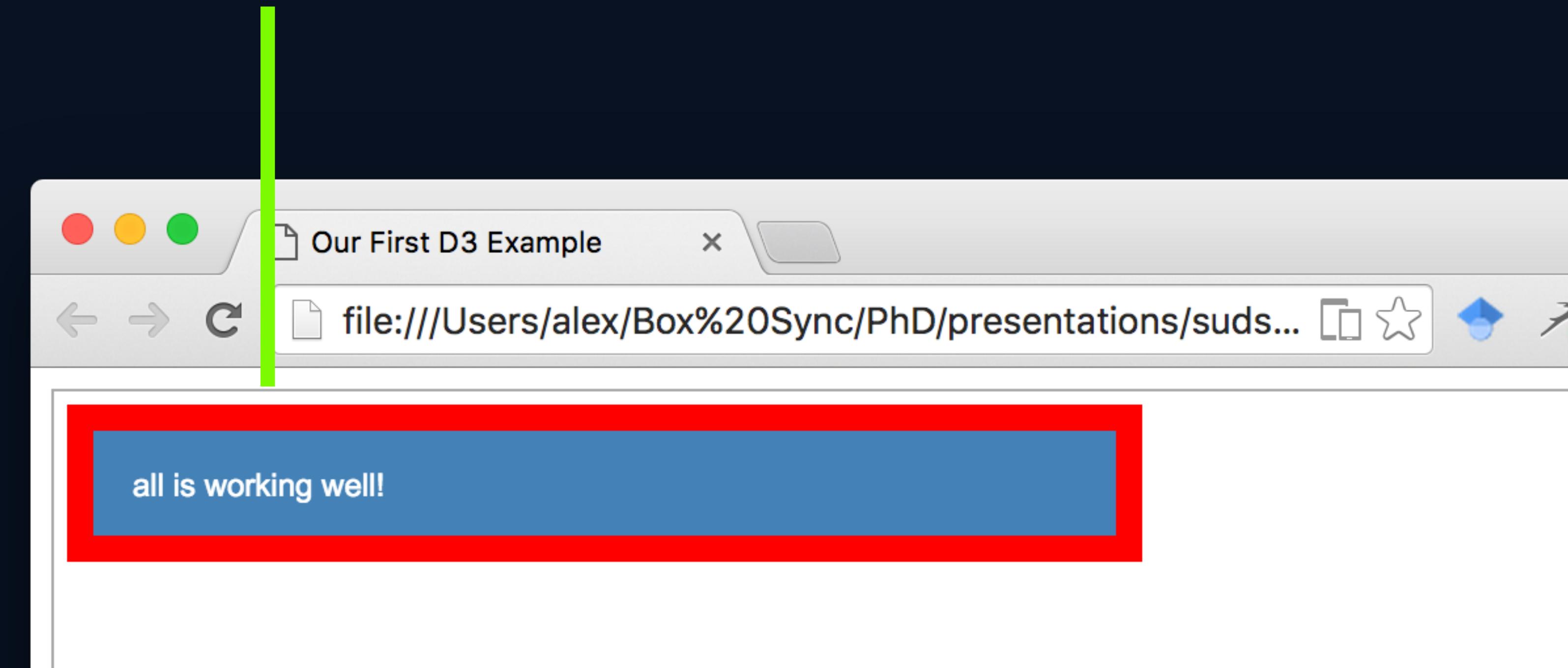


1. download
<https://github.com/SciutoAlex/viz-workshop>

2. open a text editor or download Text Wrangler
Google “textwrangler”

3. Double-click workshop-sketches/00-hello-world.html



Interactive Data Visualization For the Web.

alex sciuto

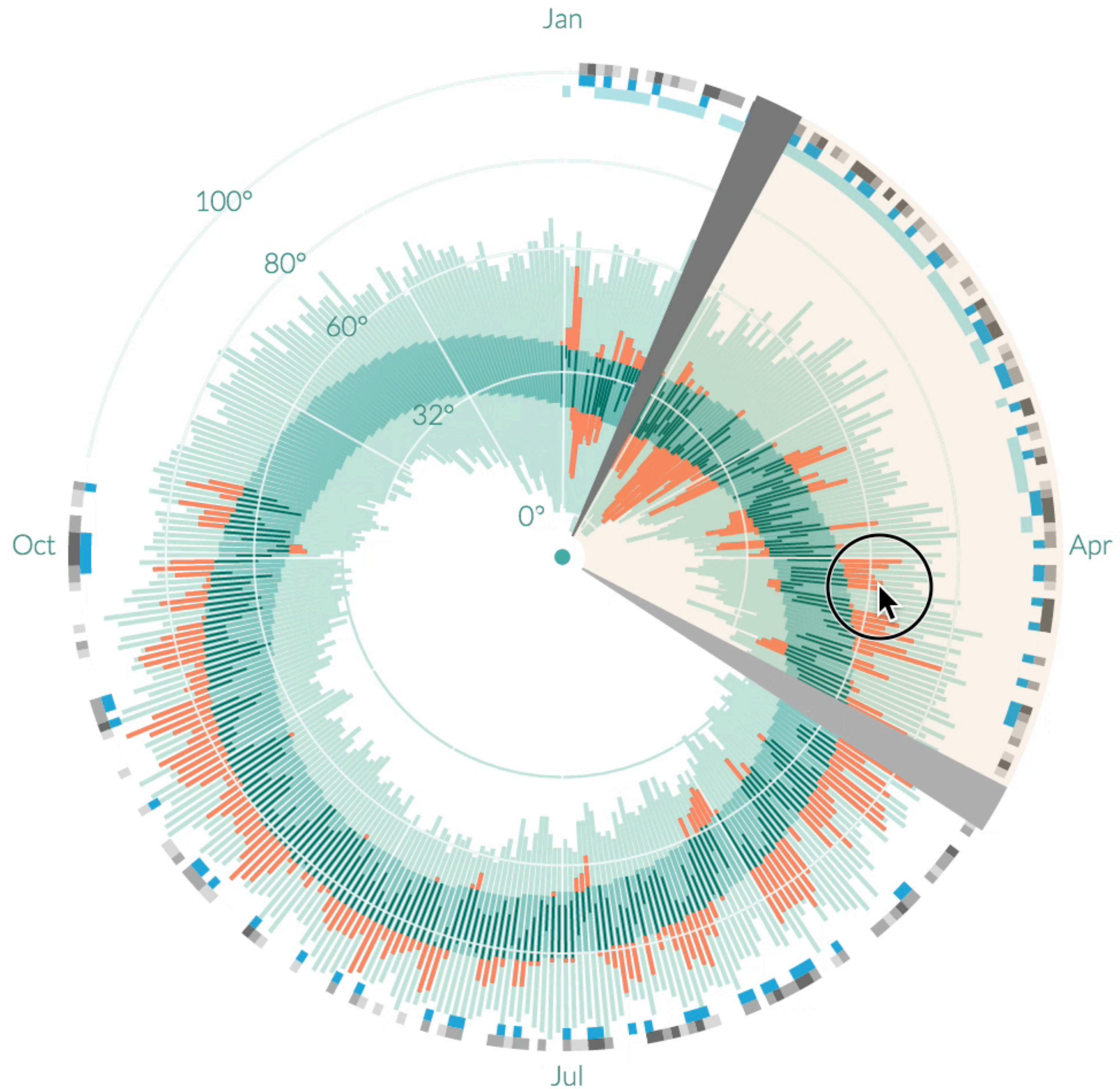
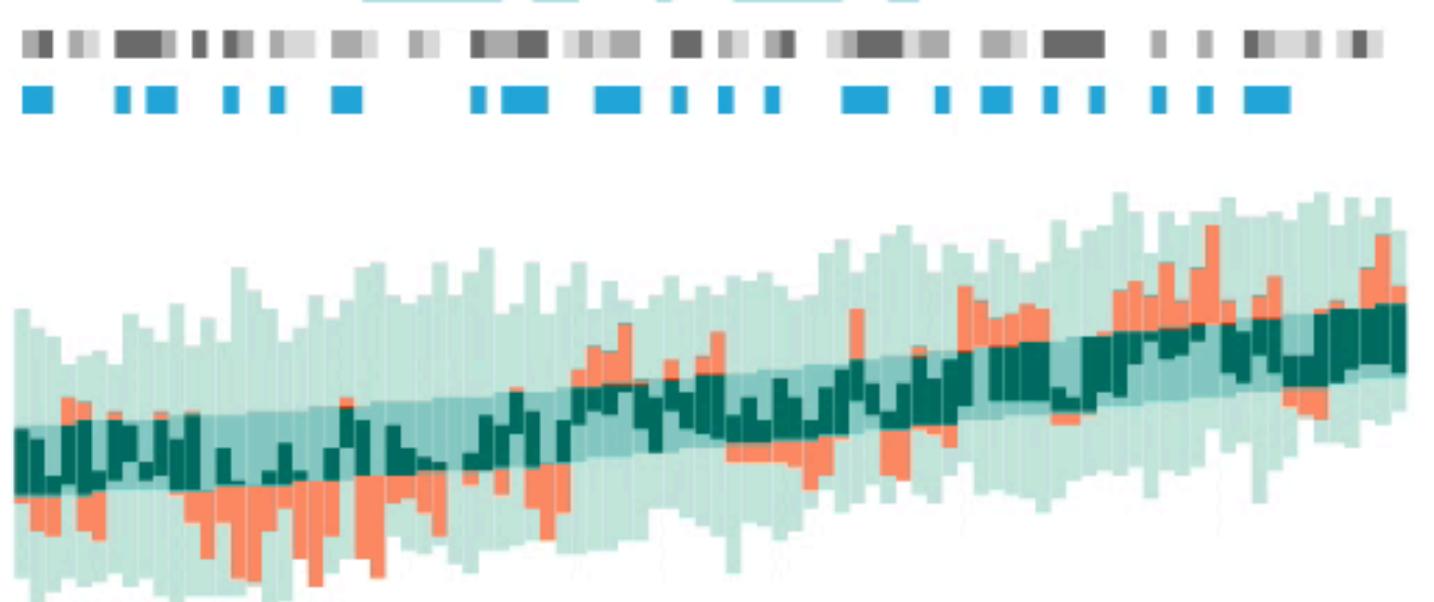
cmu human-computer interaction institute

New York

Historical Weather Data

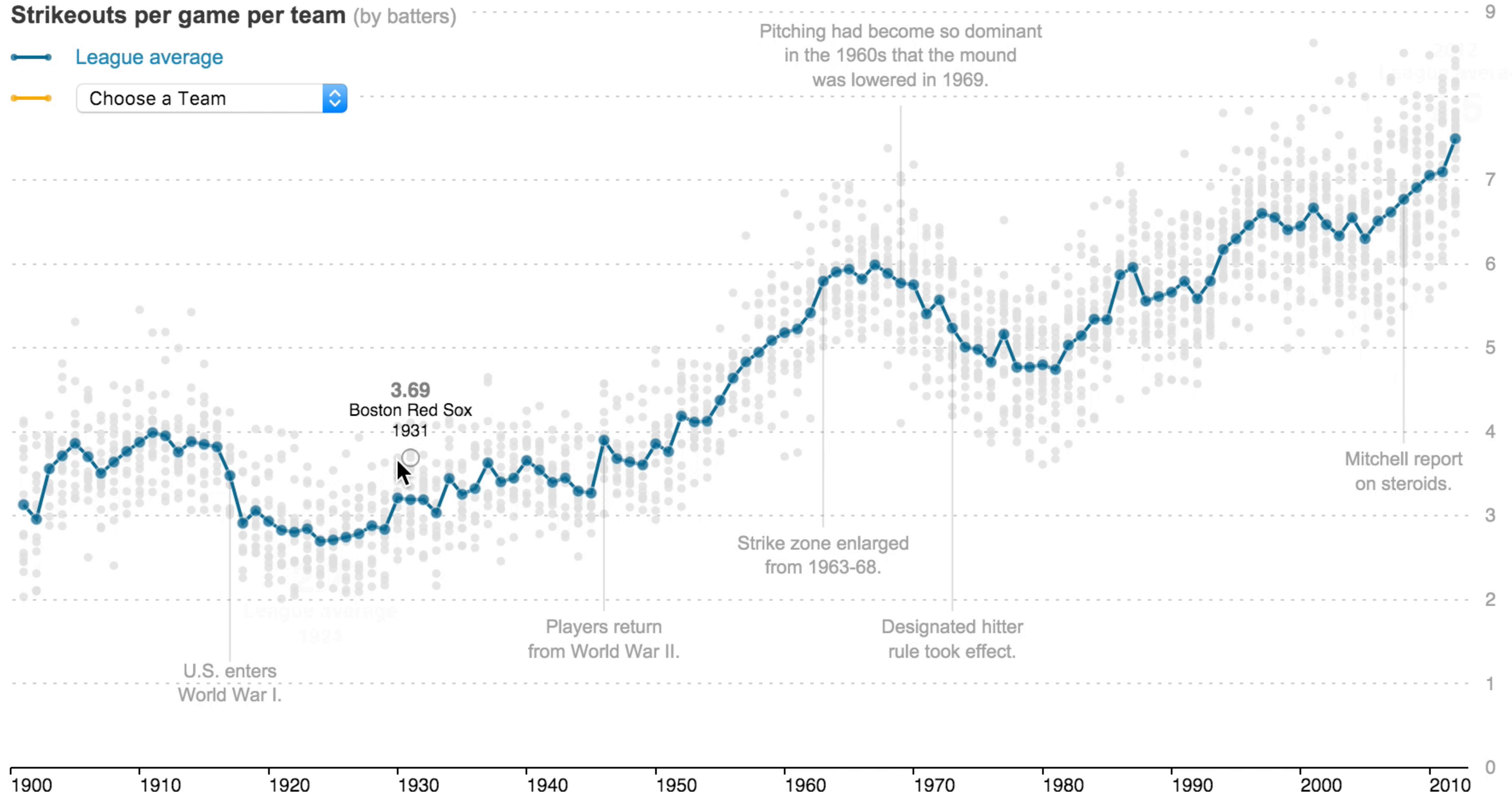
- Record
- Average
- This Year - within avg
- This Year - beyond avg
- Freezing
- Precipitation
- Scattered Clouds
- Cloudy
- Overcast

1-Feb



Strikeouts on the Rise

There were more strikeouts in 2012 than at any other time in major league history.



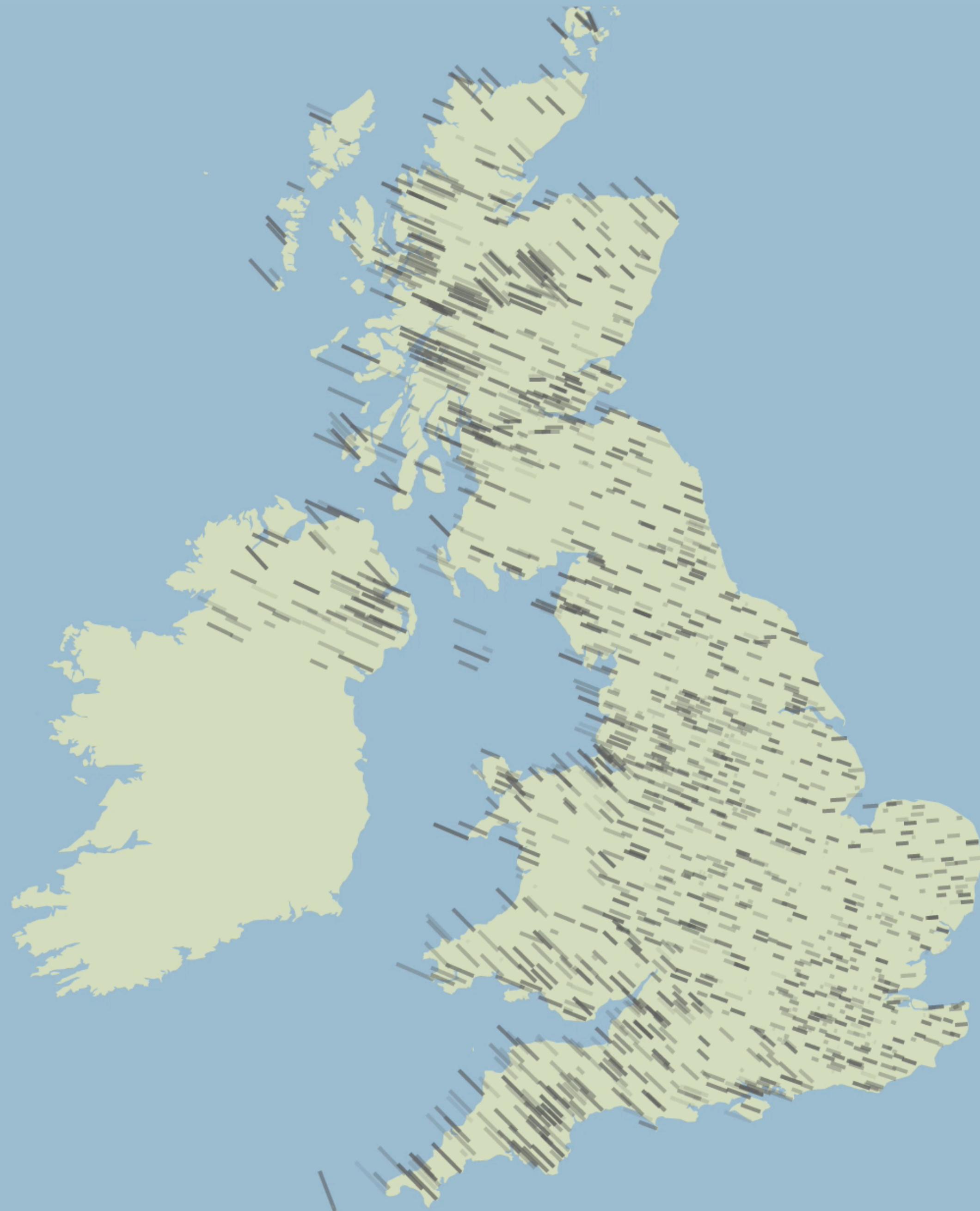
Animated UK Wind Chart

This chart shows wind speed and direction in the UK.
The more the wind blows, the faster the lines go!

The original inspiration came from the beautiful [Wind Map](#) by hint.fm. Weather data is from the Met Office's [DataPoint](#) service and the UK map was taken from Mike Bostock's [mapping tutorial](#).

Although real data, this visualisation is just for demonstration.

Built by [Peter Cook](#).

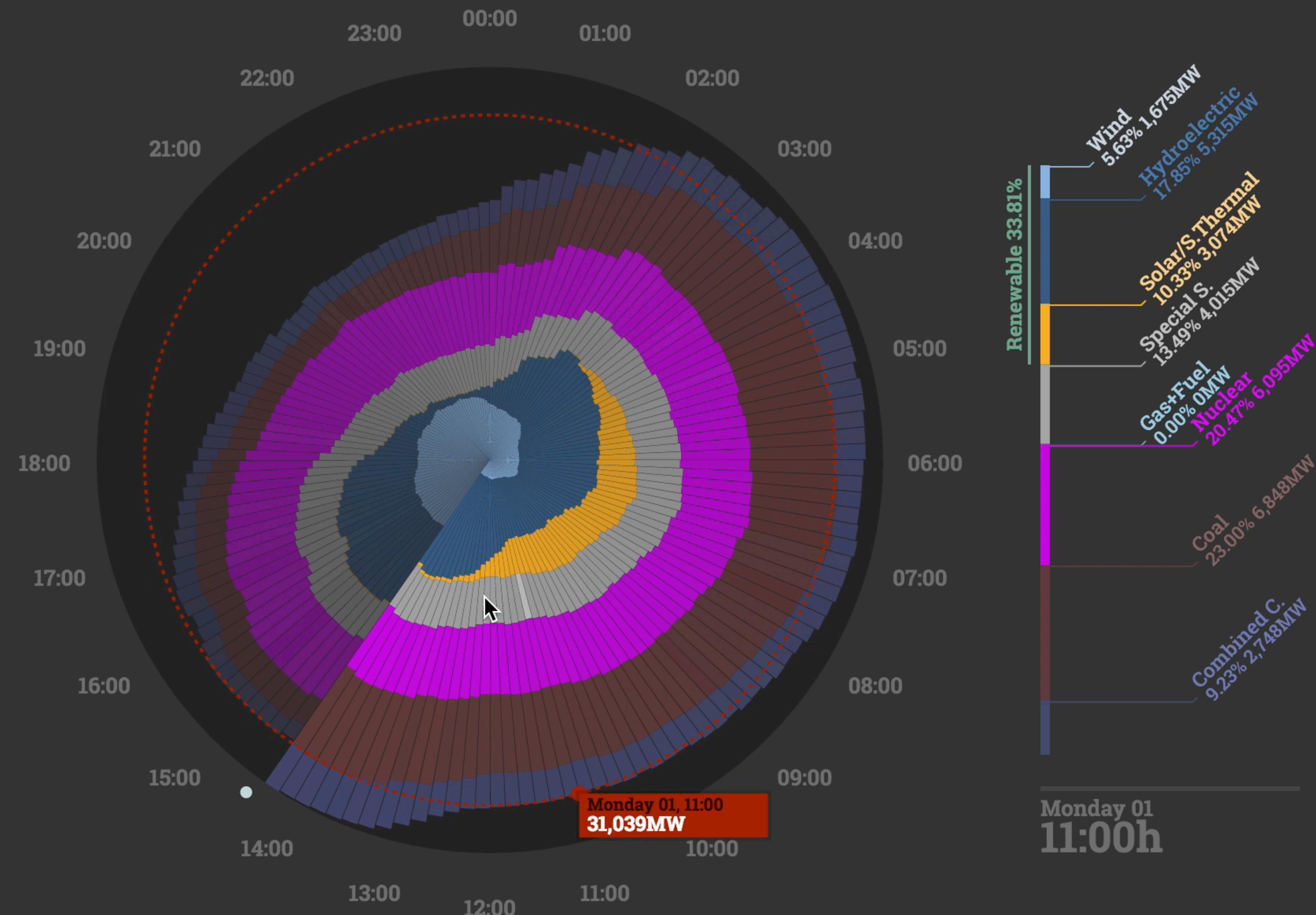


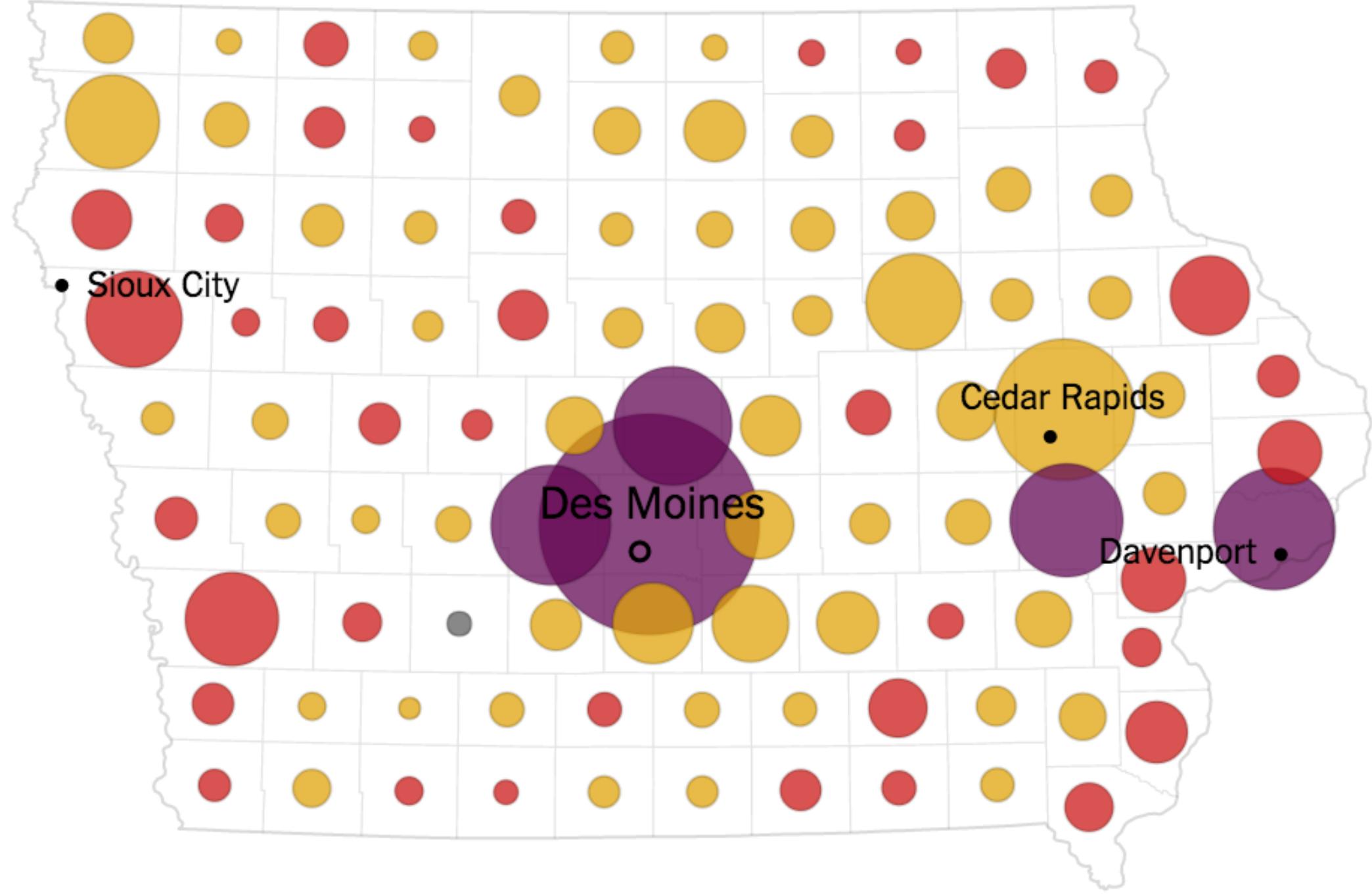


Electric Generation in Spain - Latest 24 hours^{BETA}

Versión en español

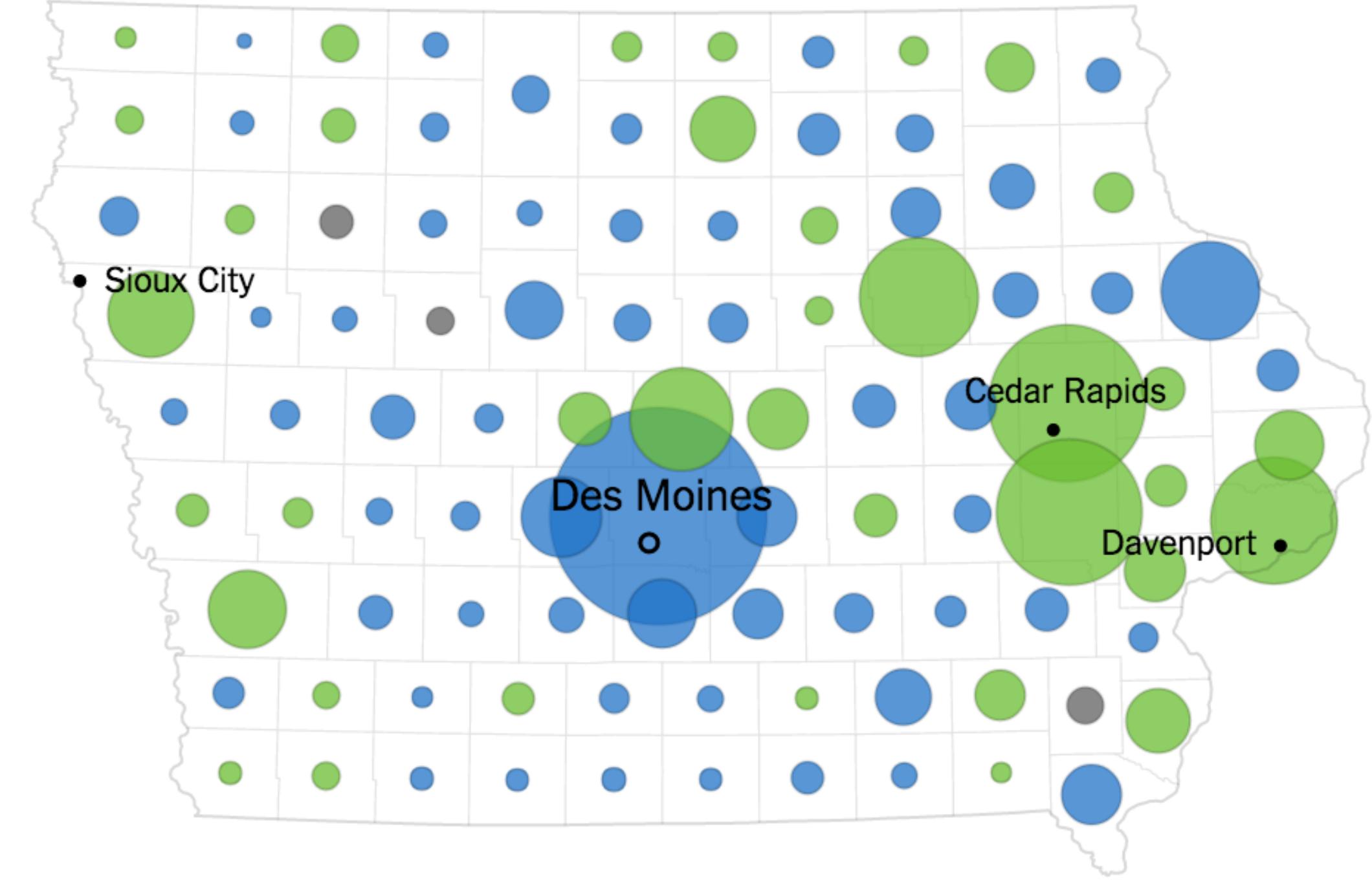
Generation Structure





Circle size is proportional to the leading candidate's vote count.

Results	Vote Leaders	Cruz	Trump	Rubio
---------	--------------	------	-------	-------



Circle size is proportional to the leading candidate's vote count.

Results	Vote Leaders	Clinton	Sanders
---------	--------------	---------	---------

Signal Selector

- none
- Esrrb Fluorescence x3
- Esrrb Fluorescence direction
- Cell Frames
- Relative Age
- Cell Area
- Cell Speed
- Cell Area Difference

zscan

1

Smooth Color Gradient Show

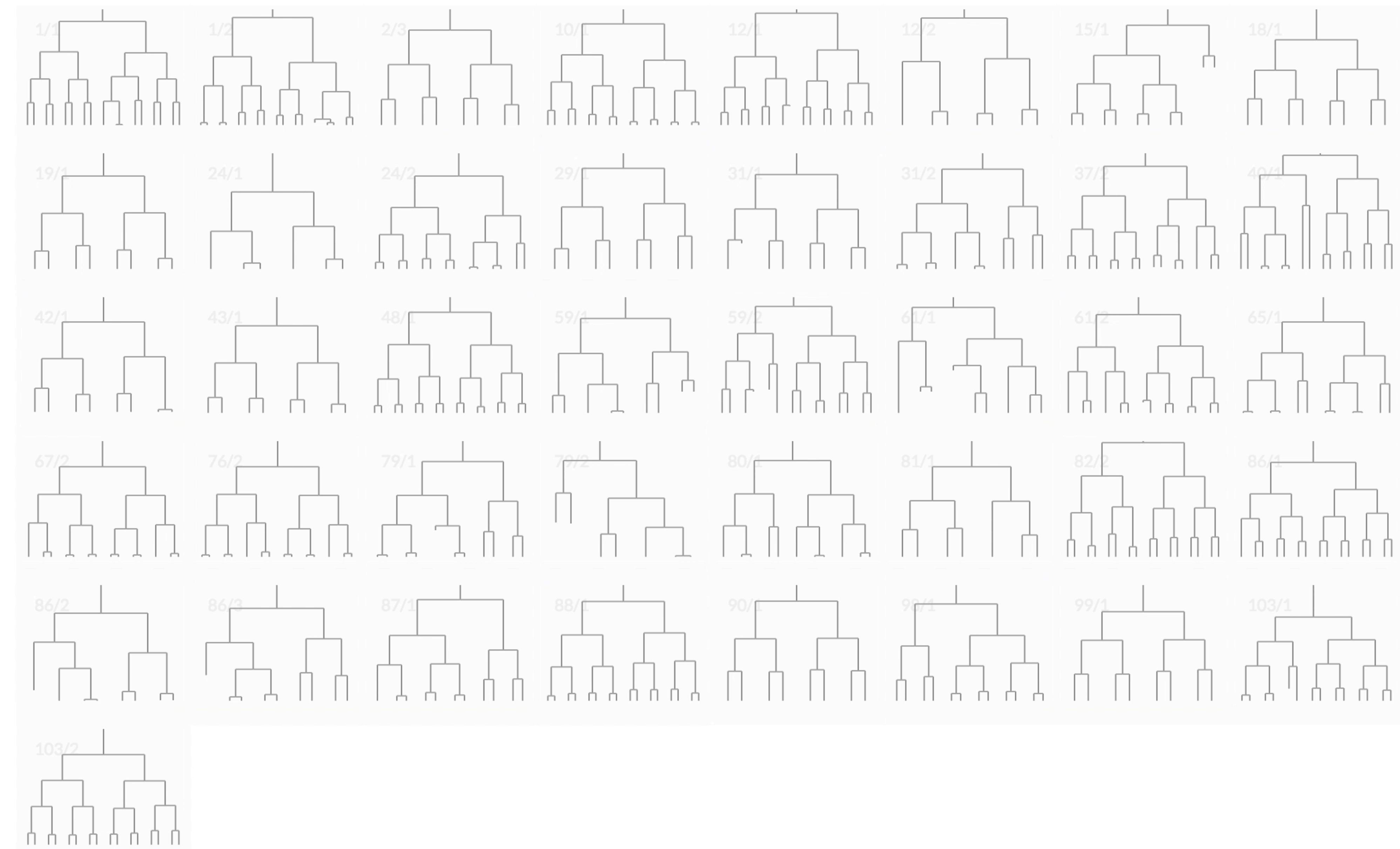
Esrrb

67

Smooth Color Gradient Show

Cell State Creator

Sort Total +



Why Use the Web for Visualizations?

- Expressive and interactive environment
- dynamic and realtime datasets
- Shareable to anyone with a Web browser
- Free, easy-to-learn, and rich resources.

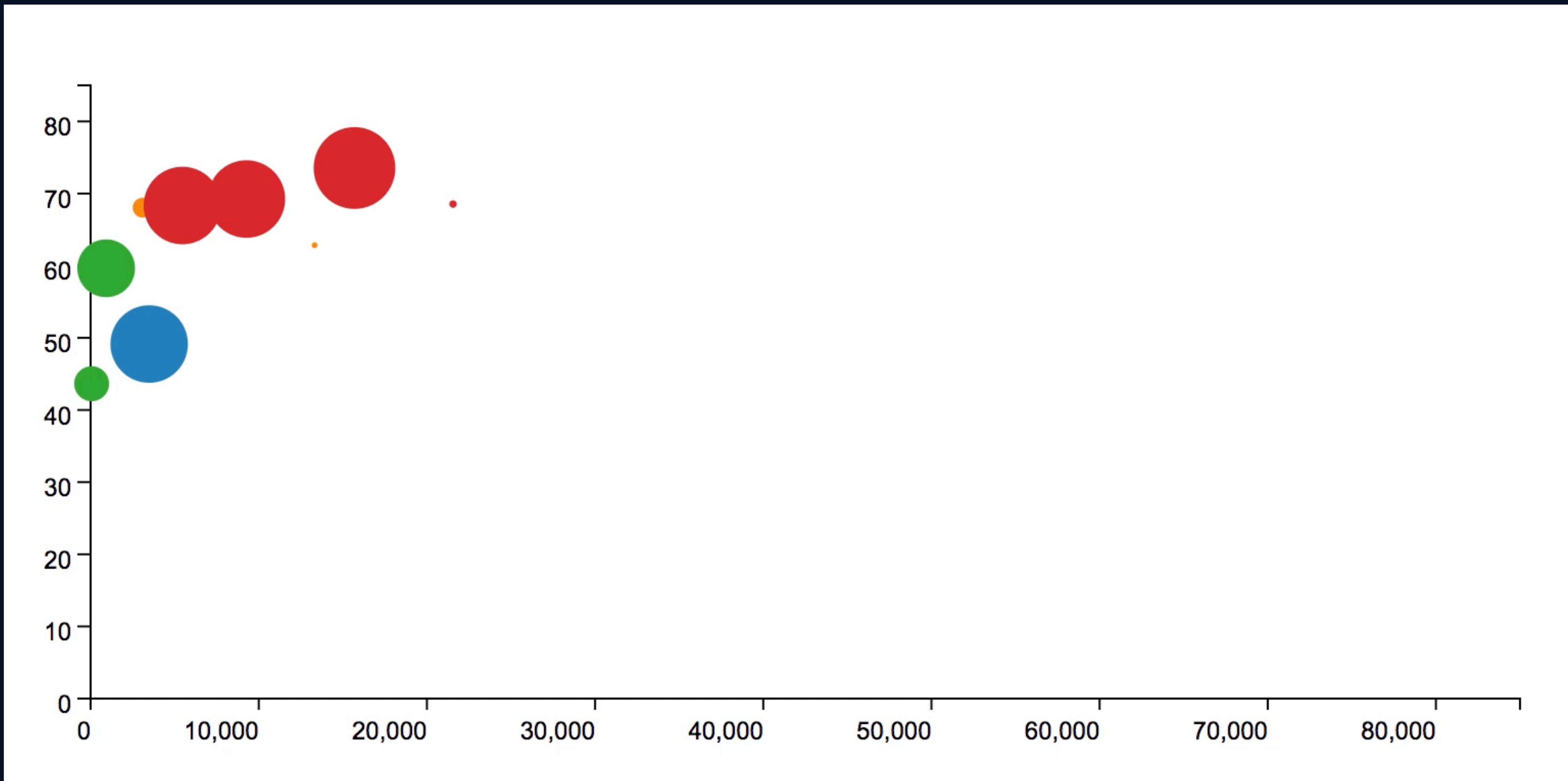
Why not use?

- Customizability requires greater work.
Some prepackaged visualizations, but great results still require lots of work.

But how do I get started?

Goals for this workshop

- Create SVG graphics format objects
- Use D3.js to create shapes
- Use D3 standard functions to manipulate data
- Play with `.data()` function
- Bring it all together in a nice graph



Let's set up our development environment

- Open a text editor of your choice
Don't have one?
Google "Text Wrangler" and download
- Open Chrome. And only Chrome so we're all on the same page.
- Download and open the github repo
<https://github.com/SciutoAlex/viz-workshop>

Anatomy of a Website

- HTML structure
- CSS style
- JAVASCRIPT behavior

```
<html>
  <head>
    <title>Our First D3 Example</title>
    <style>
      .chart rect {
        fill: steelblue;
        stroke: red;
        stroke-width: 10;
      }
    </style>
  </head>
  <body>
    <svg width="800" height="600">
      <g transform="translate(20,30)">
        <rect x="10" y="30" width="100" height="200" />
      </g>
    </svg>
    <script src="//d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <script>
      // Where we will do our coding
    </script>
  </body>
</html>
```

CSS Rules

SVG

JS Code

```
<svg width="800" height="600">
  <g transform="translate(20,30)">
    <rect x="10" y="30" width="100" height="200" />
  </g>
</svg>
```

- DOM Tree: determines structure of the webpage
- Document Object Model (*DOM*): the tree of elements
- Elements, attributes, tags

```
<svg width="800" height="600">  
  <g transform="translate(20,30)">  
    <rect x="10" y="30" width="100" height="200" />  
  </g>  
</svg>
```

Element

```
<svg width="800" height="600">
  <g transform="translate(20,30)">
    <rect x="10" y="30" width="100" height="200" />
  </g>
</svg>
```

SVG Element Attribute

```
<html>
  <head>
    <title>Our First D3 Example</title>
    <style>
      .chart rect {
        fill: steelblue;
        stroke: red;
        stroke-width: 10;
      }
    </style>
  </head>
  <body>
    <svg width="800" height="600">
      <g transform="translate(20,30)">
        <rect x="10" y="30" width="100" height="200" />
      </g>
    </svg>
    <script src="//d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <script>
      // Where we will do our coding
    </script>
  </body>
</html>
```

CSS Rules

SVG

JS Code

- CSS rules: determine how each element in the DOM will appear.

```
<style>  
  .chart rect {  
    fill: steelblue;  
    stroke: red;  
    stroke-width: 10;  
  }  
</style>
```

```
<style>
```

```
.chart rect {  
    fill: steelblue;  
    stroke: red;  
    stroke-width: 10;  
}
```

```
</style>
```

CSS Rule

```
<style>
  .chart rect {
    fill: steelblue;
    stroke: red;
    stroke-width: 10;
  }
</style>
```

CSS Selector

CSS Rules

```
<style>
  .chart rect {
    fill: steelblue;
    stroke: red;
    stroke-width: 10;
  }
</style>
```

- How do we connect the CSS to the SVG?

SVG Elements

```
<svg class="chart" width="800" height="600">
  <g transform="translate(20,30)">
    <rect x="10" y="30" width="100" height="200" />
  </g>
</svg>
```

```
<style>  
  .chart rect {  
    fill: steelblue;  
    stroke: red;  
    stroke-width: 10;  
  }  
</style>
```

```
<svg class="chart" width="800" height="600">  
  <g transform="translate(20,30)">  
    <rect x="10" y="30" width="100" height="200" />  
  </g>  
</svg>
```

- How do we connect the CSS to the SVG?

SVG Element
CSS Rule

```
<div class="red-box"></div>  
.red-box
```

SVG Element
CSS Rule

```
<div class="blue-box" id="logo"></div>  
.blue-box #logo
```

SVG Element
CSS Rule

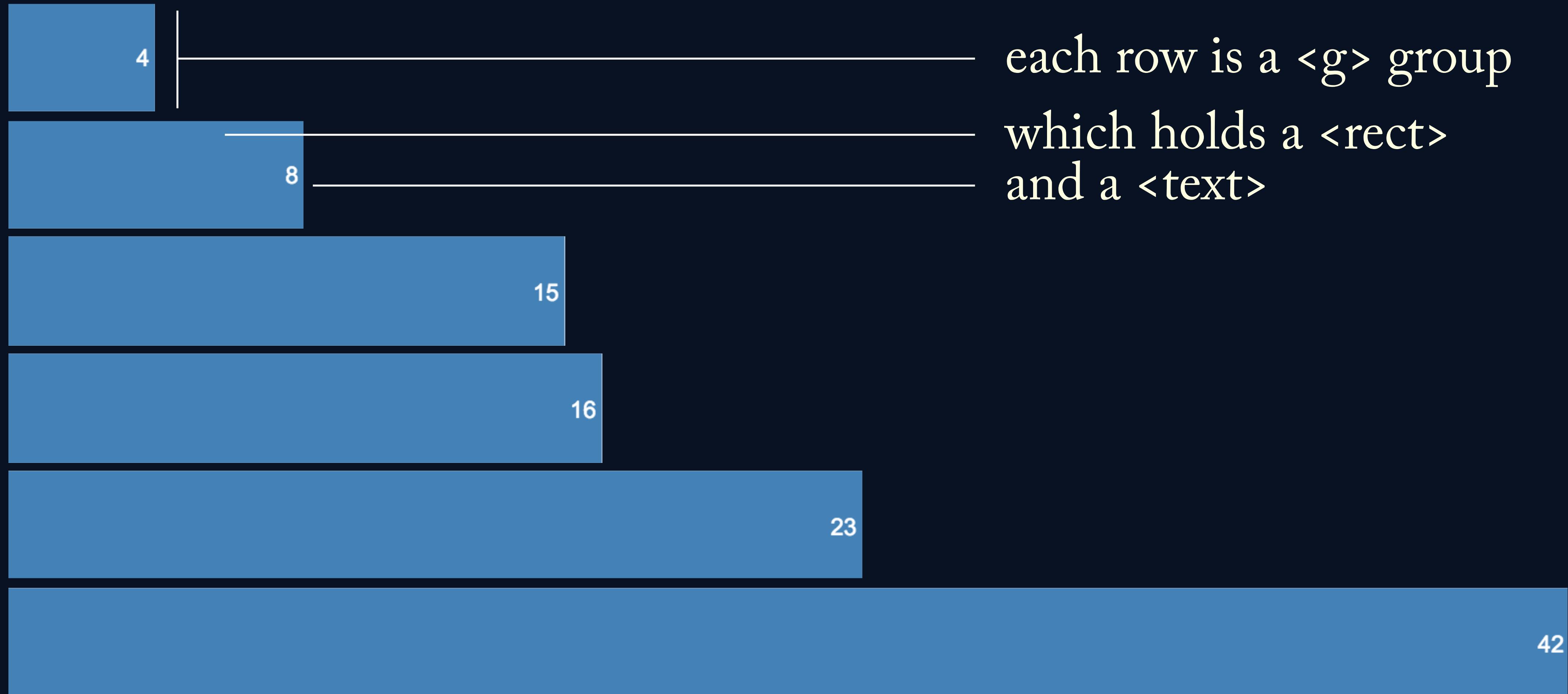
```
<div class="blue-box">  
.blue-box
```

SVG Element

```
<div class="blue-box">  
  <p>here is some text</p>  

```

CSS Rule
.blue-box p



1. Complete the Chart

- Open 01-manually-create...
 - Four bars are completed, add in the final two
 - Change the height dimensions of the bars. Change their color.
 - Become comfortable with the SVG tree structure

<svg>

The outermost parent of all the other <SVG> elements

CSS rule

none

HTML attributes

width = “500”
height = “800”

<ellipse>

draws an ellipse

CSS rule

fill: red
stroke: green
stroke-width: 4
opacity: 0.5

HTML attributes

cx = “30”
cy = “60”
rx = “10”
ry = “30”

<text>

draws a line of text

CSS rule

font-family: “arial”
font-size: “12”
fill: blue

HTML attributes

x = “30”
y = “60”
<text>Hello!</text>

<g>

No visible appearance, but allows SVG elements to be grouped and moved together

CSS rule

none

HTML attributes

transform = “translate(x,y)”

<rect>

draws a rectangle

CSS rule

fill: red
stroke: green
stroke-width: 4
opacity: 0.5

HTML attributes

width = “50”
height = “90”
x = “30”
y = “10”

<polygon>

draws a polygon

CSS rule

fill: red
stroke: green
stroke-width: 4
opacity: 0.5

HTML attributes

points =
“x1,y1 x2,y2, x3,y3 ...”
points = “60,20 100,40”

Key Structure of D3.js

- var d3rect = d3.select(".css-selector-system") returns a D3 element with lots of methods
- d3rect.append('rect') adds a child element to d3rect. <g><rect></rect></g>
- <g>, <svg> can have children. No other element can.
- d3rect.attr('width', '300') produces <g><rect width=300></rect></g>
- d3rect.style('fill', 'red') produces <g><rect width=300 style="fill:red"></rect></g>
- chain-able commands. d3rect.attr(...).style(...)

```
var chart = d3.select('.chart')
  .attr('width', width)
  .attr('height', barHeight * data.length);
```

```
var group = chart.append('g')
  .attr('transform', 'translate(40,50)');
```

```
group.append('rect')
  .attr('width', 500)
  .attr('height', 400)
```

Let's automate creation!

- Open 02-programmatically-create...
- We'll use `d3.select('.chart')` to hold a reference to the `<svg>` element
- We'll use `select.append('rect')` to add a rectangle
- We'll use `rect.style('fill', 'red')` to change CSS Rules
- We'll use `rect.attr('x', 10)` to change SVG attributes.

Let's add in real data

- Open 03-real-data-scales...
- Use `.csv()` and `.tsv()` to access data
- Dot notation to access properties,
for example `person.height`, `person.age`.
- `d3.scale.linear()` to scale the data linearly
- `d3.scale.ordinal()` to scale categorical data

The Data Join

Using d3.data() we can connect data point to data representation and unleash D3's full power.

```
var states = chart.selectAll('g') d3 selection of elements  
    .data(newDataHigh, function(d) { return d.NAME}) d3 data object  
    .enter() d3 selection of potential elements      .enter(),  
    .append('g') d3 selection of elements          .exit()  
    .attr('transform', 'translate(10, 30)'); d3 selection of elements  
  
states.append('rect')  
    .attr('width', function(d,i) {  we pass a function with data, index #  
        return xScale(d.my_value);  and return the value we want  
    });
```

Great things we didn't cover

- D3 interactivity: with SVG and D3, interactivity is trivial to add.
- D3 layouts: higher level modules that can initially layout your data
- D3 SVG shapes: Create line and area graphs.
- D3 geographic visualization.

D3.js abstraction libraries

- NVD3

This project is an attempt to build re-usable charts and chart components for d3.js without taking away the power that d3.js gives you.

- Miso Project

Miso consists of Dataset, a JavaScript client-side data management and transformation library, Storyboard, a state and flow-control management library & d3.chart, a framework for creating reusable charts with d3.js.

- C3.js

Comfort. Customizable. Controllable. D3-based reusable chart library.

Great D3.js Resources

- Scott Murray's "*Interactive Data Visualization for the Web*"
<http://alignedleft.com/work/d3-book>
- Jim Vallandingham's D3 tutorials
<http://vallandingham.me/>
- D3.js Documentation and Tutorials
<https://github.com/mbostock/d3/wiki/Tutorials>
- Stackoverflow D3.js forum
<http://stackoverflow.com/questions/tagged/d3.js>
- Introduction to d3.js
<http://mtaptich.github.io/d3-lessons/>

More infovis resources

- Flowing Data Website
<http://flowingdata.com/>
- Awesome Dataset Repository
<https://github.com/caesar0301/awesome-public-datasets>
- Infosthetics Blog
<http://infosthetics.com/>
- Information Is Beautiful
<http://www.informationisbeautiful.net/>

Twitter

News Vis

@GuardianData
@538viz
@BBGVisualData
@UpshotNYT
@nytgraphics

DataVis Studios

@Periscopic
@nyp1_labs
@stamen
@TwitterData

Practitioners

@mericson
@mbostock
@jashkenas
@amandacox
@hfairfield
@jasondavies
@visualisingdata
@feltron
@blprnt
@aaronkoblin
@hmason
@moritz_stefaner
@kleinmatic
@alignedleft
@vlandham
@jcukier
@JanWillemTulp
@arnicas
@rachelbinx
@aparrish
@lifewinning
@infobeautiful
@jkeefe
@randal_olson

Thank you!
@sciutoalex