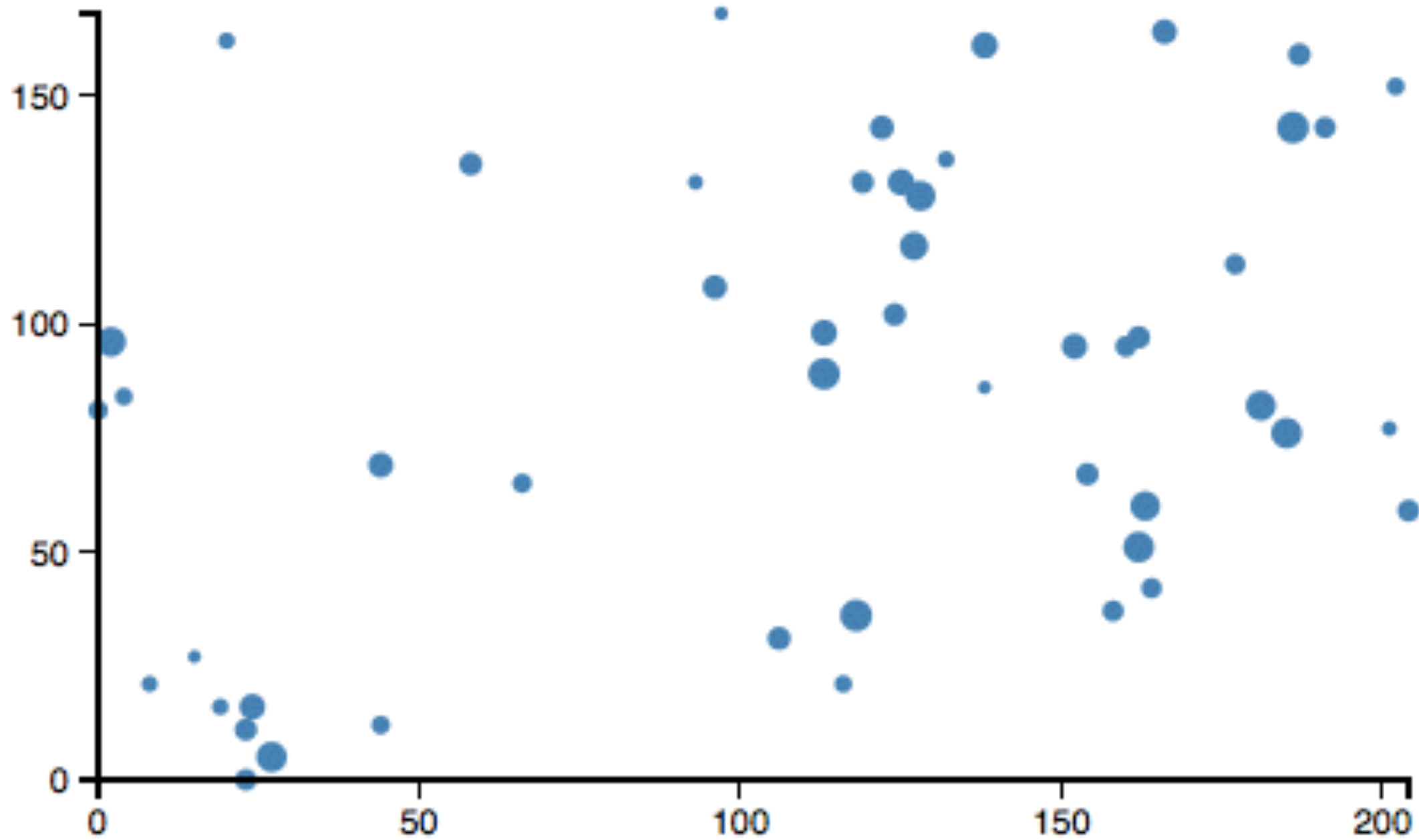


DataScience for Development and Social Change, 2015

D3

How to get started without
getting scared...

D3 (“Data Driven Documents”)



Where you're going: a D3 File

```
var xScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[0]; })])
  .range([padding, w - padding * 2]);

var yScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[1]; })])
  .range([h - padding, padding]);

var rScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[2]; })])
  .range([2, 5]);

var svg = d3.select("body")
  .append("svg")
  .attr("width", w)
  .attr("height", h);

svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle")
  .attr("fill", "steelblue")
  .attr("cx", function(d) {
    return xScale(d[0]);
  })
  .attr("cy", function(d) {
    return yScale(d[1]);
  })
  .attr("r", function(d) {
    return rScale(d[2]);
  });
```

We start with HTML

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Hello! This is my title</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Hello World!</h1>
```

```
</body>
```

```
</html>
```

DOM (Document Object Model)

- ❖ Open your html file in Chrome
- ❖ Click view->developer->javascript console
- ❖ type “document” in the new window

```
> document
< ▼ #document
  ▼ <html lang="en">
    ▼ <head>
      <title>Hello! This is my title</title>
    </head>
    ▼ <body>
      <h1>Hello World!</h1>
    </body>
  </html>
> |
```

Add some D3

```
<html lang="en">
```

```
<head>
```

```
  <title>this is where you put your title</title>
```

```
  <meta charset="utf-8">
```

```
  <script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>
```

```
</head>
```

```
<body>
```

```
  <script type="text/javascript">
```

```
    d3.select("body").append("p").text("This is my new paragraph!");
```

```
  </script>
```

```
</body>
```

```
</html>
```


Type “document” in the javascript console

```
> document
< ▼ #document
  ▼ <html lang="en">
    ▼ <head>
      <title>Hello! This is my title</title>
      <meta charset="utf-8">
      <script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>
    </head>
    ▼ <body>
      <h1>Hello World!</h1>
      <script type="text/javascript">
        d3.select('body').append('p').text('This is my new paragraph!');
      </script>
      <p>This is my new paragraph!</p>
    </body>
  </html>
```

Why do we care about the DOM?

- ❖ The DOM is created when you open a webpage
- ❖ It contains the html code for that page
- ❖ And handles things like mouse clicks on the page
- ❖ D3 works by changing the DOM.
 - ❖ Just like the last example, where “<p>New paragraph!</p>” appeared in the DOM.

Adding D3 to your HTML

- ❖ If you have stable Internet:
 - ❖ use <http://d3js.org/d3.v3.min.js>
 - ❖ `<script type="text/javascript" src="d3/d3.js"></script>`
- ❖ If you have no Internet:
 - ❖ Download the D3 library from <http://d3js.org/>
 - ❖ use `d3/d3.js`
 - ❖ `<script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>`

Drawing Shapes in HTML

SVG graphics language: draws things like circles and rectangles...

```
<svg width="100" height="100">
```

```
  <circle cx="50" cy="50" r="40" stroke="blue" stroke-width="4"  
  fill="yellow" />
```

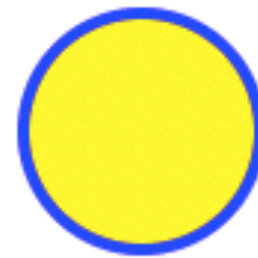
```
</svg>
```

```
<svg width="400" height="110">
```

```
  <rect width="300" height="100" style="fill:rgb(0,0,255); stroke-  
width:3; stroke:rgb(0,0,0)">
```

```
</svg>
```

Circle



Rectangle



Drawing Lines and Text with SVG

And lines and text

```
<svg height="100" width="500">  
  <line x1="0" y1="0" x2="200" y2="80"  
  style="stroke:rgb(255,0,0);stroke-width:5" />  
</svg>
```

```
<svg height="30" width="200">  
  <text x="0" y="15" fill="red">Written by SVG</text>  
</svg>
```

Line



Text

Written by SVG

CSS (Cascading StyleSheets) makes HTML pretty

- ❖ HTML classes and ids:

```
<p class="orange">This is an orange paragraph</p>
```

```
<p id="mylabel">This paragraph is labelled</p>
```

- ❖ CSS:

```
p {
```

```
  color: blue; /* Turn every paragraph blue */
```

```
}
```

```
#mylabel {
```

```
  font-style: bold; /* Make the mylabel paragraph text bold */
```

```
}
```

D3 is a Javascript library

- ❖ Javascript = “front-end” language, used in webpages
- ❖ Javascript console:
 - ❖ You can type javascript directly into the console

Javascript variables and functions

❖ Javascript variables:

```
var my_number = 500;
```

```
var my_array = [1,2,3];
```

❖ Javascript functions:

```
function mySum(a, b) {
```

```
    return a * b;
```

```
}
```

```
mySum(2,4)
```

A basic D3 visualization

- ❖ We need:
 - ❖ Styles (line colors, text fonts etc) - we have this
 - ❖ A drawing area
 - ❖ A dataset (x, y values)
 - ❖ x and y scales (e.g. a coordinate system)
 - ❖ Shapes
 - ❖ x and y axes

Start with a blank HTML page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>This is your title</title>
    <meta charset="UTF8">
    <script type="text/javascript" src="d3/d3.js"></script>
  </head>
  <body>

    <script type="text/javascript">
    </script>

  </body>
</html>
```

Viewing your D3 visualization

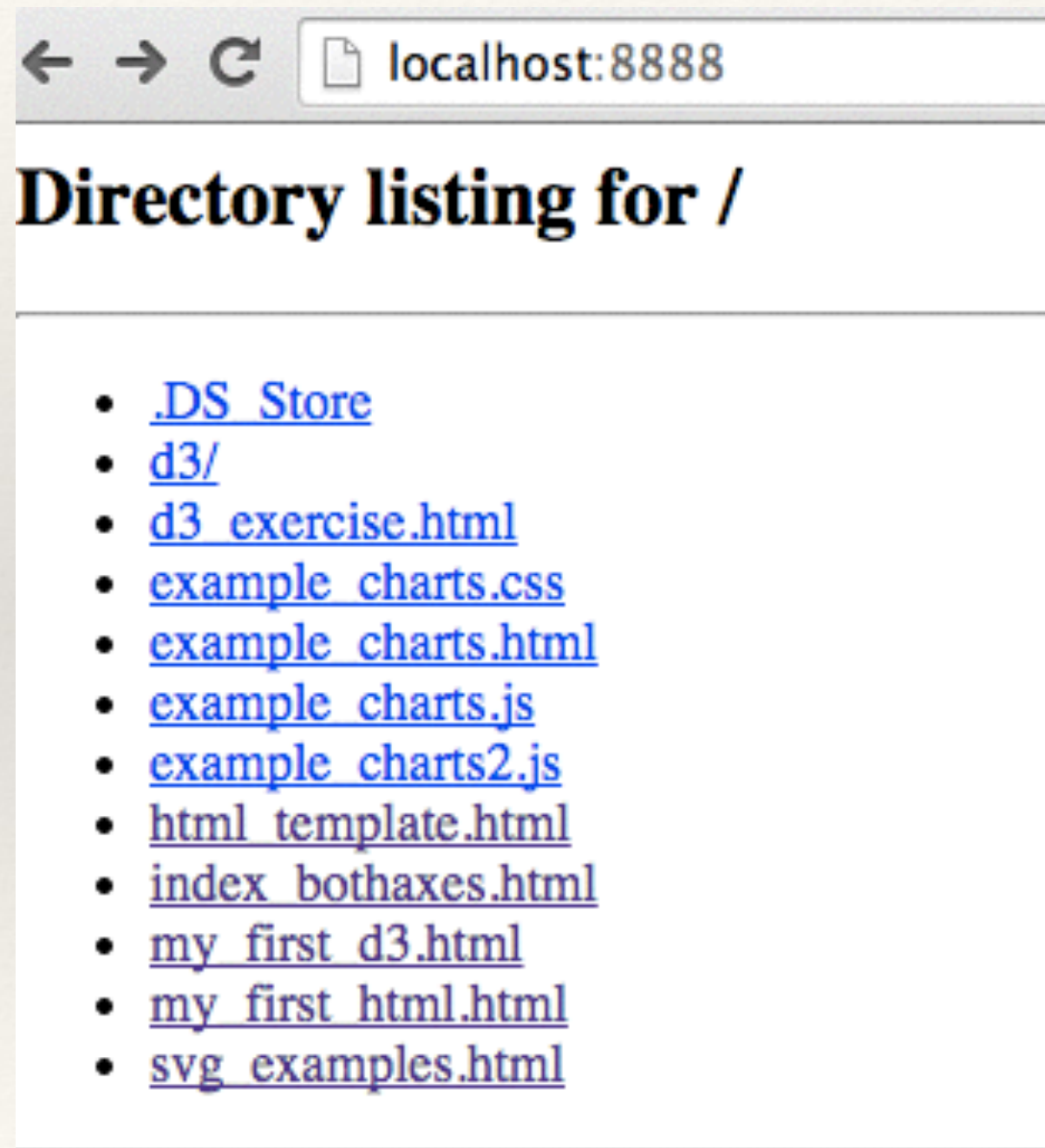
- ❖ In the directory your code is in, type:

```
python -m SimpleHTTPServer 8888
```

- ❖ Then go to <http://localhost:8888/> in your browser

- ❖ (to exit, type control-c)

You should see something like this



Add Style

```
<style>
  .axis path,
  .axis line {
    fill: none;
    stroke: black;
    shape-rendering: crispEdges;
  }
  .axis text {
    font-family: sans-serif;
    font-size: 11px;
  }
</style>
```


You should be here...

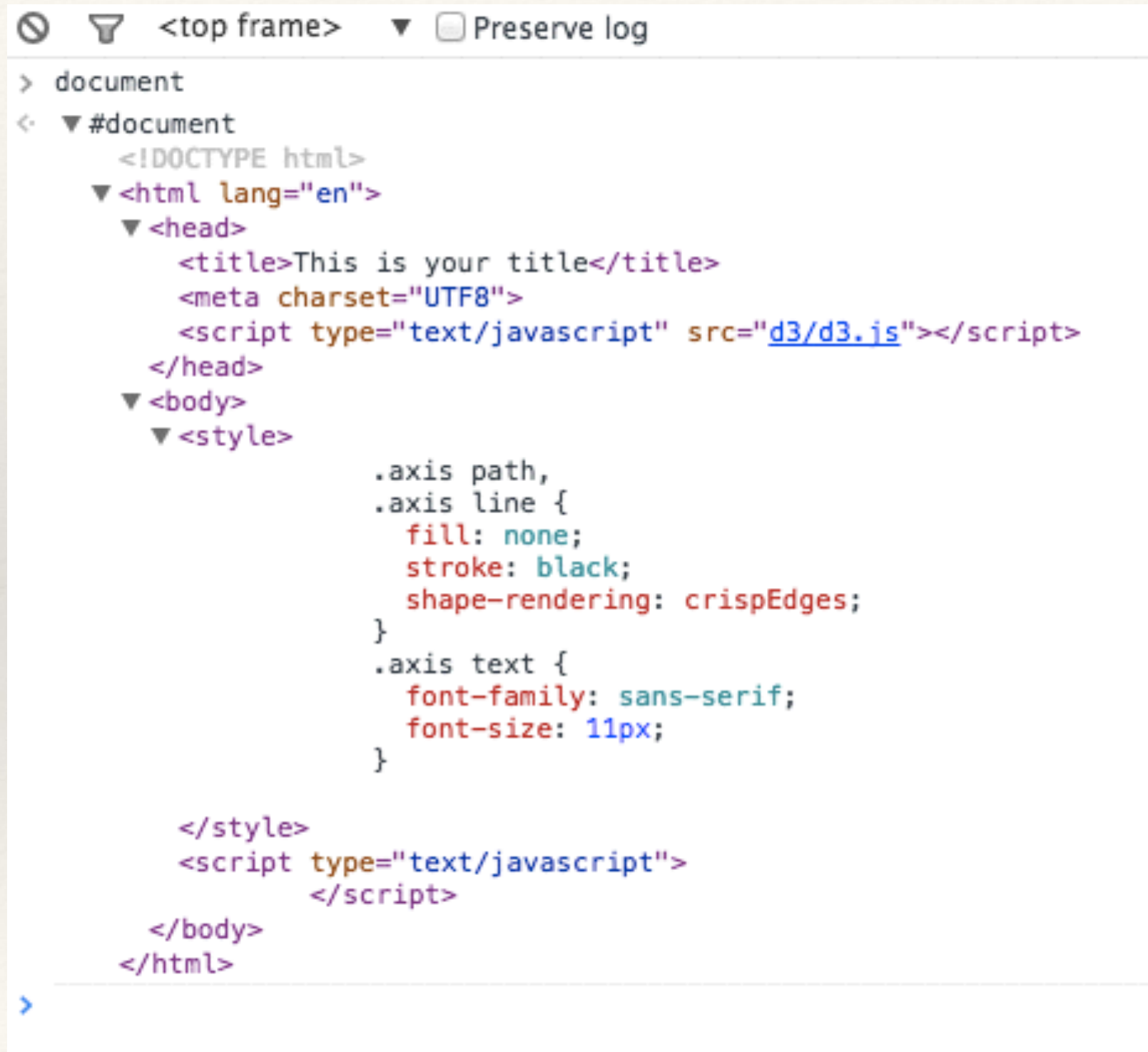
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>This is your title</title>
    <meta charset="UTF8">
    <script type="text/javascript" src="d3/d3.js"></script>
  </head>
  <body>

    <style>
      .axis path,
      .axis line {
        fill: none;
        stroke: black;
        shape-rendering: crispEdges;
      }
      .axis text {
        font-family: sans-serif;
        font-size: 11px;
      }
    </style>

    <script type="text/javascript">
    </script>

  </body>
</html>
```


Check the Javascript console



The screenshot shows a web browser's developer console with the 'document' object expanded. The console displays the HTML document structure, including the head and body sections. The head section contains a title, a meta charset, and a script tag for 'd3/d3.js'. The body section contains a style tag with CSS rules for axis paths, lines, and text. The console also shows a 'Preserve log' checkbox and a filter icon.

```
<top frame> [x] Preserve log
> document
< #document
  <!DOCTYPE html>
  <html lang="en">
    <head>
      <title>This is your title</title>
      <meta charset="UTF8">
      <script type="text/javascript" src="d3/d3.js"></script>
    </head>
    <body>
      <style>
        .axis path,
        .axis line {
          fill: none;
          stroke: black;
          shape-rendering: crispEdges;
        }
        .axis text {
          font-family: sans-serif;
          font-size: 11px;
        }
      </style>
      <script type="text/javascript">
        </script>
    </body>
  </html>
```

Create a drawing area

```
var vizwidth=500;  
var vizheight=300;  
var padding=30;  
  
var svg = d3.select("body")  
  .append("svg")  
  .attr("width", vizwidth)  
  .attr("height", vizheight);
```


Your <body> should look like this

```
<body>

  <style>
    .axis path,
    .axis line {
      fill: none;
      stroke: black;
      shape-rendering: crispEdges;
    }
    .axis text {
      font-family: sans-serif;
      font-size: 11px;
    }
  </style>

  <script type="text/javascript">
    var vizwidth=500;
    var vizheight=300;
    var padding=30;

    var svg = d3.select("body")
      .append("svg")
      .attr("width", vizwidth)
      .attr("height", vizheight);
  </script>

</body>
```


Your javascript console will look like this

```
> document
< ▼ #document
  <!DOCTYPE html>
  ▼ <html lang="en">
    ▶ <head>...</head>
    ▼ <body>
      ▶ <style>...</style>
      ▼ <script type="text/javascript">
        var vizwidth=500;
        var vizheight=300;
        var padding=30;

        var svg = d3.select("body")
          .append("svg")
          .attr("width", vizwidth)
          .attr("height", vizheight);

        </script>
        <svg width="500" height="300"></svg>
      </body>
    </html>
```

Add some data

```
var dataset = [];  
var numDataPoints = 50;  
var xRange = Math.random() * 1000;  
var yRange = Math.random() * 1000;  
var rRange = Math.random() * 1000;  
for (var i = 0; i < numDataPoints; i++) {  
    var newNumber1 = Math.floor(Math.random() * xRange);  
    var newNumber2 = Math.floor(Math.random() * yRange);  
    var newNumber3 = Math.floor(Math.random() * rRange);  
    dataset.push([newNumber1, newNumber2, newNumber3]);  
}
```


You can check that in the Console too

```
> dataset
< [▼ Array[3] ⓘ , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
  0: 208
  1: 210
  2: 229
  length: 3
  ▶ __proto__: Array[0]
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] , ▶ Array[3] ,
▶ Array[3] , ▶ Array[3] , ▶ Array[3] ]
```

>

Add x, y and radius Scales

```
var xScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[0]; })])
  .range([padding, vizwidth - padding * 2]);

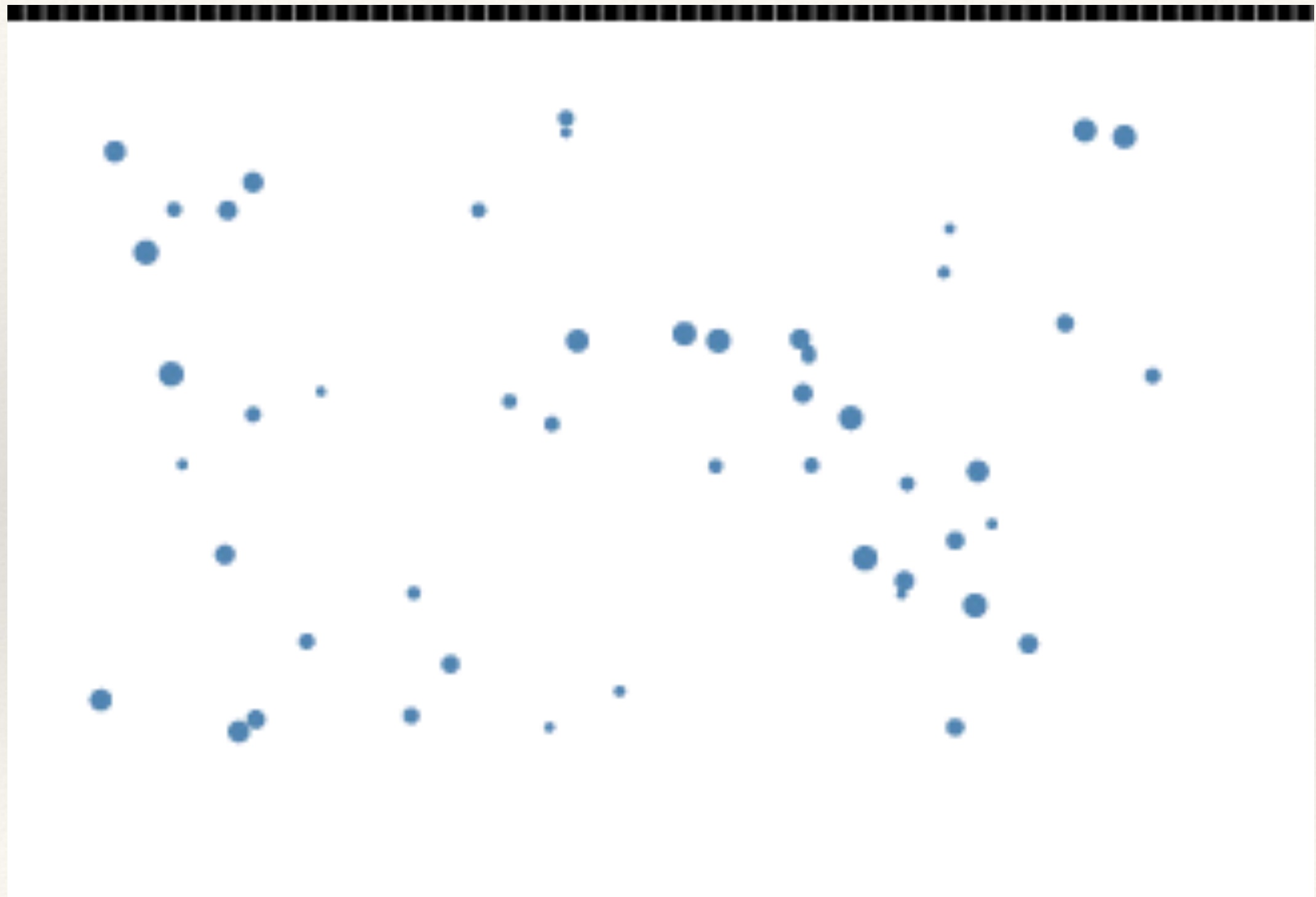
var yScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[1]; })])
  .range([vizheight - padding, padding]);

var rScale = d3.scale.linear()
  .domain([0, d3.max(dataset, function(d) { return d[2]; })])
  .range([2, 5]);
```

Add Shapes

```
svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle")
  .attr("fill", "steelblue")
  .attr("cx", function(d) {
    return xScale(d[0]);
  })
  .attr("cy", function(d) {
    return yScale(d[1]);
  })
  .attr("r", function(d) {
    return rScale(d[2]);
  });
```

Aha! in the browser



Aha! in the Console

```
<script type="text/javascript">...</script>
<svg width="500" height="300">
  <circle fill="steelblue" cx="47.58713136729223" cy="82.3076923076923" r=
    "4.900277008310249"></circle>
  <circle fill="steelblue" cx="344.3699731903485" cy="172.56410256410257" r=
    "3.0055401662049865"></circle>
  <circle fill="steelblue" cx="30" cy="257.008547008547" r=
    "4.360110803324099"></circle>
  <circle fill="steelblue" cx="371.84986595174263" cy="167.77777777777777" r=
    "4.3850415512465375"></circle>
  <circle fill="steelblue" cx="61.87667560321715" cy="165.04273504273505" r=
    "2.2243767313019394"></circle>
  <circle fill="steelblue" cx="232.25201072386056" cy="253.5897435897436" r=
    "2.299168975069252"></circle>
  <circle fill="steelblue" cx="150.91152815013402" cy="263.1623931623932" r=
    "3.263157894736842"></circle>
  <circle fill="steelblue" cx="257.5335120643432" cy="114.1025641025641" r=
    "4.717451523545706"></circle>
  <circle fill="steelblue" cx="83.86058981233245" cy="269.3162393162393" r=
    "4.3850415512465375"></circle>
  <circle fill="steelblue" cx="322.38605898123325" cy="146.92307692307693" r=
    "4.78393351800554"></circle>
  <circle fill="steelblue" cx="270.7238605898123" cy="116.83760683760684" r=
    "4.7756232686980615"></circle>
  <circle fill="steelblue" cx="215.76407506702412" cy="116.83760683760684" r=
    "4.542936288088643"></circle>
  <circle fill="steelblue" cx="429.0080428954424" cy="37.17948717948718" r=
```

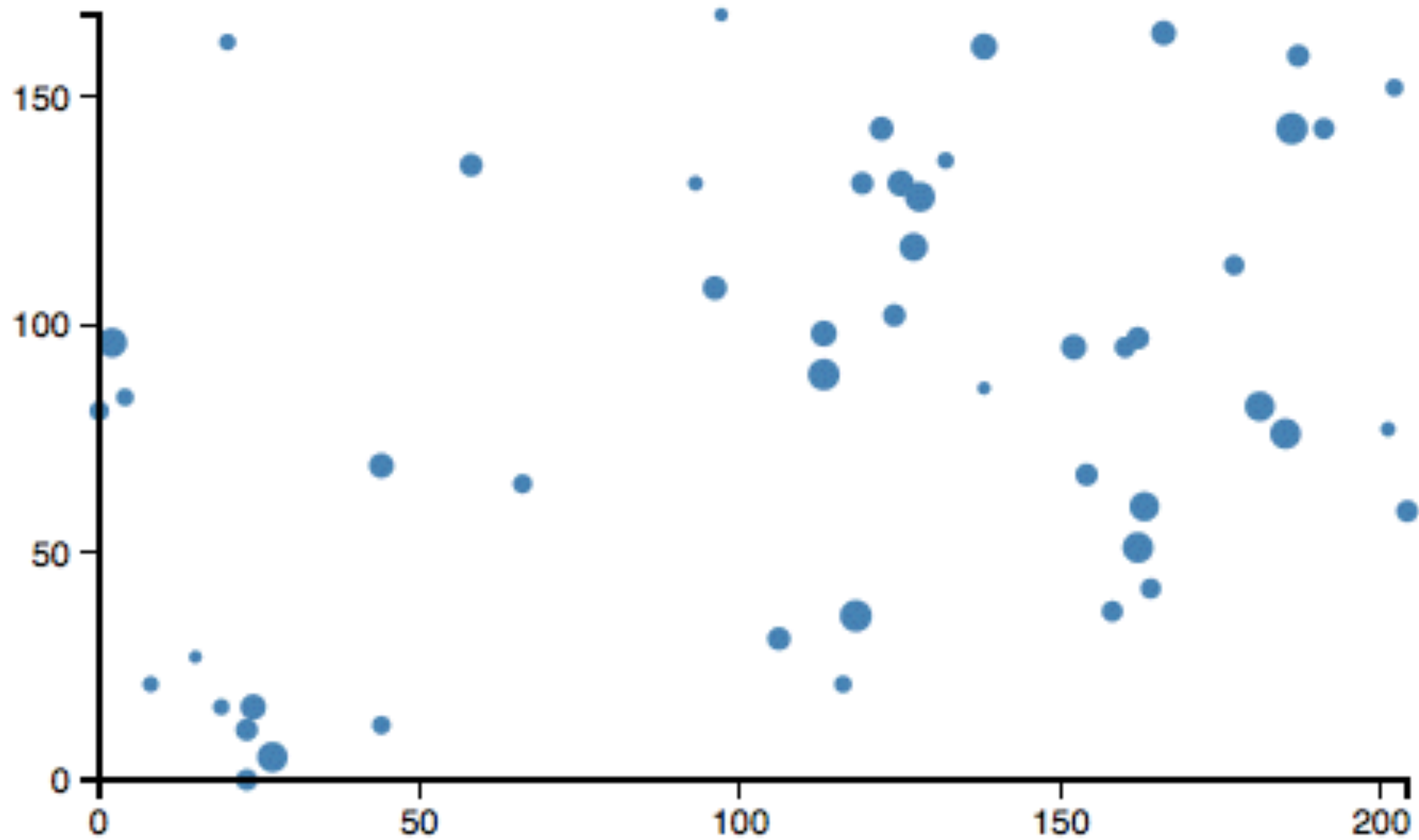

Add x and y Axes

```
var xAxis = d3.svg.axis()  
  .scale(xScale)  
  .orient("bottom")  
  .ticks(5);  
  
svg.append("g")  
  .attr("class", "axis")  
  .attr("transform", "translate(0,"+(vizheight-padding)+")")  
  .call(xAxis);  
  
var yAxis = d3.svg.axis()  
  .scale(yScale)  
  .orient("left")  
  .ticks(5);  
  
svg.append("g")  
  .attr("class", "axis")  
  .attr("transform", "translate("+padding+",0)")  
  .call(yAxis);
```

In the console:

```
▼ <g class="axis" transform="translate(0,270)">
  ▶ <g class="tick" transform="translate(30,0)" style="opacity: 1;">...</g>
  ▶ <g class="tick" transform="translate(116.3157894736842,0)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(202.6315789473684,0)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(288.9473684210526,0)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(375.2631578947368,0)" style="opacity: 1;">
    ...</g>
    <path class="domain" d="M30,6V0H440V6"></path>
  </g>
▼ <g class="axis" transform="translate(30,0)">
  ▶ <g class="tick" transform="translate(0,270)" style="opacity: 1;">...</g>
  ▶ <g class="tick" transform="translate(0,232.61682242990653)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(0,195.2336448598131)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(0,157.85046728971963)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(0,120.46728971962618)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(0,83.08411214953271)" style="opacity: 1;">
    ...</g>
  ▶ <g class="tick" transform="translate(0,45.70093457943925)" style="opacity: 1;">
    ...</g>
    <path class="domain" d="M-6,30H0V270H-6"></path>
  </g>
</svg>
```



The end product:





Putting your D3 code online


- ❖ Put your D3 code onto github pages
 - ❖ <https://pages.github.com/>
- ❖ Add your D3 code to <http://bl.ocks.org/>

<https://github.com/bodacea/bodacea.github.io>

 **3 commits**

 **1 branch**

 **0 releases**

 **1 contributor**

 branch: **master** ▾ **bodacea.github.io** / + 

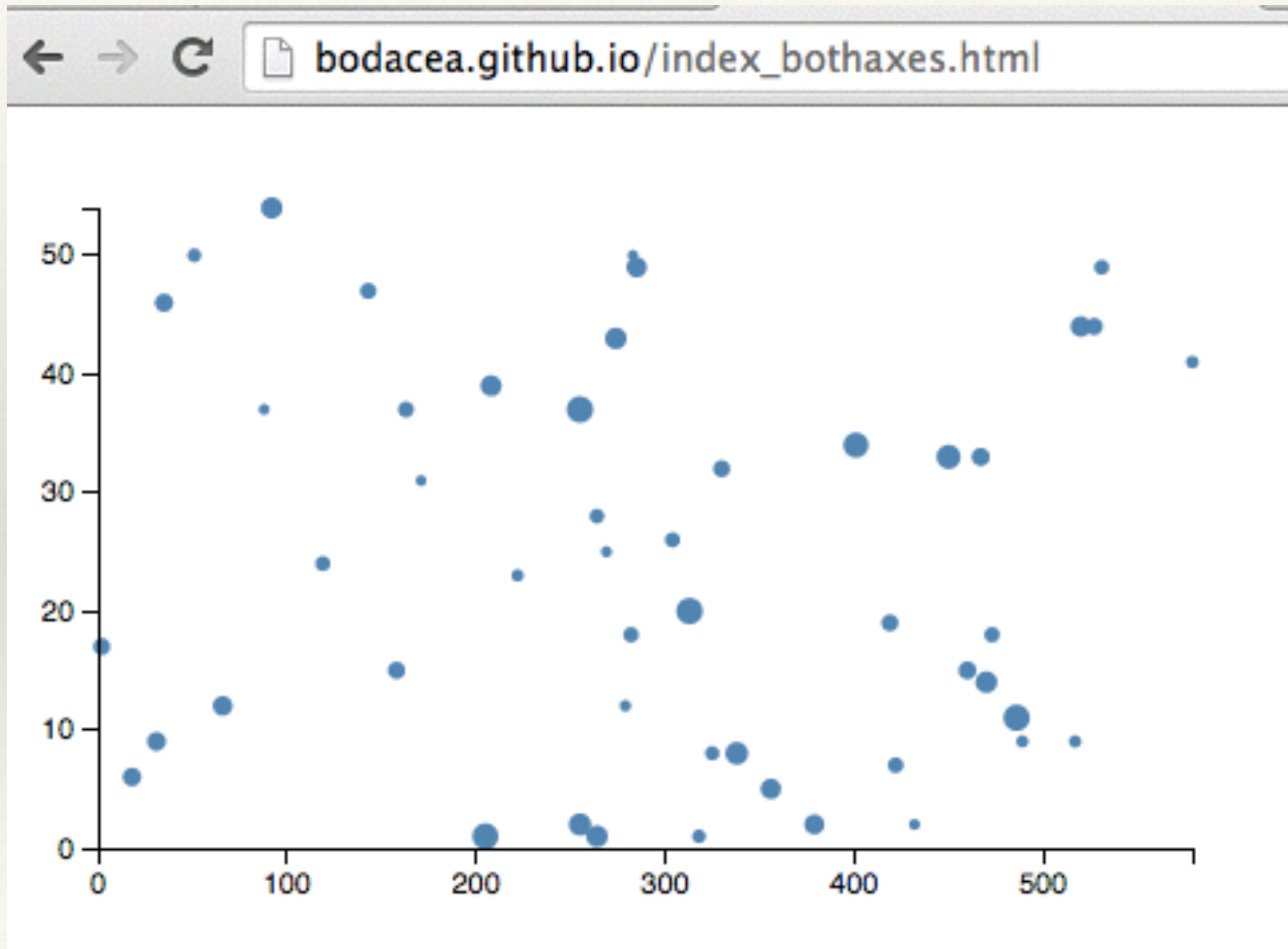
Added d3 library

 **bodacea** authored a minute ago latest commit 8fbff5abf2 

Help people interested in this repository understand your project by adding a README!

 **Add a README**

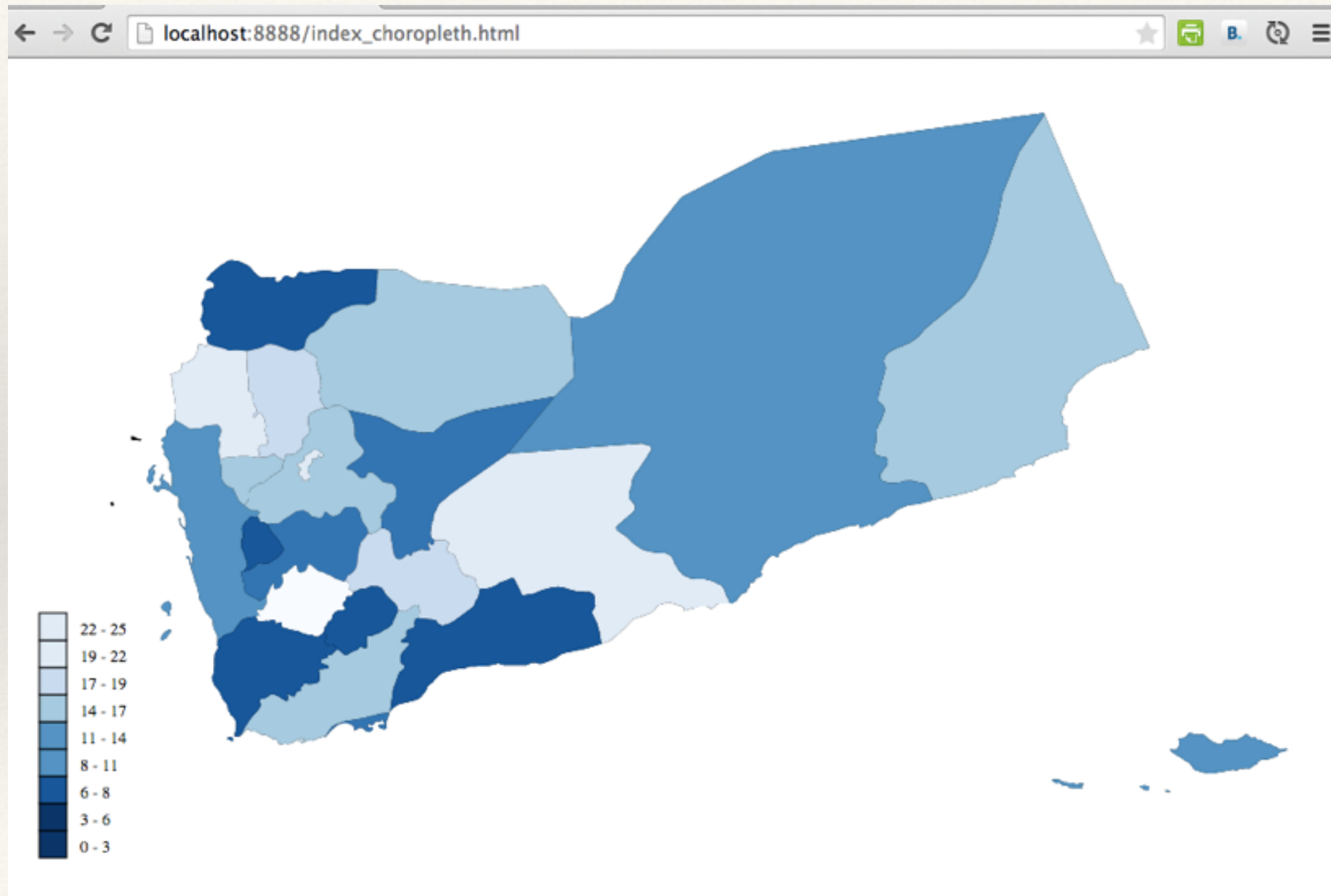
bodacea.github.io/index_bothaxes.html



A D3 Choropleth

- ❖ We need:
 - ❖ Styles (line colors, text fonts etc)
 - ❖ A set of map shapes (e.g. admin boundaries)
 - ❖ A dataset (shapename, value)
 - ❖ A color palette (value, color)
 - ❖ A legend
 - ❖ Tooltips

index_choropleth.html



Connecting D3 and Python

- ❖ Use the Flask python library to create a simple python-based website. Add your D3 code to its html pages
- ❖ The Django python library is slightly more complex, but a popular Python web framework

Making D3 Easier

- ❖ D3 helpers:

- ❖ NVD3

- ❖ xCharts

- ❖ dimple

- ❖ Vega

- ❖ D3 libraries:

- ❖ rCharts

- ❖ d3py

Easier Alternative: Highcharts

- ❖ Highcharts: charts
- ❖ Highstock: timelines
- ❖ Highmaps: maps

<http://www.highcharts.com>

Highcharts is also a Javascript Library

- ❖ `<script src="http://code.highcharts.com/adapters/standalone-framework.js"></script>`

<http://www.highcharts.com/docs/getting-started/installation>

Continuing your D3 journey

- ❖ <https://square.github.io/intro-to-d3/parts-of-a-graph/>
- ❖ Scott Murray's book: <http://chimera.labs.oreilly.com/books/12300000000345>