# https://rubenhm.org/blog/

Research, Writings + Code
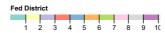
---

## *Coding Annoyances*

---

## Fed Districts Map with D3

30 March 2018 • Rubén Hernández-Murillo •

---

The goal is to generate a map of the 12 Federal Reserve Districts using d3.js.

Here is the final result:



Open ↗ (/downloads/blog/2018-03-30-fed-districts-map-with-d3/fed-districts.html)
The historical definitions of the Fed Districts are available from FRASER
(https://fraser.stlouisfed.org/files/docs/historical/federal%20reserve%20history/frdistricts/frb_districts_199603.pdf),
a service for archival information from the St. Louis Fed.

The county definitions in the PDF were manually coded into a TSV file (/downloads/blog/2018-03-30-fed-districts-
map-with-d3/counties-by-fed-district.tsv) by Chris Vecchio (https://github.com/chris-vecchio). Over time, some of
the county definitions have changed (https://www.census.gov/geo/reference/county-changes.html) but some may
have been missed, so clearly, this file does not represent official definitions.

## Choropleth map

The Fed Districts map is a straight-up modification of the recent example of a choropleth map for unemployment rates
by Mike Bostock (https://bl.ocks.org/mbostock/4060606). He uses a threshold scale and the blues color scheme which
work great for a continuous variable such as the unemployment rate, but are probably not the best choice for a
categorical map with 12 categories! For one, the blues color scheme does not even have 12 categories. So I went over to
ColorBrewer (http://colorbrewer2.org/#type=qualitative&scheme=Set3&n=12) and selected a scheme with soft colors
for qualitative data with 12 categories.

I replaced the CSV file for the unemployment rates with the Fed Districts definitions and I added code to define the
boundaries between Districts using `topojson.mesh` in the same way as Mike generated the boundaries between states.

I also learned about `map()` to define dictionary-like variables that contain useful information about each county to display on hover.

Notice that the Fed District numbers are defined in a Westerly direction.

```
1.  <!DOCTYPE html>
2.  <meta charset="utf-8">
3.  <style>
4.
5.  .counties {
6.    fill: none;
7.  }
8.
9.  .states {
10.   fill: none;
11.   stroke: #fff;
12.   stroke-linejoin: round;
13. }
14.
15. .districts {
16.   fill: none;
17.   stroke: #808080;
18.   stroke-linejoin: round;
19.   stroke-width: 1.5px;
20. }
21.
22. </style>
23. <svg width="960" height="600"></svg>
24. <script src="https://d3js.org/d3.v4.min.js"></script>
25. <script src="https://d3js.org/d3-scale-chromatic.v1.min.js"></script>
26. <script src="https://d3js.org/topojson.v2.min.js"></script>
27. <script>
28.
29. var svg = d3.select("svg"),
30.     width = +svg.attr("width"),
31.     height = +svg.attr("height");
32.
33. // Define maps with definitions and county and District names
34. var frbdefinitions = d3.map();
35. var countynames    = d3.map();
36. var districtNames  = d3.map();
37.
38. // Harcoded Fed District denominations
39. // The city name indicates the city of the main office
40. // Notice that the district numbers are defined in a Westerly direction
41. districtNames.set(1,"Boston")
42. districtNames.set(2,"New York")
43. districtNames.set(3,"Philadelphia")
44. districtNames.set(4,"Cleveland")
45. districtNames.set(5,"Richmond")
46. districtNames.set(6,"Atlanta")
47. districtNames.set(7,"Chicago")
48. districtNames.set(8,"St. Louis")
49. districtNames.set(9,"Minneapolis")
50. districtNames.set(10,"Kansas City")
51. districtNames.set(11,"Dallas")
52. districtNames.set(12,"San Francisco")
53.
54. var path = d3.geoPath();
55.
56. var x = d3.scaleLinear()
57.     .domain([0, 12])
58.     .rangeRound([600, 860]);
59.
60. var color = d3.scaleThreshold()
61.     .domain(d3.range(1, 13))
62.     .range(['#8dd3c7','#ffffb3','#bebada','#fb8072','#80b1d3','#fdb462','#b3de69','#fccde5','#d9d9d9','#bc80b
    d','#ccebc5','#ffed6f']);
63.
64. var g = svg.append("g")
65.     .attr("class", "key")
66.     .attr("transform", "translate(0,40)");
67.
```

```
 68. g.selectAll("rect")
 69.   .data(color.range().map(function(d) {
 70.       d = color.invertExtent(d);
 71.       if (d[0] == null) d[0] = x.domain()[0];
 72.       if (d[1] == null) d[1] = x.domain()[1];
 73.       return d;
 74.   }))
 75.   .enter().append("rect")
 76.     .attr("height", 8)
 77.     .attr("x", function(d) { return x(d[0]); })
 78.     .attr("width", function(d) { return x(d[1]) - x(d[0]); })
 79.     .attr("fill", function(d) { return color(d[0]); });
 80.
 81. g.append("text")
 82.     .attr("class", "caption")
 83.     .attr("x", x.range()[0])
 84.     .attr("y", -6)
 85.     .attr("fill", "#000")
 86.     .attr("text-anchor", "start")
 87.     .attr("font-weight", "bold")
 88.     .text("Fed District");
 89.
 90. g.call(d3.axisBottom(x)
 91.     .tickSize(13)
 92.     .tickFormat(function(x, i) { return i ? x : x ; })
 93.     .tickValues(color.domain()))
 94.   .select(".domain")
 95.     .remove();
 96.
 97. d3.queue()
 98.     .defer(d3.json, "https://d3js.org/us-10m.v1.json")
 99.     .defer(d3.tsv, "counties-by-fed-district.tsv", function(d) {
100.         frbdefinitions.set(d.countyfips, +d.dist_frb);
101.         countynames.set(d.countyfips, d.county_name+", "+d.state_abbr);
102.     })
103.     .await(ready);
104.
105. function ready(error, us) {
106.   if (error) throw error;
107.
108.   // Select fill color for Fed Districts using the map for the definitions
109.   // and the color scheme defined earlier
110.   // Also add information to each county to be displayed on hover
111.   svg.append("g")
112.       .attr("class", "counties")
113.     .selectAll("path")
114.     .data(topojson.feature(us, us.objects.counties).features)
115.     .enter().append("path")
116.       .attr("fill", function(d) { return color(d.dist_frb = frbdefinitions.get(d.id) -1); })
117.       .attr("d", path)
118.     .append("title")
119.       .text(function(d) {
120.         return "District: " + (+d.dist_frb + 1) + ". " +
121.                 districtNames.get(+d.dist_frb + 1) + "\n" +
122.                 "County fips: " + d.id + "\n" +
123.                 "County name: " + countynames.get(d.id)});
124.
125.   // Define boundaries for states
126.   svg.append("path")
127.       .datum(topojson.mesh(us, us.objects.states, function(a, b) { return a !== b; }))
128.       .attr("class", "states")
129.       .attr("d", path);
130.
131.   // Define boundaries for Fed Districts using counties
132.   svg.append("path")
133.       .datum(topojson.mesh(us, us.objects.counties, function(a, b) {
134.         return (frbdefinitions.get(a.id) !== frbdefinitions.get(b.id)) & (a !== b); }))
135.       .attr("class", "districts")
136.       .attr("d", path);
137.
138. }
139.
140. </script>
```

```
140.  </script>
```

Follow for updates.

🔊 (https://rubenhm.org/feed.xml) 🐦 @rubenhm_org