Aula 4- Visualização de mapas e filtragem de dados

Resumo

- Ficheiro GeoJSON
- Visualização de mapas
- Filtragem de dados – d3.nest()

Nota:
Este tutorial foi realizado usando a versão 4 do d3.

4.1 GeoJSON

Inspecione o ficheiro fornecido (d3_4_1.htm) para ver o que o ficheiro faz. Adicione o comando debugger a seguir a linha svg e abra o ficheiro no chrome para inspecionar a informação que é carregada a partir do ficheiro world_countries.json. Inspecione a variavel geo_data assim como o ficheiro que lhe deu origem.

4.2 Projeção Mercator e visualização do mapa

Baseado no ficheiro GeoJSON visualize os poligonos associados aos vários países. Para tal é necessário selecionar o tipo de projeção a usar

```
var projection = d3.geoMercator();
```

A projeção transforma um par (latitude,longitude) numas coordenadas em pixels relativas ao elemento SVG onde se vai representar a informação de acordo com uma projeção.
A seguir é necessário desenhar os poligonos correspondentes aos países no ficheiro GeoJSON através do objeto SVG path:

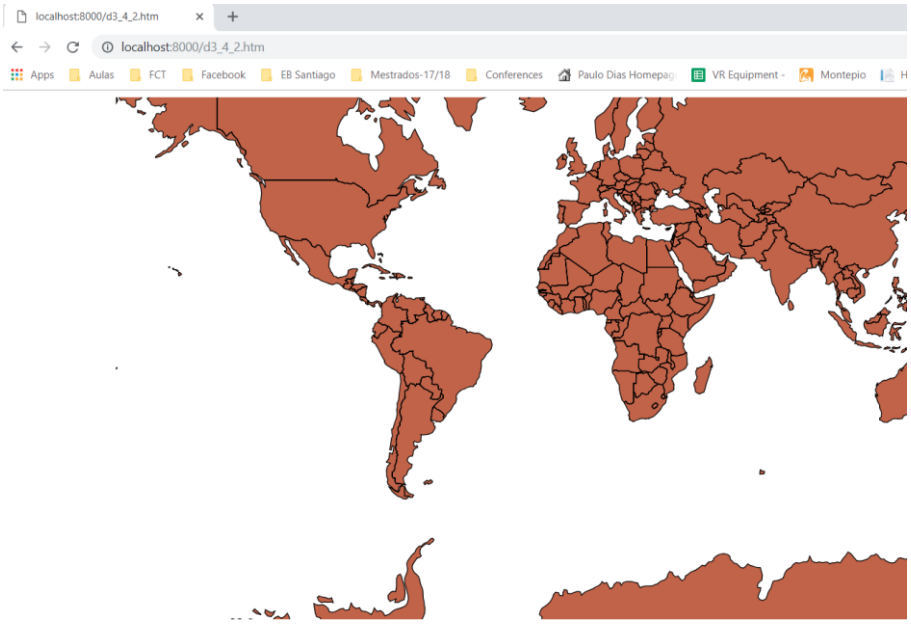
```
var path = d3.geoPath().projection(projection);

svg.append("path")
       .attr("d",path(geo_data))
```

A função `path` retorna o caminho associado ao poligono de cada país. Pode verificar isso colocando a palavra debugger a seguir a variável path e ver o que acontece quando passa um objeto do ficheiro GeoJSON (`path(geo_data.features[0])`).

Altere a escala e posição do mapa usando as funções scale e translate no objeto mercator. Ajuste a escala para 170 (a escala definida por omissão nas projeções é 150, tente valores entre 100 e 500 para ver o efeito). Centre agora o mapa no meio da janela svg.
Altere também através do style os parametros fill, stroke e stroke width de acordo com as suas preferências. Pode usar a cor "rgb(190,100,70)" e uma linha preta de 1 pixel para obter algo parecido com a imagem seguinte.

Pode testar outros tipos de projeções, por exemplo geoEquirectangular (pode ver opções em https://d3indepth.com/geographic/)



Mais informação em:
https://github.com/d3/d3-geo
https://d3indepth.com/geographic/

4.3 Carregamento dos dados de um ficheiro tsv (Nesting)

O objetivo dos exercícios seguintes é visualizar para cada país o numero de pessoas que assitiram aos jogos no ano em que la ocorreu o campeonato do mundo.
Começe por carregar a informação relativa aos campeonatos do mundo dentro da função que desenha o mapa. Pode usar o código seguinte que carrega e formata a informação do ficheiro world-cup-geo.tsv.

```
function plot_circles(data) {
}

d3.tsv("world_cup_geo.tsv", function (d) {
  return(d)
}, plot_circles);
```

Interrompa o programa dentro da função plot-circles ou adicione um console.log ou um console.table para ver os dados. Visualize só a primeira linha do ficheiro (`console.table(data[0]);`) e repare que todos os dados são string.

É possível converter os dados antes de realizar a leitura dos mesmos (ao ler o tsv) alterando a função como segue:

```
var parseTime = d3.timeParse("%d-%m-%Y (%H:%M h)");

d3.tsv("world_cup_geo.tsv", function(d) {
  d["attendance"] = +d["attendance"];
  d["date"] = parseTime(d["date"]);
  return(d);
}, plot_circles);
```

Sendo que a função timeParse permite transformar a string com a data num objeto "date" e o + transforma a string attendance num valor numérico.
Veja novamente os dados na função de desenho e confirma a conversão.

Mais informação sobre o timeParse:
https://github.com/d3/d3-time-format/blob/master/README.md#locale_format

4.4 Filtragem dos dados de acordo com o ano(Nesting)

Para permitir visualizar o nº total de espetadores por ano/campeonato do mundo, é necessário re-organizar os dados. A função `nest` do d3 permite esse tipo de operações.

A função nest é constituida por 3 blocos:

- key: critério de agrupamento dos dados.
- rollup: define que dados vão ser agregados e como.
- entries: define os dados a filtrar.

Coloque o código seguinte na função de desenho dos círculos para usar o ano como critério de associação dos dados.

```
var nested = d3.nest()
      .key(function (d) {
         return d["date"].getUTCFullYear();
      })
      .rollup(function (leaves) {

      })
      .entries(data);
```

Coloque um debugger na funçao rollup e veja o contéudo da variaveis `d["date"].getUTCFullYear();` e `leaves`. Tente perceber o que a função está fazer.

Mas informação sobre as operações de nesting em:
http://bl.ocks.org/phoebebright/raw/3176159/

4.5 Agrupamento dos dados em novas variáveis

Para somar os valores todos dos espetadores em cada ano, deve colocar dentro da funções rollup o código seguinte:

```
var total = d3.sum(leaves, function (d) {
                            return d["attendance"];
                        });
```

Dado que em cada ano o campeonato do mundo teve lugar em vários estádios cujas coordenadas estão definidas no ficheiro inicial, vamos armazenar no numa variável um array com as coordenadas dos valores de cada estádio.

```
var coords = leaves.map(function (d) {
                            return projection([+d.long, +d.lat]);
                        });
```

Note a utilização do map para permitir retornar os valores como um array a ser armazenado em coords e a utilização da função projection para ter os valores já em pixeis para representação no mapa de acordo com a projeção escolhida.

É agora possível calcular a posição "média" dos estádios (onde se vai representar o nº de espetádores através de um círculo) com o código seguinte:

```
var centerx = d3.mean(coords, function (d) { return d[0]; });
var centery = d3.mean(coords, function (d) { return d[1]; });
```

Finalmente vamos criar duas novas variáveis com os dados calculados na função rollup com o nº de espetadores e posição média entre os estádios para cada um dos anos com o código:

```
return {
  "attendance": +total,
  "x": centerx,
  "y": centery
};
```

Todo este código deve estar na função de rollup.

Coloque um "debugger" a seguir a função nest e inspecione na consola (ou através de um console.table(nested) os valores devolvido. Deve observar que a variável contêm 20 elementos (um por cada mundial) com um key (o ano) e três values (attendance, x e y).

4.6 Visualização das localizações

Represente com um círculo de raio 5 as coordenadas calculadas na alinea anterior (o que representa essa posição?). Para tal deve completar o código seguinte da melhor forma (este código foi usado na aula 3).

```
svg.append('g')
  .attr("class", "attendance")
  .selectAll("circle")
  .data(nested)
  .enter()
  .append("circle")
  .attr("cx", function (d) { ___ })
  .attr("cy", function (d) { ___ })
  .attr("r", 5);
```

4.7 Visualização do numero de espétadores

Modifique agora o código para que o tamanho dos círculos dependa do número total de espetádores em cada país.

Pode aproveitar o o código seguinte:

```
var attendance_extent = d3.extent(nested,function(d){
   return d.value["attendance"];
 })

var radius_scale=d3.scaleLinear().domain(attendance_extent).range([1,15]);
```

Visualize e o resultado. Esta escala parece-lhe a mais adequada? Porquê? Que escala faria mais sentido utilizar neste exemplo. Altere o código para usar a escala scaleSqrt e compare as duas visualizações.

Altere alguns parâmetros dos círculos para conseguir ver melhor sobreposições, por exemplo a opacidade.

No final deve obter algo semelhante a figura seguinte: