DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

Data Science Tools: Interactive visualization

10-March-2020

Sepi Chakaveh & Assel Altayeva

Overview Data Science

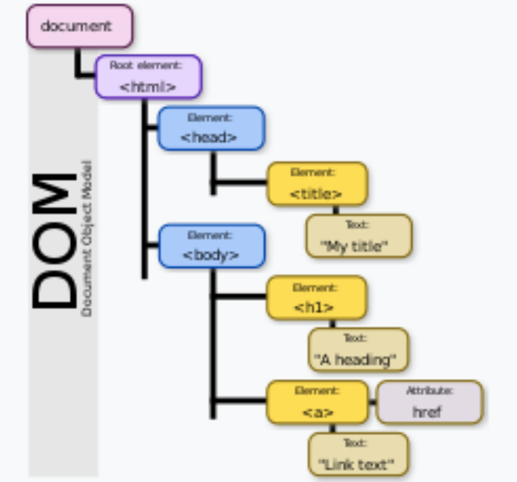
D3 –Data-Driven Document

- D3 Show Reel:

https://bl.ocks.org/mbostock/1256572

D3 is Data ↦ Elements

- Visualizing Data with Web Standards (HTML/SVG)
- Data ↦ Elements – constructing the DOM from Data
- In visualization, each data point has a corresponding element (graphical marks). D3 helps you maintain this mapping!



Example of DOM hierarchy in an HTML

SVG – Scalable Vector Graphics

DOM – Document Object Model

D3- Data-Driven Documents

D3.js is a JavaScript library for manipulating documents based on data.

d3js.org

github.com/mbostock/d3

github.com/mbostock/d3/wiki/API-Reference

D3.js is not limited to a specific canvas like the other libraries. You can use the whole page.

It doesn’t have pre-built charts that limit creativity!

+ Community, documentation, examples,...

 - Reusability

Installation of D3.js

More Options!

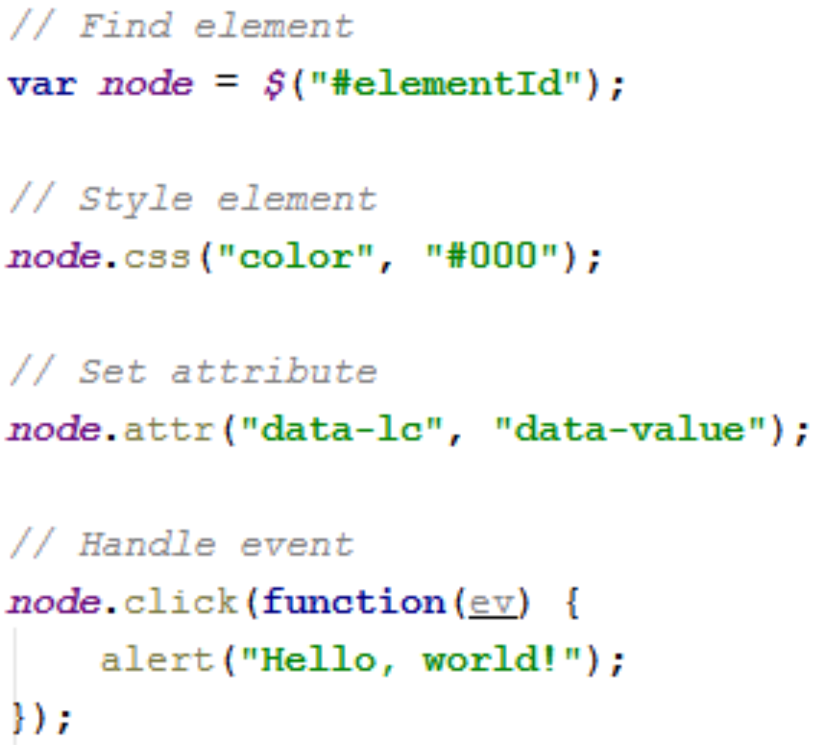
- ▷ Npm install --save d3
- ▷ Bower install --save d3
- ▷ Download D3.zip
- ▷ Use external

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

Edit and add the <script> line to your HTML file based on the installation path.

```
<script src="node_modules/d3/build/d3.min.js"></script>
```

D3 is a little like jQuery (jQuery)



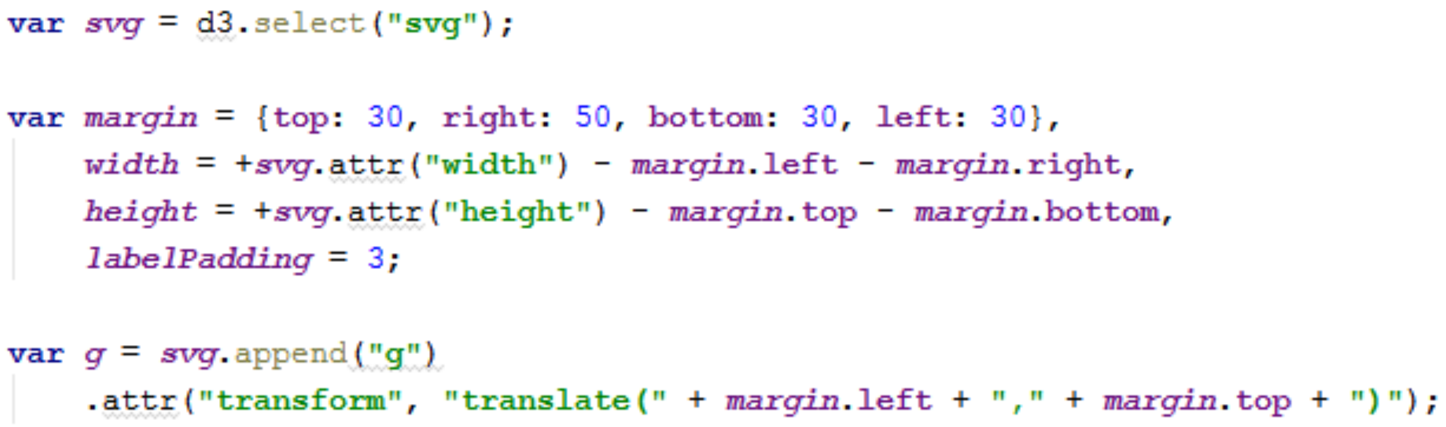
```
// Find element
var node = $("#elementId");

// Style element
node.css("color", "#000");

// Set attribute
node.attr("data-lc", "data-value");

// Handle event
node.click(function(ev) {
    alert("Hello, world!");
});
```

Selections

- A selection is an array of elements pulled from the current document. D3 uses CSS3 to select elements.
- After selecting elements, you apply operators to them to do stuff. These operators can get or set attributes, styles, properties, HTML and text content.
- d3.select(selector), d3.select(node)
- d3.selectAll(selector), d3.selectAll(nodes)

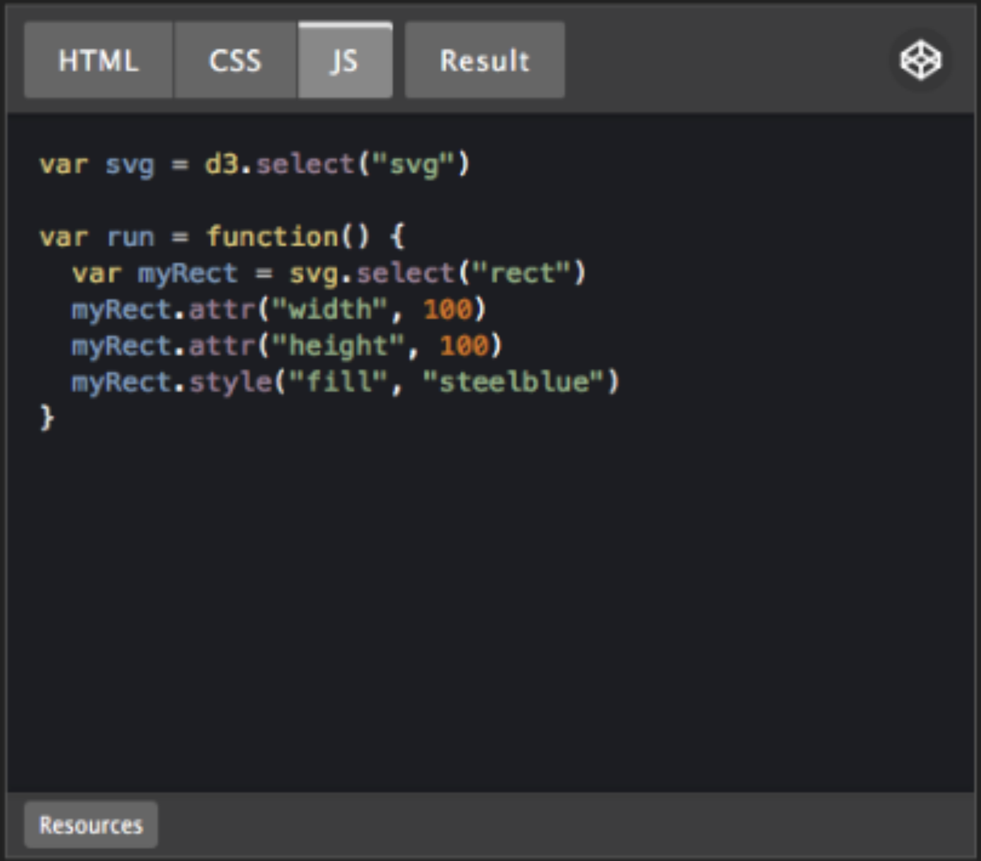
- See more: github.com/mbostock/d3/wiki/Selections

Select element and style it a bit

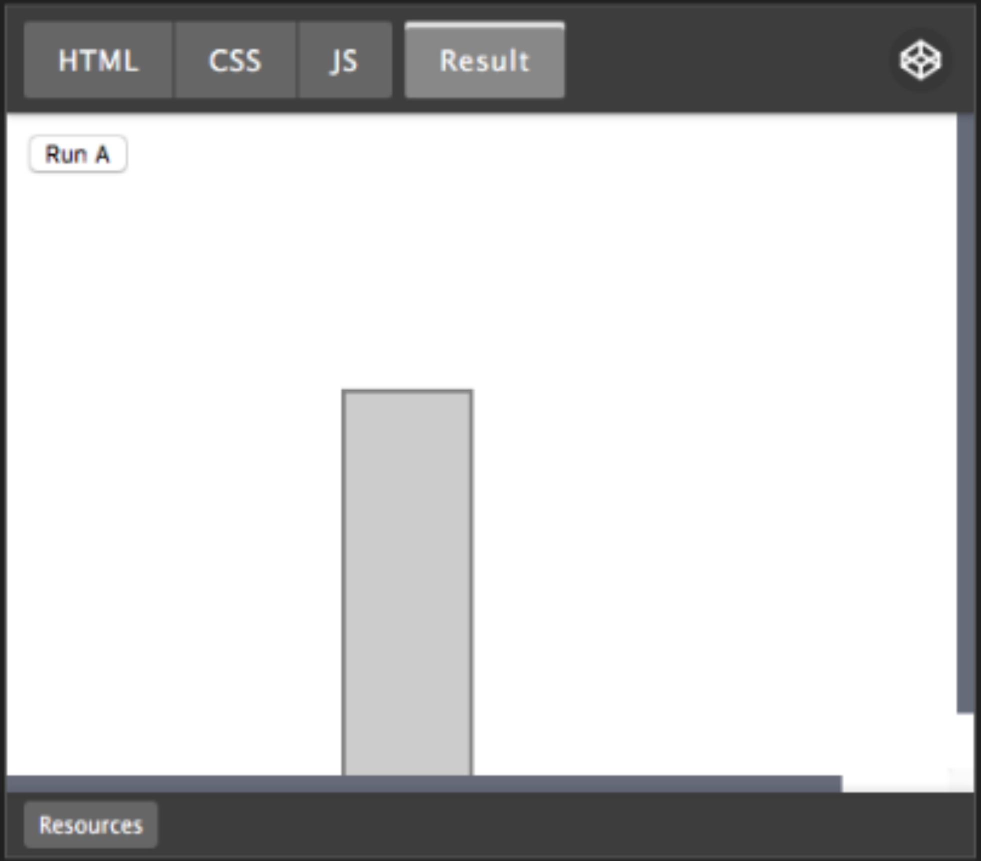


```
var svg = d3.select("svg");

var margin = {top: 30, right: 50, bottom: 30, left: 30},
    width = +svg.attr("width") - margin.left - margin.right,
    height = +svg.attr("height") - margin.top - margin.bottom,
    labelPadding = 3;

var g = svg.append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

.select()



```
var svg = d3.select("svg")

var run = function() {
  var myRect = svg.select("rect")
  myRect.attr("width", 100)
  myRect.attr("height", 100)
  myRect.style("fill", "steelblue")
}
```



External data files

- d3.csv - request a comma-separated values (CSV) file.
- d3.html - request an HTML document fragment.
- d3.json - request a JSON blob.
- d3.text - request a text file.
- d3.tsv - request a tab-separated values (TSV) file.
- d3.xhr - request a resource using XMLHttpRequest.
- d3.xml - request an XML document fragment.

- See more: github.com/mbostock/d3/wiki/Requests

Data joints

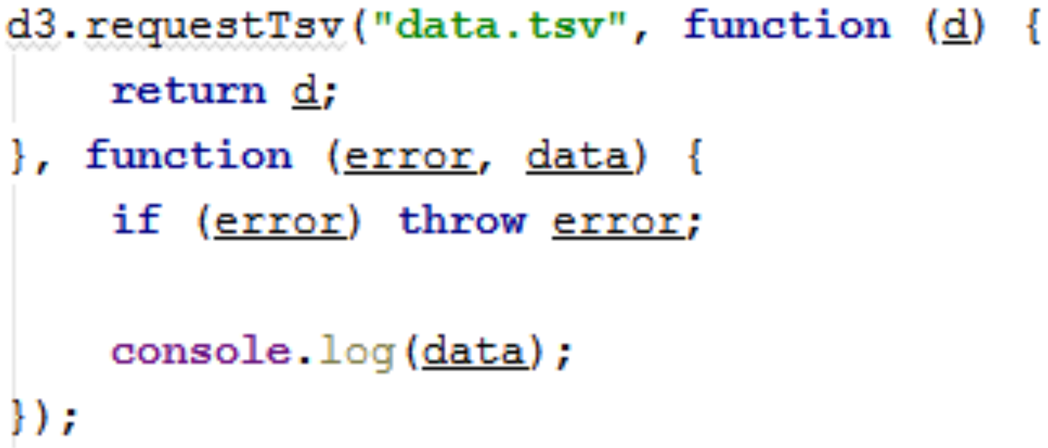
The act of creating a mapping between data points and the objects representing them.

- `[1,2,3] -> µ £ ¥`

-
  ```
  [1,2,3] -> <div>1</div>
              <div>2</div>
              <div>3</div>
  ```

- See more: github.com/mbostock/d3/wiki/Requests

Fetch data from file



```
d3.requestTsv("data.tsv", function (d) {
    return d;
}, function (error, data) {
    if (error) throw error;

    console.log(data);
});
```

Layouts

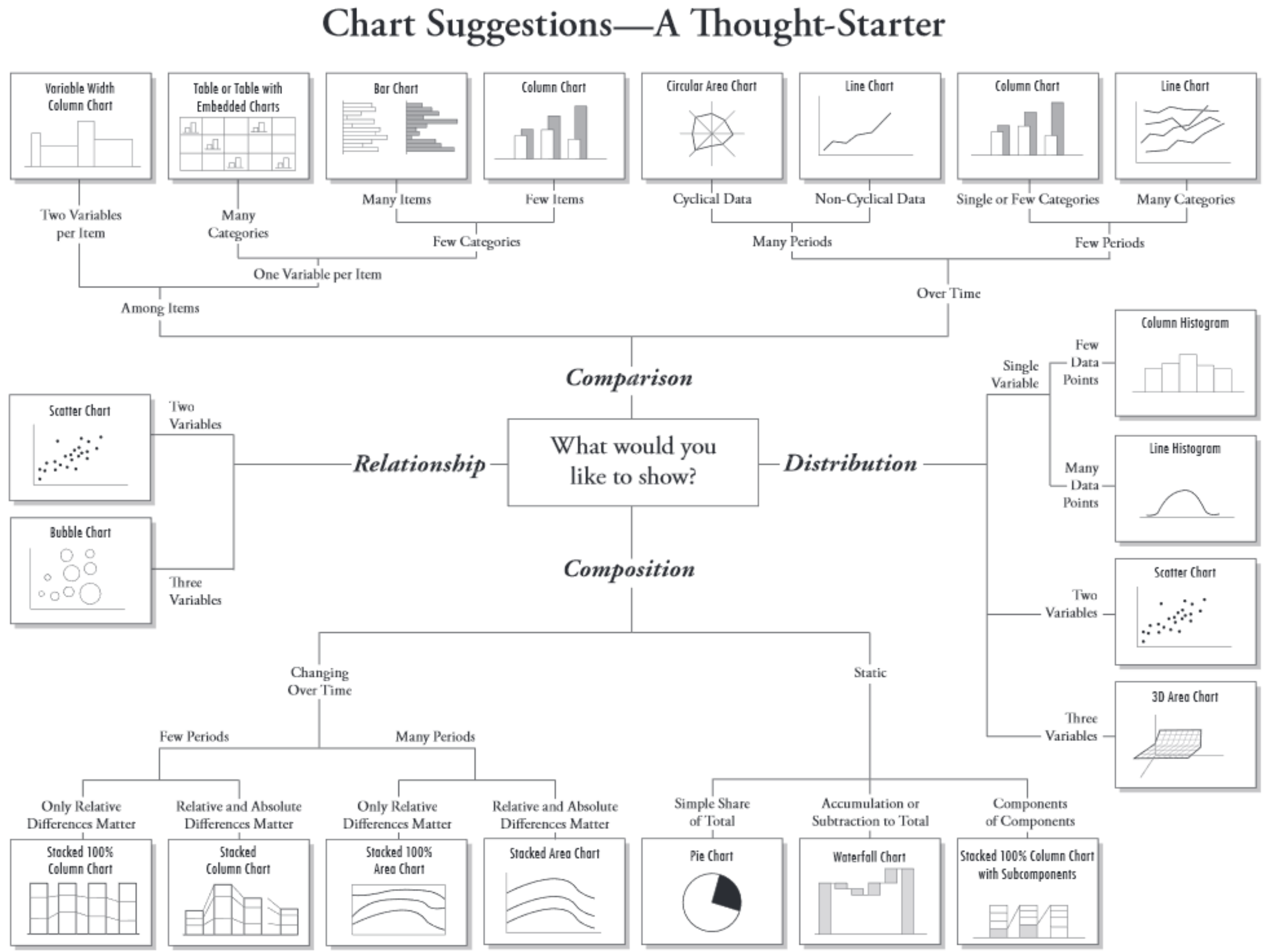
- A layout encapsulates a strategy for laying out data elements visually, relative to each other. Many layouts are built in to D3 itself:
  - Chord - produce a chord diagram from a matrix of relationships.
  - Partition - recursively partition a node tree into a sunburst or icicle.
  - Pie - compute the start and end angles for arcs in a pie or donut chart.
  - Tree - position a tree of nodes tidily.
  - etc.
- See more: github.com/mbostock/d3/wiki/Layouts

How to select a chart type?

Main purpose of data visualization:

- Comparison
- Composition
- Distribution
- Relationship

- See more: goo.gl/1jkVNk
- Try by yourself: goo.gl/gl6Q4R



Source: storytellingwithdata.com

Plotly

What is Plotly?

- Plotly is one of the finest data visualization tools available built on top of visualization library D3.js, HTML and CSS.
- It is created using Python and the Django framework.
- One can choose to create interactive data visualizations online or use the libraries that plotly offers to create these visualizations in the language/ tool of choice. It is compatible with a number of languages/ tools: R, Python, MATLAB, Perl, Julia, Arduino

Other tools that use D3.js

- Bokeh is a competitor to Plotly. It's open source, interactive, and works with Python.
- Holoviews is a higher-level wrapper around MatplotLib, Bokeh, and now Plotly. I know folks who like Holoviews.
- R folks love Shiny from RStudio. It lets' R users create interactive web application visualizations.
- Tableau and Microsoft PowerBI are two other popular drag and drop data visualization creation options that integrate with Python. Tableau is really nice, but requires you to pay money to keep your work private.

Other tools

- ❖Charts wizards (Libre Office, MS Office, Numbers)
- ❖ Google Charts
- ❖ Modest Maps (mapping tool)
- ❖ Visual.ly
- ❖ Tableau
- ❖ RAW (from Density Design)
- ❖ alternatives to D3.js