Introduction to D3

Maneesh Agrawala

CS 448B: Visualization
Fall 2021

Reading Response Questions/Thoughts

How do you balance details with overarching views? How to decide when there is too much vs. too little data in a view?

In the VDQI reading, it was stated that we can increase data density in two ways: increasing the number of data points or decreasing the size of the graph... When it is advisable to reduce the size of our graph to increase the density? ... Are there any quantitative metrics to assess visualization interpretability?

From an accessibility standpoint, I wonder if there are ways to facilitate micro-macro reading through haptic and audio means?

And more generally, what are instances where poor design choices on the micro level can obscure the types of insights gained at the macro level?

When is it appropriate to exclude outlying data points? For datasets of the Challenger launch tests, the outliers were everything. All data is prone to some degree of variation, so how much significance should we place on outliers?
D3 Notebooks

Last Time: Interaction
Baseball statistics [from Wills 95]

- how long in majors
- avg assists vs avg putouts (fielding ability)
- distribution of positions played
- select high salaries
- avg career HRs vs avg career hits (batting ability)

Dynamic Queries
Query and results

SELECT house
FROM east bay
WHERE price < 1,000,000 AND bedrooms > 2
ORDER BY price

HomeFinder

[Ahlberg and Schneiderman 92]
**Zipdecode** [from Fry 04]

https://benfry.com/zipdecode/

**NameVoyager**

http://www.babynamewizard.com/voyager
DimpVis [Kondo 14]

Parallel Coordinates [Inselberg]
Based on Wattenberg’s [2001] idea for sketch-based queries of time-series data
3D dynamic queries [Akers et al. 04]
Pros and cons

**Pros**
- Controls useful for both novices and experts
- Quick way to explore data

**Cons**
- Simple queries
- Lots of controls
- Amount of data shown limited by screen space

Summary

**Good visualizations are task dependent**
- Pick the right interaction technique

**Fundamental interaction techniques**
- Selection, Brushing & Linking, Dynamic Queries
A2: Exploratory Data Analysis

Use Tableau or Vega-Lite to formulate & answer questions

First steps
Step 1: Pick domain & data
Step 2: Pose questions
Step 3: Profile data
Iterate as needed

Create visualizations
See different views of data
Refine questions

Author a report
Screenshots of most insightful views (8+)
Include titles and captions for each view

Due before class on Oct 11, 2021
Assignment 3: Dynamic Queries

Create a small interactive dynamic query application similar to HomeFiner, but for restaurants data.

1. Implement interface
2. Submit the application and a short write-up on canvas

Can work alone or in pairs
Due before class on Oct 25, 2021
Introduction to D3

What is D3?

D3: “Data-Driven Documents”

Data visualization built on top of HTML, CSS, JavaScript, and SVG

Pros:
- Highly-customizable
- Developing and debugging tools
- Documentation, resources, community
- Integrates with the web!

Cons:
- Very “low-level”
hello-world.html

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>

<body>
  Hello, world!
</body>
</html>
```
hello-svg.html

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style>/* CSS */</style>
</head>
<body>
    <svg width="960" height="500">
        <circle cx='120' cy='150' r='60' style='fill: gold;'>
            <animate attributeName='r' from='2' to='80' begin='0' dur='3'
                repeatCount='infinite' />
        </circle>
    </svg>
</body>
</html>
```

hello-svg.html

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style>/* CSS */</style>
</head>
<body>
    <svg width="960" height="500">
        <circle cx='120' cy='150' r='60' style='fill: gold;'>
            <animate attributeName='r' from='2' to='80' begin='0' dur='3'
                repeatCount='infinite' />
        </circle>
    </svg>
</body>
</html>
```
DOM: Document Object Model

<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <div>
      <p></p>
    </div>
  </body>
</html>

[Adapted from Victoria Kirst’s cs193x slides.]
**DOM: Document Object Model**

```html
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <div>
      <svg></svg>
    </div>
  </body>
</html>
```

[Adapted from Victoria Kirst’s cs193x slides.]

---

**hello-d3.html**

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>/* CSS */</style>
  </head>
  <body>
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <script>
      // JavaScript code that handles the logic of adding SVG elements
      // that make up the visual building blocks of your data visualization
    </script>
  </body>
</html>
```
D3: Selection

```html
<html>
...  
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
```

D3: Selection

```html
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
```

```script>
// select all SVG circle elements
var circles = d3.selectAll("circle");
</script>
```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
</html>
D3: Selection & Manipulation

```html
<html>
    ...
    <svg width="960" height="500">
        <circle cx="10" cy="10" r="5"></circle>
        <circle cx="20" cy="15" r="5"></circle>
    </svg>
</html>
```

```javascript
// select all SVG circle elements
var circles = d3.select("circle");

// set attributes and styles
circles.attr("cx", 40);
circles.attr("cy", 50);
circles.attr("r", 24);
circles.style("fill", "red");
</script>

D3: Selection & Manipulation

```html
<html>
    ...
    <svg width="960" height="500">
        <circle cx="10" cy="10" r="5"></circle>
        <circle cx="20" cy="15" r="5"></circle>
    </svg>
</html>
```

```javascript
// select all SVG circle elements
var circles = d3.select("circle");

// set attributes and styles
circles.attr("cx", 40);
circles.attr("cy", 50);
circles.attr("r", 24);
circles.style("fill", "red");
</script>
D3: Selection & Manipulation

```html
<html>
  ...
  <svg width="960" height="500">
    <circle cx="10" cy="10" r="5"></circle>
    <circle cx="20" cy="15" r="5"></circle>
  </svg>
</html>
```

```script
// all together!!
d3.select("circle")
  .attr("cx", 40)
  .attr("cy", 50)
  .attr("r", 24)
  .style("fill", "red");
</script>

Binding Data & Joining DOM Elements

```javascript
const ol = d3.create('ol');

ol.selectAll('li') // select all list elements (orange circle above)
  .data(listData) // bind all our data values (blue circle above)
  .join()
    .enter => enter.append('li'), // append an li element for each entering item
    .update => update, // do nothing with items that match an existing element
    .exit => exit.remove() // remove li elements whose backing data is now gone
  .text(d => `$d.country: $d.pop`);

return ol.node();
```
A join creates three sub-selections:

- **Enter**: selection containing placeholders for every data value that did not have a corresponding DOM element in the original selection.
- **Update**: selection containing existing DOM elements that match a bound data value.
- **Exit**: selection that also contains existing DOM elements, but for which a matching data value was not found.

---

**Exercise**

Modify the code and run it to make sure the changes that you make to the variables, `num1`, `num2`, and `num3`, are reflected in the output as expected.
Let’s make a scatterplot 🐾 🐶

https://observablehq.com/@stanfordvis/lets-make-a-scatterplot