

Drawing Maps with D3



Ben Sullins

DATA GEEK

@bensullins www.bensullins.com



What You'll Learn



Building Choropleth Maps

Building Bubble Maps

Final Project Map





Cogsley Services Inc.

Founded in 2008

Technical Consulting Firm

**Needs to understand customer locations
for targeted marketing efforts**



Building Choropleth Maps



Demo



Map Template

Drawing the Base

Adding Layers

Adding Color



Map Template

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>
  /* insert css here */
</style>
<body>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script src="http://d3js.org/topojson.v1.min.js"></script>
<script>
  /* insert D3 here */
</script>
```



Drawing the Base

```
//setup our dimensions (could be done dynamically)
var width = 960,
    height = 600;

//our topojson already has a projection
var path = d3.geo.path()
    .projection(null);

//add svg to body
var svg = d3.select("body").append("svg")
    .attr("width", width)
    .attr("height", height);
```



Drawing the Base

```
//read in us.json we created in previous module
d3.json("us.json", function(error, us) {
  if (error) return console.error(error);

  //append path to svg and draw map!
  svg.append("path")
    .datum(topojson.feature(us, us.objects.states))
    .attr("class", "land")
    .attr("d", path);
});
```



Drawing the Base

```
//add style info
.land {
  fill: #e5e5e5;
  stroke: #fff;
  stroke-linejoin: round;
  stroke-linecap: round;
}
```



Adding Layers

```
//draw base  
svg.append("path")  
    .datum(topojson.feature(us, us.objects.nation))  
    .attr("class", "land")  
    .attr("d", path);
```



Adding Layers

```
//add state borders
svg.append("path")
    .datum(topojson.mesh(us, us.objects.states))
    .attr("class", "border state")
    .attr("d", path);
```



Adding Layers

```
//add county borders
svg.append("path")
    .datum(topojson.mesh(us, us.objects.counties))
    .attr("class", "border county")
    .attr("d", path);
```



Adding Layers

```
//add style info
```

```
.border {  
  fill: none;  
  stroke: #fff;  
}
```

```
.county {  
  stroke: #999;  
  stroke-width: .5;  
}
```



Adding Color

```
// get colorbrewer
<script src="colorbrewer.js"></script>

//define color palette using colorbrewer
var color = d3.scale.linear()
    .domain([-100000, 500000])
    .range(colorbrewer.Greens[7]);
```



Adding Color

```
//draw counties
svg.append("g")
  .attr("class", "counties")
  .selectAll("path")
  .data(topojson.feature(us, us.objects.counties).features)
  .enter().append("path")
  .attr("class", "county")
  .attr("d", path)
//add color
.attr("fill", function(d) { return color(d.properties.profit); });
```



Building Bubble Maps



Demo



Adding Bubbles

Sorting Bubbles

Sizing the Bubbles

Adding Tooltips

Formatting Numbers



Adding Bubbles

```
//starting with a base map
svg.append("g")
  .attr("class", "bubble")
  .selectAll("circle")
    .data(topojson.feature(us, us.objects.counties).features)
  .enter().append("circle")
//find the center of the county to place the bubble at
  .attr("transform", function(d) {
    return "translate(" + path.centroid(d) + ")"; })
//choose the radius size
  .attr("r", function(d) {
    return Math.sqrt(parseFloat(d.properties.profit)* 0.00005) });
```



Sorting Bubbles

```
//starting with a base map
svg.append("g")
  .attr("class", "bubble")
  .selectAll("circle")
    .data(topojson.feature(us, us.objects.counties).features)
    //sort so larger bubbles are in background
    .sort(function(a, b) {
      return b.properties.profit - a.properties.profit;
    })
    ...rest of code...
```



Sizing Bubbles

```
//add function for calculating size
```

```
var radius = d3.scale.sqrt()  
    .domain([0, 1e6])  
    .range([0, 15]);
```

```
//before
```

```
.attr("r", function(d) {  
    return Math.sqrt(parseFloat(d.properties.profit)* 0.00005) });
```

```
//after
```

```
.attr("r", function(d) { return radius(d.properties.profit); })
```



Adding Tooltips

```
//add tooltips
.append("title")
  .text(function(d) {
    return d.properties.name
      + "\nProfit " + d.properties.profit;
  });
```



Formatting Numbers

```
//add function for formatting
function formatSales(val) {
    var prefix = d3.formatPrefix(val),
        format = d3.format(".1f");
    return format(prefix.scale(val)) + prefix.symbol;
}

//add tooltips
.append("title")
    .text(function(d) {
        return d.properties.name
            + "\nProfit " + formatSales(d.properties.profit);
    });
```



Final Project Map



Demo



Adding a Legend

Adding Styling

Reading Data with Queue

Adding a Hovercard

Adding a Bar Chart to the Hovercard



Adding a Legend

```
//add legend
var legend = svg.append("g")
  .attr("class", "legend")
  .attr("transform", "translate(" + (width - 50) + "," + (height - 20)
    + ")")
  .selectAll("g")
  .data([1e6, 5e6, 1e7])
  .enter().append("g");
```



Adding a Legend

```
legend.append("circle")  
  .attr("cy", function(d) { return -radius(d); })  
  .attr("r", radius);
```

```
legend.append("text")  
  .attr("y", function(d) { return -2 * radius(d); })  
  .attr("dy", "1.3em")  
  .text(d3.format(".1s"));
```



Adding Styling

```
/*body formatting*/  
body { background-color: #181818;}  
  
/*base map formatting*/  
.land { fill: #3A3A3A;}  
.border {  
    fill: none;  
    stroke: #606060;  
    stroke-linejoin: round;  
    stroke-linecap: round;  
}
```



Adding Styling

```
/*bubble format*/  
.bubble {  
    fill: #F15B2A; /* PS Orange */  
    fill-opacity: .5;  
    stroke: #606060;  
    stroke-width: .5px;  
}  
  
.bubble :hover { stroke: #606060; }
```



Adding Styling

```
/*legend format*/  
.legend circle {  
    fill: none;  
    stroke: #606060;  
}  
.legend text {  
    fill: #777;  
    font: 10px sans-serif;  
    text-anchor: middle;  
}
```



Reading Data with Queue

```
<script src="http://d3js.org/queue.v1.min.js"></script>
```

```
//read in data asynchronously then call our ready function  
queue()
```

```
  .defer(d3.json, "us.json") //our topojson from before  
  .defer(d3.csv, "category-sales.csv") //our data for the bar chart  
  .await(ready);
```

```
function ready(error, us, catSales) {  
  if (error) throw error;  
  //us = us.json  
  //catSales = category-sales.csv
```



Adding a Hovercard

```
//add dynamic tooltip
var barTooltip = d3.select("body").append("div")
    .attr("class", "tooltip")
    .style("opacity", 0)
    .style("width", 600);

//add it to a DOM event
.on("mouseover", function(d) {
    var circleId = d.id;
    barTooltip.transition()
        .duration(500)
        .style("opacity", .7);
```



Adding a Hovercard

```
var tip = "<h3>" + d.properties.name + "</h3>";  
var tip = tip+"<strong>Orders:</strong>" +  
    formatNum(d.properties.orders) + "<br/>";  
var tip = tip+"<strong>Profit:</strong> $" +  
    formatSales(d.properties.profit)+ "<br/>";  
var tip = tip+"<h4>Category Sales</h4>";
```

```
barTooltip.html(tip)  
    .style("left", (d3.event.pageX) + "px")  
    .style("top", (d3.event.pageY) + "px");
```

```
//need to add styling info for correct positioning
```



Adding a Hovercard

```
.tooltip {  
  position: absolute;  
  max-width: 400px;  
  height: auto;  
  padding: 5px;  
  background-color: #606060;  
  /* rounded border & shadow */  
  pointer-events: none; /* ¡MUY IMPORTANTE! */  
  font-family: sans-serif;  
  font-size: 12px; color: #fff;  
}
```



Adding a Hovercard

```
//remove hovercard on mouseout  
.on("mouseout", function(d) {  
    barTooltip.transition()  
        .duration(500)  
        .style("opacity", 0);  
});
```



Adding a Bar Chart to the Hovercard

```
/* setup already added */
var chart = barTooltip.append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
.append("g")
    .attr("transform", "translate(" + margin.left + ","
        + margin.top + ")");
//make sure to filter to the current ID
x.domain(catSales.map(function(d) { return d.category; }));
y.domain([0, d3.max(catSales.filter(
    function(d) { return d.id == circleId },
    function(d) { return d.sales; })
    ]);
```



Adding a Bar Chart to the Hovercard

```
//add axes
chart.append("g")
    .attr("class", "x-axis")
    .attr("transform", "translate(0," + height + ")")
    .call(xAxis);

chart.append("g")
    .attr("class", "y-axis")
    .call(yAxis);
```



Adding a Bar Chart to the Hovercard

```
//build bar chart
chart.selectAll("#barChart")
  .data(catSales)
  .enter().append("rect")
  //apply filter for the state we're currently looking at
  .filter(function(d) { return d.id == circleId })
  .attr("class", "bar")
  .attr("x", function(d) { return x(d.category); })
  .attr("y", function(d) { return y(d.sales); })
  .attr("height", function(d) { return height - y(d.sales); })
  .attr("width", x.rangeBand());
```



Where to Find More



Where to Find More



Pluralsight Courses

- D3.js Fundamentals

Code School

- JavaScript Road Trip
- D3 Course (future)

External

- <https://bl.ocks.org/mbostock>
- <http://bit.ly/bs-d3-tips>

