

0. Ejemplos

US Homicides 2010 – Jérôme Cukier

Population Choropleth – Mike Bostock

Situación de los supermercados Walmart –
Zachari Forest (indiemaps)

Plant Hardiness Zones – Mike Bostock

Map Projection Transitions – Jason Davies

Un Mundo de Migrantes – Outliers Collective

1. Datos

Cartográficos

polígonos que forman las CCAA
tiles

Cuantitativos asociados a Geo

población

incendios

tweets geolocalizados

1. Datos

¿Dónde conseguir datos cartográficos?

Natural Earth

<http://www.naturalearthdata.com/>

Spatial Data Repository

<http://spatialdata.dhsprogram.com>

UCLA Spatial Data Repository

<http://gis.ats.ucla.edu>

2. ShapeFile

Formato de archivo informático propietario de **datos espaciales** desarrollado por la compañía ESRI

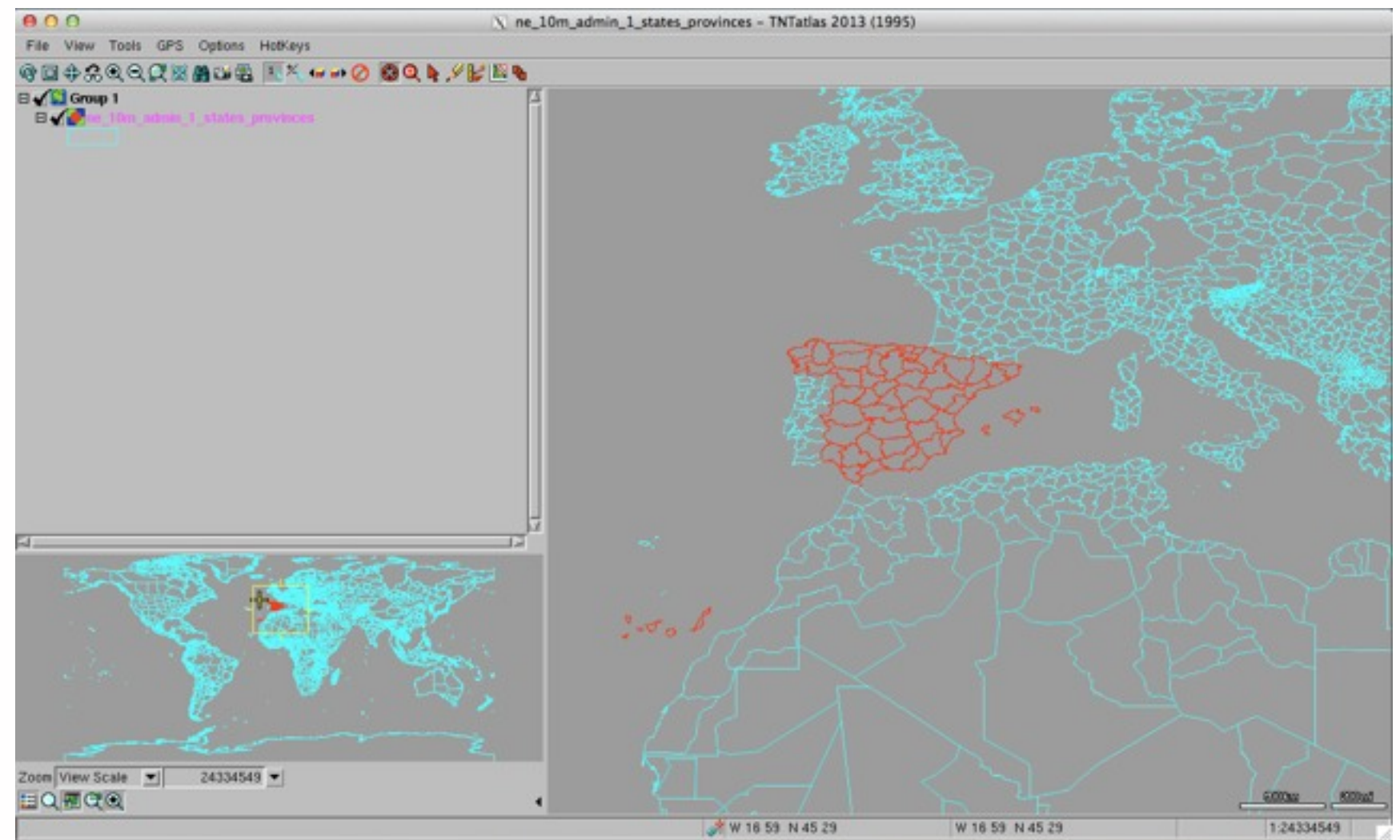
Actualmente se ha convertido en formato estándar de facto

Permite almacenar atributos asociados a las diferentes geometrías

<http://es.wikipedia.org/wiki/Shapefile>

2. ShapeFile

Visor SHP TNTAtlas



<http://www.microimages.com/products/tntatlas/>

3. GeoJSON y TopoJSON

GeoJSON

<http://en.wikipedia.org/wiki/GeoJSON>

TopoJSON

<https://github.com/mbostock/topojson>

De SHP a TopoJSON

Descargar los **SHP**:

Natural Earth, Admin 1 – States & Provinces

Natural Earth, Populated Places

Utilizando **ogr2ogr** y **topojson** generar
TopoJSON:

```
ogr2ogr -f GeoJSON -where "iso_a2 IN ('ES', 'PT', 'FR', 'MA')" subunits.json  
ne_10m_admin_1_states_provinces.shp
```

```
ogr2ogr -f GeoJSON -where "ISO_A2 = 'ES' AND SCALERANK < 8" places.json  
ne_10m_admin_1_states_provinces.shp
```

```
topojson --id-property adm1_code -p name=name -p name -p  
population=POP_MAX -p population -o es.json subunits.json places.json
```

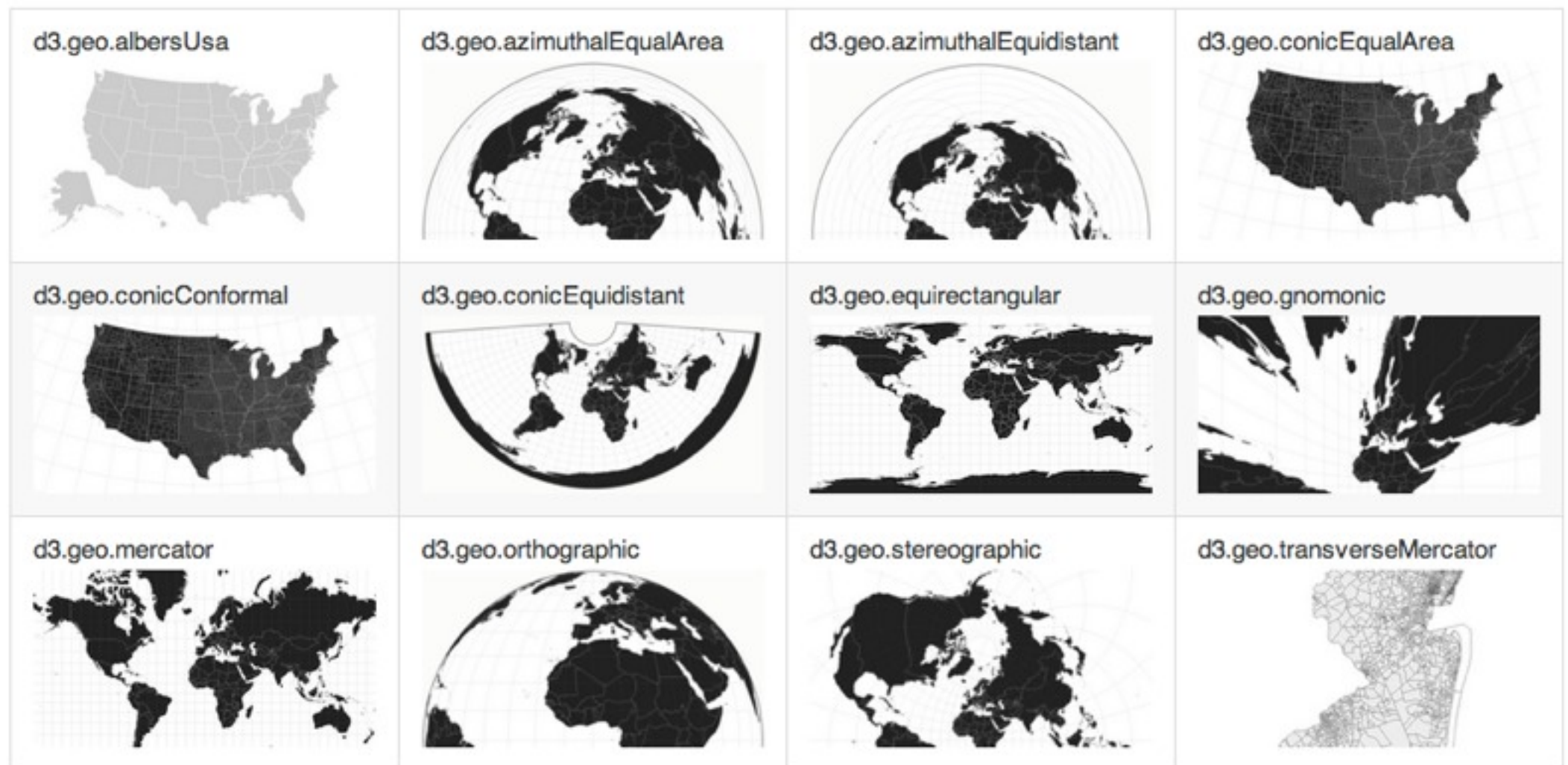
Para renderizar cartografías necesitaremos una **proyección** y un **generador de paths**

Una **proyección** proyecta coordenadas esféricas al plano cartesiano

Un **generador de paths** toma la geometría 2D proyectada y la formatea adecuadamente a SVG

Proyecciones en d3.geo

```
var mercatorProjection = d3.geo.mercator();  
/* Albers centrada sobre el Reino Unido */  
var albersProjections = d3.geo.albers()  
    .center([0, 55.4])  
    .rotate([4.4, 0])  
    .parallels([50, 60])  
    .scale(6000);
```



3. d3.geo

<https://github.com/mbostock/d3/wiki/Geo-Projections>

Obtener las coordenadas de un punto en una proyección

```
var projection = d3.geo.mercator();  
var punto = {  
  "latitude": 40.4165000,  
  "longitude": -3.7025600  
};  
var x = projection([punto.longitude, punto.latitude])[0];  
var y = projection([punto.longitude, punto.latitude])[1];
```

[ejemplo1.html](#)

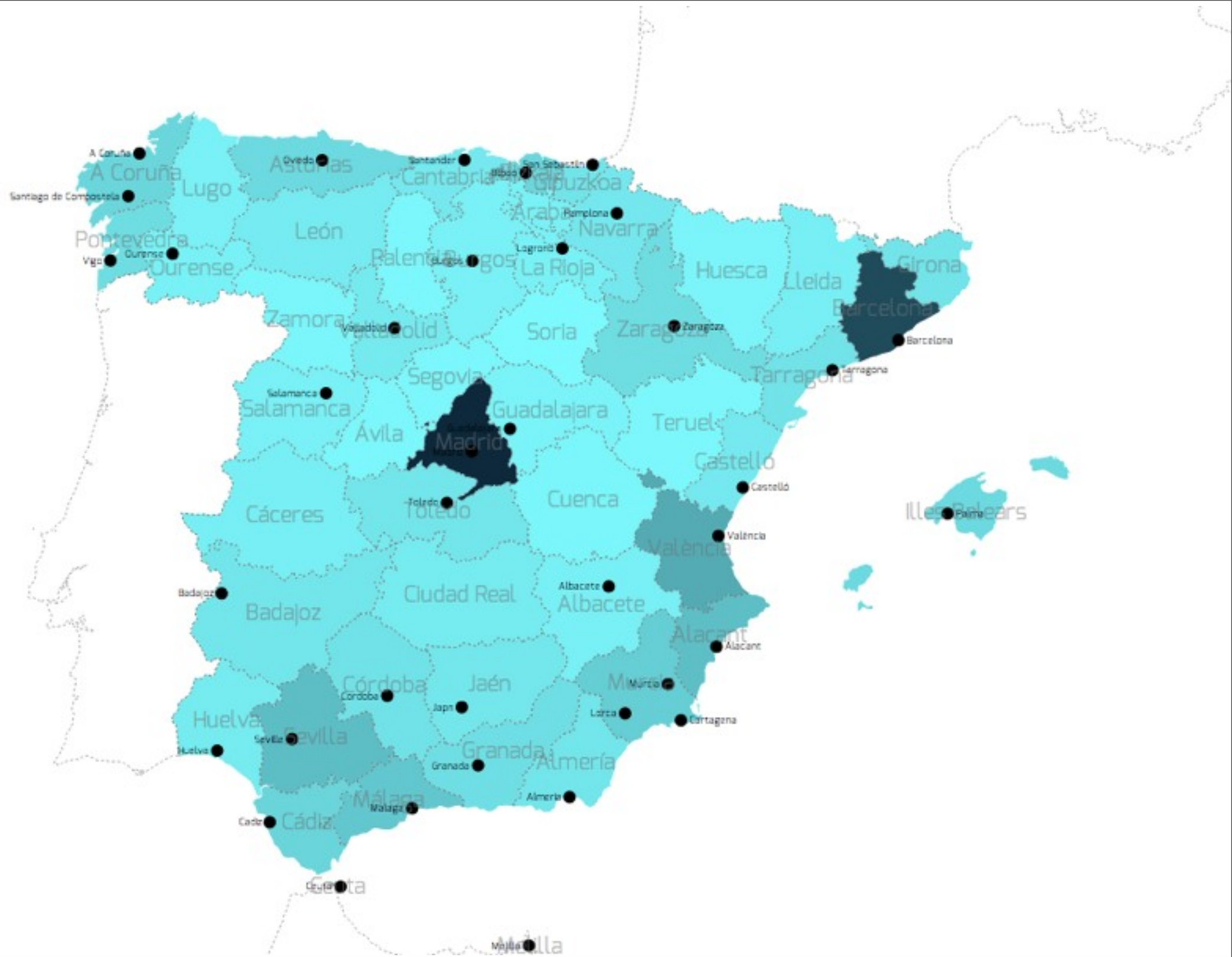
Generadores de paths en d3.geo

<https://github.com/mbostock/d3/wiki/Geo-Paths>

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", 1200)
    .attr("height", 700);
d3.json("es.json", function (error, es) {
    var projection = d3.geo.mercator();
    var generador = d3.geo.path()
        .projection(projection);
    svg.selectAll("path")
        .data(topojson.feature(es, es.objects.subunits).features)
        .enter()
        .append("path")
        .attr("d", function (d) { return generador(d); });
});
```

ejemplo2.html
ejemplo3.html

4. Ejercicio guiado



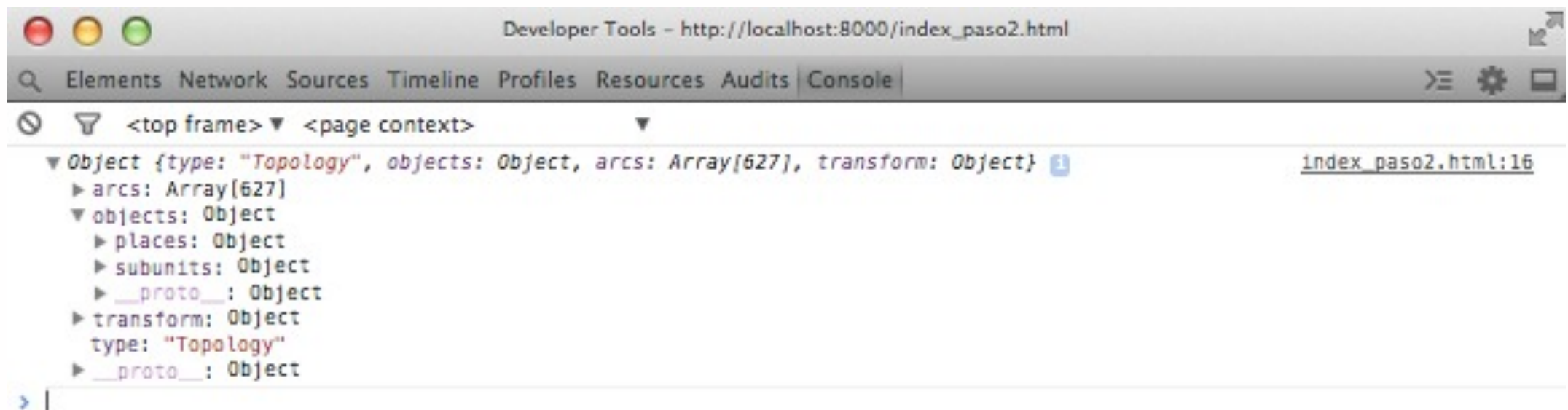
Vamos a dibujar el mapa de España con sus provincias y a colorearlo según diferentes estadísticas del INE

4. Ejercicio guiado

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      /* Código CSS */
    </style>
  </head>
  <body>
    <script src="http://d3js.org/d3.v3.min.js"></script>
    <script src="http://d3js.org/queue.v1.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <script>
      /* Código JavaScript */
    </script>
  </body>
</html>
```

PASO 1: Fichero HTML de partida


4. Ejercicio guiado



```
<script>
  /* Código JavaScript */
  d3.json("es.json", function(error, es) {
    console.log(es);
  });
</script>
```

PASO 2: Cargar los datos

4. Ejercicio guiado



```
<script>
  /* Código JavaScript */
  var width = 1200,
      height = 700;
  var svg = d3.select("body")
    .append("svg")
    .attr("width", width)
    .attr("height", height);
  d3.json("es.json", function (error, es) {
    svg.append("path")
      .datum(topojson.feature(es, es.objects.subunits))
      .attr("d", d3.geo.path().projection(d3.geo.mercator()));
  });
</script>
```

PASO 3: Dibujar los polígonos

4. Ejercicio guiado



```
d3.json("es.json", function (error, es) {  
  var subunits = topojson.feature(es, es.objects.subunits),  
  projection = d3.geo.albers()  
    .center([0, 40.23])  
    .rotate([3.43, 0])  
    .parallels([40, 60])  
    .scale(4000)  
    .translate([width / 2, height / 2]),  
  path = d3.geo.path()  
    .projection(projection);  
  svg.append("path")  
    .datum(subunits)  
    .attr("d", path);  
});
```

PASO 4: Ajustar la proyección

4. Ejercicio guiado



```
<style>
  /* Código CSS */
  .subunit.ESP { fill: #ddc; }
  .subunit.FRA, .subunit.MAR, .subunit.PRT { display: none; }
</style>
...
svg.append("path")
— .datum(subunits)
— .attr("d", path);
svg.selectAll(".subunit")
  .data(subunits.features)
  .enter()
  .append("path")
  .attr("class", function (d) { return "subunit " + d.id.slice(0, 3); })
  .attr("d", path);
```

PASO 5: Aplicando estilos a los polígonos

4. Ejercicio guiado

```
<style>
  ...
  .subunit-boundary {
    fill: none;
    stroke: #777;
    stroke-dasharray: 2,2;
    stroke-linejoin: round;
  }
  .subunit-boundary.OTHER { stroke: #aaa; }
</style>
...
svg.append("path")
  .datum(topojson.mesh(es, es.objects.subunits, function(a, b) { return a !== b && a.id
=== "ESP"; }))
  .attr("d", path)
  .attr("class", "subunit-boundary");
svg.append("path")
  .datum(topojson.mesh(es, es.objects.subunits, function(a, b) { return a === b && a.id !
=== "ESP"; }))
  .attr("d", path)
  .attr("class", "subunit-boundary OTHER");
```

PASO 6: Dibujar los bordes



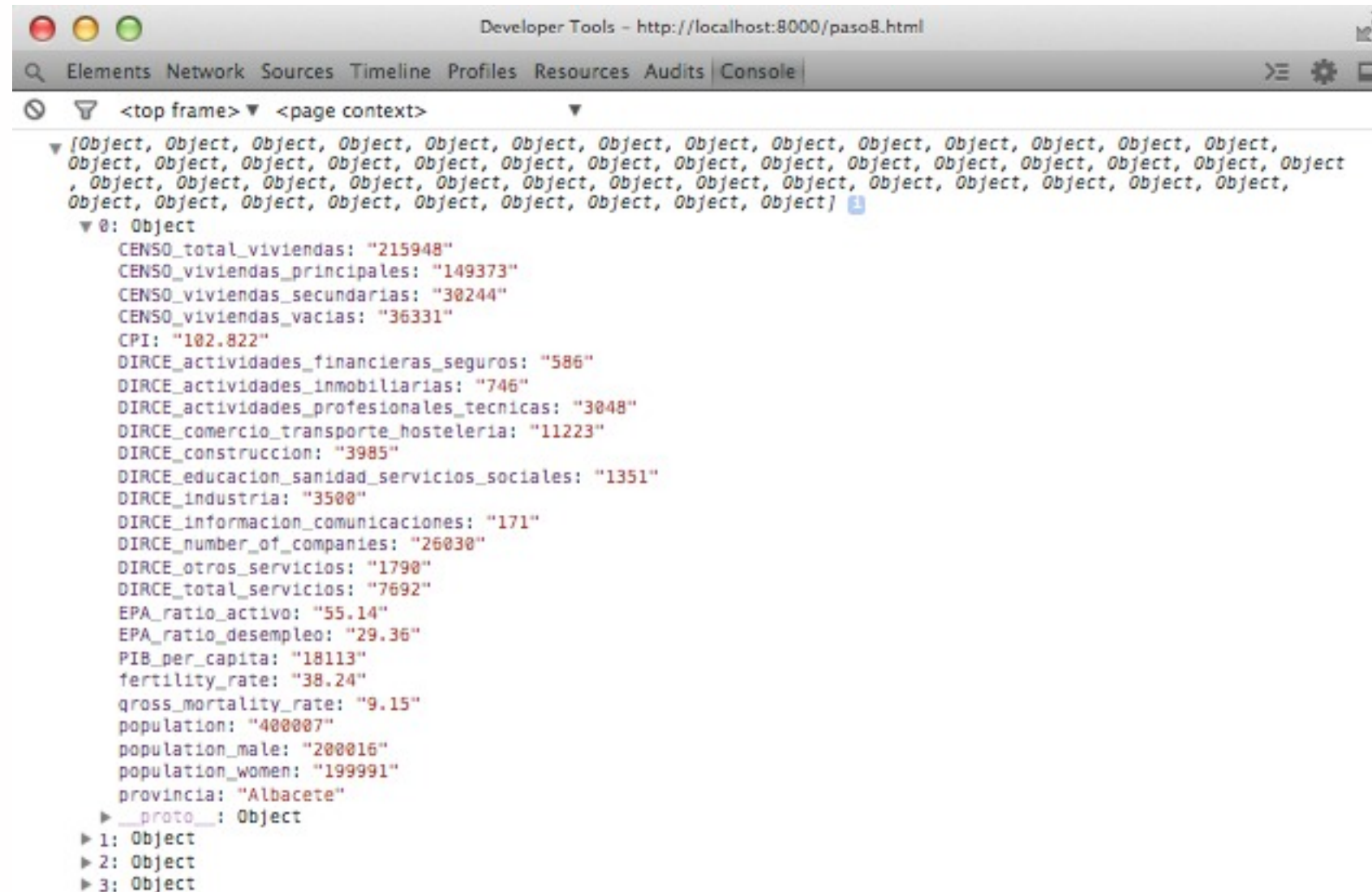
4. Ejercicio guiado

```
svg.append("path")
  .datum(topojson.feature(es, es.objects.places))
  .attr("d", path)
  .attr("class", "place");
svg.selectAll(".place-label")
  .data(topojson.feature(es, es.objects.places).features)
  .enter()
  .append("text")
  .attr("class", "place-label")
  .attr("transform", function(d) { return "translate(" + projection(d.geometry.coordinates)
+ ")"; })
  .attr("dy", ".35em")
  .text(function(d) { return d.properties.name; });
svg.selectAll(".place-label")
  .attr("x", function(d) { return d.geometry.coordinates[0] > -1 ? 6 : -6; })
  .style("text-anchor", function(d) { return d.geometry.coordinates[0] > -1 ? "start" :
"end"; });
```



PASO 7: Mostrar las ciudades más pobladas

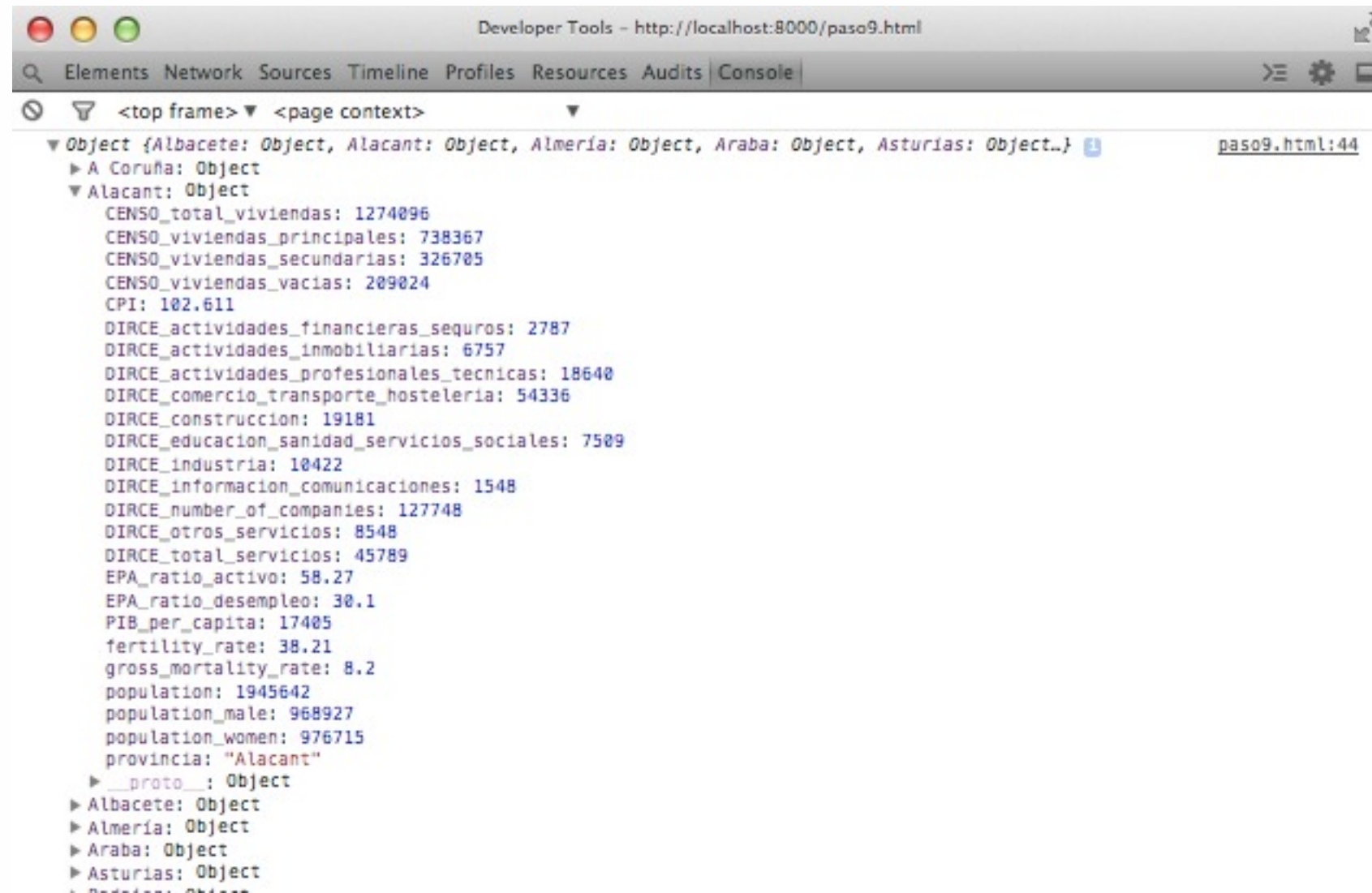
4. Ejercicio guiado



```
queue().defer(d3.json, "json/es.json")
      .defer(d3.tsv, "tsv/ine.tsv")
      .await(ready);
function ready (error, es, ine) {
  console.log(ine);
  /* Todo el código que teníamos dentro de la llamada a d3.json */
};
```

PASO 8: Cargar el TSV con los datos del INE

4. Ejercicio guiado



```
/* Al principio de la función ready */  
var provincesData = {};  
ine.forEach(function (d, i) {  
    Object.keys(d).forEach(function (k) {  
        var v = parseFloat(d[k]);  
        if (!isNaN(v)) { d[k] = v; }  
    });  
    provincesData[d.provincia] = d;  
});  
console.log(provincesData);
```

PASO 9: Estructura auxiliar con los datos del INE

4. Ejercicio guiado

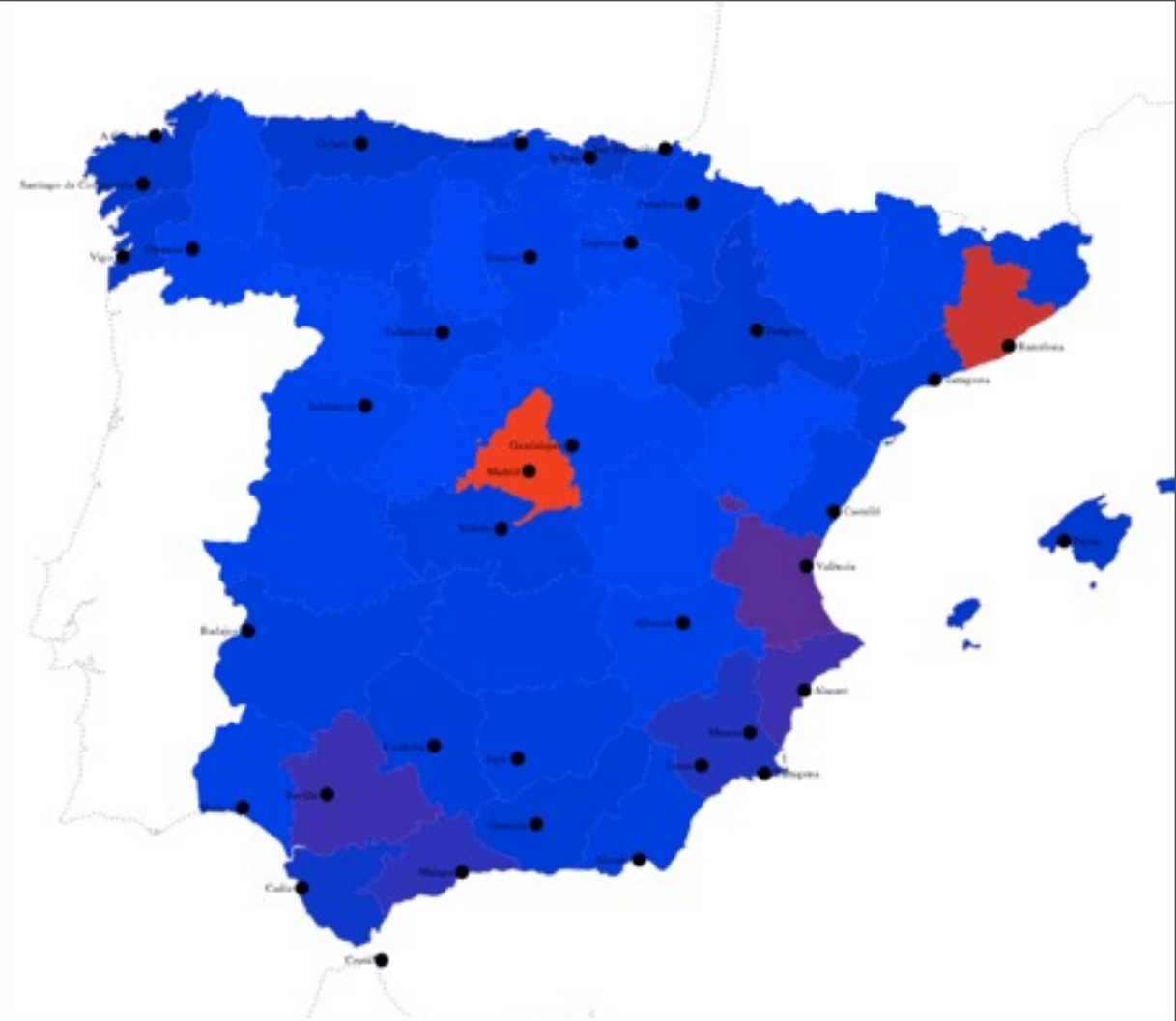


```
/* Primero definimos una escala de color */  
var colorScale = d3.scale  
    .linear()  
    .domain([  
        d3.min(ine, function (d) { return d.population; }),  
        d3.max(ine, function (d) { return d.population; }),  
    ])  
    .range(["#0000ff", "#ff0000"]);
```

PASO 10: Colorear las provincias según su población

4. Ejercicio guiado

```
/*
Debemos aplicar el color a los polígonos
de las provincias
*/
svg.selectAll(".subunit")
  .data(subunits.features)
  .enter()
  .append("path")
  .attr("class", function (d) { return "subunit " + d.id.slice(0, 3); })
  .attr("d", path)
  .style("fill", function (d) {
    if ("properties" in d){
      if ("name" in d.properties) {
        if (d.properties.name in provincesData) {
          return colorScale(provincesData[d.properties.name].population);
        }
      }
    }
    return "none";
  });
```



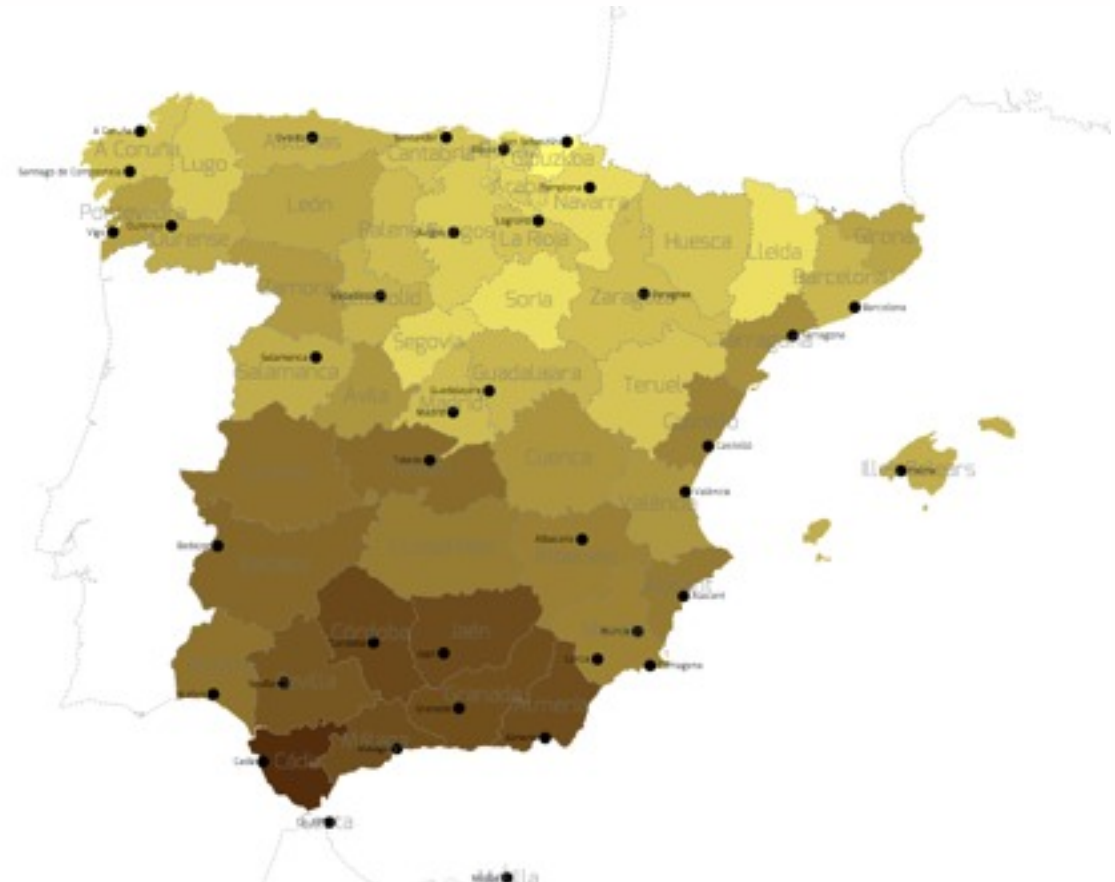
PASO 10: Colorear las provincias según su población

4. Ejercicio guiado

Índice de mortalidad

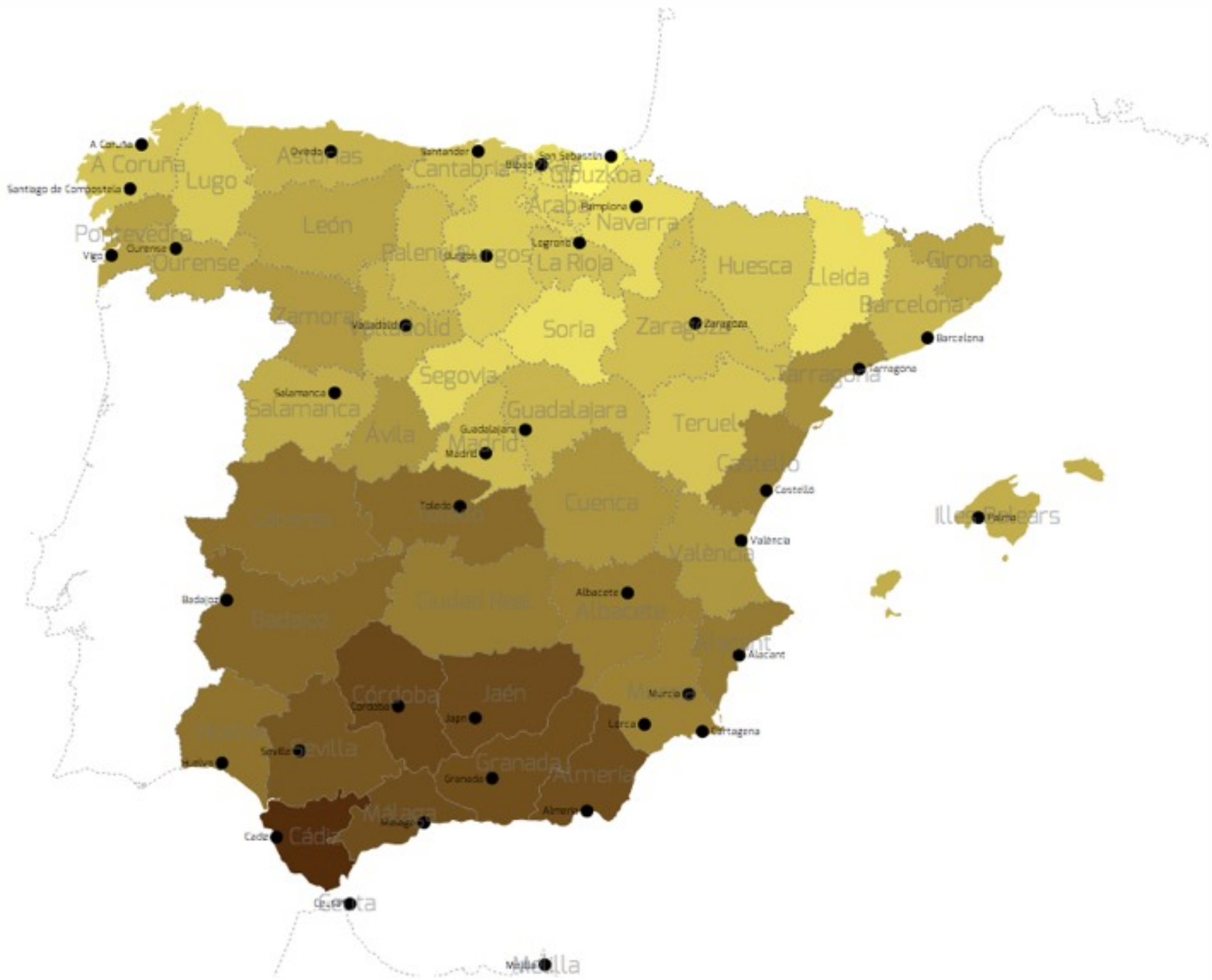


Ratio de desempleo



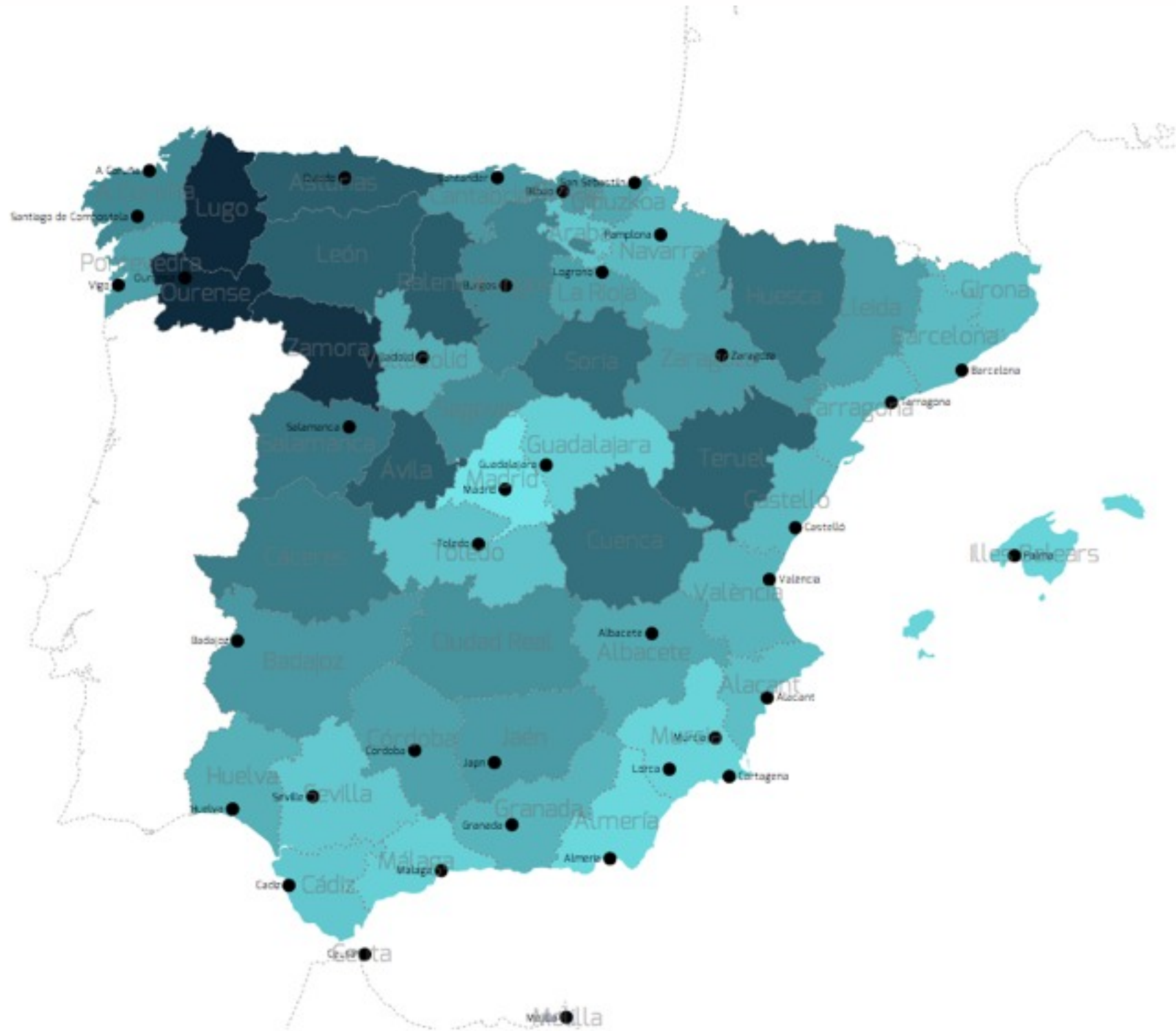
**Colorear las provincias
utilizando otras estadísticas
del INE**

4. Ejercicio guiado



Añadir zoom al mapa

4. Ejercicio guiado



Añadir tooltips al mapa que muestren el valor de la estadística actual para la provincia señalada

5. Ejercicio propuesto

Datos:

Ubicación de aeropuertos en Estados Unidos

Densidad poblacional en Estados Unidos

Objetivo:

