

D3js Basics

Information Visualization

Kalpathi Subramanian

Computer Science
The University of North Carolina at Charlotte

Last Modified: Feb 2, 2023

References: [D3 Link: D3 Home](#)

What is D3JS

- D3.js is a JavaScript library for manipulating documents based on data. (<https://d3js.org/>)
- At it's core, D3 is a graphics library for the web
- D3 is a targeted library – data visualization
- D3 visualizations can be embedded into any web page

Running D3

- What does D3 require to run?
 - ▶ A web browser
 - ▶ D3 source code
 - ▶ Valid HTML document
 - ▶ Server
- D3 source code can be included in any HTML file as follows (we are using version 7):

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

- **Server:**
 - ▶ Server: when using data files, use a **local server**
 - ▶ **From python:** Run the following command from a terminal (MacOS/Unix) or command prompt (Windows):
python3 -m http.server
 - ▶ **Needed when you are processing local files**

Selections

Binding elements and changing attributes

- Selections allow html elements to be selected for processing, changing attributes
- Alternately, when you are drawing new elements, use an element name that is not defined in the DOM
- Commands: *d3.select*, *d3.selectAll*, *d3.append*
- **Example:** *d3.selectAll('circle')*, will select all circle elements in the HTML page
- **Example:** *d3.selectAll('circle').style('fill', 'steelblue')* selects all circles and changes their color

Binding Data

- Can attach data to a selection and use the data using anonymous functions
- **Example:** `d3.selectAll('circle').data([32, 57, 112]);` attaches an array of integers to the selections
- Use the data **Example:**

```
d3.selectAll('circle')  
  .data([32, 57, 112])  
  .attr('r', function(d) { return Math.sqrt(d); });
```

Entering/Exiting Selections

- Enter/Exit selection is a useful means to process data when you don't have enough elements or have too many.
- When joining data to elements, any extra elements are added to **enter** selection - allows us to create new elements for the extra data values
- **Example:**

```
var svg = d3.select("svg");  
var circle =  
  svg.selectAll("circle")  
    .data([32, 57, 112, 293]);  
  .enter()  
  .append("circle");
```

- Exiting selections does the opposite, removes elements not used by the data values

Processing Data Files (CSV, JSON)

- D3 can read CSV and JSON files
- **Commands:** `d3.csv()`, `d3.json()`
- Refer to this [*D3 Link: Loading CSV and JSON Files*](#)