

Ajout d'un nouveau type de graphique

La première tâche nous ayant été confié fut de rajouter un nouveau type de graphique au projet déjà existant. Ce nouveau graphique a été un nuage de points. Pour cela, on a d'abord mise en place le nuage de point dans sur un fichier a part mais se basant sur le même format que le projet. Afin de pouvoir afficher le nuage de point sur la page web, il a fallu effectuer plusieurs étapes.

La première étape a été de mettre en place les échelles des différents axes pour le nuage de points. Pour cela on a utilisé une fonction de la bibliothèque D3 permettant de créer une échelle en fournissant l'étendue des valeurs, c'est-à-dire la valeur maximum et minimum des valeurs x et y présent dans l'ensemble des points afficher dans le nuage, et la taille souhaitait pour l'affichage des axes. Une fois ces échelles construites, il est nécessaire de créer les axes via la fonction `axis()` de D3.js permettant de créer un objet SVG comportant tous les éléments nécessaire à l'affiche dans la page HTML, c'est-à-dire la balise `PATH` permettant de dessiner le trait de l'axe et les balises affichant les valeurs à côté de l'axe. Afin de créer cet objet il faut lors de l'appel de la fonction fournir l'échelle correspondante afin de pouvoir afficher les bonnes valeurs pour l'axe et définir sa taille. Il est aussi possible de définir l'épaisseur souhaitait pour l'axe, la position de l'axe afin de choisir si l'axe doit être l'axe x ou y, et il est aussi possible de définir le nombre de valeurs maximal à afficher sur l'axe.

```
var echelleY = d3.scale.linear()
    .domain([nuageRange[1][0],nuageRange[1][1]])
    .range([maxy,miny]);

var echelleX = d3.scale.linear()
    .domain([nuageRange[0][0],nuageRange[0][1]])
    .range([0,axe]);

var xAxe = d3.svg.axis()
    .scale(echelleX)
    .tickSize(0.5*(cellSize/100))
    .ticks(4)
    .orient("bottom");

var yAxe = d3.svg.axis()
    .scale(echelleY)
    .tickSize(0.5*(cellSize/100))
    .ticks(4)
    .orient("left");
```

Une fois les axes créés, il a fallu mettre en place pour chaque cellule une balise qui recevra les différents éléments nécessaire à l'affichage du graphique dans la cellule. Pour cela, on ajoute à la zone dédié au SVG (balise `SVG` : zone comportant les cellules et les graphiques) autant de balise `g` (une balise permettant de regrouper différents objets tel que les axes et les points du nuage) qu'il y a de cellule présente. Pour chacune de ces balises, on lui définit ses dimensions afin de s'afficher correctement dans la cellule. On lui fournit aussi un ID afin de

pouvoir plus facilement accéder aux balises de chaque cellule lors de l'ajout des axes et des points du nuage. En plus de cela il faut aussi effectuer une translation à ces balises afin de pouvoir faire en sorte que les balises soient bien affichées à leur cellule correspondante. Pour cela lors de la création des cellules, un tableau comportant le centre de chaque cellule (c'est-à-dire son positionnement x et y dans la balise SVG) est alimenté. On ajoute alors l'attribut transform (attribut permettant de manipuler des éléments HTML dans l'espace 2D ou 3D) avec comme valeur translate et les valeurs x et y de chaque cellule à chaque balise g.

```
for (var indice = 0; indice < nbRows*nbColumns; indice++) {  
    svg.append("g")  
    .attr("id", "ggraph" + (indice + 1) )  
    .attr("width", (width) + "px")  
    .attr("height", (height) + "px")  
    .attr("transform", "translate(" + coordinatesArrayX[indice] + ',' +  
    (coordinatesArrayY[indice]) + ')');  
}
```

Une fois ces zones dédiées aux graphiques, il a fallu y ajouter les axes et les points du nuage. Pour cela, on parcourt l'ensemble des balises g créées précédemment et on ajoute une nouvelle balise g qui contiendra les éléments nécessaires à l'affichage de l'axe. Une fois cette balise créée, l'ajout des éléments de l'axe s'ajoute automatiquement en utilisant la fonction call() de D3 et en y fournissant en paramètre l'objet comportant les informations sur l'axe créé au début.

```
g.append("g")  
  .attr("class", "axis")  
  .attr("font-size", cellSize/10 + "px")  
  .attr("transform", "translate("+ 0 + "," + (axe + transpoX) + ")")  
  .call(xAxe);
```

Une fois les axes ajoutés, il a fallu ajouter les points du nuage. Pour cela, les données relatives aux points étaient stockées dans un objet composé d'une liste. Chaque élément de cette liste était composé de l'ensemble des points à ajouter à la cellule selon l'indice de l'élément. Chaque point possédait trois informations : son label, sa valeur x et sa valeur y. Afin de pouvoir ajouter les points à leur cellule attribuée, la méthode utilisée a été d'extraire l'élément comportant les points dans la liste. Une fois cet élément récupéré, il a fallu le parcourir afin de récupérer chaque point devant être ajouté. À chaque point récupéré, on récupère la balise g de la cellule correspondante et on y ajoute une nouvelle balise qui comportera les informations du point. Cette balise est une balise circle permettant de dessiner un cercle dans le graphique. Il a été nécessaire de fournir différents attributs à cette balise afin de pouvoir la placer correctement dans le graphique. Il a fallu fournir sa dimension r afin de définir sa taille. Et il a fallu aussi fournir sa position x et y mais en les modifiant via les échelles produites précédemment afin qu'elle ne sorte pas de la zone définie.

```

for (var indice=0; indice<nbRows*nbColumns; indice++){
  var dataCell=nuageData[indice+1];

  for(var j=0; j<dataCell.x.length; j++){
    var g=d3.select("#ggraph" + (indice+1));
    g.append("circle")
      .attr("cx",function(){
        return (echelleX(dataCell.x[j]));
      })
      .attr("cy",function(){
        return (echelleY(dataCell.y[j]));
      })
      .attr("r",function(){
        return 3*(cellSize/100);
      })
      .style("stroke", "black")
      .style("fill","none");
  }
}

```

Une fois ces différentes étapes effectuées, on obtient une page HTML avec un nuage de points dans chaque cellule.

