Vizualizacija podataka

Bruno Miletić

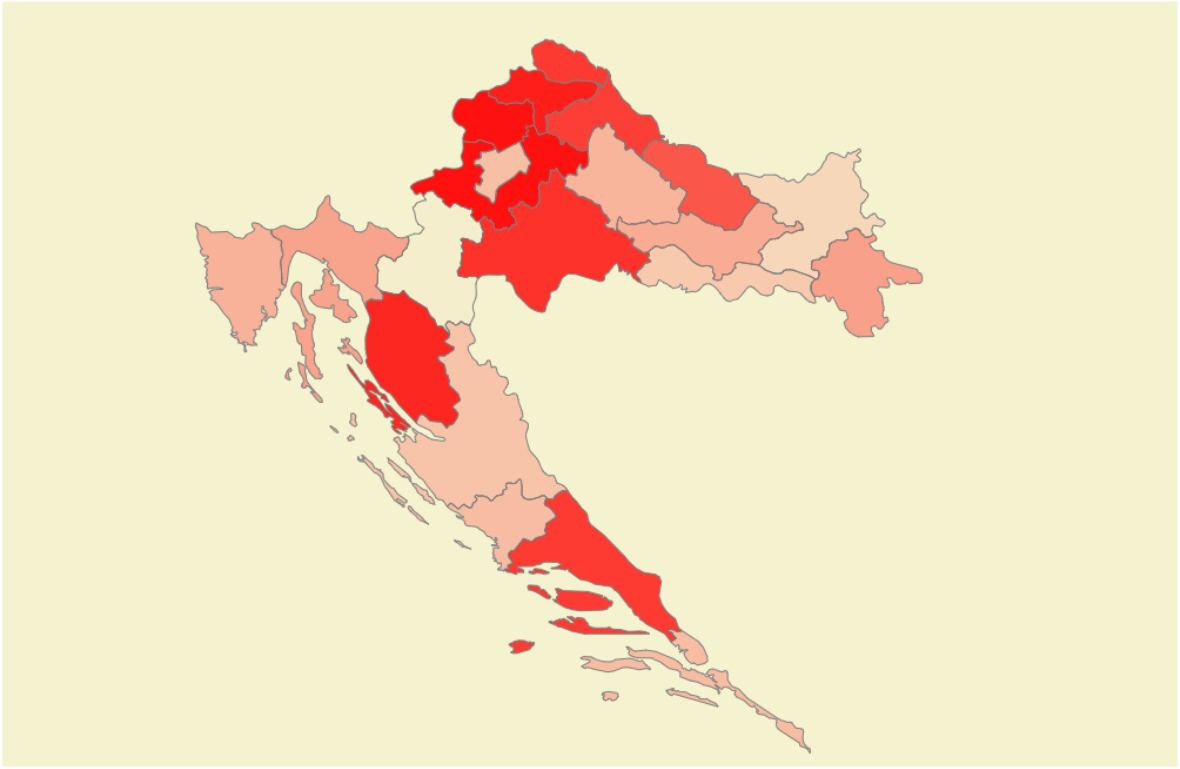
LV5

Napomena: GIF datoteke se iz nekog razloga unutar worda ne prikazuju nego stoje kao slike, pa sam unutar zip datoteke prilozio I gif datoteke za drugi I cetvrti zadatak.

1. Izraditi kartu RH te dodati element unutar kojega ce se ispisati naziv zupanije prilikom prelaska misa iznad karte.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <title>Prvi zadatak</title>
</head>

<body>
    <script>
        var width = 1000;
        var height = 650;

        var projection = d3.geo.mercator()
            .center([0, 10])
            .scale(6000)
            .translate([17600, 4500])
            .rotate([-180, 0]);

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

        var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background", "#F5F2D0");

        d3.select("body")
            .append("div")
            .attr("id", "name");

        d3.json("cro.json", function (error, cro) {
            var data = topojson.feature(cro, cro.objects.layer1);
            var states = svg.selectAll("path.county")
```

```
                .data(data.features)
                .enter()
                .append("path")
                .attr("class", "county")
                .attr("id", function (d) { return d.id; })
                .attr("d", path).style("fill", "#ff0000")
                .style("stroke", "gray")
                .style("stroke-width", 1)
                .style("fill-opacity", function () { return Math.random(); })
                .on("mouseover", function (d) {
                    console.log(d.properties.name);
                    d3.select("#name").text(d.properties.name);
                });
        });
    </script>
</body>
```

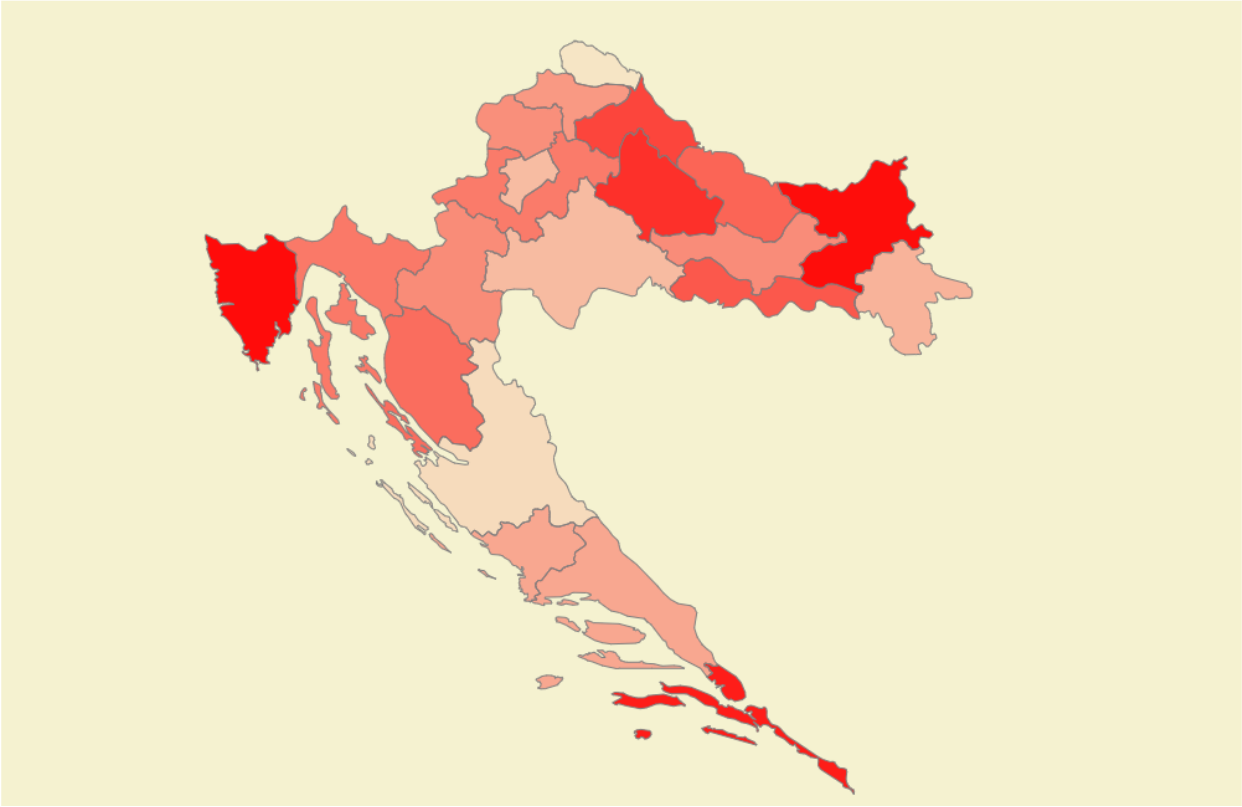


Istarska

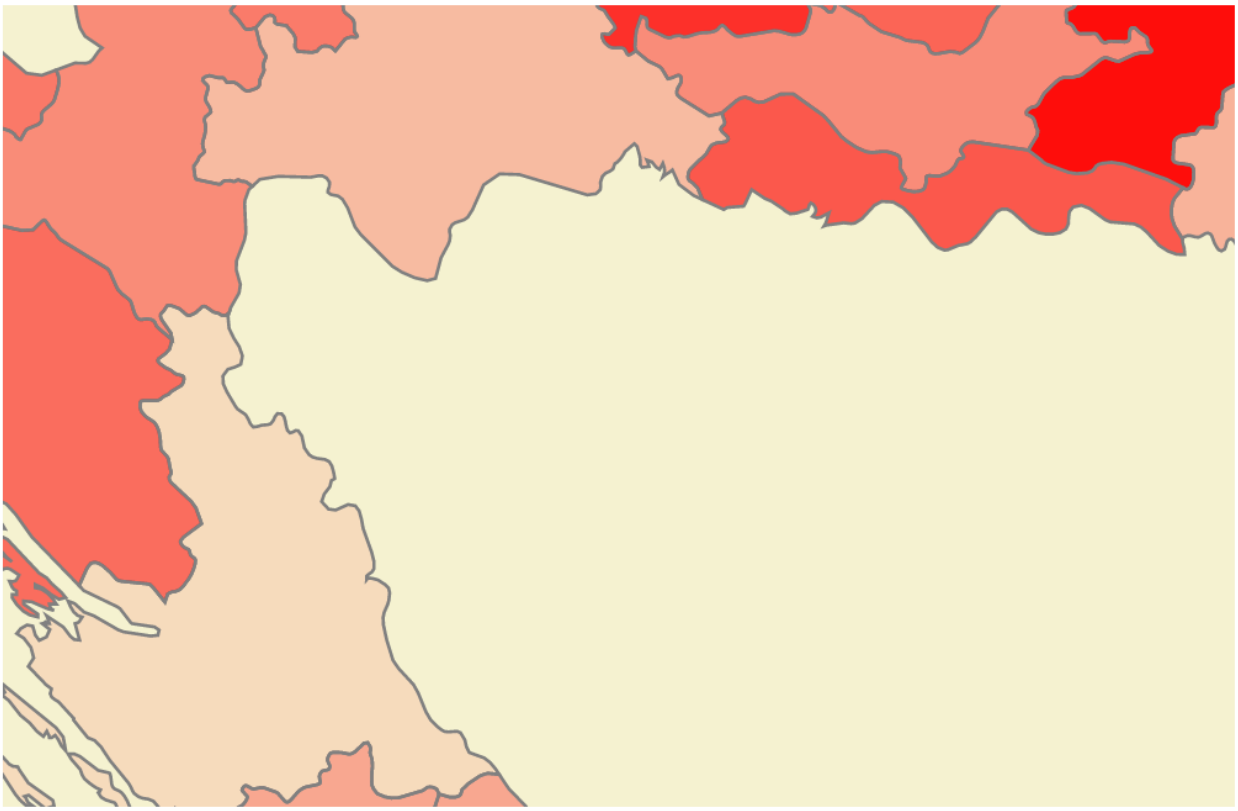
2. Prouciti primjere na internetu te omoguciti zoom-in/out ponasanje SVG elementa.

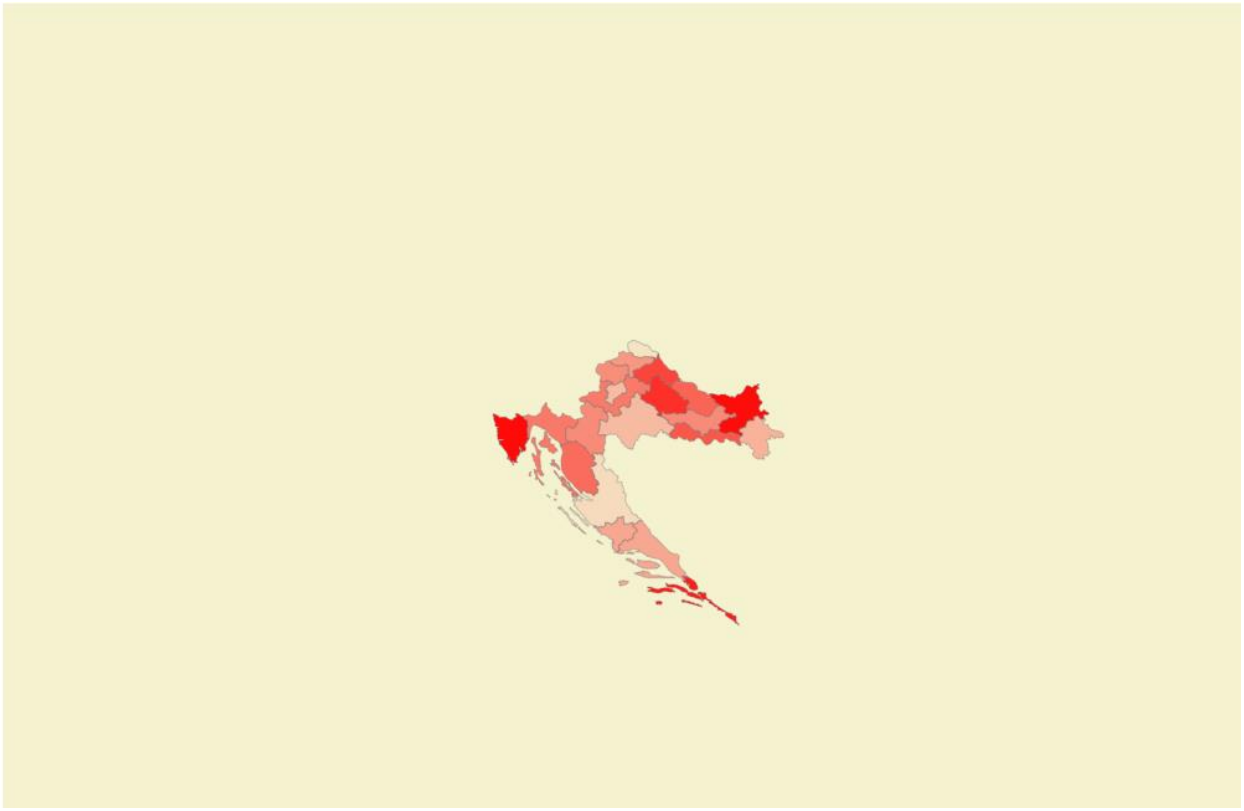
```
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <title>Drugi zadatak</title>
</head>

<body>
    <script>
        var width = 1000;
        var height = 650;

        var projection = d3.geo.mercator()
            .center([0, 10])
            .scale(6000)
            .translate([17600, 4500])
            .rotate([-180, 0]);


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

        var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background", "#F5F2D0")
            .call(d3.behavior.zoom().scaleExtent([0.3, 5])
                .on("zoom", onZoom))
            .append("g");

        d3.select("body")
            .append("div")
            .attr("id", "name");

        d3.json("cro.json", function (error, cro) {
            var data = topojson.feature(cro, cro.objects.layer1);
            var states = svg.selectAll("path.county")
                .data(data.features)
                .enter()
                .append("path")
                .attr("class", "county")
                .attr("id", function (d) { return d.id; })
```

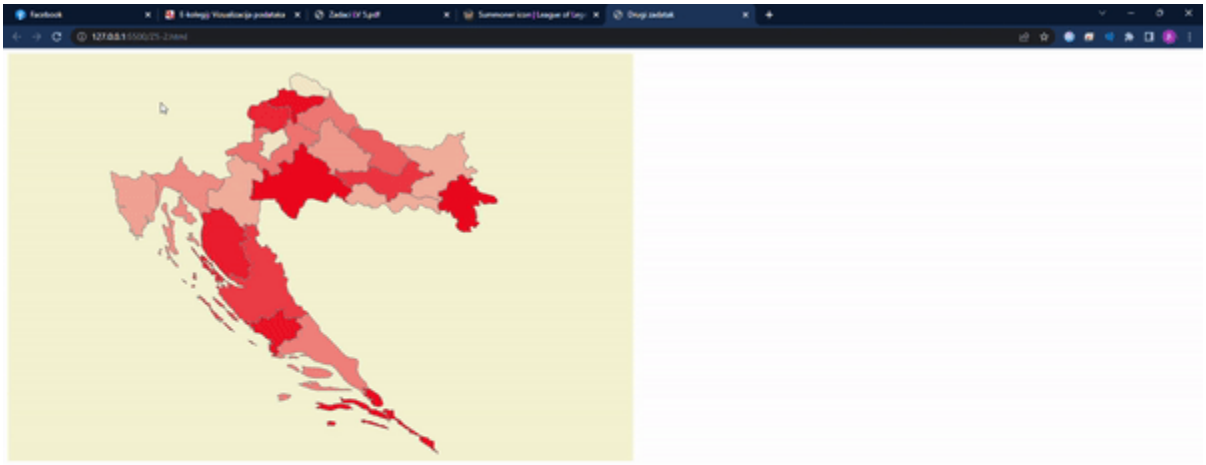
```
                .attr("d", path).style("fill", "red")
                .style("stroke", "gray")
                .style("stroke-width", 1)
                .style("fill-opacity", function () { return Math.random(); })
                .on("mouseover", function (d) {
                    console.log(d.properties.name);
                    d3.select("#name").text(d.properties.name);
                });
        });

        function onZoom() {
            svg.attr("transform", "translate (" + d3.event.translate + ") scale
(" + d3.event.scale + ")");
        }
    </script>
</body>

</html>
```



Bez zooma



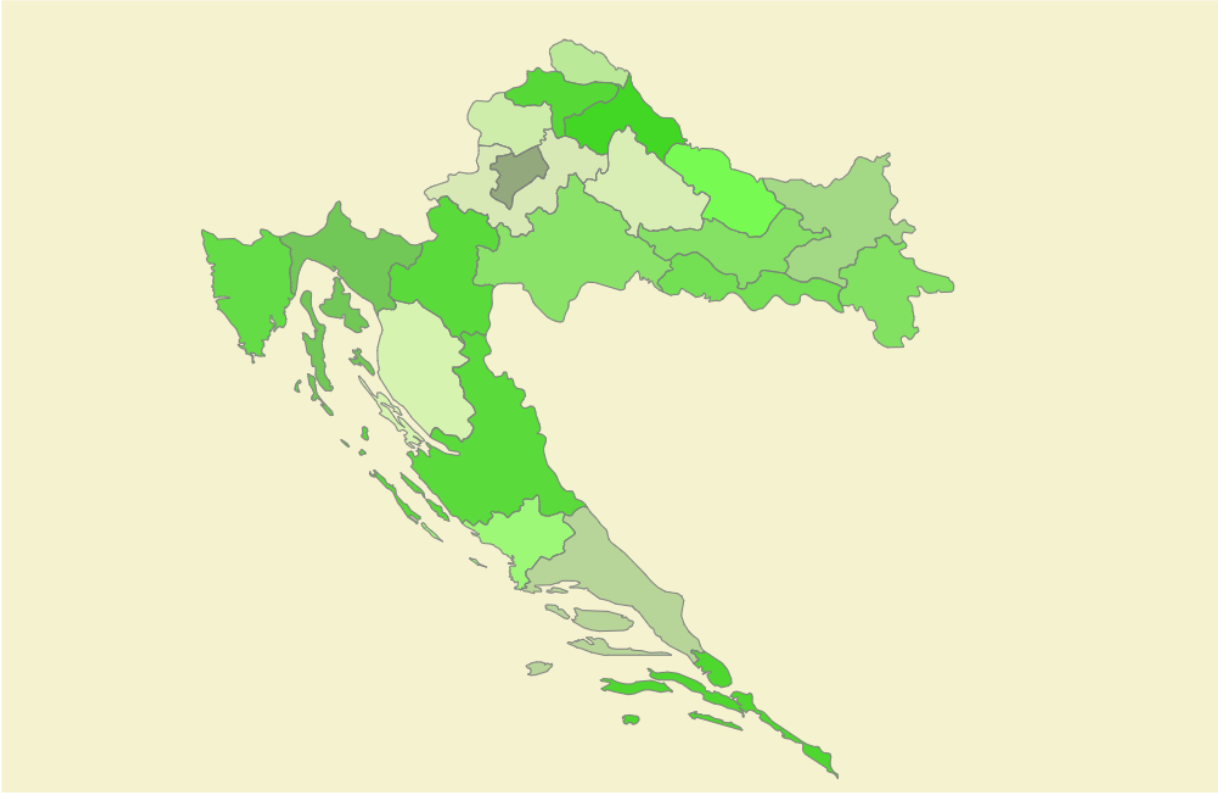




3. Pronaci podatke za RH koji su grupirani po zupanijama. Podatke povezati s kartom te omoguciti prikaz tih podataka prilikom odabira odgovarajuce zupanije na karti. Potrebno je i izraditi skalu koju ce se koristiti kod odabira boje svake od zupanija.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <title>Treci zadatak</title>
</head>

<body>
    <script>
        var width = 1000;
        var height = 650;

        var skala = d3.scale.linear()
            .domain([50000, 800000])
            .range([0, 6]);

        var colors = ["#26ff00", "#30d413", "#35b51f", "#369925", "#368729",
"#35702c", "#2f5c28"];

        var projection = d3.geo.mercator()
            .center([0, 10])
            .scale(6000)
            .translate([17600, 4500])
            .rotate([-180, 0]);

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

        var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background", "#F5F2D0")
            .call(d3.behavior.zoom().scaleExtent([0.3, 5])
                .on("zoom", onZoom))
            .append("g");

        d3.select("body")
            .append("div")
            .attr("id", "name");
```

```
        d3.select("body")
            .append("div")
            .attr("id", "info");

        d3.json("cro_data.json", function (error, cro) {
            var data = topojson.feature(cro, cro.objects.layer1);
            var states = svg.selectAll("path.county")
                .data(data.features)
                .enter()
                .append("path")
                .attr("class", "county")
                .attr("id", function (d) { return d.id; })
                .attr("d", path)
                .style("fill", function (d) {
                    var value = Math.round(skala(d.properties.population));
                    return colors[value];
                })
                .style("stroke", "gray")
                .style("stroke-width", 1)
                .style("fill-opacity", function () { return Math.random(); })
                .on("mouseover", function (d) {
                    console.log(d.properties.population);
                })
                .on("click", function (d) {
                    console.log(d.properties.population);
                    d3.select("#info").html(`Zupanija: ${d.properties.name} <br/>
Broj stanovnika: ${d.properties.population} <br/> Povrsina: ${d.properties.area}
km2 <br/> Zupan: ${d.properties.mayor}`);
                });;
        });

        function onZoom() {
            svg.attr("transform", "translate (" + d3.event.translate + ") scale
(" + d3.event.scale + ")");
        }
    </script>
</body>

</html>
```



Zupanija: Sisacko-Moslavacka
Broj stanovnika: 172439
Povrsina: 4468 km2
Zupan: Ivo Žinić

4. Odabirom svake od zupanija pribliziti zupaniju (zoom-in) i ispisati osnovne podatke o zupaniji. Nakon 10 sekundi zupaniju vratiti u prvobitan polozaj te ukloniti navedeni tekst.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <title>Cetvrti zadatak</title>
</head>

<body>
    <script>
        var width = 1000;
        var height = 650;


        var skala = d3.scale.linear()
            .domain([50000, 800000])
            .range([0, 6]);


        var colors = ["#26ff00", "#30d413", "#35b51f", "#369925", "#368729",
"#35702c", "#2f5c28"];


        var projection = d3.geo.mercator()
            .center([0, 10])
            .scale(6000)
            .translate([17600, 4500])
            .rotate([-180, 0]);


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


        var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background", "#F5F2D0")


        d3.select("body")
            .append("div")
```

```
        .attr("id", "name");


    d3.select("body")
        .append("div")
        .attr("id", "info");
    var zoom = d3.behavior.zoom()
        .translate(projection.translate())
        .scale(projection.scale())
        .scaleExtent([height, 20 * height])
        .on("zoom", onZoom);


    var g = svg.append("g")
        .call(zoom);


    d3.json("cro_data.json", function (error, cro) {
        var data = topojson.feature(cro, cro.objects.layer1);
        var states = g.selectAll("path.county")
            .data(data.features)
            .enter()
            .append("path")
            .attr("class", "county")
            .attr("id", function (d) { return d.id; })
            .attr("d", path)
            .style("fill", function (d) {
                var value = Math.round(skala(d.properties.population));
                return colors[value];
            })
            .style("stroke", "gray")
            .style("stroke-width", 1)
            .style("fill-opacity", function () { return Math.random(); })
            .on("mouseover", function (d) {
                console.log(d.properties.population);
            })
            .on("click", onClick);
    });


    function onZoom() {
        projection.translate(d3.event.translate).scale(d3.event.scale);
        g.selectAll("path").attr("d", path);
    }
```

```
        function onClick(d) {
            var a, b, c;
            var centroid = path.centroid(d);
            a = centroid[0];
            b = centroid[1];
            c = 4;
            centered = d;


            g.selectAll("path")
                .classed("active", centered && function (d) { return d ===
centered; });


            d3.select("#info").html(`Zupanija: ${d.properties.name} <br/> Broj
stanovnika: ${d.properties.population} <br/> Povrsina: ${d.properties.area} km2
<br/> Zupan: ${d.properties.mayor}`);


            g.transition()
                .duration(3000)
                .attr("transform", `translate(${width / 2}, ${height / 2})
scale(${c}) translate(${-a}, ${-b})`)
                .style("stroke-width", 1.5 / c + "px");


            a = width / 2;
            b = height / 2;
            c = 1;
            centered = null;


            g.transition()
                .delay(200000)
                .duration(1000)
                .attr("transform", `translate(${width / 2}, ${height / 2})
scale(${c}) translate(${-a}, ${-b})`)
                .style("stroke-width", 2 / c + "px")
                .each('end', function () {
                    d3.select("#info")
                        .html("");
                }
                );
        }
    </script>
</body>
```

```
</html>
```