

Aula 3- Escalas, eixos, interações e transições

Resumo

- Gráficos de dispersão (scatter plot)
- Leitura de dados de um ficheiro JSON
- Escalas, eixos e linhas

3.1 Scatter Plot

Neste exemplo os dados vão ser bidimensionais permitindo especificar coordenadas x,y no plano.

Acrescente os dados seguintes no exemplo 2.3.

```
var dataset = [
  [ 5, 20 ],
  [ 480, 90 ],
  [ 250, 50 ],
  [ 100, 33 ],
  [ 330, 95 ],
  [ 410, 12 ],
  [ 475, 44 ],
  [ 25, 67 ],
  [ 85, 21 ],
  [ 220, 88 ]
];
```

Represente os dados utilizando círculos SVG com raio 5, atributo cx d[0] e atributo cy d[1].



Opcional:

Pode ainda adicionar as etiquetas para obter algo semelhante a figura anterior. Finalmente altere o exemplo para gerar aleatoriamente um array a 3 dimensões usando os dois primeiros valores para as coordenadas e o terceiro para o raio (entre 10 e 50). Visualize os gráficos com esses novos dados.

Aproveite o código seguinte:

```
dataset[i] = [];
dataset[i][0] = Math.floor((Math.random() * 500));
```

3.2 Leitura de dados de um ficheiro

Considere o exemplo seguinte que permite carregar dados a partir de um ficheiro json e visualizar estes dados com d3 modificando o conteúdo da função draw. Note que para aceder ao conteúdo do ficheiro JSON, é necessário ter os dados num servidor. É possível usar o firefox que permite ler diretamente ficheiros locais no computador. Em alternativa, pode se usar um servidor, por exemplo em python. Para configurar um servidor em python, basta instalar o python (<http://python.org>) e correr o código seguinte numa linha de comandos a partir do diretório onde estão os ficheiros html: `python -m SimpleHTTPServer 8000` ou `python -m http.server`. Poderá então ver os exemplos apontando o navegador para o endereço `http://localhost:8000`.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <script src="http://d3js.org/d3.v5.js"></script>
  <script>
    var width = 500;
    var height = 300;
    var margin = 50;
    function draw(data) {
      // código de visualização
    }
  </script>
</head>

<body>
  <script>
    d3.json("MIECT_Alunos_Ano.json", draw);
  </script>
</body>

</html>
```

Nota:

Na versão 5 do d3 tem que usar o código:

```
d3.json("MIECT_Alunos_Ano.json").then(draw);
```

Utilize o código para criar um scatter plot baseado no exemplo anterior que represente os alunos do curso de ECT em função do ano (ficheiro `MIECT_Alunos_Ano.json` fornecido). Utilize no eixo dos x a variável contadora do d3 multiplicada por 50 (ver última aula) com valores adequados e no eixo dos y, os dados `d.Numero_de_Alunos` e um raio de 10.

3.3 Utilização de uma escala

O D3.js fornece funções que permitem realizar transformações dos dados. Estas funções permitem mapear um domínio de entrada (domain) para um intervalo de saída (range). Por exemplo imagine que quer mapear dados correspondentes a idades entre 15 e 90 anos para um intervalo de pixels 0 e 100. Isso pode ser realizado com a função `ScaleLinear`.

Modifique o exemplo para utilizar escalas nos eixos adaptando o código que segue. Comece por definir a gama de valores a representar com a função `extent`:

```
var y_extent = d3.extent(data, function(d){return d.Numero_de_Alunos });
```

Pode agora definir a escala a usar (note a utilização de uma margem que pode ser definida com 20 pixéis por exemplo):

```
var y_scale = d3.scaleLinear()
    .range([margin, height - margin])
    .domain(y_extent);
```

É agora possível mapear o nº de alunos usando a escala assim definida:

```
.attr("cy", function (d) {return y_scale(d. Numero_de_Alunos);})
```

Da mesma forma, defina uma escala adequada para o eixo dos x, usando a gama de valores definida da forma seguinte:

```
var x_extent = [0, data.length];
```

3.4 Visualização de eixos

O d3 fornece facilidades para a criação e visualização de eixos. Para criar e visualizar o eixo das abcissas utilize o código seguinte que permite criar um eixo com as linhas e etiquetas adequadas a partir duma escala.

```
var x_axis = d3.axisBottom(x_scale);
```

Pode agora adicionar o eixo na representação:

```
d3.select("svg")
    .append("g")
    .attr("class", "x axis")
    .attr("transform", "translate(0," + (height - margin) + ")")
    .call(x_axis);
```

Este código agrupa o eixo com uma transformação para permitir desenhar o mesmo na posição correta. Comente a linha da transformação veja o que ocorre.

Da mesma forma adicione agora o eixo das ordenadas utilizando a função `axisLeft`. Não se esqueça que o eixo dos abcissas está invertido em SVG. Como pode resolver este problema usando a escala?

Pode utilizar o código css seguinte para formatar os eixos:

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

```

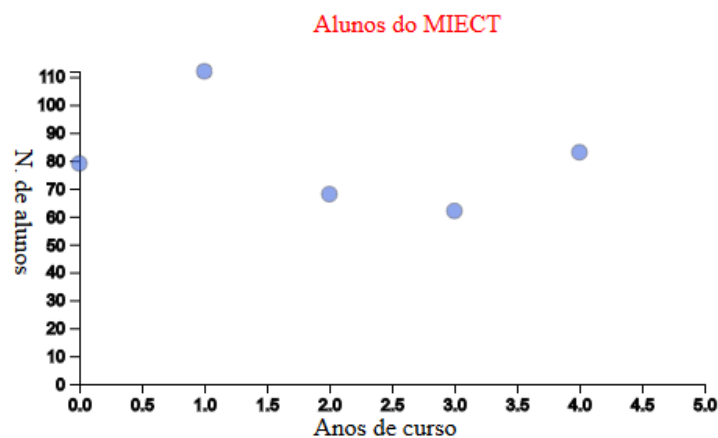
circle{
  stroke:black;
  stroke-width:0.5px;
  fill:RoyalBlue;
  opacity:0.6;
}
</style>

```

Por fim, adapte o código para adicionar um título nos eixos usando um texto (ver 2.6). Modifique o necessário para obter um resultado semelhante a figura seguinte. Pode também modificar os estilos para obter um gráfico ao seu gosto.

Opcional: pode usar uma escala ordinal para mapear os anos de curso (“1º Ano”, ...”5ºano”).

<https://bl.ocks.org/lhstock/8394a9bef240321c34f3c13576d50ee5>



3.5 Adição de uma linha

Utilize o código seguinte para criar um *path* em svg unindo os vários círculos definidos pelos dados:

```

var line = d3.line()
  .x(function (d,i) { return x_scale(i) })
  .y(function (d) { return y_scale(d.Numero_de_Alunos) });

d3.select("svg")
  .append("path")
  .attr("d", line(data))
  .attr("class", "linha_alunos");

```

Veja o resultado. Pode alterar os vários atributos da linha e dos círculos usando o CSS seguinte:

```

path {
  fill:none;
  stroke:black;
  stroke-width:2px;
}
path.linha_alunos{
  stroke:DeepPink;
}

```