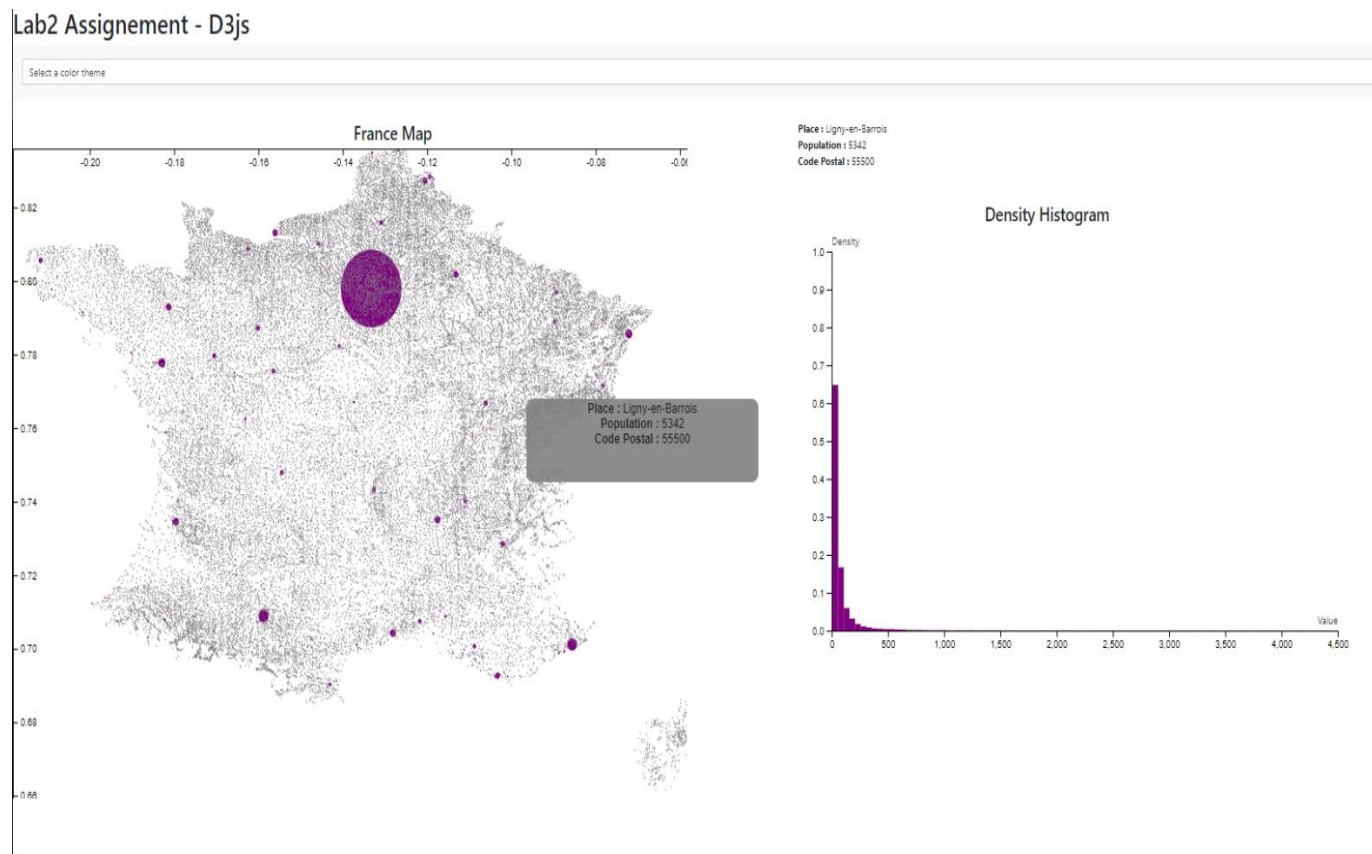


- JavaScript
- The HTML Document Object Model (DOM)
- The SVG vector graphics format
- Cascading Style Sheets (CSS)

Below the visualisation we created using D3js :



As we can see , the dashboard contains the following components :

- France Map that shows the population and density per postal code , with a tooltip containing the place , the postal code and the population
- A span that shows the same information displayed in the tooltip
- A density histogram
- A selection menu through which we can change the color of the map and the histogram

In the next section, we will explain briefly how we the implementation of these elements was done !

II. Code Organization

1- France Map

In order to draw the map , we implemented the function `draw ()` in javascript . This function takes as an argument the color that the user selects and draw the map

```
function draw(selectedcolor)
```

In this function , we add the population as circles whose radius is proportional to the population value , and the density as `rect` whose color intensity is proportional to the density value :

```
svg.selectAll("circle")
```

```
svg.selectAll("rect")
```

We add then « `mouseover` » and « `mouseout` » events to all rectangles in order to display and hide the tooltip while hovering with the mouse

```
.on("mouseover", function(d) {
```

```
.on("mouseout", function(d) {
```

We add then the y and x axis

```
svg.append("g")  
.attr("class", "x axis")
```

```
svg.append("g")  
.attr("class", "y axis")
```

2- Density Histogram

For the histogram , we implemented the function « `drawhist` »

```
function drawhist(selectedcolor){
```

We add the x and y axis as we did for the map , then we decalre a d3 histogram component with the density as a variable:

```
let histogram = d3.histogram()  
    .value(d=>d.density)  
    .domain(x.domain())  
    .thresholds(x.ticks(100));
```

We compute the bins with the histogram variable

```
let bins = histogram(dataset);
```

after adding the labels of the axis , we generate the histogram with the following function :

```
svghist.insert("g", "*")  
    .attr("fill", "#bbb")  
    .selectAll("rect")  
    .data(bins)  
    .enter().append("rect")  
    .attr("x", function(d) { return x(d.x0) + 1; })  
    .attr("y", function(d) { return y(d.length / n); })  
    .attr("width", function(d) { return x(d.x1) - x(d.x0); })  
    .attr("height", function(d) { return y(0) - y(d.length / n); })  
    .style("fill", selectedcolor);
```

3- Color theme

In order to change the color of the graphs , we added a dropdown menu containing three colors , when the user select a color , the function « update » is called . This function calls « drawhist » and « drawmap » with the selected color

```
////////// update color theme //////////  
function update(selectedGroup) {  
  
    // Create new data with the selection?  
    //var dataFilter = data.map(function(d){return {time: d.time,  
value:d[selectedGroup]} })  
  
    // Give these new data to update line  
    console.log(selectedGroup);  
    svghist.selectAll('rect').remove();  
    drawhist.call(this,selectedGroup);  
}
```

```
svg.selectAll("circle").remove()  
draw.call(this, selectedGroup);  
  
}
```

For the html and css part , it was pretty simple , we create for each graph a html component and the select it with javascript and append whatever we want (span , title , label ,...)

For the design, we using Bootstrap template for the nav header and the menu selection .