

LAB EXERCISE 5

Learning Outcomes:

1. To create choropleth map using d3.js
2. To add legend to choropleth map

Lesson 1: Creating Choropleth Map

Create new HTML file in VS Code. Type ! and press Tab key to get skeleton of HTML code.

Step 1: Load d3.js libraries in <head> section of HTML

<head>

```
<!-- Load d3.js -->
<script src="https://d3js.org/d3.v4.js"></script>
<script src="https://d3js.org/d3-scale-chromatic.v1.min.js"></script>
<script src="https://d3js.org/d3-geo-projection.v2.min.js"></script></head>
```

Step 2: Create SVG with dimension to put the graph in <body> section of HTML

<body>

```
<svg id="my_dataviz" width="1000" height="800"></svg>
```

...

Start writing the scripts in <body> section of HTML. There are few code fragments within the <script> compound.

1. Open script
2. Set SVG
3. Create path and projection
4. Set data and color range
5. Load external data
6. Boot the data
7. Draw the map
8. Set the color for each country
9. Close script

Step 3: Set the dimensions and margins of the graph

```
<script>
var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");</script>
```

Step 4: Set the path and projection

```
var path = d3.geoPath();
var projection = d3.geoMercator()
  .scale(20)
  .center([30, 5])
  .translate([width / 2, height / 2]);
```

***Refer to lecture note for more details about path and projection**

Step 5: Set data and color Scale

Values in .domain represents different colors in the map based on population.

```
var data = d3.map();
var colorScale = d3.scaleThreshold()
  .domain([100000, 1000000, 10000000, 30000000, 100000000, 500000000])
  .range(d3.schemeBlues[7]);
```

Step 6: Load external two external data files using d3.queue()

```
d3.queue()
  .defer(d3.json, "https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson")
  .defer(d3.csv, "https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world_population.csv", function(d) { data.set(d.code, +d.pop); })
  .await(ready);
```

A **queue** evaluates zero or more *deferred* asynchronous tasks with configurable concurrency: you control how many tasks run at the same time. When all the tasks complete, or an error occurs, the queue passes the results to your *await* callback.

When a task completes, it must call the provided callback. The first argument to the callback should be null if the task is successful, or the error if the task failed.

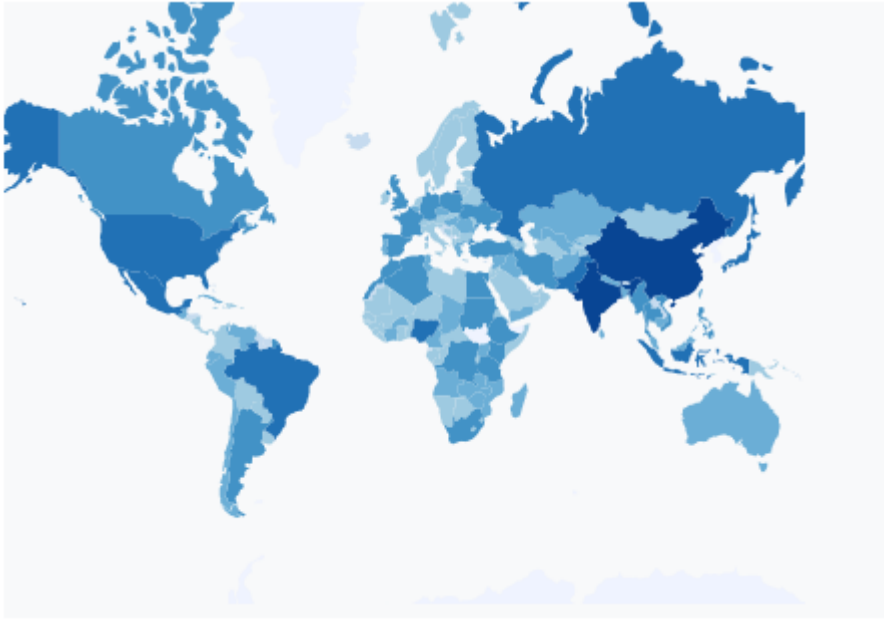
Step 7: Boot the data and draw the map

```
function ready(error, topo) {
  // Draw the map
  svg.append("g")
    .selectAll("path")
    .data(topo.features)
    .enter()
    .append("path")
    // draw each country
    .attr("d", d3.geoPath()
      .projection(projection))
    // set the color of each country
    .attr("fill", function(d) {
      d.total = data.get(d.id) || 0;
      return colorScale(d.total);
    });
}
```

Step 8: Close the script.

```
</script>
```

Output



Reference: https://d3-graph-gallery.com/graph/choropleth_basic.html

Lesson 2: Creating Legend for Choropleth Map

Insert additional code to the script that you have created in Lesson 1.

Step 1: Create a diverging colour scale

D3.scaleLinear() is used to create a diverging colour scale that interpolates between two colours.

```
const x = d3.scaleLinear()
    .domain([2.6, 75.1])
    .rangeRound([600, 860]);
```

scale.rangeRound sets the range as scale.range, while making sure that the values returned will be integers.

Step 2: Create SVG for Legend

```
const legend = svg.append("g")
    .attr("id", "legend");
```

Step 3: Create Legend entry

```
const legend_entry = legend.selectAll("g.legend")
    .data(colorScale.range().map(function(d) {
        d = colorScale.invertExtent(d);
        if (d[0] == null) d[0] = x.domain()[0];
        if (d[1] == null) d[1] = x.domain()[1];
        return d;
    }))
    .enter().append("g")
    .attr("class", "legend_entry");
```

Step 4: Set width and height of legend rectangle

```
const ls_w = 20,
    ls_h = 20;
```

Step 5: Append rectangle to the legend entry

```
legend_entry.append("rect")
    .attr("x", 250)
    .attr("y", function(d, i) {
        return height - (i * ls_h) - 2 * ls_h;
    })
    .attr("width", ls_w)
    .attr("height", ls_h)
    .style("fill", function(d) {
        return colorScale(d[0]);
    })
    .style("opacity", 0.8);
```

Set the position of the legend. Fille the rectangle with different colours based on colorScale.

Step 6: Append rectangle text to legend

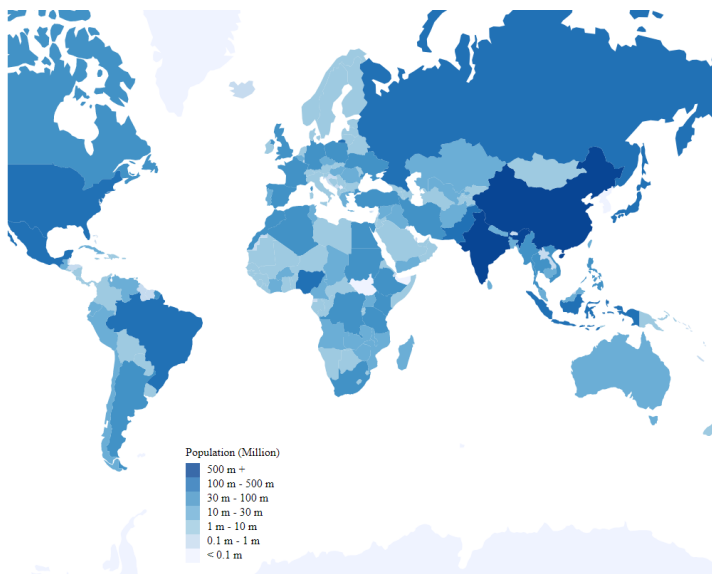
```
legend_entry.append("text")
    .attr("x", 280)
    .attr("y", function(d, i) {
        return height - (i * ls_h) - ls_h - 6;
    })
    .text(function(d, i) {
        if (i === 0) return "< " + d[1] / 1000000 + " m";
        if (d[1] < d[0]) return d[0] / 1000000 + " m +";
        return d[0] / 1000000 + " m - " + d[1] / 1000000 + " m";
    });
```

Set the position and calculate the values based on million.

Step 7: Append the Legend Title

```
legend.append("text").attr("x", 250).attr("y", 630).text("Population  
(Million)");
```

Output



LAB TASK

Find the data about any disease cases in the world.

- a. Create choropleth map to differentiate the total number of cases for all countries.
- b. Consider appropriate pre-attentive attributes such as label, color, legend, etc.
- c. Add interactive elements such as mouse hover, tooltips and bubble.

Submission item: HTML file.

Submission platform: MS Teams

Deadline: 23rd October 2023