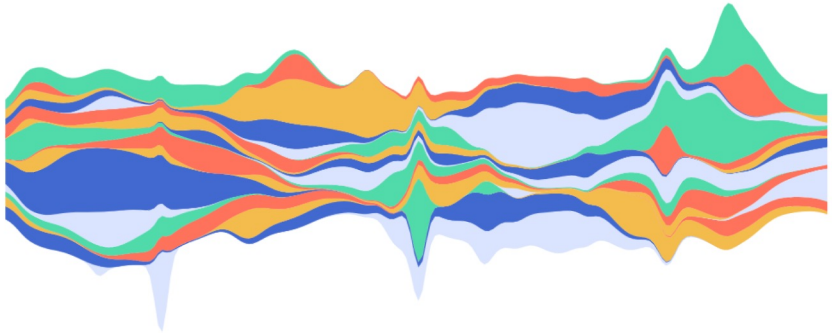
by Observable

Introduction to D3

Session 5

Observable notebook

YouTube video



These slides are provided with the course “Introduction to D3” by Paul Buffa – Head of Product Education at Observable

Agenda

1. Data structures and visualization structures
2. Interaction and event listeners
3. Thinking in D3
4. How to keep learning
5. Course assignment & certificate

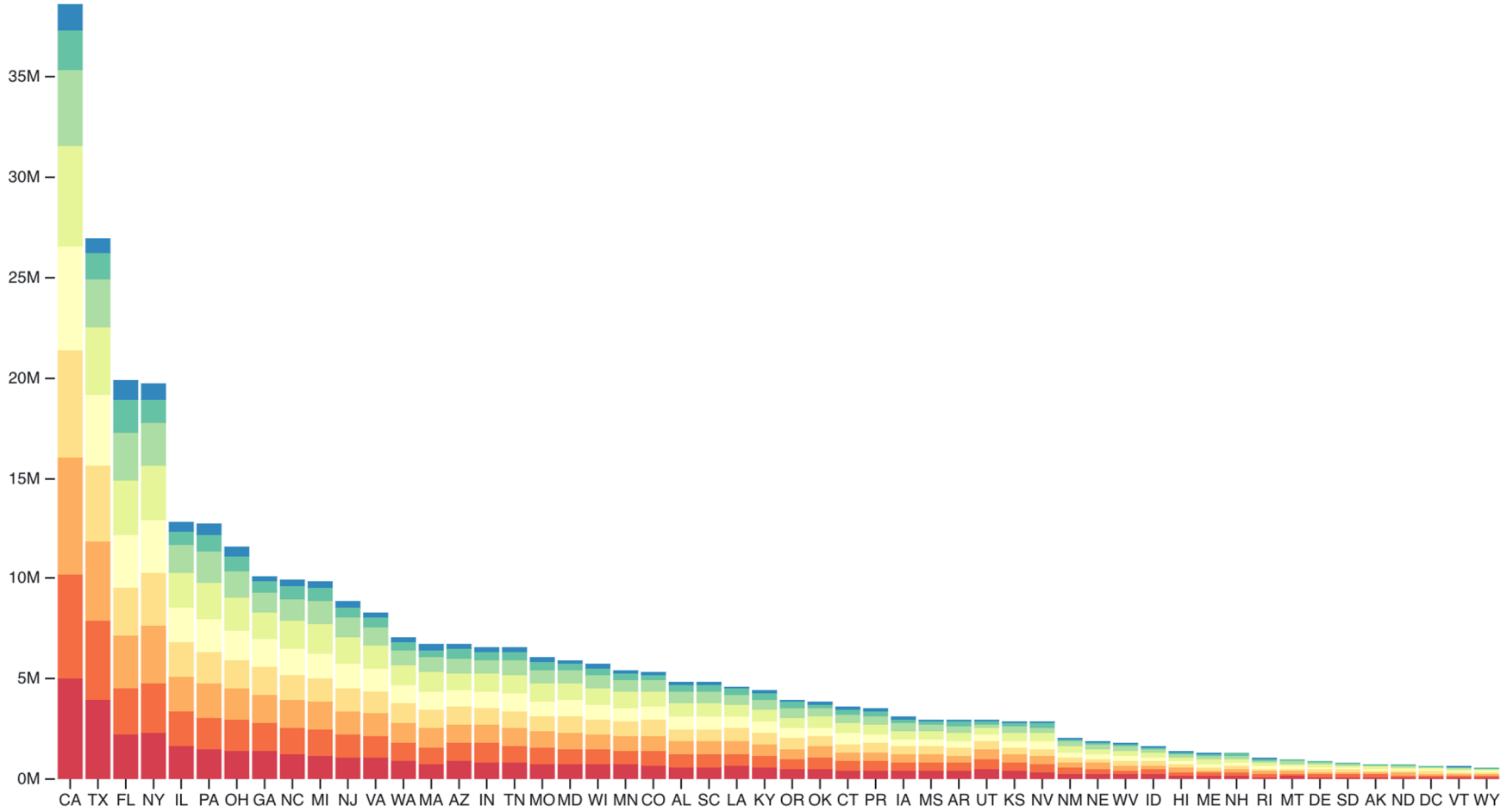
Data Structures

Data Structures

So far in this course, we've worked with tabular data, i.e. data from a spreadsheet. This sort of dataset is common, but for more advanced visualization forms, we often need more complex data structures. One such structure is the hierarchical data structure.

Data Structures

To demonstrate how we work with hierarchical data, we're going to revisit the stack layout from session 3. Let's walk through how they work and think about them from the perspective of a nested structure.


35M
30M
25M
20M
15M
10M
5M
0M
CA TX FL NY IL PA OH GA NC MI NJ VA WA MA AZ IN TN MO MD WI MN CO AL SC LA KY OR OK CT PR IA MS AR UT KS NV NM NE WV ID HI ME NH RI MT DE SD AK ND DC VT WY

Stack Layouts

Given the complexity making a bar chart, to make a stacked bar chart using this method would be very difficult. Fortunately there is d3.stack() for the win!

Stack Layouts

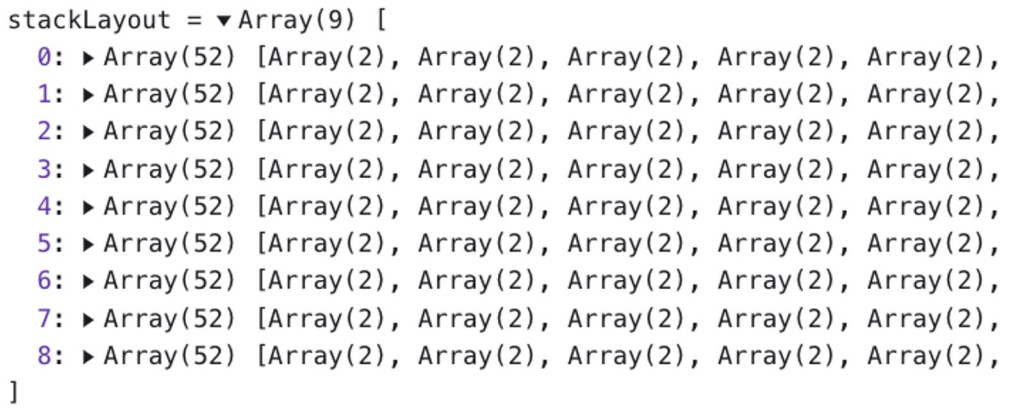
We need to tell the d3.stack() function what we're stacking. These are called the keys.

```
stackLayout = ƒ()
```

```
stackLayout = d3.stack()
  .keys()
```

Stack Layouts

When we enter a dataset into our stack function, we are returned an array of arrays, i.e. hierarchical data.



Stack Layouts

Given we're stacking values, we'll need to figure out how to find the max y-scale value correctly.

```
d3.scaleLinear()
  .domain([0, d3.max(stackLayout, d => d3.max(d, d => d[1]))])
  .range([100, 0])
```

Stack Layouts

And since our stacked data is hierarchical, that means we'll need our elements to be hierarchical.

```
const bars = g.selectAll(".group")
  .data(stackLayout)
  .join("g")
  .attr("class", "group")
  .selectAll("rect")
  .data(d => d)
  .join("rect")
```

Activity 1

Event Listeners

Event Listeners

For interaction, selections allow for listening and dispatching of events. We did a simple example of this yesterday when we had a click event listener on our buttons.

Event Listeners

Listening means we're waiting for an event to trigger on an element, and we're going to do something in response. We can listen for things like click events, hover events, keyboard events, etc.

Event Listeners

Here is an example of a rectangle with a click event listener. We use the selection.on method.

```
{
  const width = 200;
  const height = 200;
  const margin = {top: 20, bottom: 20, left: 20, right: 20};

  const svg = d3.create("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom);

  const g = svg.append("g")
    .attr("transform", `translate(${margin.left}, ${margin.top})`);

  const rectangle = g.append("rect")
    .attr("width", width)
    .attr("height", height)
    .style("fill", "#ddd")
    .style("stroke", "#000")
    .on("click", (event) => {
      rectangle.style("fill", "red")
    })

  return svg.node()


}
```

Event Listeners

We can change the event listener to hover for a hover event to trigger this action.

```
{
  const width = 200;
  const height = 200;
  const margin = {top: 20, bottom: 20, left: 20, right: 20};

  const svg = d3.create("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom);

  const g = svg.append("g")
    .attr("transform", `translate(${margin.left}, ${margin.top})`);

  const rectangle = g.append("rect")
    .attr("width", width)
    .attr("height", height)
    .style("fill", "#ddd")
    .style("stroke", "#000")
    .on("click", (event) => {
      rectangle.style("fill", "red")
    })

  return svg.node()


}
```

Event Listeners

Note that we also have the event data passed to us as the first parameter in the callback function.

```
{
  const width = 200;
  const height = 200;
  const margin = {top: 20, bottom: 20, left: 20, right: 20};

  const svg = d3.create("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom);

  const g = svg.append("g")
    .attr("transform", `translate(${margin.left}, ${margin.top})`);

  const rectangle = g.append("rect")
    .attr("width", width)
    .attr("height", height)
    .style("fill", "#ddd")
    .style("stroke", "#000")
    .on("click", (event) => {
      rectangle.style("fill", "red")
    })

  return svg.node()


}
```

Activity 2 & 3

Thinking in D3

Thinking in D3

When we work with D3, we have to build (almost) all the various visualization components ourselves. This forces us to think more deeply about the visualizations we make and what it entails.

Thinking in D3

- Thinking in D3 means:
  - Structure of our visualization
  - Structure of our data
  - Scales, layouts
  - Binding our data
  - Styling from data

What's Your Pay Gap?

Women earn less than men in 439 of 446 major U.S. occupations, a Wall Street Journal examination of the gender pay gap found.

They earn more than men in seven occupations

Women working as

physicians and surgeons

earn **64%** of their male counterparts on average.

Enter your profession here

Median earnings by occupation

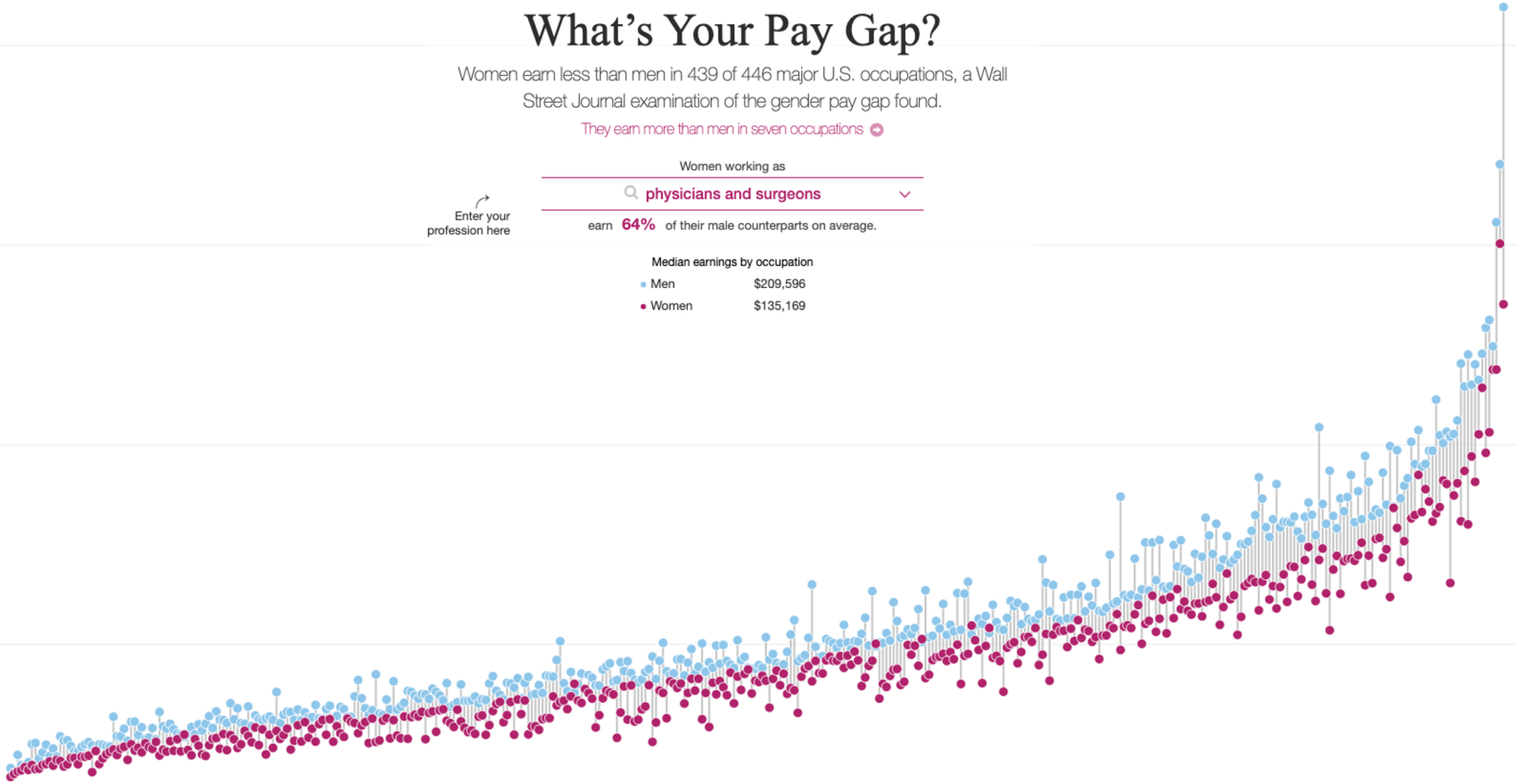
Men	$209,596
Women	$135,169

$200,000

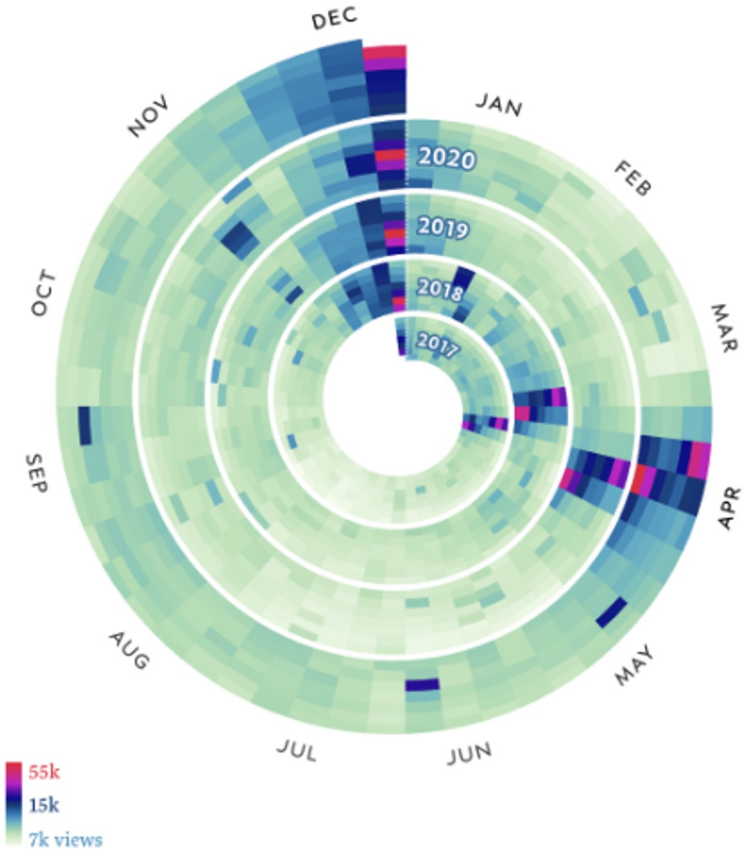
$150,000

$100,000

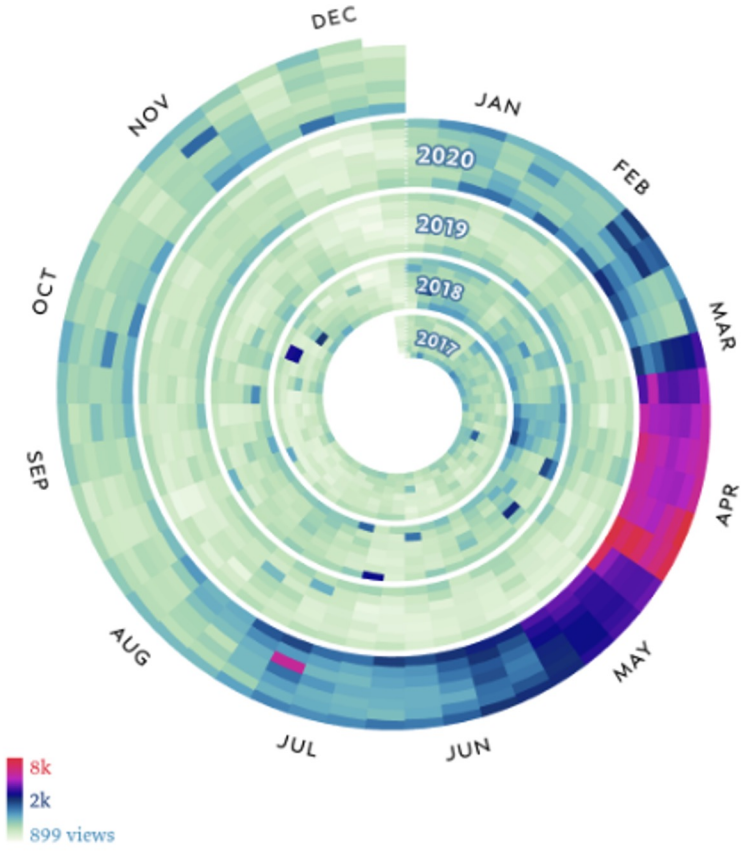
$50,000



Related: Women in Elite Jobs Face Stubborn Pay Gap »

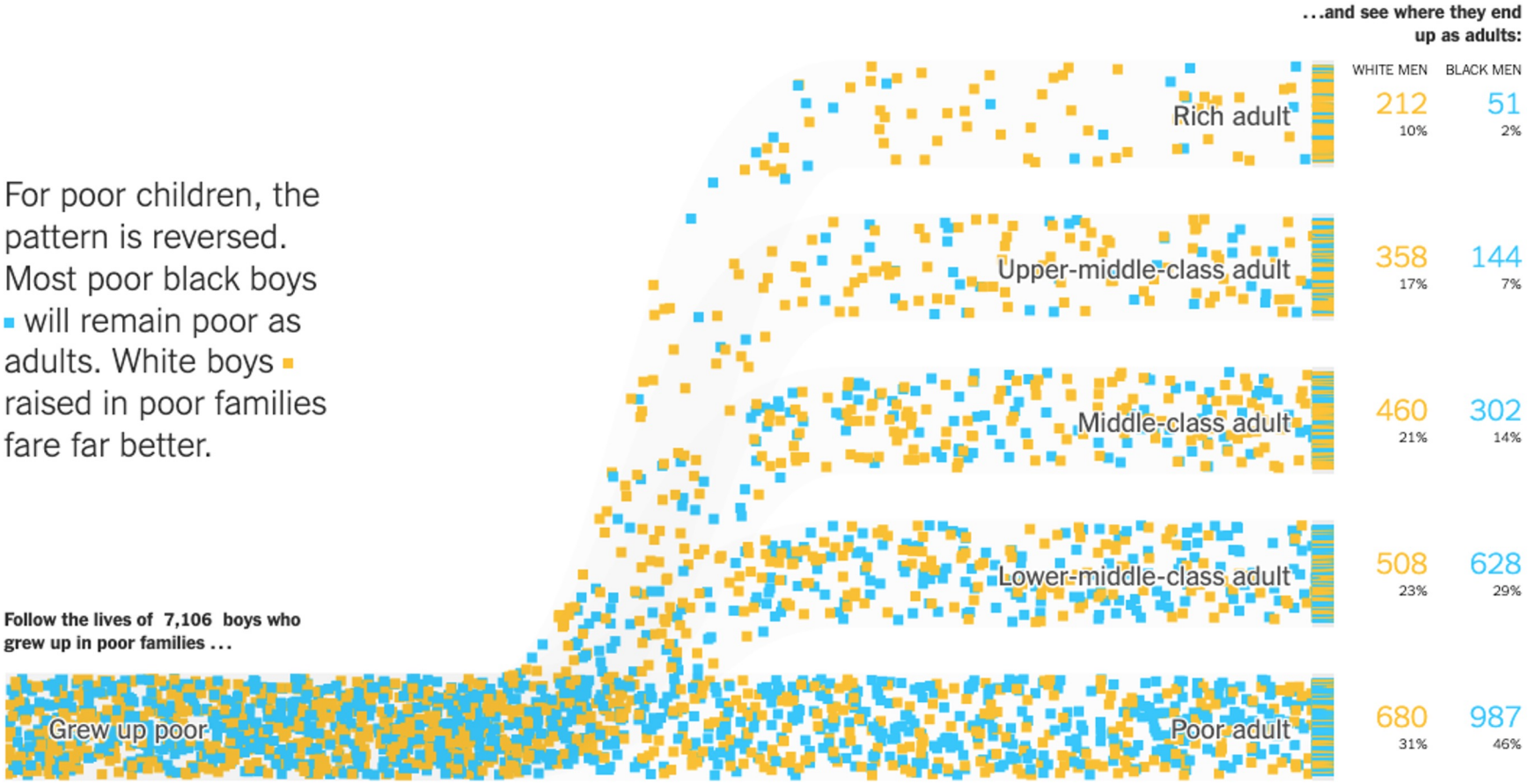


Jesus is popular on Christmas and Easter.



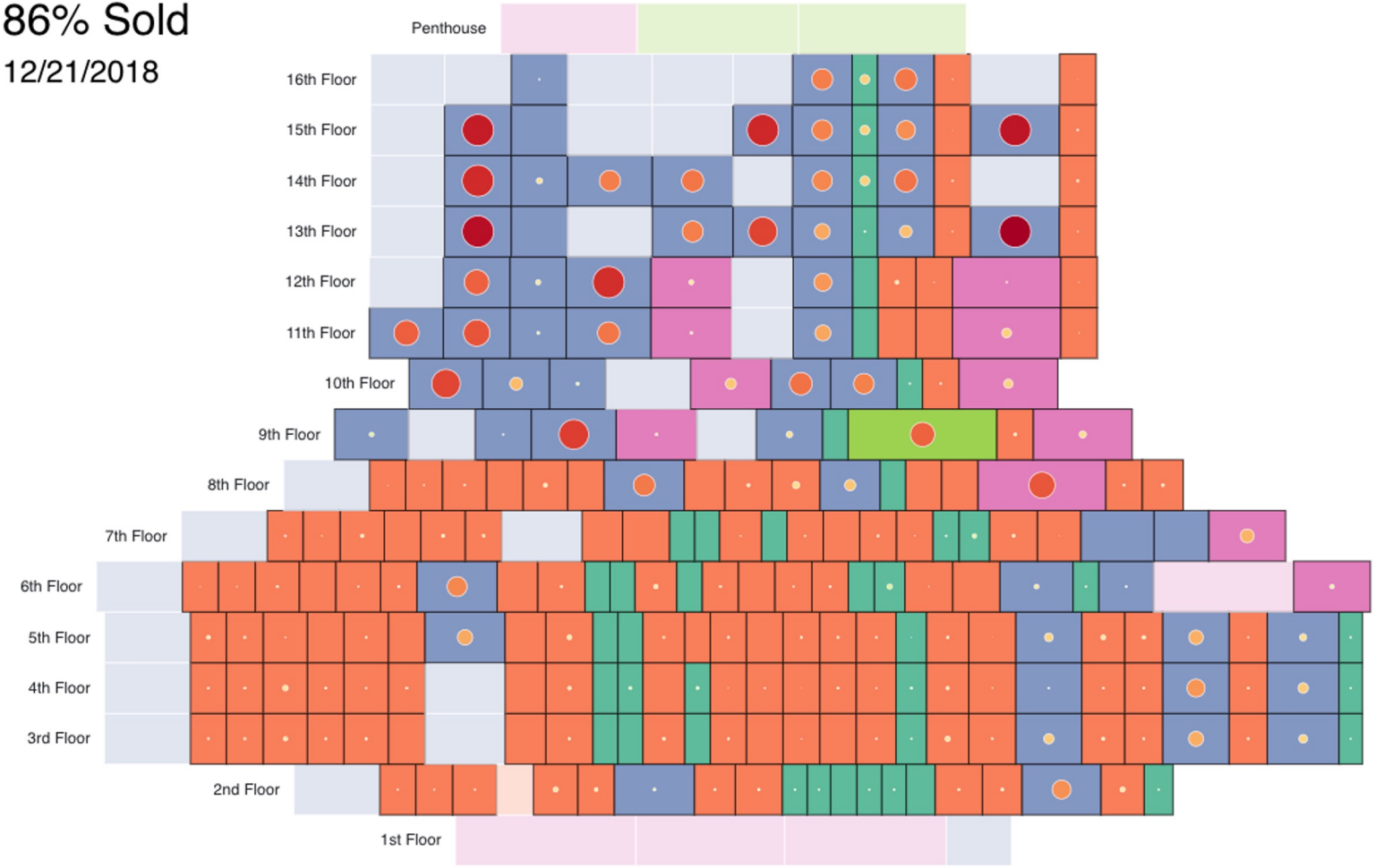
Sourdough. I wonder what happened?

For poor children, the pattern is reversed. Most poor black boys will remain poor as adults. White boys raised in poor families fare far better.



86% Sold

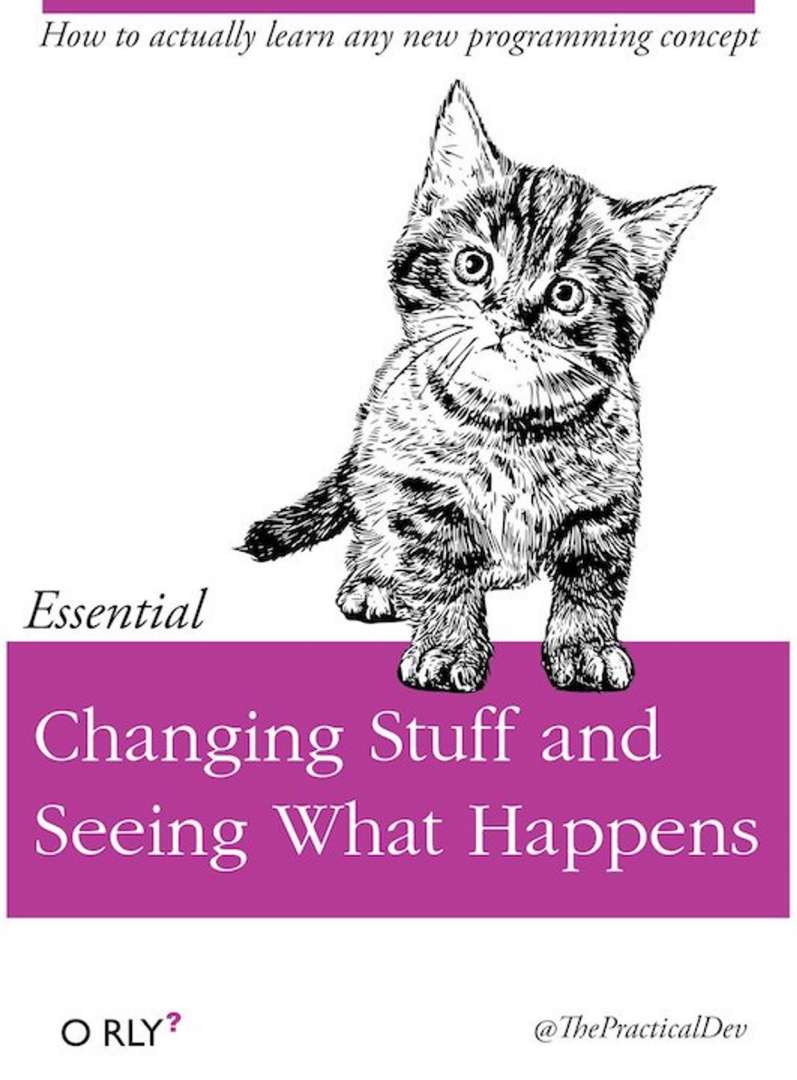
12/21/2018



How to Keep Learning

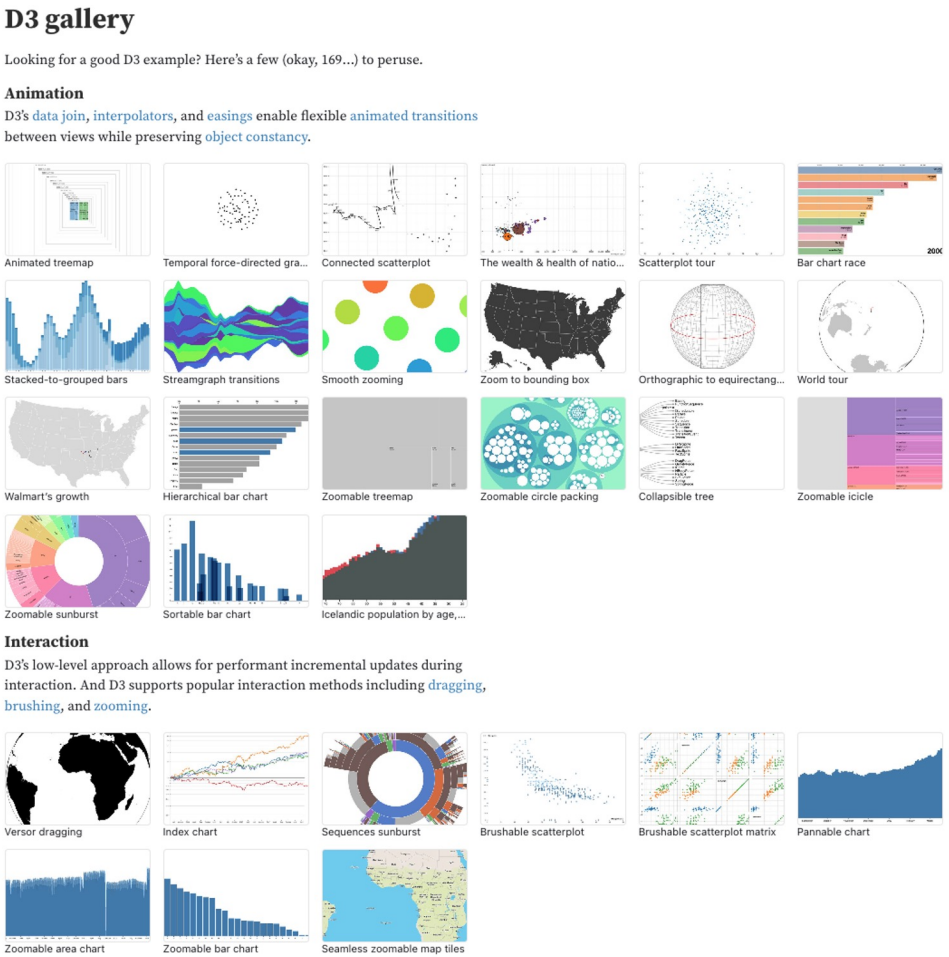
Keep Learning!

Using D3 examples is the best way to learn more D3. Tweaking and seeing what happens is a great method for learning.



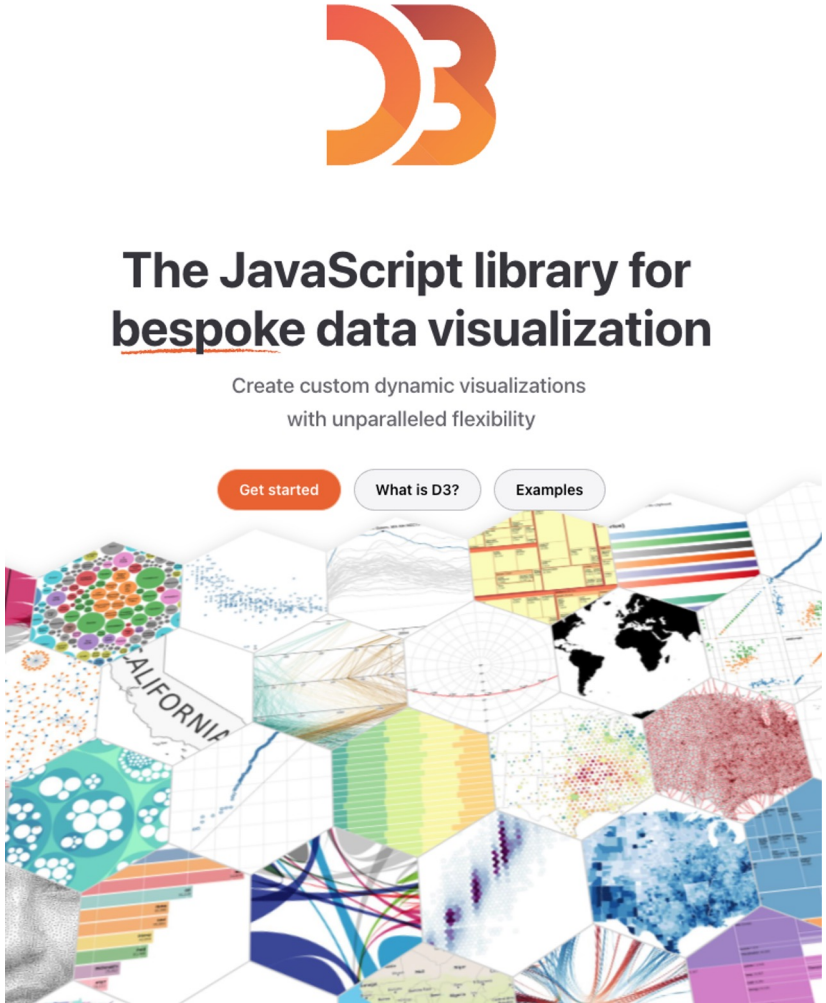
Keep Learning!

The D3 gallery has a 169 examples that you can use to get started with.



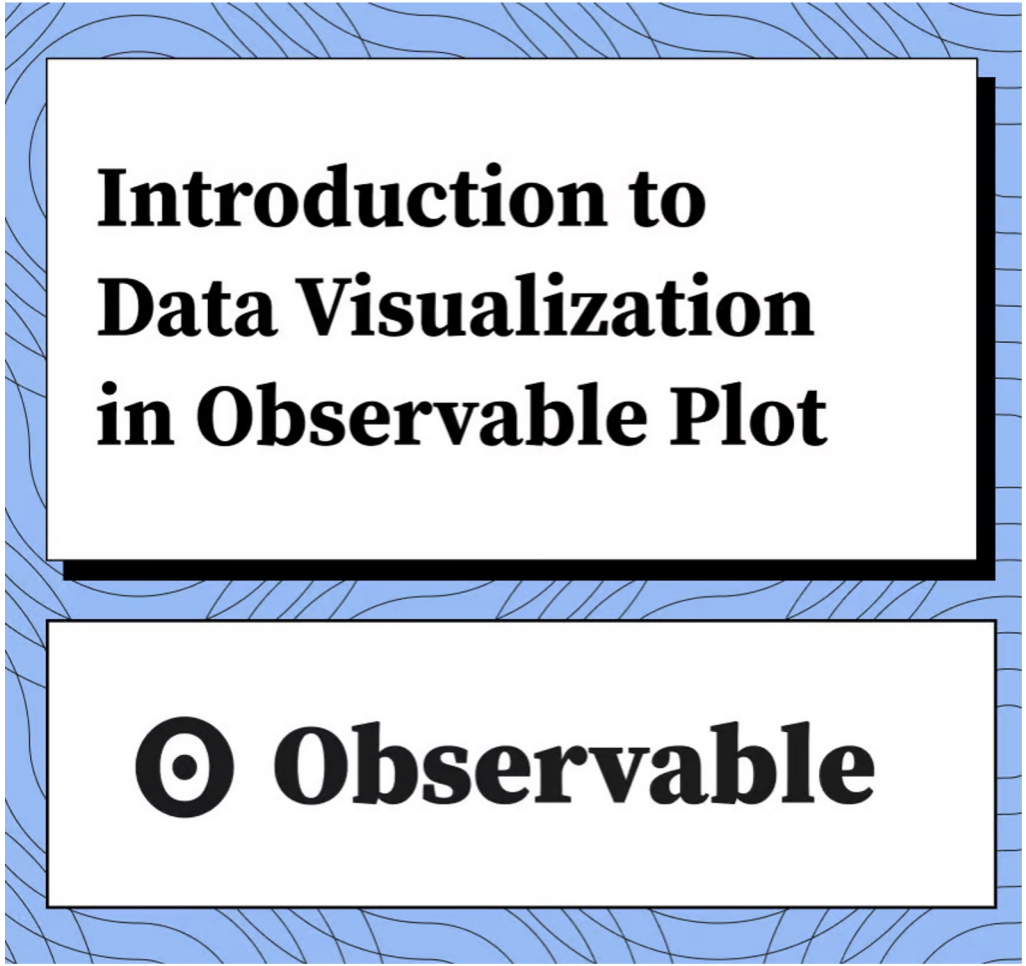
Keep Learning!

Of course there is the D3 website, which is the resource for looking up how specific functions work.



Keep Learning!

We also have an upcoming course on Observable Plot, which is built on top of D3.



Course Assignment

In the next few days we'll send you the course assignment for you to complete to get certification.

The goal will be to know how to find examples, and repurpose them for a new dataset.

You will have two weeks to complete the assignment.

Session 5 is a wrap!

Observable