Introduction to data visualization with D3.js

Aleksander Fabijan

Why do we visualize data?

?

Data Visualization

Data visualization is the presentation of data in a pictorial or graphical format.

Reasons to visualize things:

- helps people see **things that were not obvious** to them before;
- **patterns** can be spotted quickly and easily;
- conveys information in a universal manner;
- answer questions like “What would happen if we make an adjustment?”.

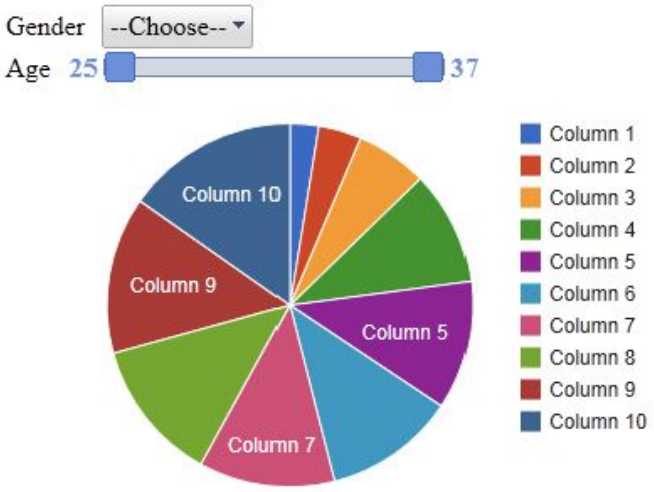
Visualization Libraries

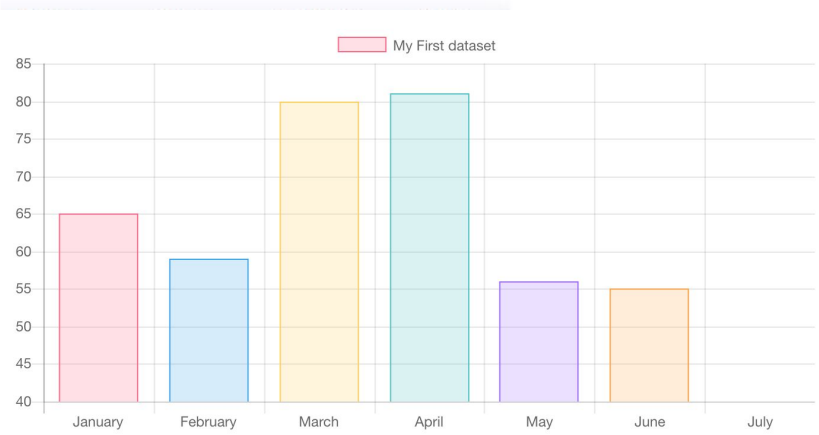
There exists lots of visualization libraries.

See e.g. the following:

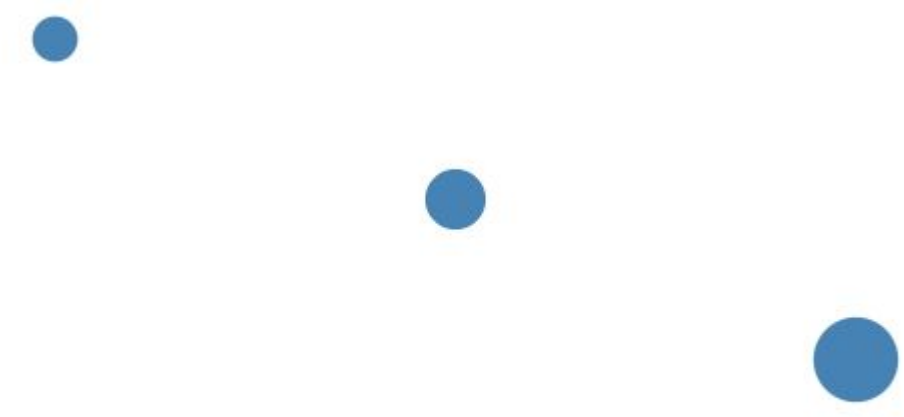
- Grafana : http://play.grafana.org/
- Chart.js: http://www.chartjs.org/
- Google graphs,
- D3.js: https://d3js.org/
- ...

1.
Motivation for D3.JS





Is there a more innovative way to visualize information?



At the end of the lesson...

D3 - Motivation

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

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

Main motivation

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

Uses of D3

It is: https://bl.ocks.org/mbostock

2.
Data Visualization with D3
The basics

The three Main concepts

1. **Selections**
2. **SVG**
3. **Data Binding**

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>
```

Selection

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.

Jquery vs. D3

```
// 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!");
});
```

Jquery vs. D3

```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!");
});
```

```d3
// Find element
var node = d3.select("#elementId");

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

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

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

Jquery vs. D3

```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!");
});
```

```d3
// Find element
var node = d3.select("#elementId");

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

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

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

SVG

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

SVG Example

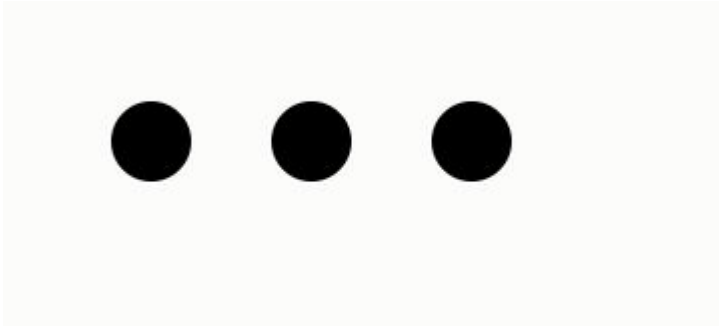
```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
   <circle cx="50" cy="50" r="40" stroke="green"
stroke-width="4" fill="yellow" />
   Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

https://www.w3schools.com/html/html5_svg.asp

Another Example

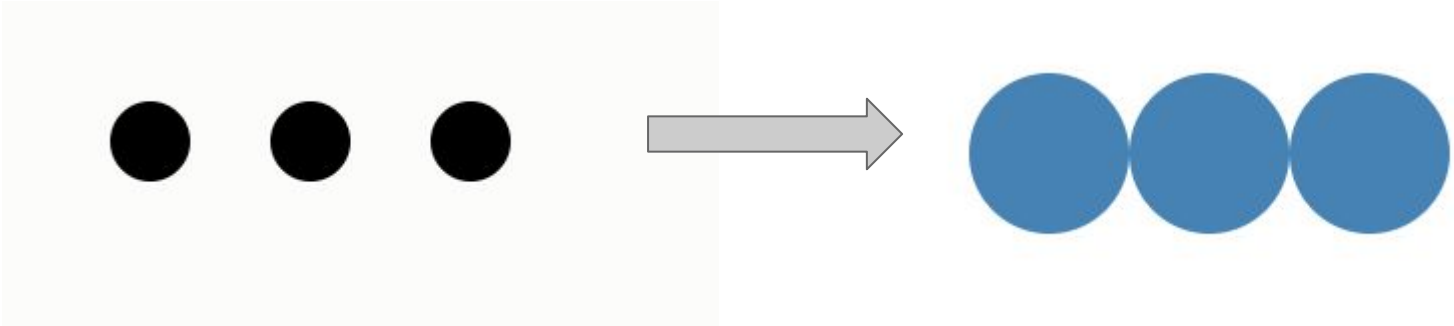
```
<svg width="720" height="120">
  <circle cx="40" cy="60" r="10"></circle>
  <circle cx="80" cy="60" r="10"></circle>
  <circle cx="120" cy="60" r="10"></circle>
</svg>
```

Location X Location Y Radius

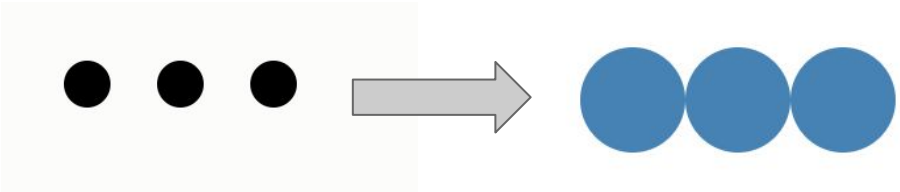


D3 Selecting

Challenge for the audience:



D3 Selecting



```
var circle = d3.selectAll("circle");

circle.style("fill", "steelblue");

circle.attr("r", 20);
```

Data-binding

More commonly, we use data to drive the appearance of our elements.

Let’s say we want these circles represent the numbers 32, 57 and 112.

Data-binding Cnt.

```
//Define an input dataset
var inputDataset = [32, 57, 112];

//Bind the element with the data
circle.data(inputDataset);

//Set the attribute based on the dataset value
circle.attr("r", function(d) { return Math.sqrt(d); });
circle.attr("cx", function(d, i) { return i * 100 + 30; });
```

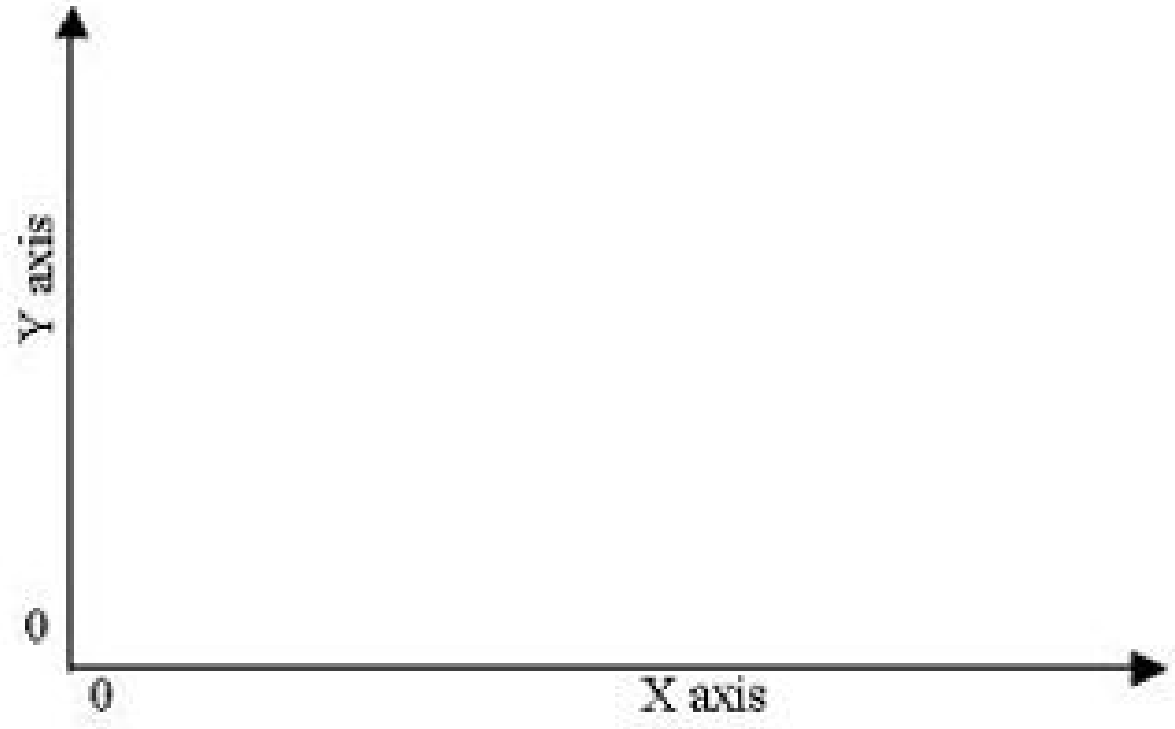
3.
Coordinate System

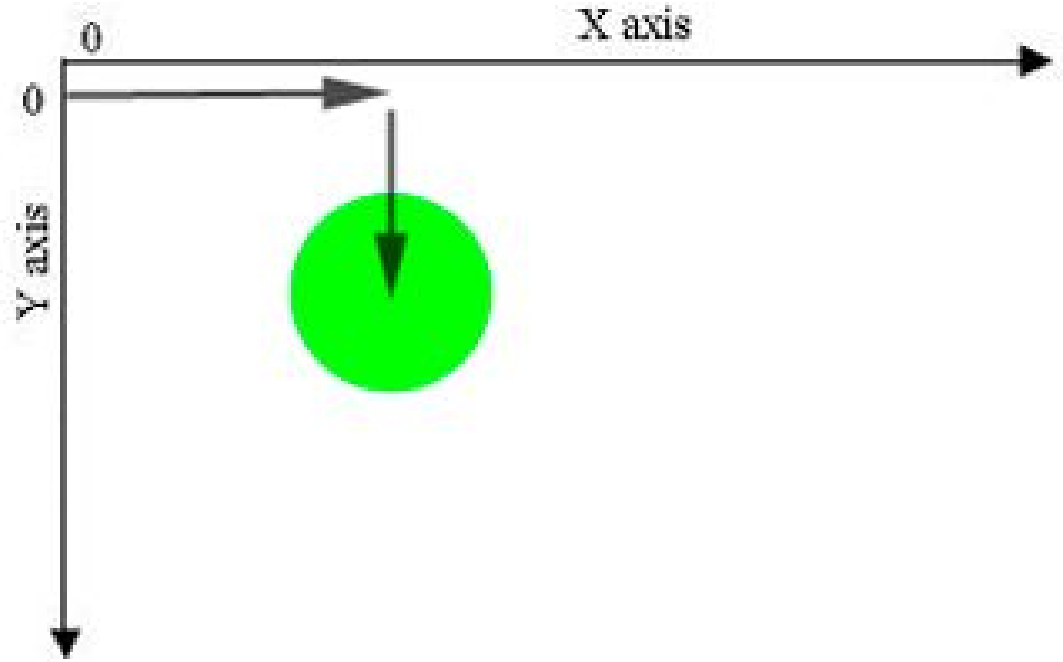
Coordinate System

SVG Coordinate space has x=0 and y=0 coordinates fall on the top left.

SVG Coordinate space has the **Y coordinate growing from top to bottom**.

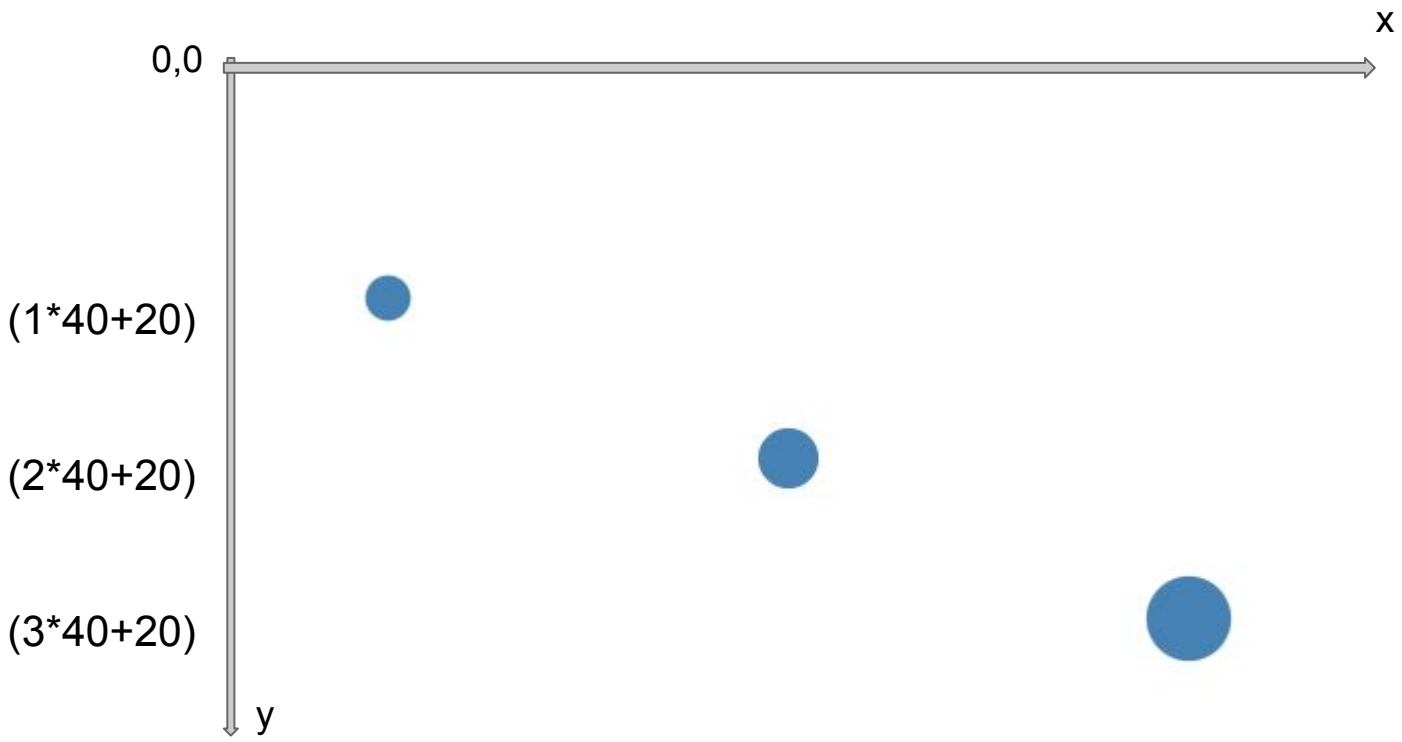
A typical coordinate system…. Not D3!!





D3 coordinate system!

```
circle.attr("cy", function(d, i) { return i * 40 + 20; });
```



4.
Scales

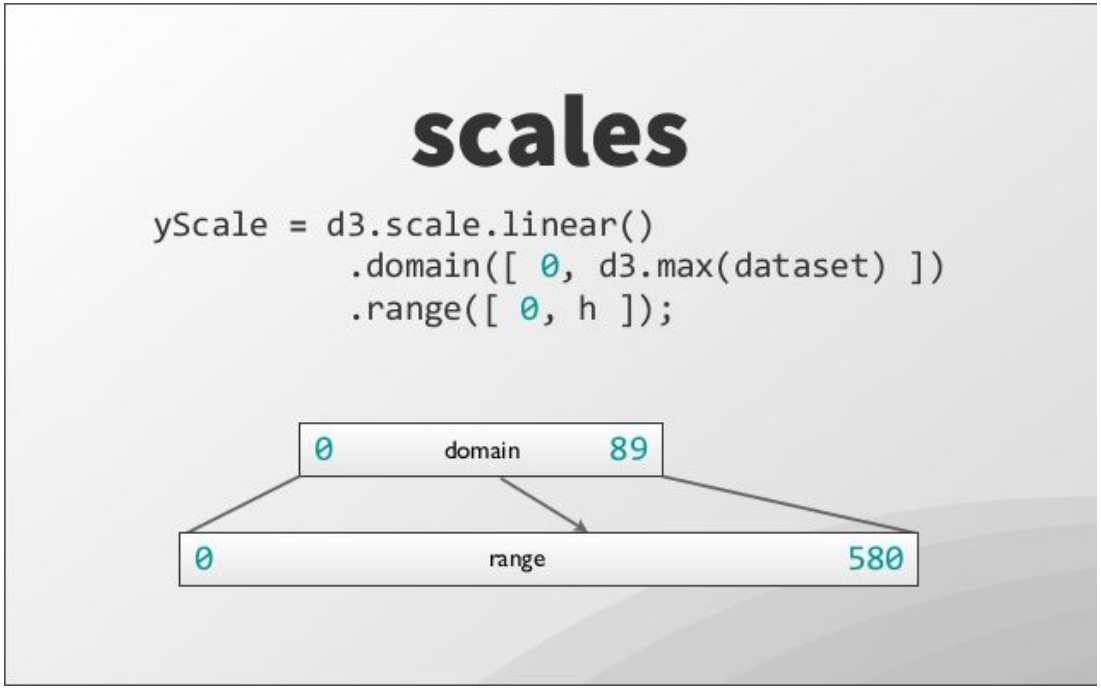
Problem with Data

Sometimes data value will be out of scale!!! How do we plot them?

var dataset3 = [{height: 2000}, {height: 4000}, {height: 3000}, {height: 6000}, {height: 8000}, {height: 9000}, {height: 6000}, {height: 1000}];

D3 Scale for the rescue!

Our input data range domain() is transformed into our drawing range: range();



d3.scaleLinear() for V4 of D3

Example

Let's Assume that my frame is 100pix high. I need to place my data that is between 0 and 30000 into this range 0,100.

```
var y = d3.scaleLinear()
        .domain([0,30000]
        .range([0,100]);
```

4.
Resources

Useful resources

https://www.dashingd3js.com/using-the-svg-coordinate-space

https://d3js.org/

**http://www.storytellingwithdata.com/blog/2013/04/chart-chooser

bl.ocks.org/mbostock

christopheviau.com/d3list

“D3 Tutorials” by Scott Murray

“Introduction to D3” by Irene Ros

"Data Visualization with D3.js" by Lynda.com

"Data Visualization and D3.js" by Udacity

Thanks!

You can find me at:
aleksander.fabijan@mah.se