

## Serie de travaux pratiques n° 2

### Exercice 1.

Ecrire un code en D3 pour visualiser en scatterplot les valeurs des données (place, population, longitude, latitude) du fichier us-cities.csv.

#### 1. Lecture du fichier CSV :

```
const scatterData = (callback) => {  
  d3.csv("us-cities.csv").then((data) => {  
    let dataset = data.map((d) => {  
      return {  
        place: d.place,  
        population: parseInt(d.population),  
        latitude: parseInt(d.lat),  
        longitude: parseInt(d.lon)  
      };  
    });  
    callback(dataset);  
  });  
};
```

#### 2. Afficher les données sous forme de paragraphes :

```
scatterData((data) => {  
  d3.select("body").selectAll("p")  
    .data(data)  
    .enter()  
    .append("p")  
    .text((d) => {  
      return d.place+" "+d.population+" "+d.latitude+" "+d.longitude;  
    });  
});
```

#### 3. Création d'un SVG :

```
let h = 500;  
let w = 1000;  
let svg1 = d3.select("body")  
  .append("svg")  
  .attr("height", h)  
  .attr("width", w)
```

4. Définir les cercles qui représente le nuage de points a l'intérieur du SVG :

```
svg.selectAll("circle")
  .data(data)
  .enter()
  .append("circle")
  .attr("cx", (d) => {return d.longitude; })
  .attr("cy", (d) => {return d.latitude; })
  .attr("r", (d) => {return d.population; })
  .attr("fill", (d) => {return d.population; })
```

5. Afin de bien représenter les points a l'intérieur du SVG, on doit changer l'échelle de nos données selon les dimensions du SVG.

- ✓ L'échelle de l'axe x change selon le domaine de l'attribut longitude :

```
let padding = 30;
let xDomainMin = d3.min(data, (d) => {return d.longitude;});
let xDomainMax = d3.max(data, (d) => {return d.longitude;});
let xScale = d3.scaleLinear()
  .domain([xDomainMin-10, xDomainMax+10])
  .range([padding, w - padding]);
```

- ✓ L'échelle de l'axe y change selon le domaine de l'attribut latitude :

```
let yDomainMin = d3.min(data, (d) => {return d.latitude;});
let yDomainMax = d3.max(data, (d) => {return d.latitude;});
let yScale = d3.scaleLinear()
  .domain([yDomainMin-10, yDomainMax+10])
  .range([h - padding, padding]);
```

- ✓ La taille et la couleur des points du scatter plot changent selon l'attribut population.

```
let rDomain = d3.max(data, (d) => {return d.population;});
let rScale = d3.scaleLinear()
  .domain([0, rDomain])
  .range([0, rDomain/100000]);

let cScale = d3.scaleSequential(d3.interpolate("purple", "yellow"))
  .domain([0, rDomain]);
```

6. Redéfinir les cercles qui représente le nuage de points a l'intérieur du SVG en utilisant les échelles définies précédemment :

```
svg.selectAll("circle")
  .data(data)
  .enter()
  .append("circle")
  .attr("cx", (d) => {return xScale(d.longitude); })
  .attr("cy", (d) => {return yScale(d.latitude); })
```

```
.attr("r", (d) => {return rScale(d.population); })  
.attr("fill", (d) => {return cScale(d.population); })
```

7. Ajouter du texte (l'attribut place) sur le nuage de points :

```
svg.selectAll("text")  
  .data(data)  
  .enter()  
  .append("text")  
  .text((d) => {return d.place;})  
  .attr("x", (d) => {return xScale(d.longitude);})  
  .attr("y", (d) => {return yScale(d.latitude);})  
  .attr("fill", "black")  
  .attr("font-size", "8px")
```

8. Définir les axes x et y :

```
let xAxis = d3.axisBottom(xScale).ticks(5);  
let yAxis = d3.axisLeft(yScale).ticks(5);
```

9. Ajouter les axes x et y au SVG :

```
svg.append("g")  
  .attr("class", "axis")  
  .attr("transform", "translate(0, "+ (h - padding) + ")")  
  .call(xAxis);  
  
svg.append("g")  
  .attr("class", "axis")  
  .attr("transform", "translate("+ padding + ", 0)")  
  .call(yAxis);
```

10. Comme l'élément g (group) du SVG est défini par la classe axis, on doit donner les caractéristiques de cette classe en CSS :

```
<style type="text/css">  
  .axis path,  
  .axis line {  
    fill: none;  
    stroke: black;  
  }  
  .axis text {  
    font-family: sans-serif;  
    font-size: 8px;  
  }  
</style>
```

**Exercice 2.**

Ecrire un code en D3 pour visualiser en histogramme les valeurs des données du fichier us-productivity.csv.