<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>D3: Loading GeoJSON data and generating SVG paths</title>

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

<style type="text/css">

/\* No style rules here yet \*/

</style>

</head>

<body>

<script type="text/javascript">

//Width and height

var w = 500;

var h = 300;

//Define default path generator

var projection = d3.geoAlbersUsa()

.translate([w/2, h/2])

.scale(500);

var path = d3.geoPath().projection(projection);

//Create SVG element

var svg = d3.select("body")

.append("svg")

.attr("width", w)

.attr("height", h);

//Load in GeoJSON data

d3.json("us-states.json", function(json) {

d3.csv("us-ag-productivity-2004.csv", function(csv)

{

for (var i = 0; i < csv.length; i++) {

var dataState = csv[i].state;

var dataValue = parseFloat(csv[i].value);

for (var j = 0; j < json.features.length; j++) {

var jsonState = json.features[j].properties.name;

if (dataState == jsonState) {

json.features[j].properties.value = dataValue;

break;

}

}

}

color = d3.scaleQuantize()

.range(["rgb(237,248,233)", "rgb(186,228,179)",

"rgb(116,196,118)", "rgb(49,163,84)","rgb(0,109,44)"])

.domain([

d3.min(csv, function(d) { return d.value; }),

d3.max(csv, function(d) { return d.value; })

]);

//Bind data and create one path per GeoJSON feature

svg.selectAll("path")

.data(json.features)

.enter()

.append("path")

.attr("d", path)

.style("fill", function(d) {

//Get data value

var value = d.properties.value;

if (value) {

//If value exists…

return color(value);

} else {

//If value is undefined…

return "#ccc";

}

});

d3.csv("us-cities.csv", function(data) {

svg.selectAll("circle")

.data(data)

.enter()

.append("circle")

.attr("cx", function(d) {

return projection([d.lon, d.lat])[0];

})

.attr("cy", function(d) {

return projection([d.lon, d.lat])[1];

})

.attr("r", function(d) {

return Math.sqrt(parseInt(d.population) \* 0.00004);

})

.style("fill", "yellow")

.style("stroke", "gray")

.style("stroke-width", 0.25)

.style("opacity", 0.75)

.append("title") //Simple tooltip

.text(function(d) {

return d.place ;

});

});

});

});

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<title>D3: Stacked area chart</title>

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

<style type="text/css">

h1 {

font-family: Helvetica, sans-serif;

font-size: 14px;

font-weight: bold;

}

.area {

stroke: none;

}

.area:hover {

fill: yellow;

}

</style>

</head>

<body>

<h1>Monthly Number of Electric Vehicles Sold in the U.S.: December 2010&ndash;June 2013</h1>

<script type="text/javascript">

var rowConverter = function(d, i, cols) {

var row = {

date: parseTime(d.Date)

};

for (var i = 1; i < cols.length; i++) {

var col = cols[i];

if (d[cols[i]]) {

row[cols[i]] = +d[cols[i]];

} else {

row[cols[i]] = 0;

}

}

return row;

}

var w = 800;

var h = 500;

var svg = d3.select("body")

.append("svg")

.attr("width", w)

.attr("height", h);

//For converting strings to Dates

var parseTime = d3.timeParse("%Y-%m");

var dataset;

var series;

var area, xScale, yScale;

var stack = d3.stack().order(d3.stackOrderDescending); ;

d3.csv("ev\_sales\_data.csv", rowConverter, function(data) {

dataset = data;

//Set keys

var keys = dataset.columns;

keys.shift();

stack.keys(keys);

series = stack(dataset)

xScale = d3.scaleTime()

.domain([

d3.min(dataset, function(d) { return d.date; }),

d3.max(dataset, function(d) { return d.date; })

])

.range([0, w]);

yScale = d3.scaleLinear()

.domain([

0,

d3.max(dataset, function(d) {

var sum = 0;

for (var i = 0; i < keys.length; i++) {

sum += d[keys[i]];

};

return sum;

})

])

.range([h, 0]);

area = d3.area()

.x(function(d) { return xScale(d.data.date); })

.y0(function(d) { return yScale(d[0]); })

.y1(function(d) { return yScale(d[1]); });

svg.selectAll("path")

.data(series)

.enter()

.append("path")

.attr("class", "area")

.attr("d", area)

.attr("fill", function(d, i) {

return d3.schemeCategory20[i];

})

.append("title") //Make tooltip

.text(function(d) {

return d.key;

});

});

</script>

</body>