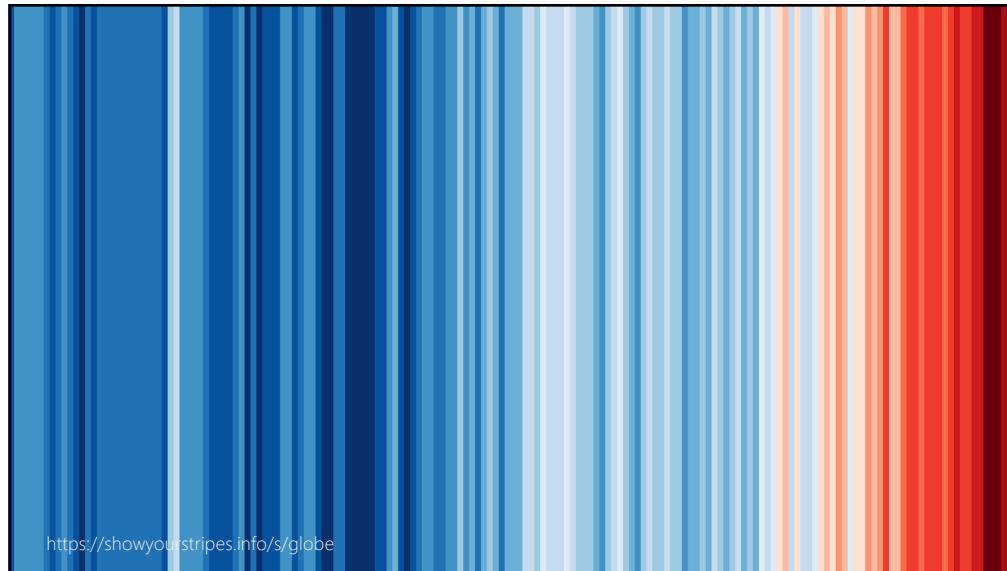


INTRODUCTION TO D3

CS 448B | Fall 204

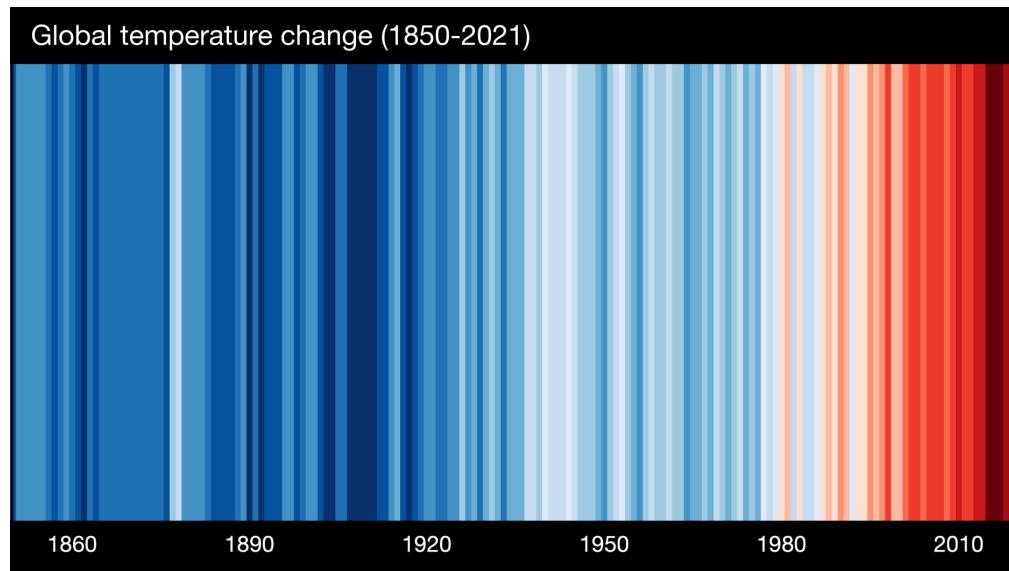
MANEESH AGRAWALA

1

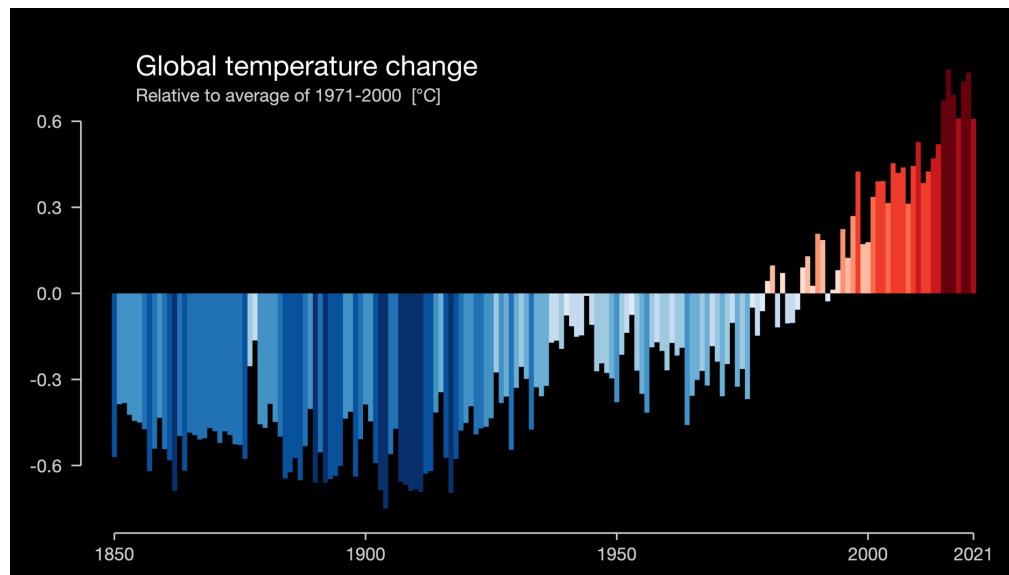


2

1



3



4

READING RESPONSE: QUESTIONS/THOUGHTS

- Heer and Shneiderman's ... While interactivity enhances exploration, I think there is significant room to develop new guidelines for balancing user freedom with narrative structure. A possible ***future direction*** could be designing "***guided***" ***interactivity***, where ***users are free to explore*** but are ***subtly directed towards important insights***, ensuring that they engage with key patterns before moving into the details.
- ... the ***integration of AR*** into visual analytics research has the possibility of revolutionizing interaction with data in a three-dimensional space. AR can do so by taking data representation ***beyond the two-dimensional*** screen to ***a spatially interactive visualization*** that ***allows users to more intuitively and naturally manipulate data***.

5

ANNOUNCEMENTS

6

ASSIGNMENT 2: EXP. DATA ANALYSIS

Due NOW

Use **Tableau** or **Vega-Lite** to formulate & answer data questions

First steps

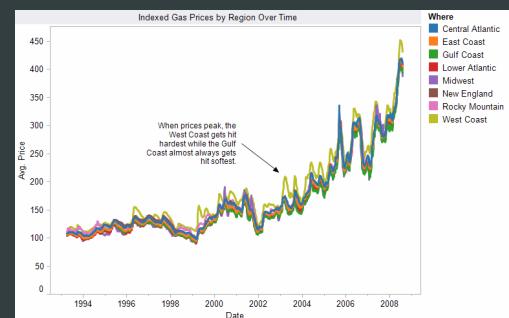
- Step 1: Pick domain & data
- Step 2: Pose questions
- Step 3: Profile data
- Iterate as needed

Create visualizations

- See different views of data
- Refine questions

Author a report

- Screenshots of most insightful views (8+)
- Include titles and captions for each view



7

ASSIGNMENT 3: INTERACTION

Due 10/28 10:30am

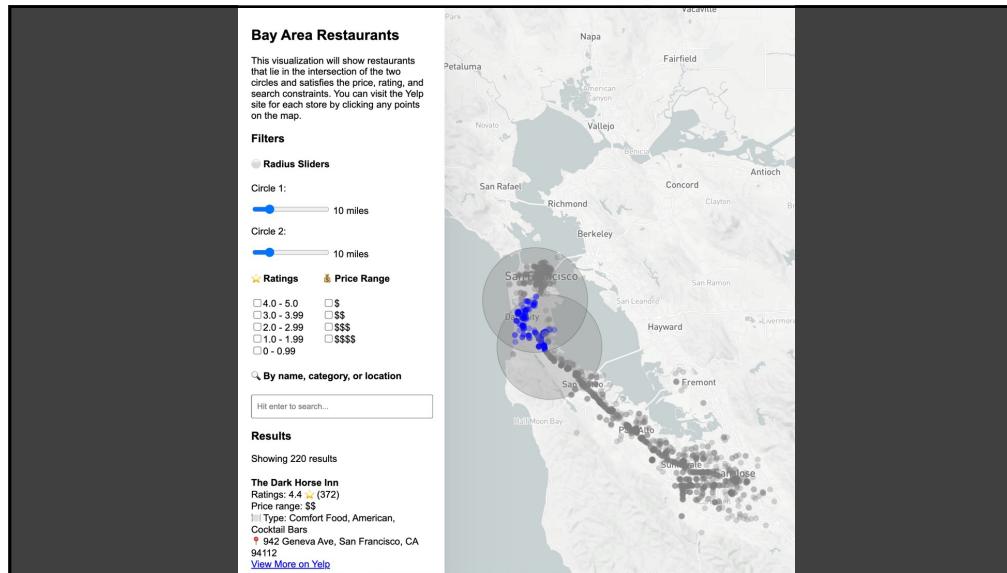
Create a small interactive dynamic query application similar to HomeFinder, but for Bay Area restaurants.

1. Implement interface
2. Submit the application as a website and a short write-up on canvas

Can work alone or in pairs



8



9

D3 NOTEBOOKS TODAY & WED

Data Elements

Enter Update Exit

Team · Published
Introduction to D3 ·
You republished 14 hours ago

1955

You · Published
Making D3 Charts Interactive ·
You republished 14 hours ago

10

TODAY

Learning Objectives

1. Get started with D3 and web technologies it is based on.
2. D3 binding data and joining it with DOM elements.

11

INTRODUCTION TO D3

12

WHAT IS D3?

D3: “Data-Driven Documents”

Data visualization API built on top of **HTML, CSS, JavaScript, & SVG**

Pros:

Highly-customizable

Development and debugging tools

Good documentation, many resources, large community

Integrates with the web

Cons:

Very “*low-level*”

13

hello-world.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>

<body>
  Hello, world!
</body>

</html>
```

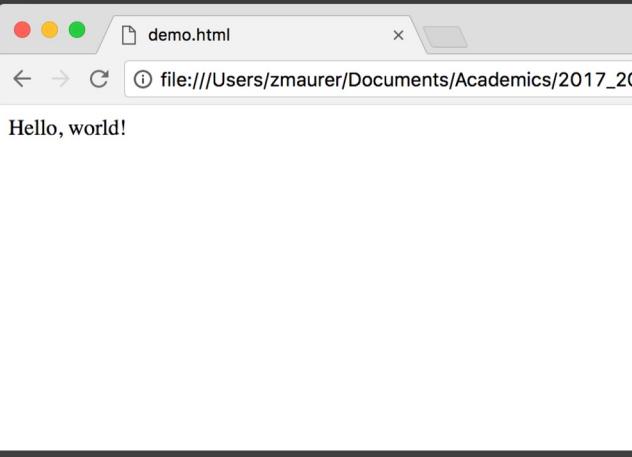
14

hello-world.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>

<body>
Hello, world!
</body>

</html>
```



The screenshot shows a Mac OS X style window titled "demo.html". The address bar indicates the file is located at "file:///Users/zmaurer/Documents/Academics/2017_20". The main content area of the browser displays the text "Hello, world!".

15

hello-css.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">

<style>
body { background: steelblue; }

</style>

</head>

<body>
Hello, world!
</body>

</html>
```

16

hello-css.html

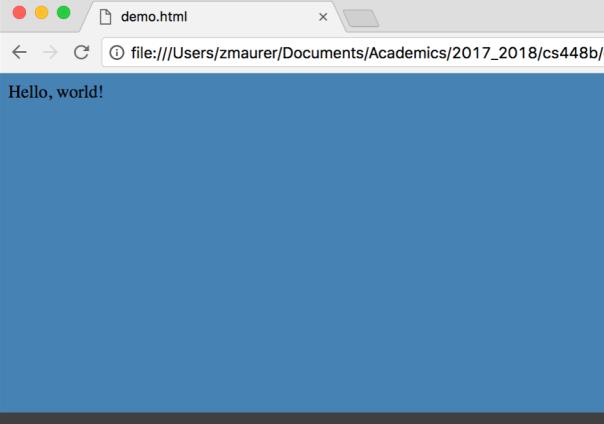
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">

<style>
  body { background: steelblue; }
</style>

</head>

<body>
  Hello, world!
</body>

</html>
```



17

hello-svg.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style> /* CSS */ </style>
</head>

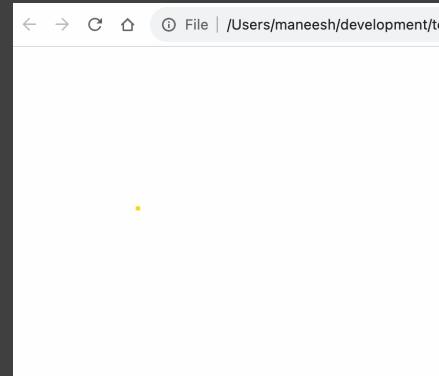
<body>
  <svg width="960" height="500">
    <circle cx='120' cy='150' r='60' style='fill: gold;'>
      <animate
        attributeName='r'
        from='120' to='60' begin='0' dur='3'
        repeatCount='indefinite' />
    </circle>
  </svg>
</body>
</html>
```

18

hello-svg.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style> /* CSS */ </style>
</head>

<body>
<svg width="960" height="500">
<circle cx='120' cy='150' r='60' style='fill: gold;'>
<animate
attributeName='r'
from='2' to='80' begin='0' dur='3'
repeatCount='indefinite' />
</circle>
</svg>
</body>
</html>
```



19

DOCUMENT OBJECT MODEL (DOM)

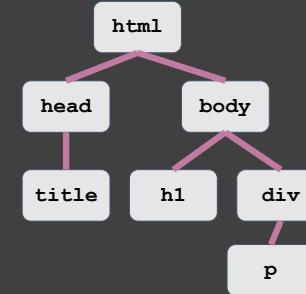
```
<html>
<head>
<title></title>
</head>
<body>
<h1></h1>
<div>
<p></p>
</div>
</body>
</html>
```

Adapted from Victoria Kirst's cs193x [slides](#).

20

DOCUMENT OBJECT MODEL (DOM)

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <div>
      <p></p>
    </div>
  </body>
</html>
```

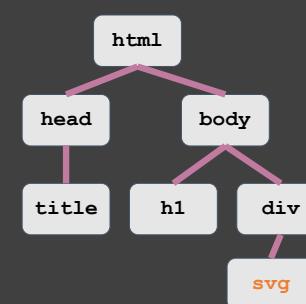


Adapted from Victoria Kirst's cs193x [slides](#)

21

DOCUMENT OBJECT MODEL (DOM)

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <div>
      <svg></svg>
    </div>
  </body>
</html>
```



Adapted from Victoria Kirst's cs193x [slides](#)

22

hello-javascript.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style> /* CSS */ </style>
</head>

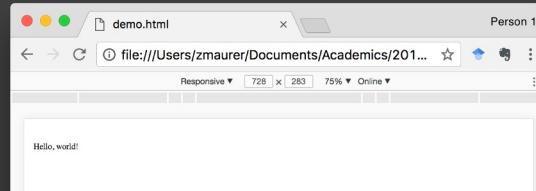
<body>
Hello, world!
<script>
console.log("Hello, console!");
function add2(x) {
    return x + 2;
}
console.log("2 + 2 is " + add2(2));
</script>
</body>
</html>
```

23

hello-javascript.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style> /* CSS */ </style>
</head>

<body>
Hello, world!
<script>
console.log("Hello, console!");
function add2(x) {
    return x + 2;
}
console.log("2 + 2 is " + add2(2));
</script>
</body>
</html>
```



24

hello-d3.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style> /* CSS */ </style>
</head>

<body>
  <script src="https://d3js.org/d3.v7.min.js"></script>
  <script>

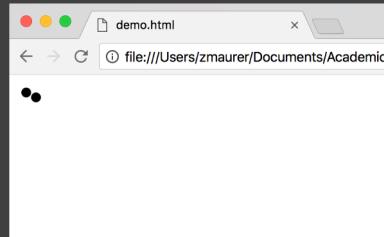
    // JavaScript code that handles the logic of adding SVG elements
    // that make up the visual building blocks of your data visualization

  </script>
</body>
</html>
```

25

D3 SELECTION

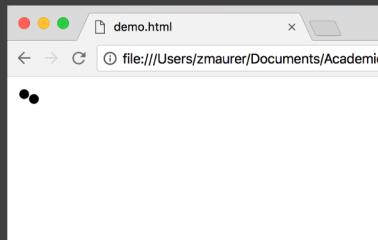
```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
...
```



26

D3 SELECTION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
...
<script>
// select all SVG circle elements
var circles = d3.selectAll("circle");
</script>
```



27

D3 SELECTION AND MANIPULATION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
...
<script>
// select all SVG circle elements
var circles = d3.selectAll("circle");

// set attributes and styles
circles.attr("cx", 40);
circles.attr("cy", 50);
circles.attr("r", 24);
circles.style("fill", "red");

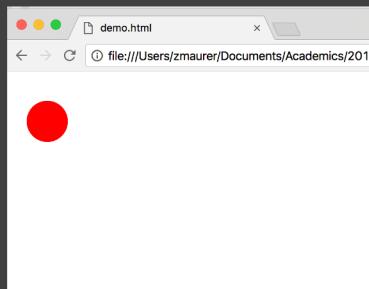
</script>
```

28

D3 SELECTION AND MANIPULATION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"/></circle>
  <circle cx="20" cy="15" r="5"/></circle>
</svg>
...
<script>
  // select all SVG circle elements
  var circles = d3.selectAll("circle");

  // set attributes and styles
  circles.attr("cx", 40);
  circles.attr("cy", 50);
  circles.attr("r", 24);
  circles.style("fill", "red");
</script>
```



29

D3 SELECTION AND MANIPULATION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"/></circle>
  <circle cx="20" cy="15" r="5"/></circle>
</svg>
...
<script>
  // select SVG circle element
  var circles = d3.select("circle");

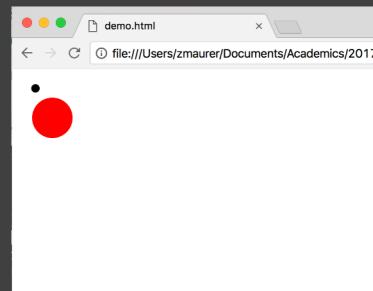
  // set attributes and styles
  circles.attr("cx", 40);
  circles.attr("cy", 50);
  circles.attr("r", 24);
  circles.style("fill", "red");
</script>
```

30

D3 SELECTION AND MANIPULATION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
...
<script>
  // select SVG circle element
  var circles = d3.select("circle");

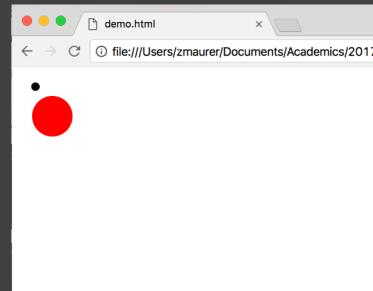
  // set attributes and styles
  circles.attr("cx", 40);
  circles.attr("cy", 50);
  circles.attr("r", 24);
  circles.style("fill", "red");
</script>
```



31

D3 SELECTION AND MANIPULATION

```
<html>
...
<svg width="960" height="500">
  <circle cx="10" cy="10" r="5"></circle>
  <circle cx="20" cy="15" r="5"></circle>
</svg>
...
<script>
  // all together!
  d3.select("circle")
    .attr("cx", 40)
    .attr("cy", 50)
    .attr("r", 24)
    .style("fill", "red");
</script>
```



32

D3 BINDING DATA & JOINING DOM ELEMENTS

year	country	cluster	pop	life_expect	fertility
1955	"Afghanistan"	0	8891909	30.332	7.7
1960	"Afghanistan"	0	9939450	31.997	7.7
1965	"Afghanistan"	0	10987885	34.02	7.7
1970	"Afghanistan"	0	12430623	36.088	7.7
1975	"Afghanistan"	0	14132019	38.438	7.7
1980	"Afghanistan"	0	15121149	39.854	7.8
1985	"Afghanistan"	0	13796928	40.822	7.9
1990	"Afghanistan"	0	14669339	41.674	8
1995	"Afghanistan"	0	20881480	41.763	8
2000	"Afghanistan"	0	23898198	42.129	7.4792

Note that we have put a `Imports` section at the end of this document where we import various utility functions such as the `printTable()` function.

Let's also extract a subset of this data for the year 2005, sort it by population and slice off the top 10 countries. Expand the cell below to see how we obtain this subset. We will use this subset in our first few D3 examples.

```
listData = ► Array(10) [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object]
  // extract the 10 most populous countries in 2005
  listData = gapminder
    .filter(d => d.year === 2005)
    .sort((a, b) => b.pop - a.pop)
    .slice(0, 10)
```

33

D3 BINDING DATA & JOINING DOM ELEMENTS

```
1. China: 1303182268
2. India: 1080264388
3. United States: 295734134
4. Indonesia: 218465000
5. Brazil: 186112794
6. Pakistan: 162419946
7. Bangladesh: 144319628
8. Nigeria: 128765768
9. Japan: 127417244
10. Mexico: 106202903

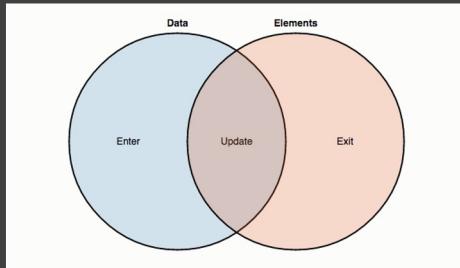
{
  const ol = d3.create('ol');

  ol.selectAll('li') // select all list elements (orange circle above)
    .data(listData) // bind all our data values (blue circle above)
    .join(
      enter => enter.append('li'), // append an li element for each entering item
      update => update, // do nothing with items that match an existing element
      exit => exit.remove() // remove li elements whose backing data is now gone
    )
    .text(d => `${d.country}: ${d.pop}`)

  return ol.node();
}
```

34

D3 BINDING DATA & JOINING DOM ELEMENTS



A join creates three sub-selections:

Enter: selection containing placeholders for every data value that did not have a corresponding DOM element in the original selection

Update: selection containing *existing* DOM elements that match a bound data value

Exit: selection that also contains *existing* DOM elements, but for which a matching data value was not found

35

D3 BINDING DATA & JOINING DOM ELEMENTS

Exercise:
Modify the `enter`, `update`, and `exit` functions in the code below such that entering items are colored green, updating items are colored blue, and exiting items are not removed but rather colored red.

```

1. China: 1303126268
2. India: 1080264588
3. United States: 295734134
4. Indonesia: 260266000
5. Brazil: 180112794
6. Pakistan: 162419946
7. Bangladesh: 141192828
8. Nigeria: 128763768
9. Japan: 127417244
10. Mexico: 102029020

```

```

    .selectAll('l1') // select all list elements (orange circle above)
    .data(medals) // bind all our data values (blue circle above)
    .join(
      enter => enter.append('li').style('color', 'green'),
      update => update.style('color', 'blue'),
      exit => exit.remove()
    )
    .text(d => `id: ${d.country}: ${d.pop}`);
  
```

You can use the variables below to change the elements in the medals extracted in the code above to test your changes to `enter`, `update`, and `exit`. Note that we are using Observable's ability to automatically figure out the dependency structure between cells so that changes to the cells below correctly update the cells above.

```

year = 2005
year = 2005

n = 10
n = 10

```

37

18