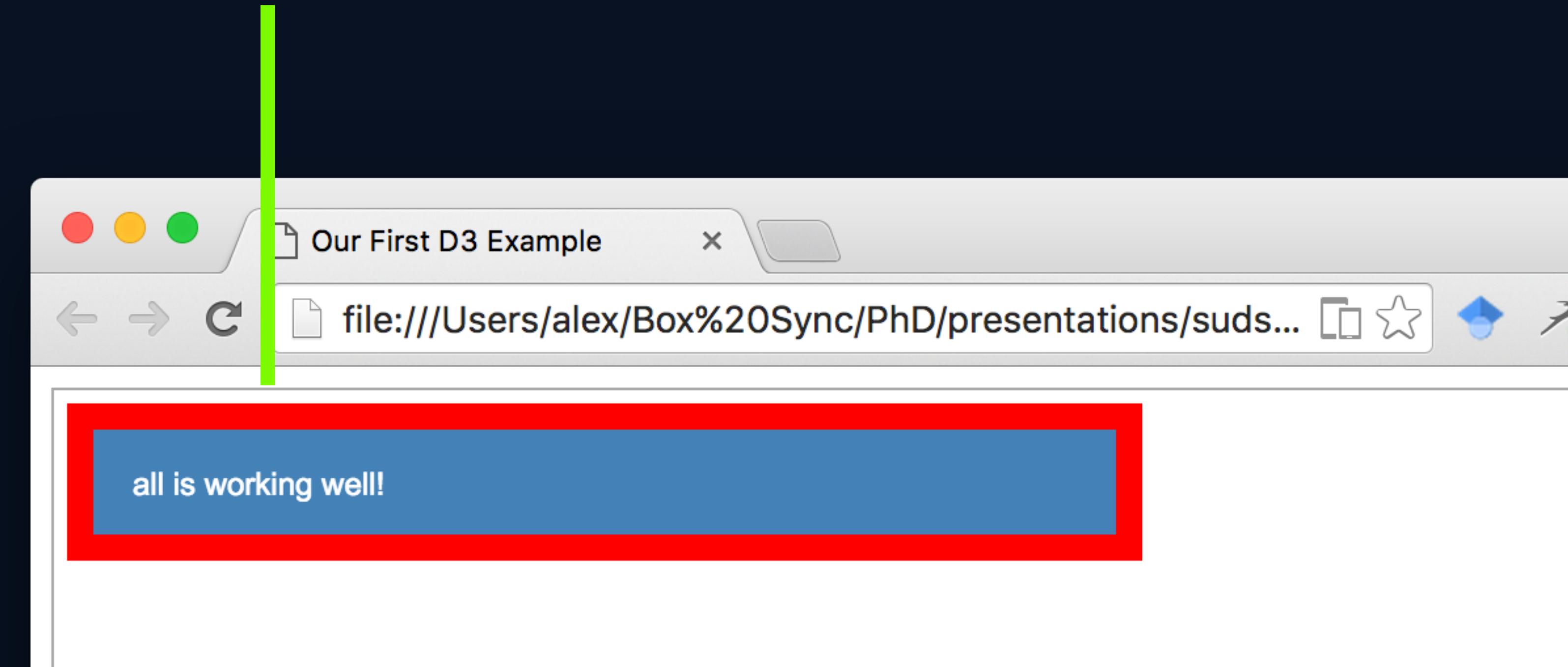


1. download  
<https://github.com/SciutoAlex/viz-workshop>

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

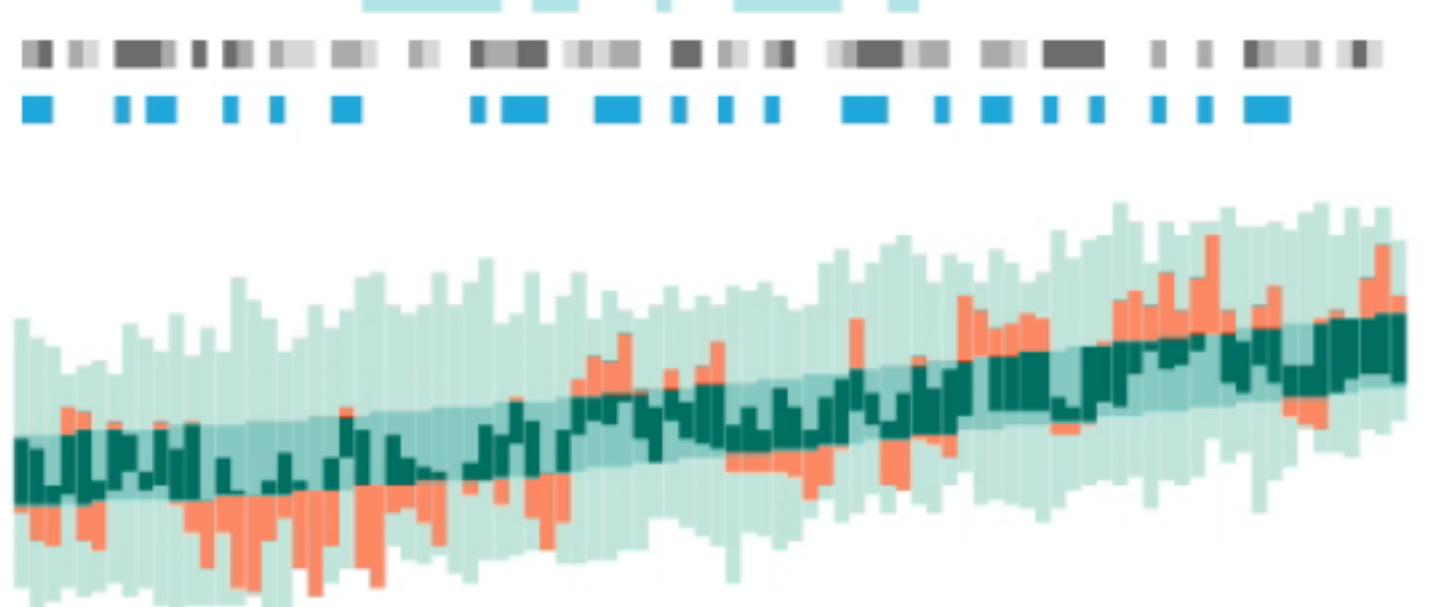
cmu human-computer interaction institute

# New York

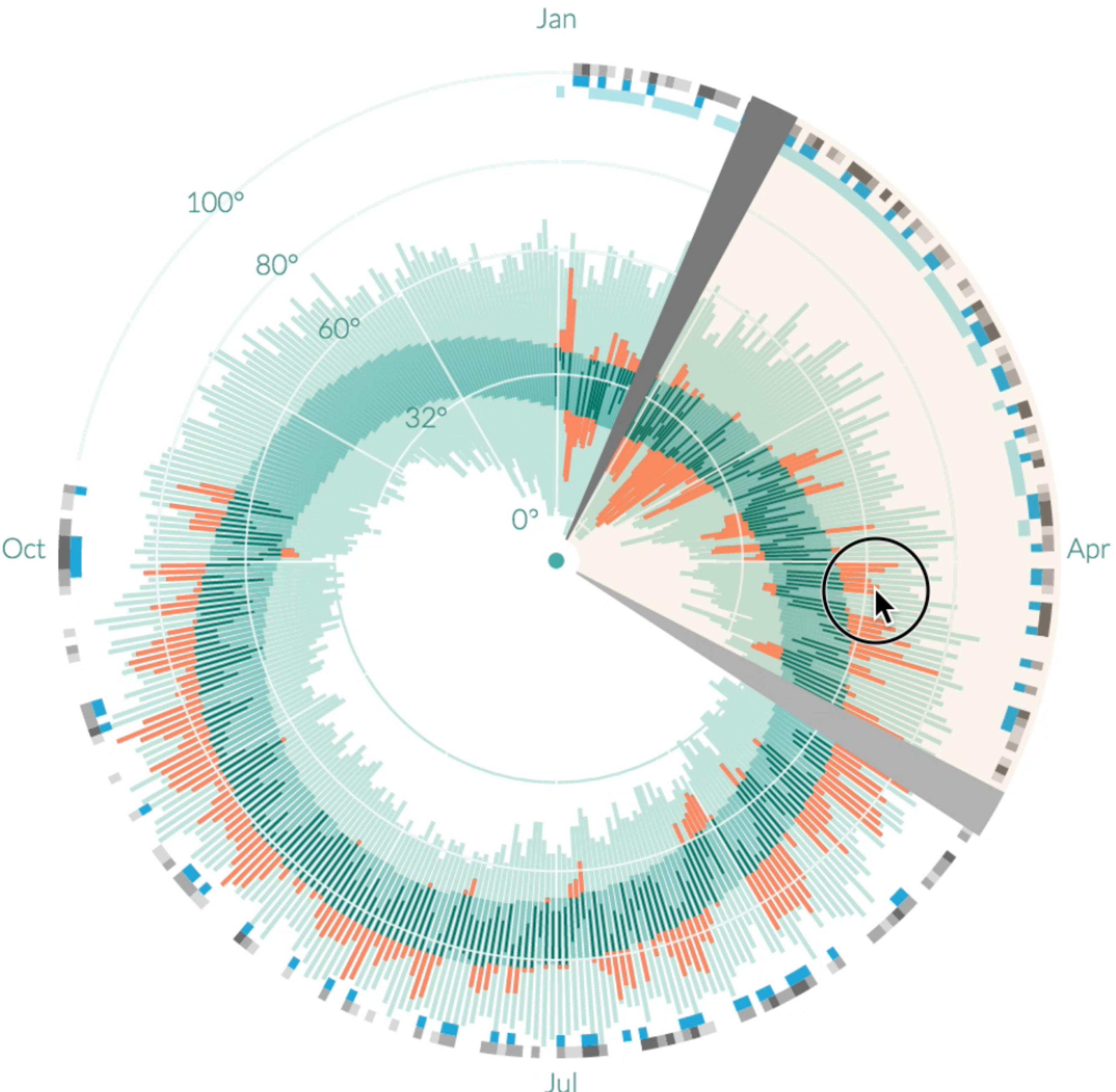
## Historical Weather Data

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

1-Feb



30-Apr



# Strikeouts on the Rise

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

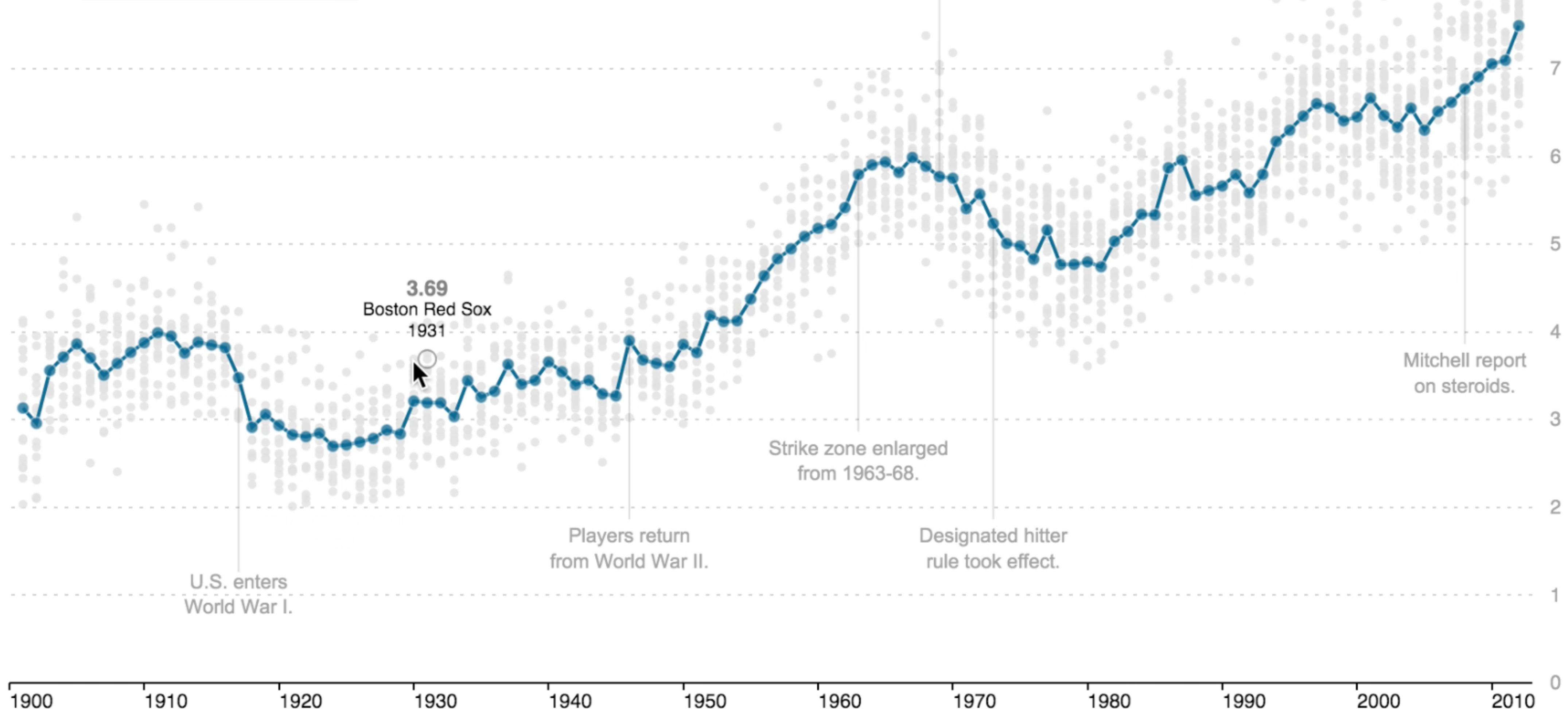
## Strikeouts per game per team (by batters)

League average

Choose a Team



Pitching had become so dominant  
in the 1960s that the mound  
was lowered in 1969.



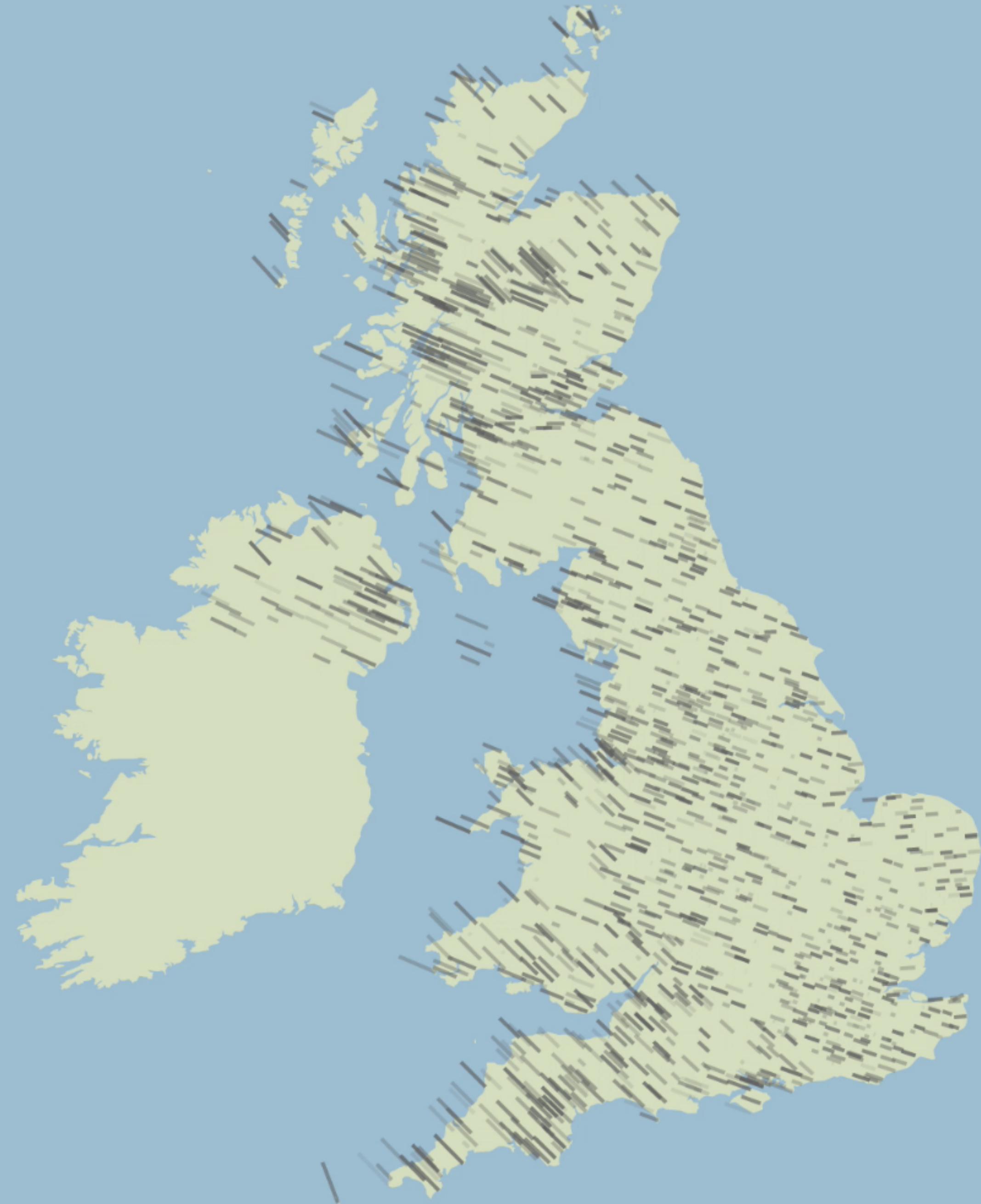
# Animated UK Wind Chart

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](#).

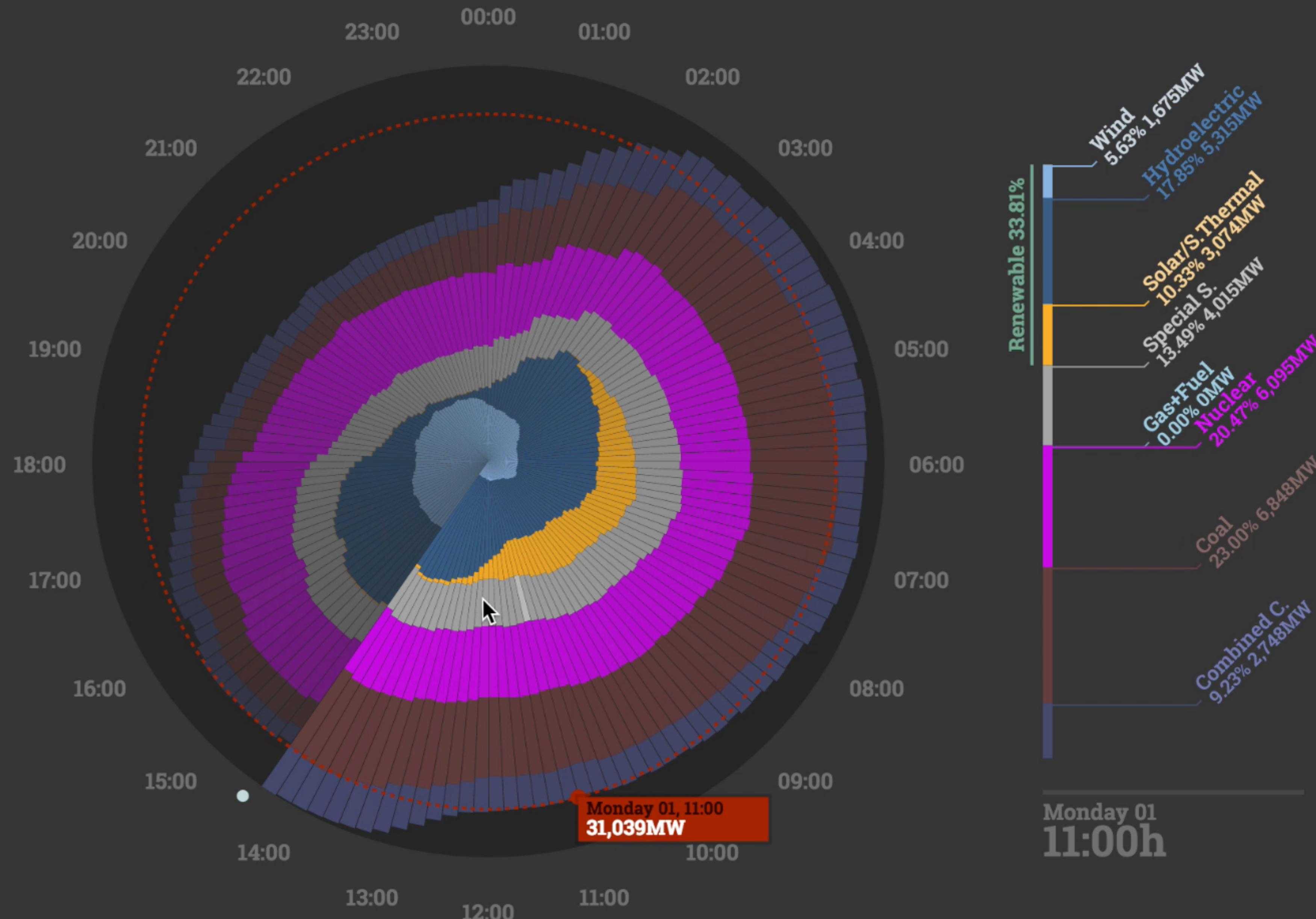


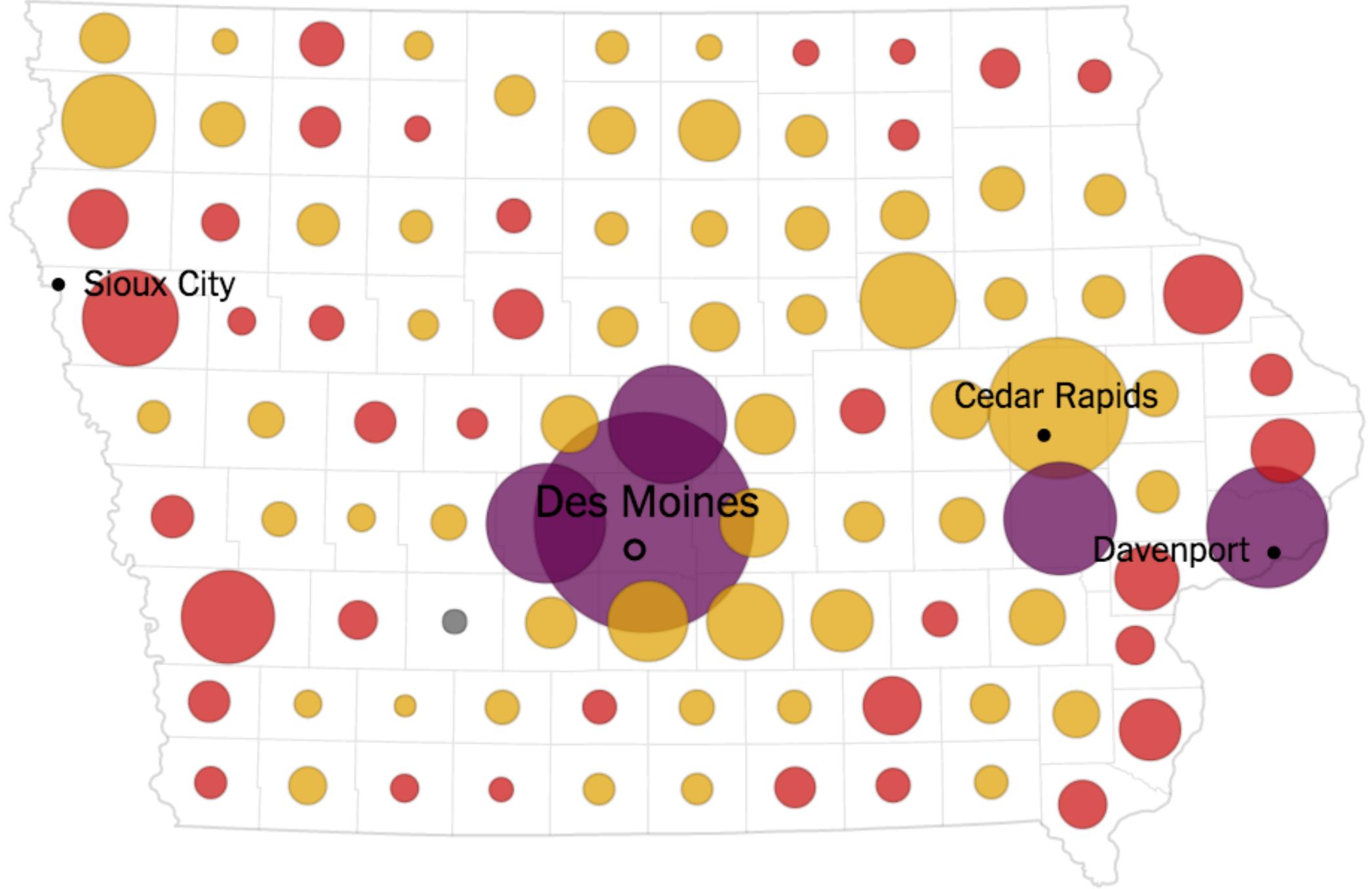


# Electric Generation in Spain - Latest 24 hours<sup>BETA</sup>

Versión en español

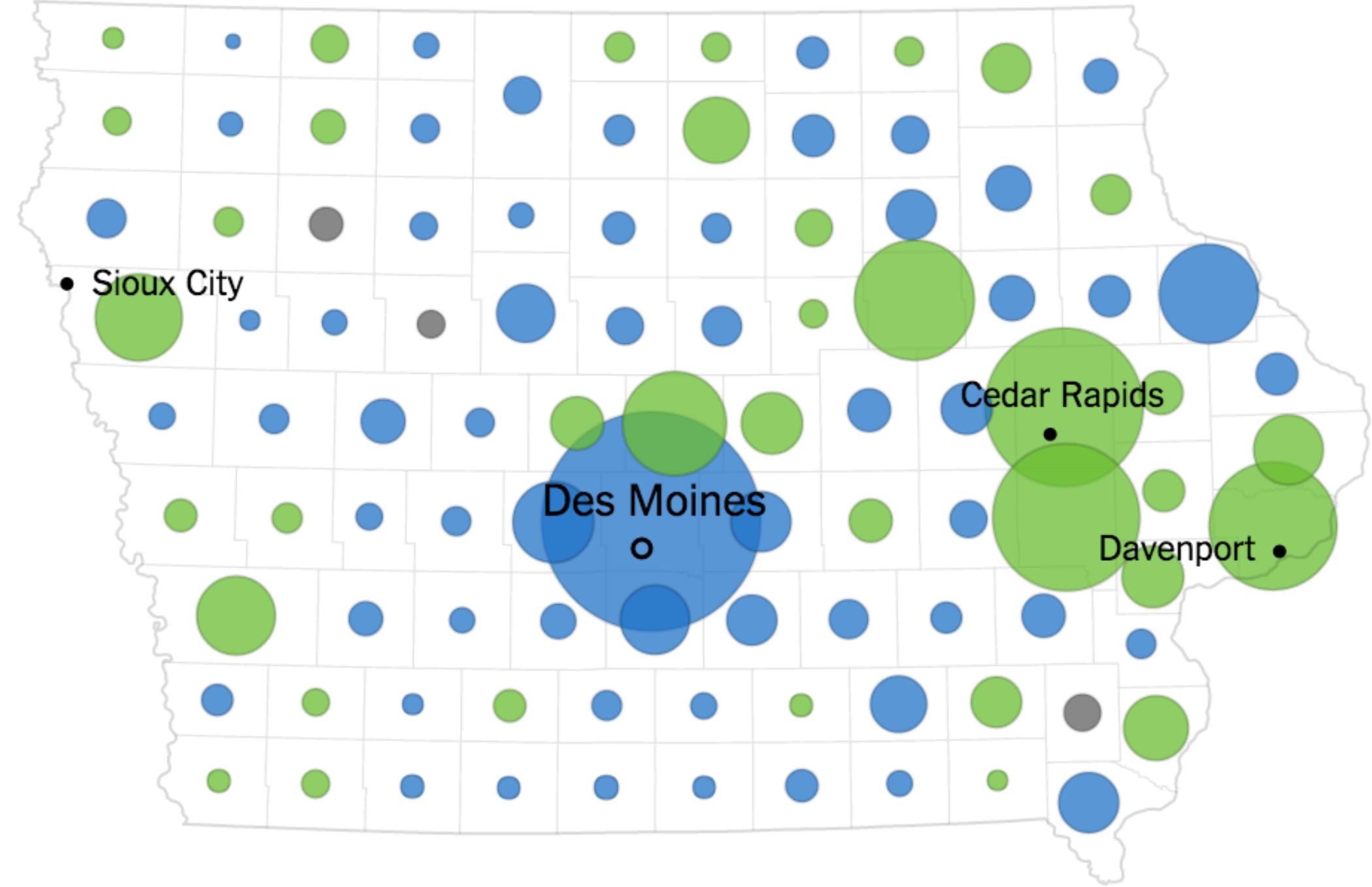
## Generation Structure





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
---------	--------------	---------	---------

# CELL LINES

31 positions / 41 trees / 449 boundary cells / 903 total cells / 929 excluded ce  
Zoom, Hide Details, Hide Overlay, Toggle Fi:

Signal Selector

Esrrb Florescence

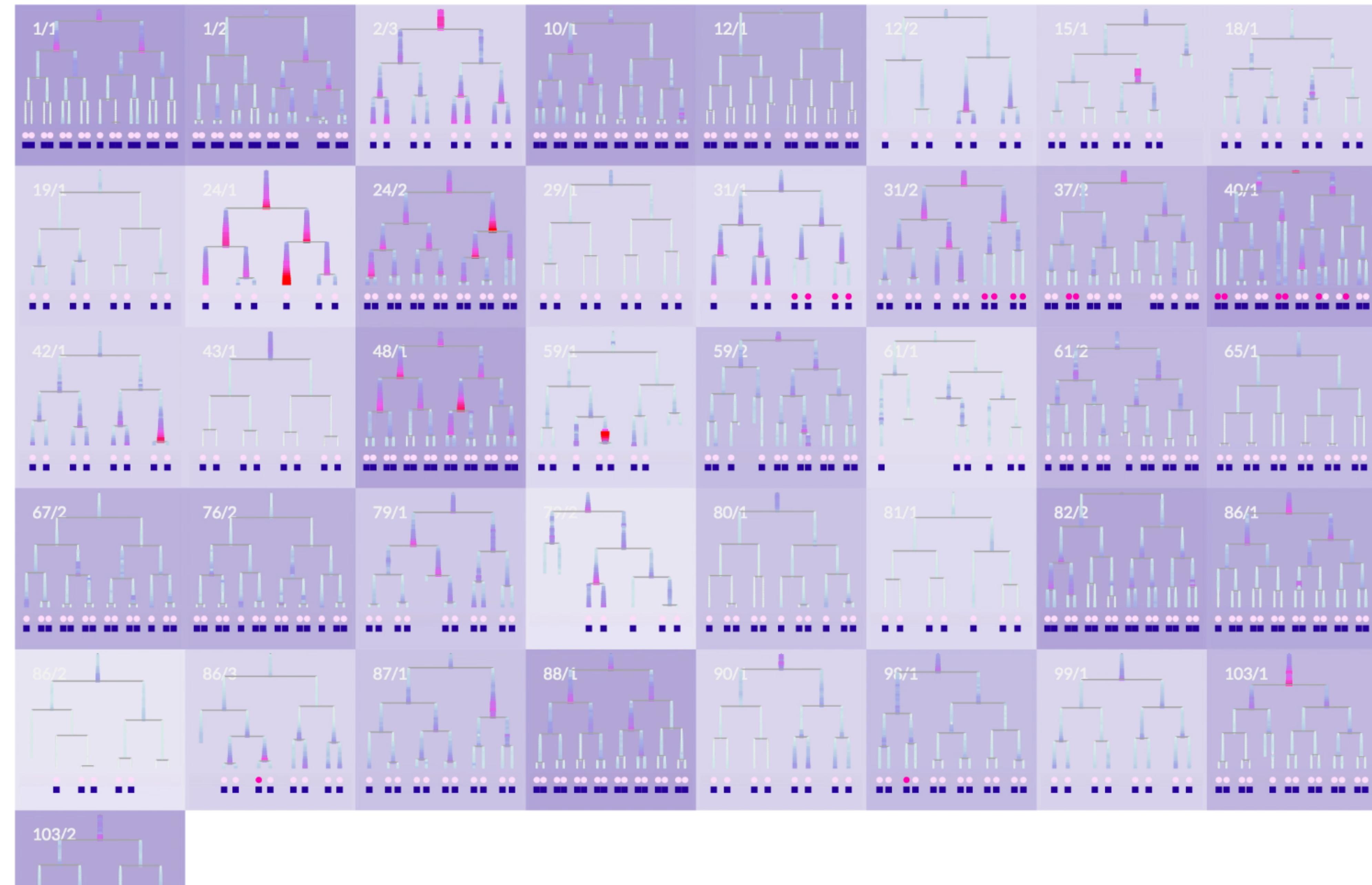
Expression Editor

**tbx3**  Smooth Color Gradient  Show

**zscan**  Smooth Color Gradient  Show

**Esrrb**  Smooth Color Gradient  Show

Cell State Creator    
 order  attribute



# 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

- This is an introductory workshop and is intended for everyone. Participants will actually work with these tools and will walk away from the workshop being able to create a simple graph in Javascript. These tools are among the most widely used for data visualizations and are in high demand. Come and check it out!
- Specific Goal: create an interactive bar chart using D3.js's data-join concept.

# 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>
```

- SVG 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

## <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>

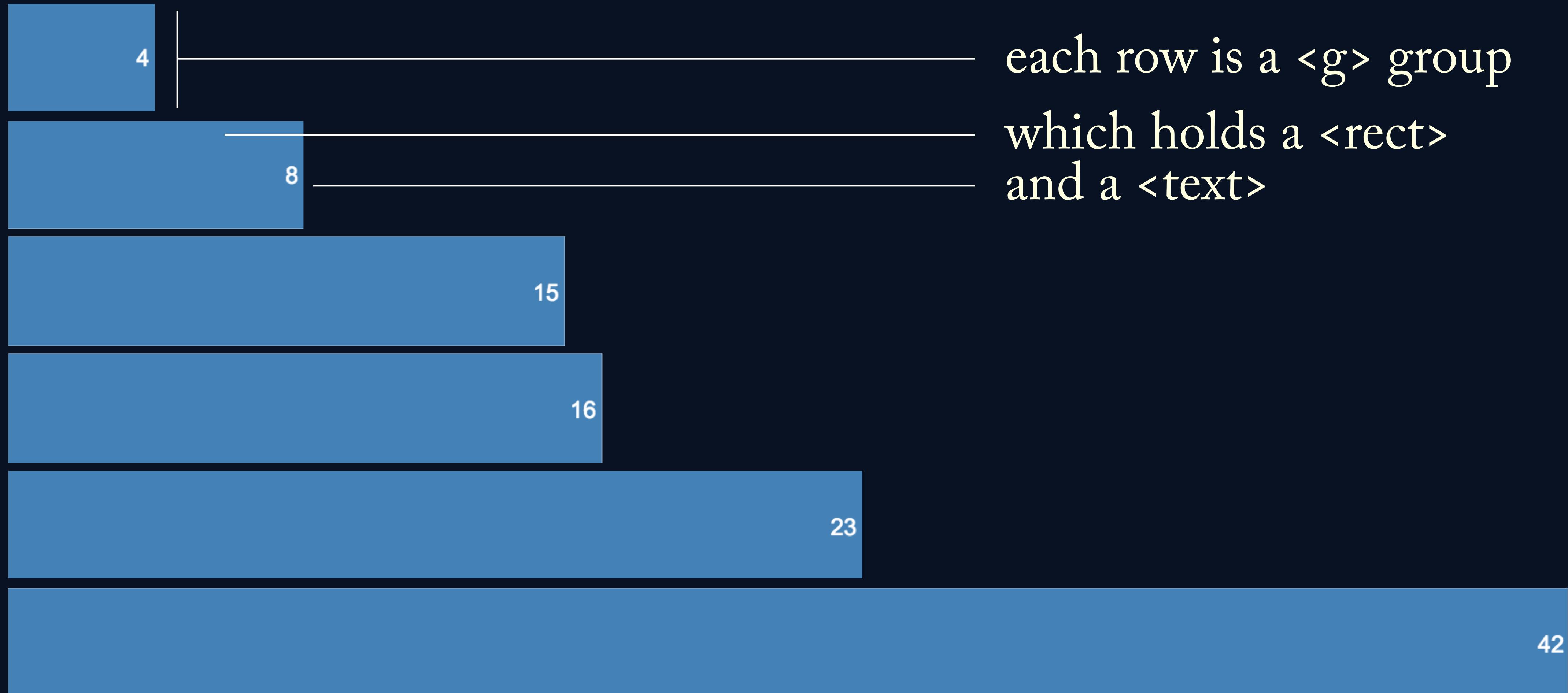
*draws a polygon*

### 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”



# Let's automate creation!

- Let's loop through a list [4,8,15,...]
- We'll use `d3.select('.chart')` to hold a reference to the `<svg>` element
- We'll use `d3.append('rect')` to add a rectangle
- We'll use `d3.style('fill', 'red')` to change CSS Rules
- We'll use `d3.attr('x', 10)` to change SVG attributes.

---

## <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>

### d3 .text() functions

.text('here is some text')

.attr('x', 34)

# Let's add in real data

- 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

```
parentGroup.append( 'rect' )
    .attr( 'width', 0 )
    .transition()
        .duration(1000)
        .delay(i * 100)
    .attr( 'width', 100 )
```

# The Data Join

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

# 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

j.mp/sudsworkshop