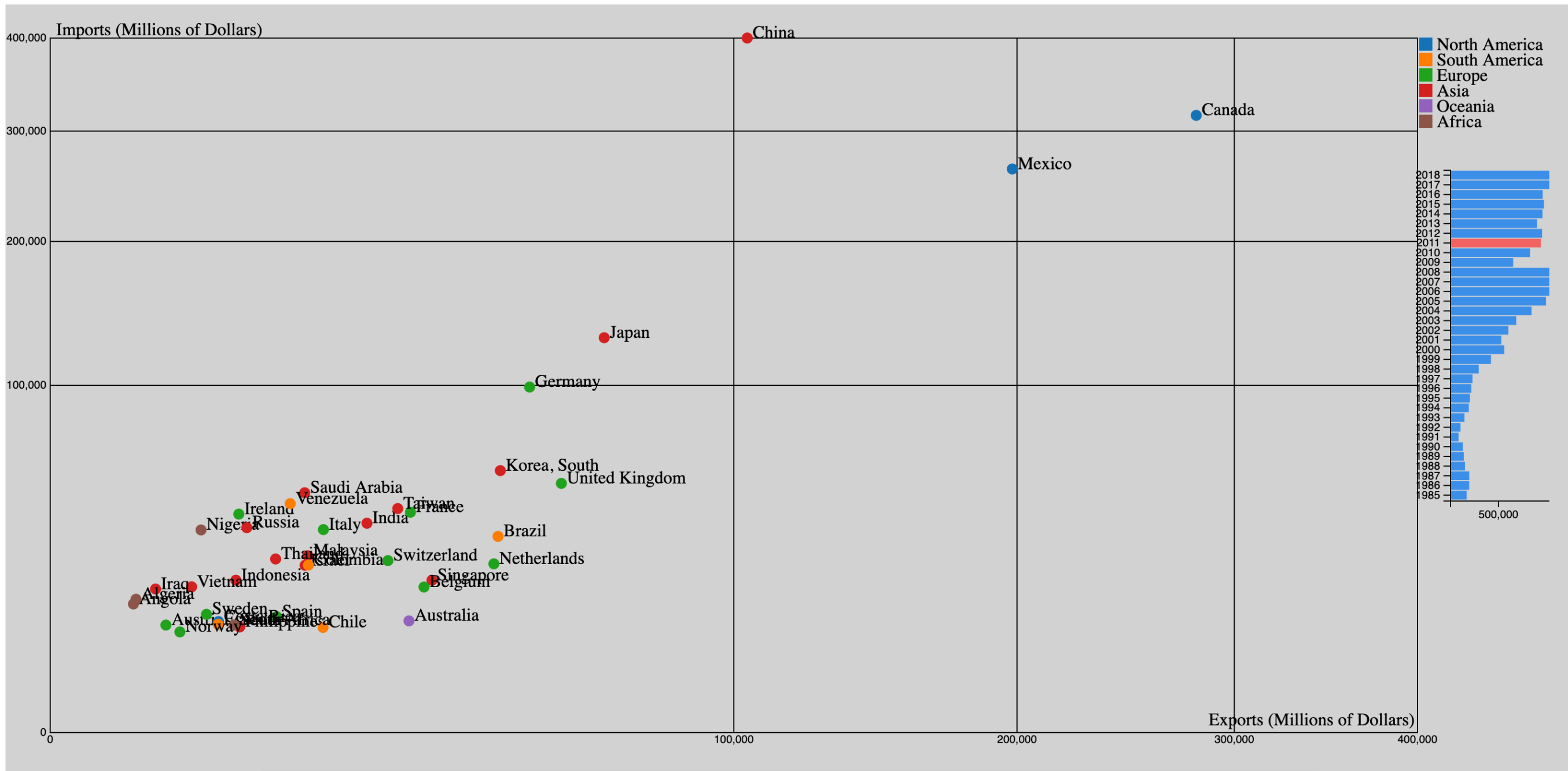


D3: Enter, Update, Exit Pattern

Matt Britton

Our example for today

<https://github.com/MattJBritton/4460EnterUpdateExitLecture>

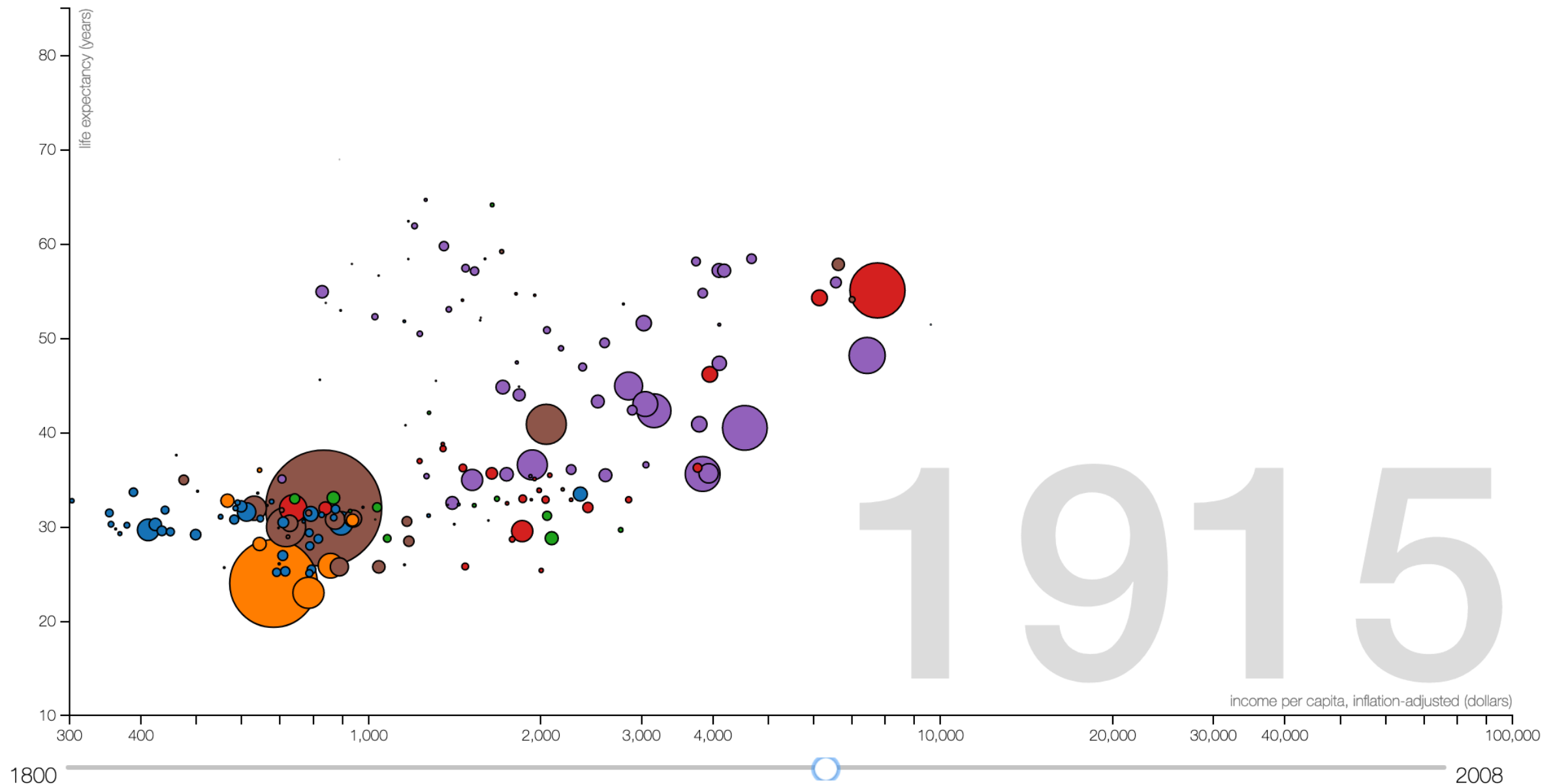


Inspired By

<http://romsson.github.io/dragit/example/nations.html>

A Re-Recreation of Gapminder's Wealth & Health of Nations

Use your mouse to click and drag countries (represented as circles) to explore +200 years of life expectancy and income indicators.



Structure of a D3 Program

* CSS

```
text {  
  font: bold 48px monospace;  
}
```

* Import Scripts

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

* Some HTML for titles, menus, sidebars, SVGs

```
<p id="menu"><b>Top States by Age Bracket, 2008</b><br>Age: <select></select>
```

* Load Data

```
d3.csv("states-age.csv", function(data) {  
  states = data;  
  
  var ages = d3.keys(states[0]).filter(function(key) {  
    return key !== "State" && key !== "Total";  
  });  
  
  states.forEach(function(state) {  
    ages.forEach(function(age) {  
      state[age] = state[age] / state.Total;  
    });  
  });  
});
```

* Margins

```
var margin = {top: 20, right: 40, bottom: 10, left: 40},  
    width = 960,  
    height = 250 - margin.top - margin.bottom;
```

* Axes/Scales

```
var y = d3.scale.ordinal()  
    .rangeRoundBands([0, height], .1);
```

* Enter Update Exit

What is Enter Update Exit?

- * Basically the heart of D3
- * Binds data to a visual encoding on the screen
- * Manages the process of updating which data is shown on the screen (for filtering, sorting, etc.)
- * Easy to just copy/paste this block for static vis...

```
function build_scatterplot() {  
  var bubbleSelection = svg.selectAll("g.bubble")  
    .data(filteredData, d => d.Country)  
    .enter()  
    .append("g")  
    .attr("class", "bubble")  
    .attr("transform", function(d) {  
      return "translate(" + xScale(d.Exports) + "," +  
        + yScale(d.Imports) + ")";  
    });  
  
  bubbleSelection  
    .append("circle")  
    .attr("r", 5)  
    .style("fill", d => colorScale(d.Continent))  
}
```


Static vs Interactive Visualizations

* Static

* Easy to copy/paste this section

```
1 function update(data) {  
2  
3  
4   g.selectAll("text")  
5  
6     .data(data)  
7  
8     .enter()  
9  
10    .append("text")  
11  
12      .attr("class", "enter")  
13  
14      .attr("dy", ".35em")  
15  
16      .text(function(d) { return d; })  
17  
18      .attr("x", function(d, i) { return i * 32; });  
19 }
```

* Interactive

* Need to know the details!

```
112 function build_scatterplot() {  
113  
114   var bubbleSelection = svg.selectAll("g.bubble")  
115     .data(filteredData, d => d.Country);  
116  
117   bubbleSelection.exit().transition().duration(animation_duration)  
118     .attr("transform", "translate(0,0)")  
119     .style("fill-opacity", 0) |  
120     .remove();  
121  
122   var enter = bubbleSelection  
123     .enter()  
124     .append("g")  
125     .attr("class", "bubble");  
126  
127   enter  
128     .append("circle")  
129     .attr("r", 5)  
130     .style("fill", d => colorScale(d.Continent))  
131  
132   enter  
133     .append("text")  
134     .attr("x", 5)  
135     .attr("alignment-baseline", "middle")  
136     .text(d => d.Country);  
137  
138   enter.merge(bubbleSelection)  
139     .transition().duration(animation_duration)  
140     .attr("transform", function(d) {  
141       return "translate(" + xScale(d.Exports) + ","  
142         + yScale(d.Imports) + ")";  
143     });  
144 }  
145
```


Line-by-line breakdown

Get all matching elements
on the screen

Outer join the data
to the elements

Subtract sets to get
the new elements only

Give each new data case
a text element

```
1  function update(data) {  
2  
3  
4    g.selectAll("text")  
5  
6    .data(data)  
7  
8    .enter()  
9  
10   .append("text")  
11  
12   .attr("class", "enter")  
13  
14   .attr("dy", ".35em")  
15  
16   .text(function(d) { return d; })  
17  
18   .attr("x", function(d, i) { return i * 32; });  
19 }
```

Style to your heart's content

How does this work when there is data on the screen already?

Save selection
w/ data as variable

Use that var for
update,
enter,
and exit

```
1  function update(data) {  
2  
3      var text = g.selectAll("text")  
4          .data(data, function(d) { return d; });  
5  
6  
7      text.attr("class", "update");  
8  
9  
10     text.enter().append("text")  
11         .attr("class", "enter")  
12         .attr("dy", ".35em")  
13         .text(function(d) { return d; })  
14  
15     .merge(text)  
16         .attr("x", function(d, i) { return i * 32; });  
17  
18  
19     text.exit().remove();  
20 }
```


Other neat things to do with Enter Update Exit

- * `D3.Transition()`
 - * Animation effect, works best with object constancy
- * Lots of CSS selectors for complex elements
- * Multiple coordinated views
- * You can use Enter, Update, Exit to build all kinds of interface elements, such as menus.

```
1 menu.selectAll("option")
2   .data(ages)
3   .enter().append("option")
4   .text(function(d) { return d; });
5
6 menu.property("value", "18 to 24 Years");
```