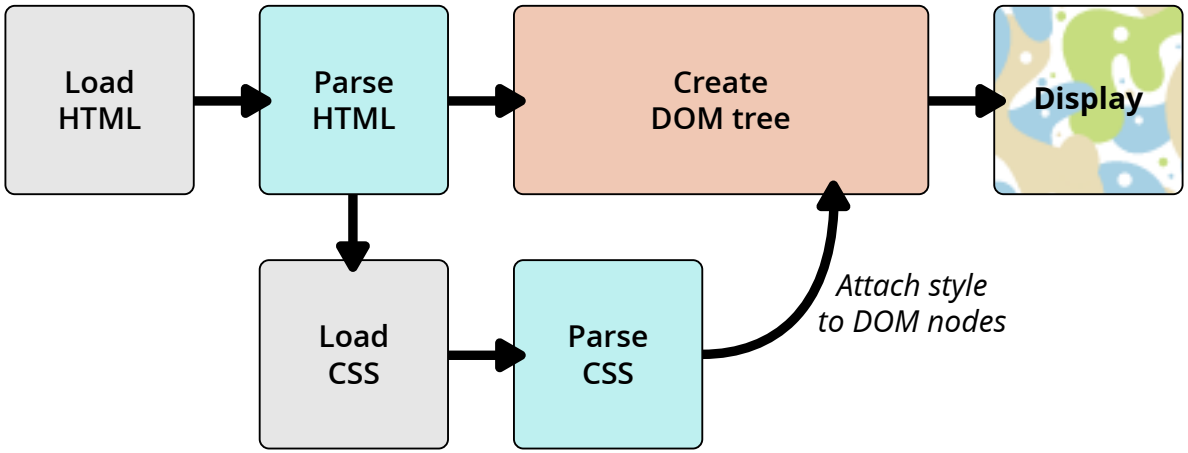
- Not a programming or markup language
- Role » Describes presentation of a document written in HTML or XML (e.g. SVG)
- Selectors, properties, and values
- Properties and Attributes

```
#map {
  height: 500px;
}
h1 {
  font-family: "Helvetica", "sans-serif";
  font-weight: bold;
  background-color: #000;
}
.chart {
  float: left;
}
```

- CSS3 Resources
  - Mozilla Developer Network

Document Object Model (DOM)

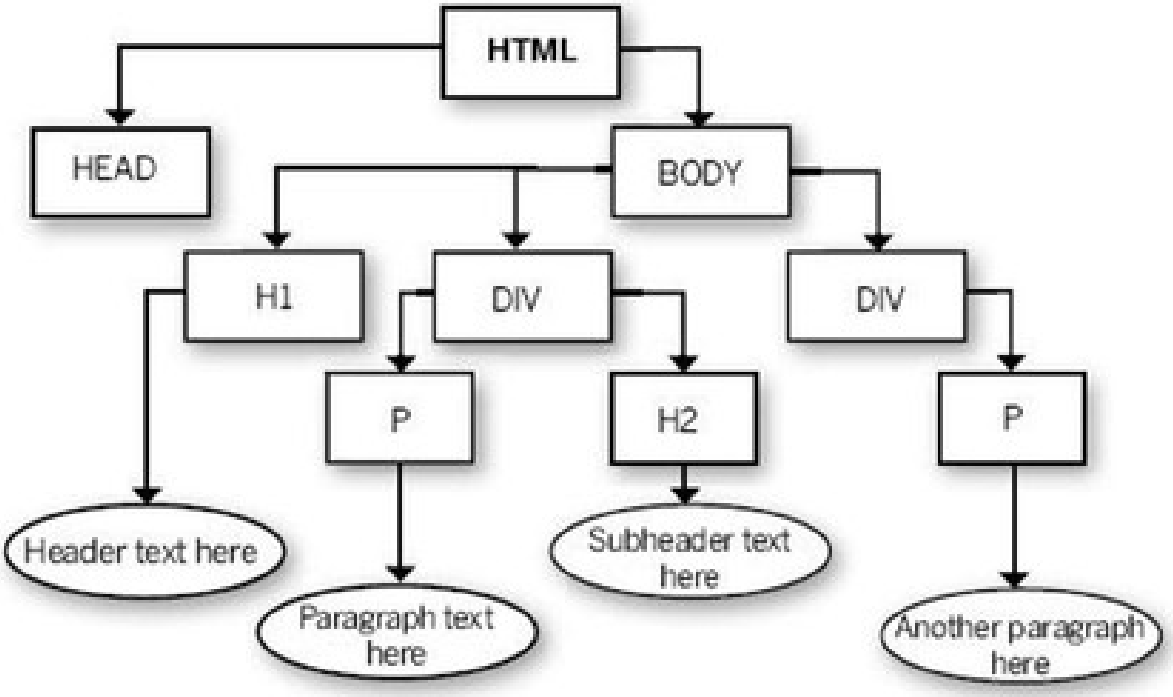
- Structured representation of the document (a tree) created by the browser
- HTML you write is parsed by the browser and turned into the DOM
- Programming interface for HTML and SVG documents
- JavaScript manipulates the DOM



Conceptual Web Page

```
<!DOCTYPE html>
<html>
<head>
    <!-- head content -->
    <style type="text/css">
      div {
        color: red;
      }
    </style>
</head>
<body>
    <h1>Header text here</h1>
    <div>
        <h2>Subheader text here</h2>
        <p>Paragraph text here</p>
    </div>
    <div>
        <p>Another paragraph here</p>
    </div>
    <script>
        <!-- JavaScript content -->
    </script>
</body>
</html>
```

The DOM for conceptual web page



- Understading the DOM

JavaScript

- "Easy to learn, hard to master"
- Role » Creating interaction
- Interpreted by your browser
- Asychronous (code executes in background after client-browser recieves data from server)
- Get familiar with:
  - Working with json objects
  - Array functions (forEach, filter, map)
  - Method chaining, Callbacks, Closures, Modules
- JS Resources
  - Mozilla Developer Network

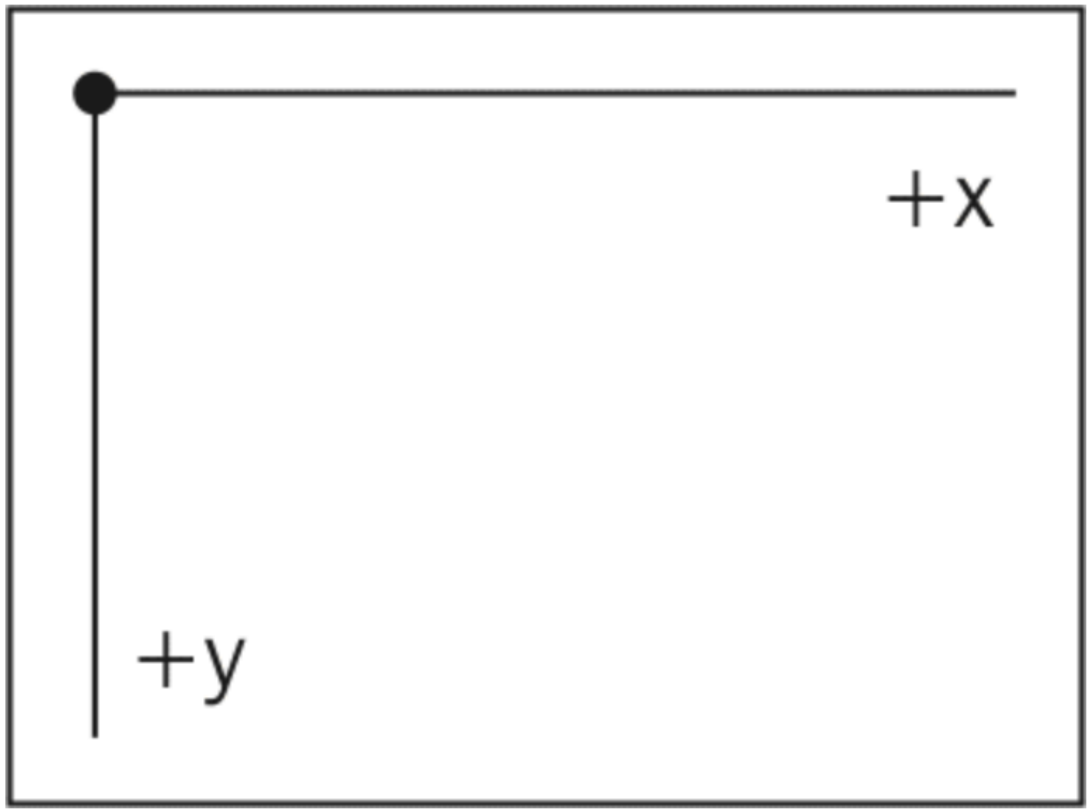
SVG

```
<svg width="300" height="180">
  <rect x="10" y="20" width="20" height="50" fill="blue"
              stroke="red" stroke-width="1"/>

  <circle cx="100" cy="100" r="25" fill="red"
              stroke="#ddd" stroke-width="5"></circle>

  <g transform="translate(5, 15)">
    <text x="0" y="0">My graphic</text>
  </g>

  <g transform="translate(5, 55)">
    <!-- M: move to (jump)
         L: line to
         Q: curve to (quadratic) -->
    <path d="M0,50 L50,0 Q100,0 100,50"
      fill="none" stroke-width="3" stroke="black" />
  </g>

  <g transform="translate(5, 105)">
    <!-- C: curve to (cubic)
         Z: close shape -->
    <path d="M0,100 C0,0 25,0 125,100 Z" fill="black" />
  </g>
</svg>
```

- Play with this code on SVG Graphic Primitives - JSFiddle

SVG (contd.)

- Coordinate system, top-left is (0, 0)
- Depth depends on order in which shapes are drawn



SVG v/s Canvas

- HTML5 Canvas is a raster based format for drawing on the web
  - You can draw raster graphcis with D3
  - You can only get pixel values on click event
  - Faster
- SVG (Scalable Vector Graphics) is a vector based format for drawing on the web
  - HTML has div and span, etc.; SVG has circle and rect, etc.
  - SVG is a DOM for graphical elements
  - You can attach events to SVG elements
  - Most D3 examples work with SVG

HTML, CSS and JavaScript playgrounds

- Codepen, JSFiddle, JS Bin
- Open this Codepen link in a new tab to explore how HTML, CSS and JavaScript work together.
- In-browser code editor built for creating and sharing d3.js examples
  - Blockbuilder
  - Integrated with Github and bl.ocks.org
- Or set up a local testing server on your machine

Start a local testing server and review final project

- Set up a simple local HTTP server on Windows
  - How do you set up a local testing server?
  - Python library comes with a `SimpleHTTPServer` module
  - For today we will use Python installed by ArcGIS on the lab computers
- Open Windows Command Prompt
- Change directory to the workshop folder

```
cd Documents\workshop-intro-d3-leaflet
```

- Start the local server using python library installed by ArcGIS

```
C:\Python27\ArcGIS10.4\python -m SimpleHTTPServer
```

- Open a browser. Type `localhost:8000` into the url bar at top. You should see a list of directories & files. Click on the `final` directory

Web Development Resources

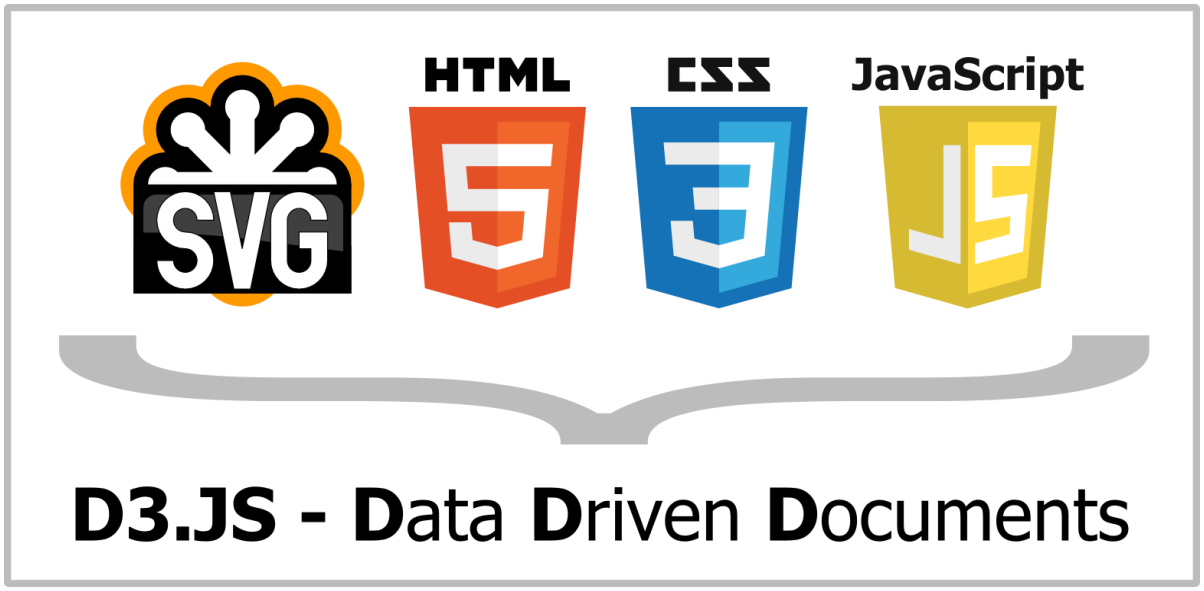
- If you are going to develop medium to large scale web apps learn more about using front-end development tools.
  - Getting Started Web Development Guide
  - Front-end Handbook
- Google Search
  - Favor results from Stack Exchange, Mozilla Developer Network, CSS-Tricks
- caniuse - Check which browsers support what features
- Web Design
  - A List Apart
  - Webdesigner News - curated stories
- Eloquent Javascript
- Egghead.io - Bite size web dev video training
- Lynda.com, Frontend Masters, CodeAcademy, Udacity, Coursera, etc.

Explore D3 Core Concepts and build a bar chart

Data-Driven Documents

D3 (Data Driven Documents)

- D3.js is a javascript for producing dynamic, interactive data visualizations in web browsers.
- Built on the fundamental web standards of HTML, CSS, SVG, and JavaScript.



- Created by Mike Bostock, Vadim Ogievetsky and Jeffrey Heer (all at that time were part of the Stanford Visualization Group).

What does it do?

- It is *not* a graphics library, it is *not* a charting library, it is a general purpose data visualization library
- Has everything you need for visualizing complex data BUT you will not find commands like barchart, scatterplot, or piechart, or even map
- Visualizes data using web standards (HTML, Javascript, CSS, SVG)
- Heavily used in data journalism and visualization. New York Times has been the pioneer in data visualization and data journalism.
- Most and foremost a DOM manipulation library (in this regard similar to jquery).
- Provides handy utilities for processing data (array, time series, geo data)
- Comes with a lot of functions and methods to create mapping functions that will map your data directly to properties on html elements
- Filled with algorithms (voronoi, quadtrees, circle fitting, convex hull, projections)

Learning D3

One of the most interesting aspects about d3 is that it is the intersection between design and code, math and art, data and story - Ian Johnson

- Very active user community in Bay Area.
- Bay Area d3 User Group.
- Rich ecosystem of examples showing visualization types, coding techniques:
  - blockbuilder.org/
  - bl.ocks.org/
- Navigating the D3 Ecosystem
- How do I learn D3.js? Quora thread
- D3 version 3 not the same as D3 version 4

Resources

- Books
  - Interactive Data Visualization for the Web
  - D3 Tips and Tricks
- Curated lists of data viz and data viz research
  - visualisingdata.com
  - The Chartmaker Directory
- Mike Bostock Blog
- For geographic visualizations
  - Learning D3.js 4 Mapping
  - Jason Davies, Jason Davies Code Blocks
- API documentation

Core Concepts - Declarative API

- Declarative: Tell D3 what you want, not how you want it done
- API: Methods and objects you use to interact with software
  - e.g. keyboard interface for a computer
- Describe a relationship between data and UI (User Interface) elements, D3 handles implementation details

```
d3.selectAll('p').style('color', 'red');
```

- Exercise: D3- Declarative API

Core Concepts - Selections

- D3 provides shorthand for selecting and manipulating DOM objects (similar to jQuery)
    - `d3.select()`, `d3.selectAll()`
- Modify properties of selections
    - Create/Read/Update/Delete (CRUD)
    - Using Getters and setters

Name	Behaviour	Example
.style	Update the style	d3.selectAll('circle').style('fill', 'red')
.attr	Update an attribute	d3.selectAll('rect').attr('width', 10)
.classed	Add/remove a class attribute	d3.select('.item').classed('selected', true)
.property	Update an element's property	d3.selectAll('.checkbox').property('checked', false)
.text	Update the text content	d3.select('div.title').text('My new book')
.html	Change the html content	d3.select('.legend').html('0 - 10')

- Exercise: D3 - Selections

Core Concepts - Selections (contd.)

- Append/insert elements to elements to existing selections
  - selection.append(type) && selection.insert(type[, before])

```
d3.select('body').append('p');
```

- Remove items from selections
  - selection.remove()

```
d3.select('body').remove('p');
```

- Handle events
  - selection.on()

```
d3.selectAll('button').on('click', handleButtonClicks);
d3.select('body').on('mousemover', trackMousePosition);
```

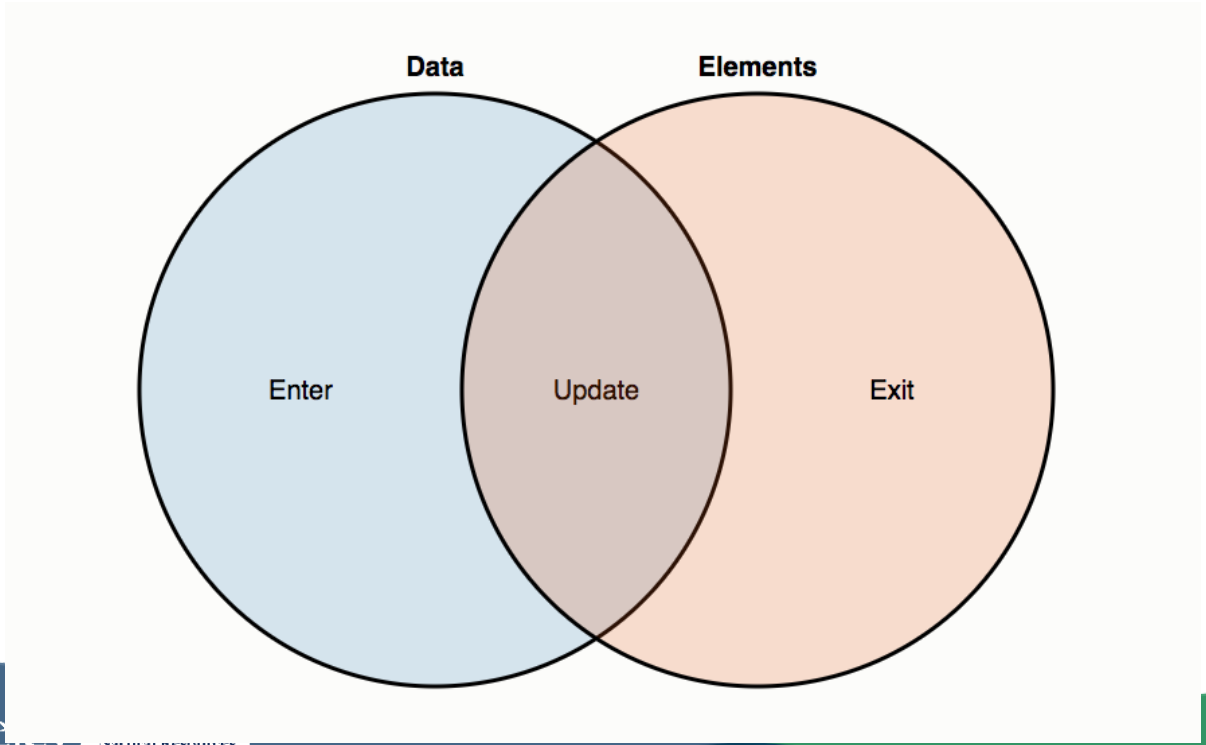
- Read documentation d3-selection

Core Concepts - Data Binding

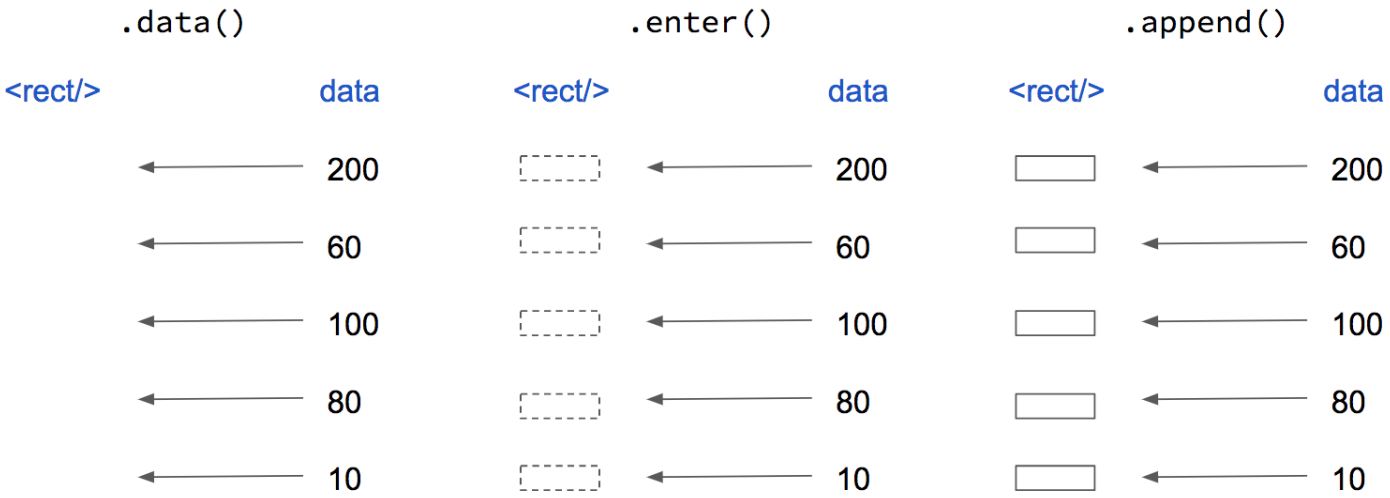
- Real power of D3 is joining data and selections
- `selection.data()` - accepts an array of data items and joins them to the DOM elements in the selection
- Exercise: D3 - Data Binding 1
- Exercise with SVG: D3 - Data Binding 2

Core Concepts - Data Binding (contd.)

- What if we have more data than existing elements? Enter...
- Exercise: D3 - Enter Selection
- ENTER, UPDATE and EXIT SELECTIONS
  - Mike Bostock - Thinking with Joins
  - Mike Bostock - General Update Pattern examples
  - Also D3 In Depth book (in progress) is great!



Core Concepts - Data Binding (contd.)



We need 5 rectangles to match data

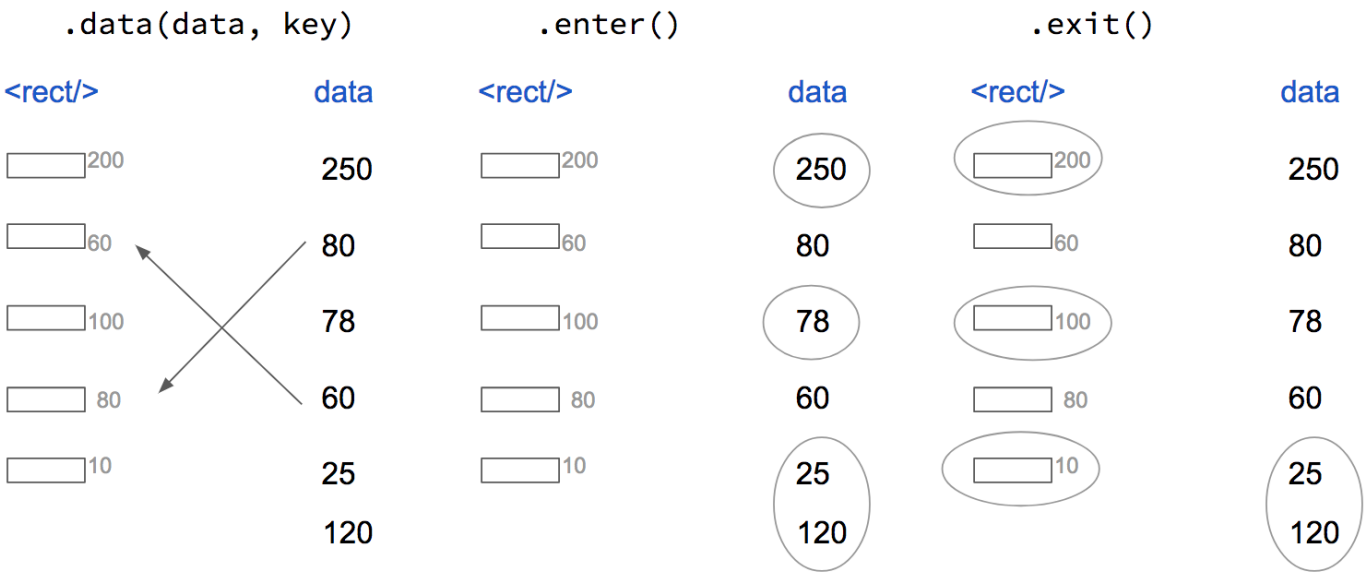
Put in 5 placeholders

Create a rect element for each placeholder

Core Concepts - Data Binding (contd.)

- Use with dynamic data
  - Exercise: D3 - Enter, Update, Exit
- Use with data from Cal-Adapt API
  - Exercise: D3 - Get Data from API

Core Concepts - Data Binding (contd.)



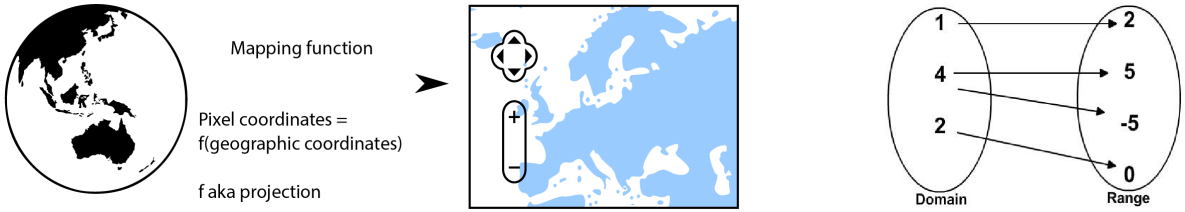
Update Selection
These 2 rects will remain on the screen.

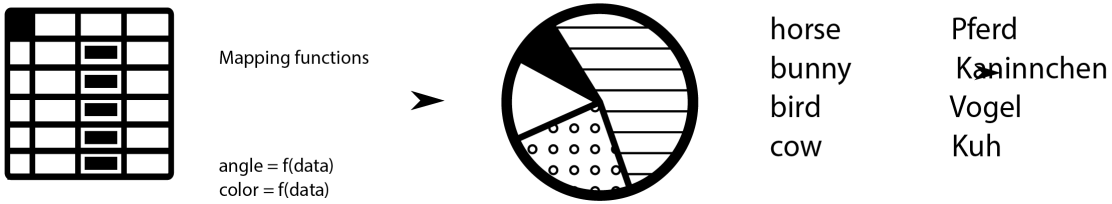
Enter Selection
Need to create 4 new rects for these data values

Exit Selection
Need to get rid of these 3 rects to match data

Core Concepts - Scales

- Mapping: An operation that associates each element of a given set (the domain) with one or more elements of a second set (range)
- Any (computer) visualization problem can be described as a mapping problem.





- Documentation - d3.scales

Core Concepts - Scales (contd.)

- Mapping from data attributes (domain) to display (range)
  - e.g. date -> x value, data value -> opacity, data value -> y value

```
  d3.scaleLinear()
    .domain([min, max]) // input
    .range([min, max]); // output
```

- Linear scale

```
const percentToDecimal = d3Scale.scaleLinear()
  .domain([0, 100])
  .range([0, 1])

percentToDecimal(0)   // 0
percentToDecimal(50)  // 0.5
percentToDecimal(100) // 1
```

- Exercise - d3.scales

Core Concepts - Axes

- Human readable reference for scales
- Documentation - d3-axis
- Exercise - Bar Chart
- Exercise - Bar Chart using Cal-Adapt data

D3 - Other concepts

- Lots of convenience functions for loading and manipulating data, working with arrays
- Shapes - `d3.line()`, `d3.circle()`, `d3.pie()`, `d3.arc()`
- Colors
- Manipulate structure of your data - d3.nest()
- D3 API Reference
- Writing reusable components
  - A good place to start is by reading Towards Resuable Charts where Mike Bostock proposes a convention for creating reusable charts.
  - This blog post and this book do a great job on clarifying the resuable charts api with examples and explanations.

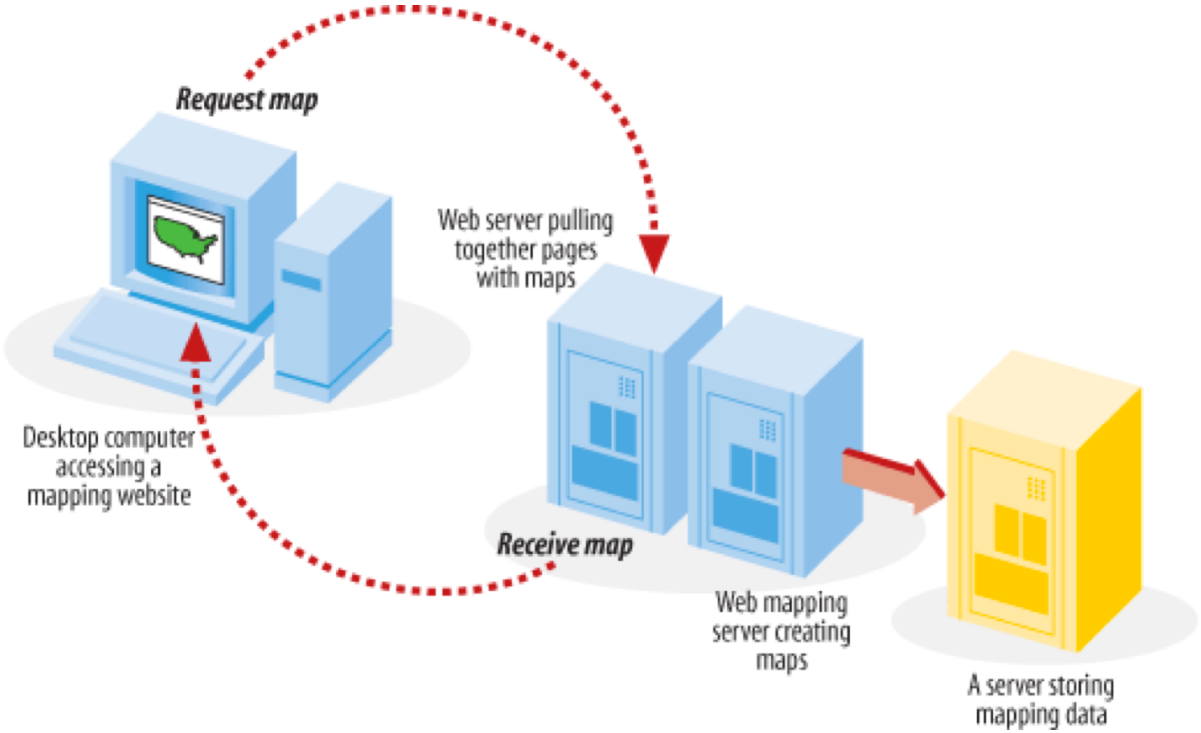
Other options for using D3

- Rickshaw, Highcharts, NVD3 are libraries built on top of D3
- D3 and Crossfilter (Fast Multidimensional Filtering for Coordinated Views)
- Open-source tools binding D3 to R, Python
- Vega - higher-level visualization specification language on top of D3

Web Mapping

What is a Web Map?

- Digital map that works in a web browser (as opposed to PDF/Image files)
  - Interactive
  - Developed using web application technologies – Not specialized mapping software



A Quick Recap of Web Mapping History

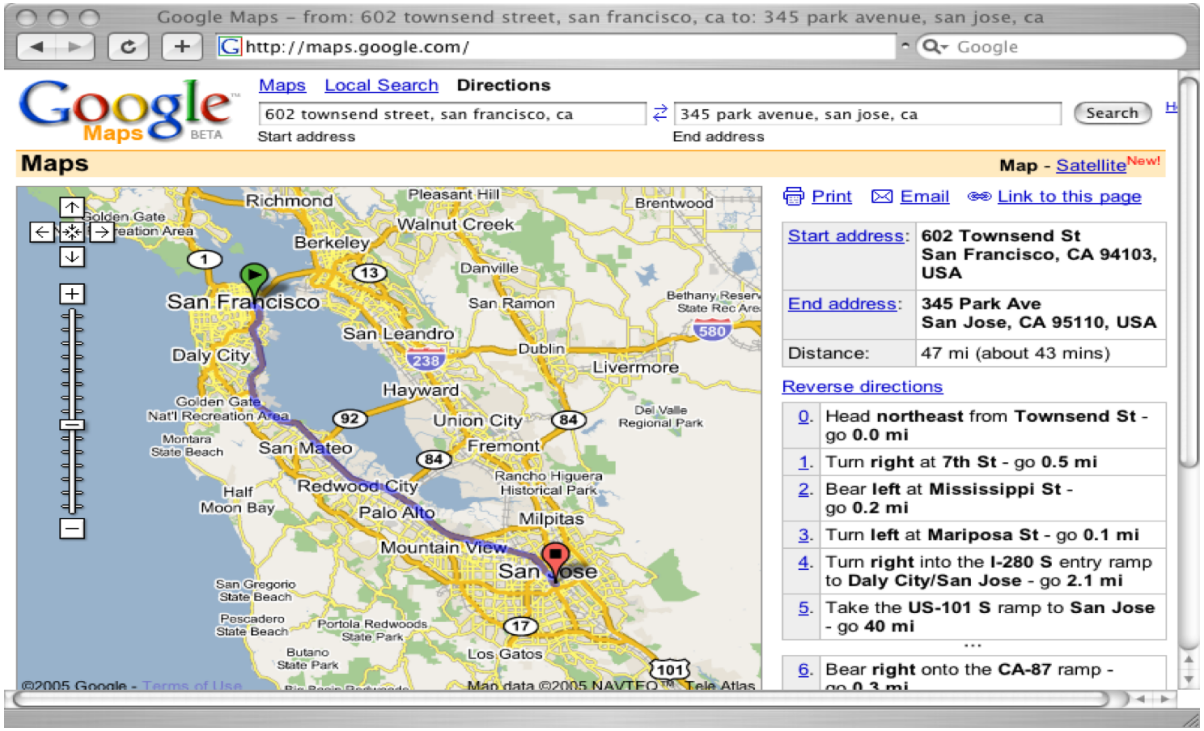
- 1996 MapQuest

A Quick Recap of Web Mapping History (contd.)

- 1997 - 2004
  - MapInfo, AOL buys MapQuest
  - GPS Selective Availability ends (2000)
  - Microsoft MapPoint
  - Open Street Map (OSM) founded in UK (2004)
  - NASA releases WorldWind
  - Yahoo! Maps
  - Google buys Where2, Keyhold & ZipDash

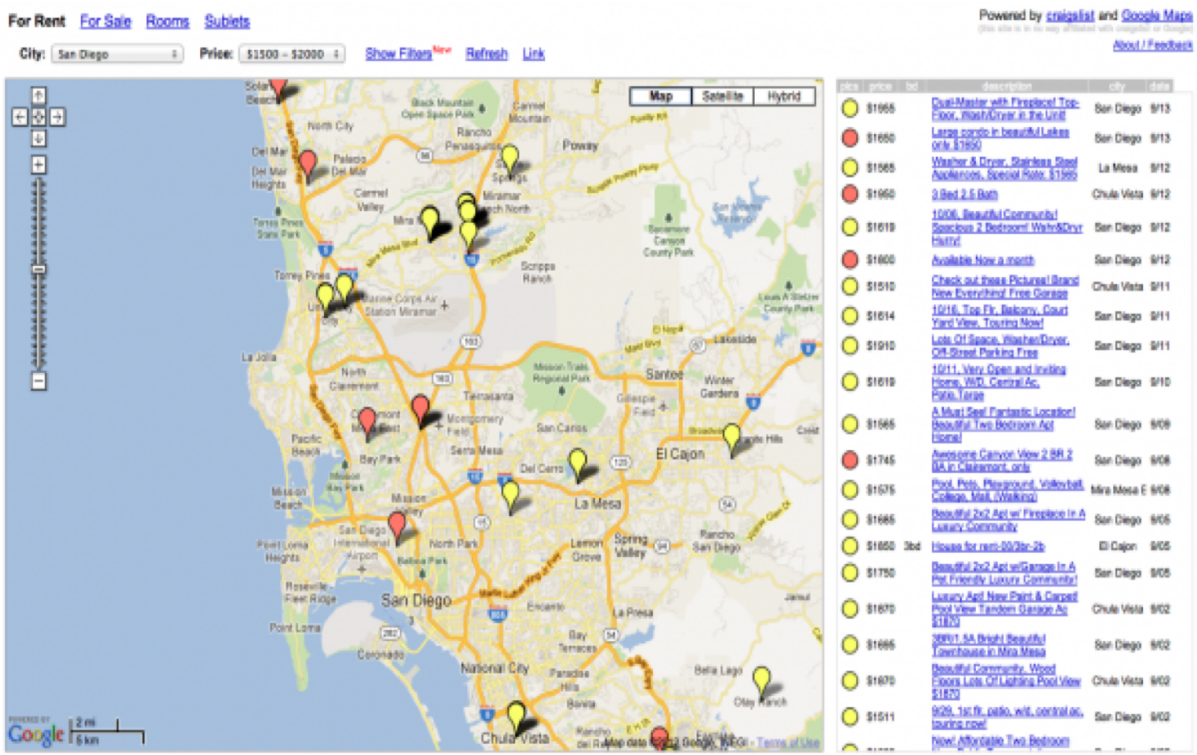
A Quick Recap of Web Mapping History (contd.)

- 2005 - Google Maps



A Quick Recap of Web Mapping History (contd.)

- 2005 - First Google Maps Mashup

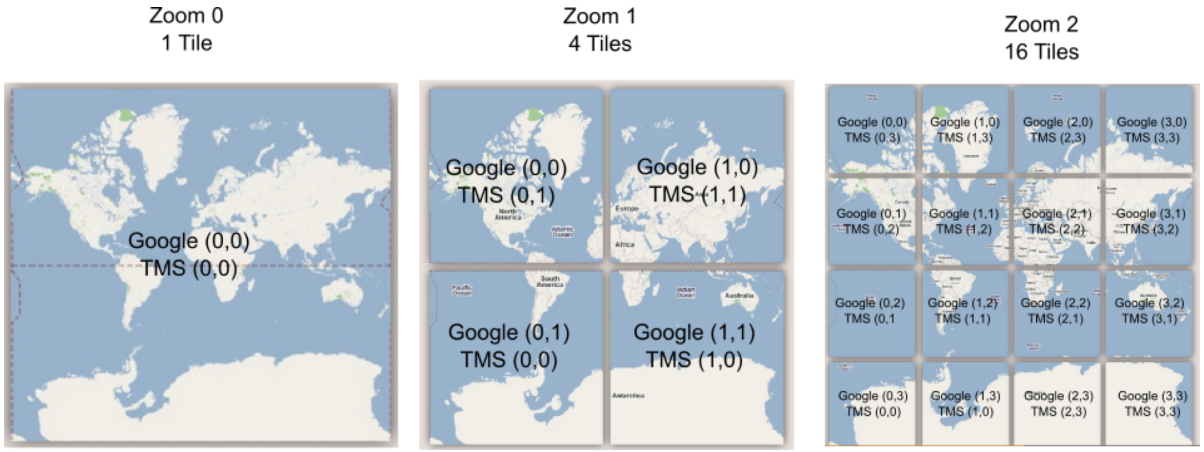


Technological Breakthroughs

- Slicing a square map into smaller square image tiles – Web Mercator projection
- Slippy maps with smooth zoom and pan – Web development techniques (AJAX) used for requesting data (map tiles) from server asynchronously (in the background)
- Open API
- No plugins, seamless user experience
- Mapping as a service

Map Tiles

- 256 pixels x 256 pixels image files
- Pre-rendered or rendered on the fly
- 360 billion tiles to cover the whole world at 20 zoom levels



Fast forward to present

- Web Services, Open Data, Open Standards and Formats, Free and Open Source Software

Web Services

- APIs (Application Programming Interface)
  - Basemaps, Data, Geocoding, Navigation
- Commercial and non-commercial data and service providers
  - OpenStreetMap
  - Google, ESRI, CARTO, MapBox, Planet … many more – Free and paid tiers
  - Data.gov, NPMap, SF Open Data, NASA
  - Cal-Adapt, Landcarbon, Ecoengine @berkeleyGIF

Open Standards and Data Formats

- Data Formats
  - GeoJSON, TopoJSON, KML, Shapefile, WaterML, NetCDF…
- Specifications
  - Web Map Services, Web Feature Services, Vector Tiles

Leaflet

- Lightweight, simple & flexible open source JavaScript mapping library
- Created by Vladimir Agafonkin. Great speaker, checkout some of his talks on Leaflet
- Mobile-friendly. Well documented API, huge amount of plugins.
- Use with other JS mapping libraries (like Esri-Leaflet) or by itself. Similar libraries - OpenLayers, ModestMaps, Polymaps.
- Carto.js and Mapbox.js libraries are built on top of Leaflet.
- Leaflet FAQ

What does it do?

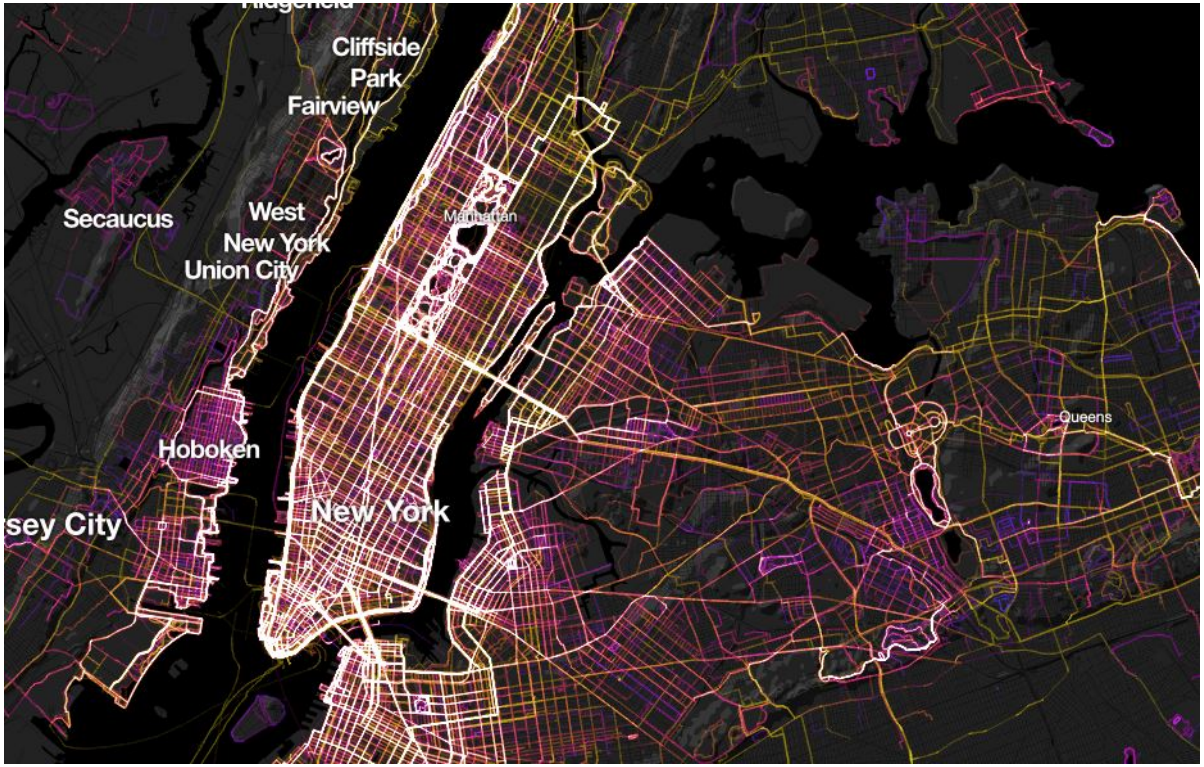
- Slippy maps with panning and zooming
- Provides functions for converting data into map layers
- Provides mouse interaction
- Does not provide any data
- You provide tile basemaps and data for overlays
- Easy to extend with plugins

Overlays - GeoJSON

```
var obj =
{ "type": "FeatureCollection",
    "features": [
      { "type": "Feature",
        "geometry": {"type": "Point", "coordinates": [102.0, 0.5] },
        "properties": {"prop0": "value0"}
       },
      { "type": "Feature",
         "geometry": {
           "type": "Polygon",
           "coordinates": [
             [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],
               [100.0, 1.0], [100.0, 0.0] ]
             ]
         },
         "properties": {
           "prop0": "value0",
           "prop1": {"this": "that"}
           }
      }
    ]
}
```

- GeoJSON Specification, Validate your geojson
- Convert from shapefile - geojson.io, Mapshaper, GIS
- GeoJSON data coordinates should be in latitude, longitude (WGS 84)!!!

Basemaps & Overlays - Map Tiles



- How Web Maps Work
- Web Mercator - Standard projection for map tiles

Leaflet Exercises

- Link to Github Repo
- Before we begin:
  - Start a local server
  - Code Editor and Browser
  - Workflow for the exercises

Exercise 01 - Make a map, add basemap

- This exercise covers the basics for creating a Leaflet map element and adding a basemap to it.
- Four things you absolutely need for adding Leaflet map to a web page
  - Leaflet CSS file
  - Leaflet JavaScript file
  - A div element with an `id` attribute
  - A `css` height for your map div

Exercise 02 - Add and style data

- Add an overlay of tick collection point locations and style it

Exercise 03 - Working with Larger Datasets

- Geojson files with lots of data can take longer to load depending on size and network speed. And on the web you never want to keep your user waiting :-) There are couple of ways you can deal with larger files.
- Option 1 - Use the topojson format. Topojson is an extension of GeoJSON that encodes topology.
    - Convert geojson to topojson using mapshaper.org. You can also further reduce file size by simplifying the geometry. It depends on your app, how much detail you need to show.
- Option 2 - Host the data on service providers like CARTO, ArcGIS Online, Mapbox. These companies also provide their own web mapping libraries (some built on top of Leaflet)
- Great mapmakers cheatsheet on other options for dealing with large files with Leaflet and web maps in general.

Review Final Project with Leaflet Map and D3 chart

- In browser go to `localhost:8000/final`

Additions

- Content and styling
- Legend
- An example of using the Turf.js library for spatial analysis
- Integrated D3 bar chart

Discussion

You Pick!

Visualization, Web Development, Learning Strategies.

Deeper dive into some examples

- (D3 In Depth Examples](http://d3indepth.com/introduction/)