

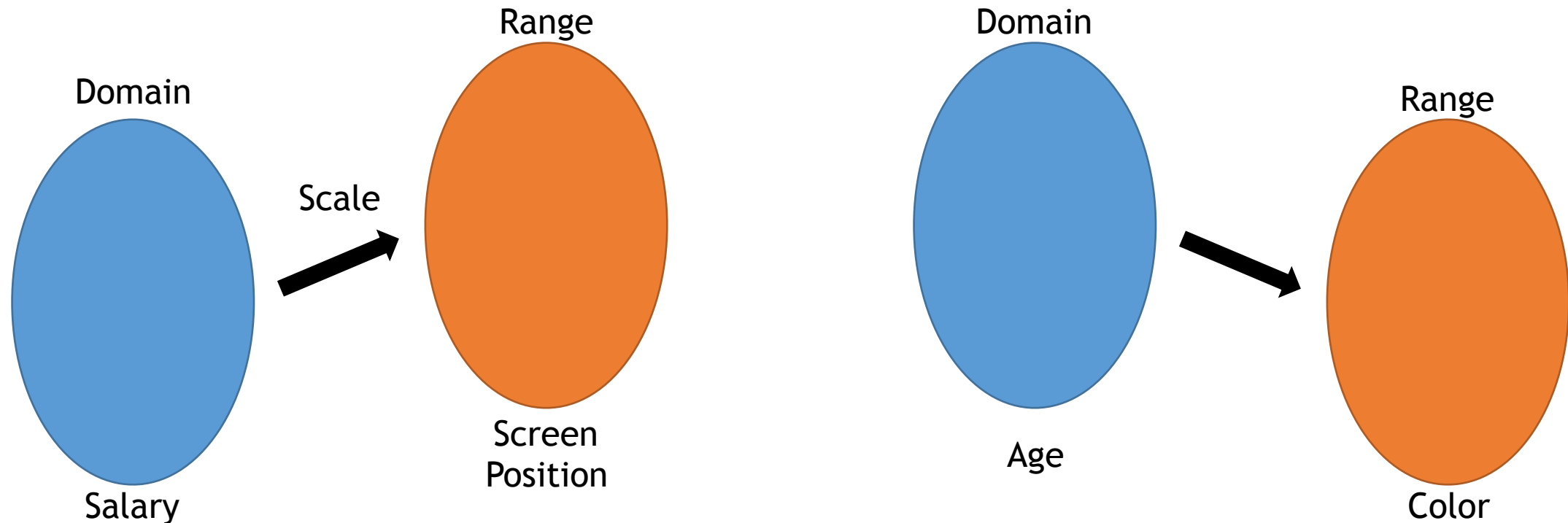
Treinamento Visualização de Dados em D3

[Nivan Ferreira](mailto:nivan@cin.ufpe.br)
nivan@cin.ufpe.br



Escalas

- “Escalas são funções que mapeiam um domínio de entrada (usualmente dados) para uma domínio de saída (imagem da função) (usualmente propriedades visuais)”



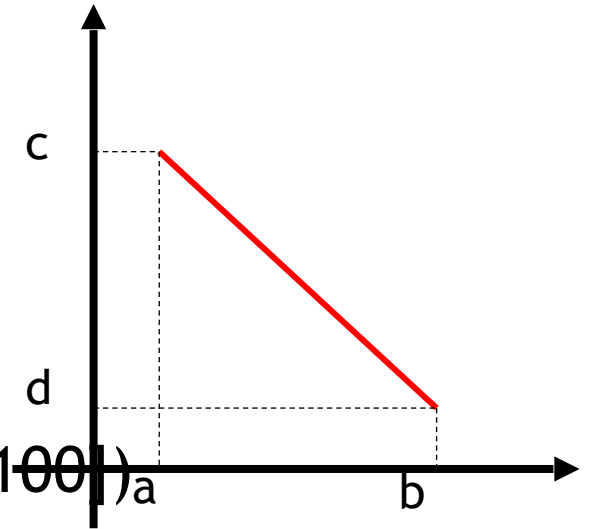
Escalas Lineares

- Funções de Interpolação Linear

- `var scale = d3.scaleLinear().domain([a,b]).range([c,d])`

- Exemplo

- `var scale = d3.scaleLinear().domain([20,50]).range([0,100])`
 - `scale(20) = ?`
 - `scale(50) = ?`
 - `scale(35) = ?`
 - `scale(0) = ?`
 - `scale(100) = ?`



Escalas Lineares

- Usando escalas lineares, podemos mapear dados para coordenadas svg de maneira adequada (evitando que dados sejam plotados fora do svg)
- Processo
 - Encontrar o intervalo dos dados
 - Criar uma escala que mapeia o intervalo de dados em coordenadas de tela apropriadas

Escalas Lineares

- Encontrar o intervalo dos dados
 - `d3.extent`

- Exemplo

```
var dataset = [5,1,-1,9,100,50];  
console.log(d3.extent(dataset));
```

- Para dados complexos, podemos usar uma função que acessa o elemento com o qual desejamos achar o intervalo

```
var dataset = [[5, 20], [480, 90], [250, 50], [100, 33], [330, 95]];  
console.log(d3.extent(dataset, d=>d[0]));
```

Escalas Lineares

- Agora precisamos criar a escala que mapeia o intervalo em coordenadas de tela
- `var dataXInterval = d3.extent(dataset, d=>d[0]);`
`var xScale = d3.scaleLinear().domain(dataXInterval).range([0,500]);`
- `var dataYInterval = d3.extent(dataset, d=>d[1]);`
`var yScale = d3.scaleLinear().domain(dataYInterval).range([0,500]);`

Escalas Lineares

- Agora usaremos essas escalas

mySVG

.selectAll("circle")

.data(dataset)

.enter()

.append("circle")

.attr("r","10")

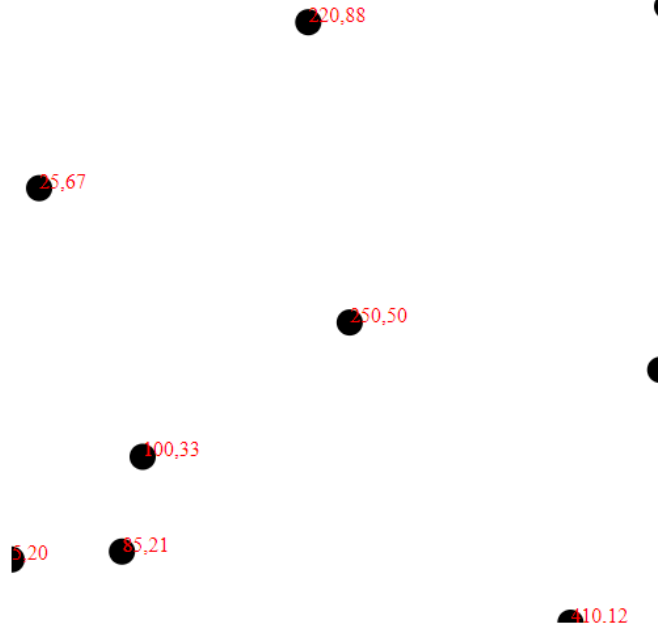
.attr("cx",function(d){return xScale(d[0]);})

.attr("cy",function(d){return yScale(d[1]);});

Escalas Lineares

- Como corrigir o problema da orientação do eixo y?

```
var yScale =  
d3.scaleLinear().domain(dataYInterval).range([500,0]);
```



Escalas de Cores (Colormaps)

- Podemos usar escalas lineares para mapear cores
- D3 automaticamente interpola cores
- Exemplo
 - `var cScale = d3.scaleLinear().domain([0,100]).range(["black","red"])`
 - `cScale(0)?`
 - `cScale(100)?`
 - `cScale(50)?`



Escalas de Cores (Colormaps)

- Exercício

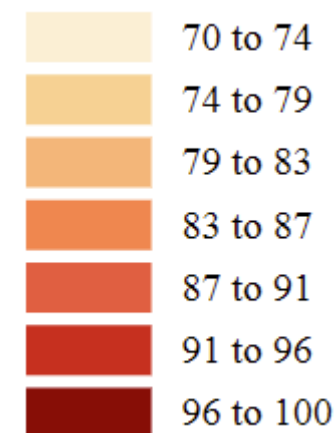
```
var dataset = [[5, 20, 9], [480, 90, 5], [250, 50, 1], [100, 33, 0], [330, 95, 10], [410, 12, 3], [475, 44, 7], [25, 67, 9], [85, 21, 2], [220, 88, 2]];
```

- Defina a cor dos pontos do scatterplot proporcional à terceira coordenada
- Solução [third_scatterplot.html](#)

Escalas Quantizadas

- Muito útil para criar escalas de cores
- ```
var colorset = ['#fef0d9' , '#fdd49e' , '#fdbb84' , '#fc8d59' ,
 '#ef6548' , '#d7301f' , '#990000'];
```

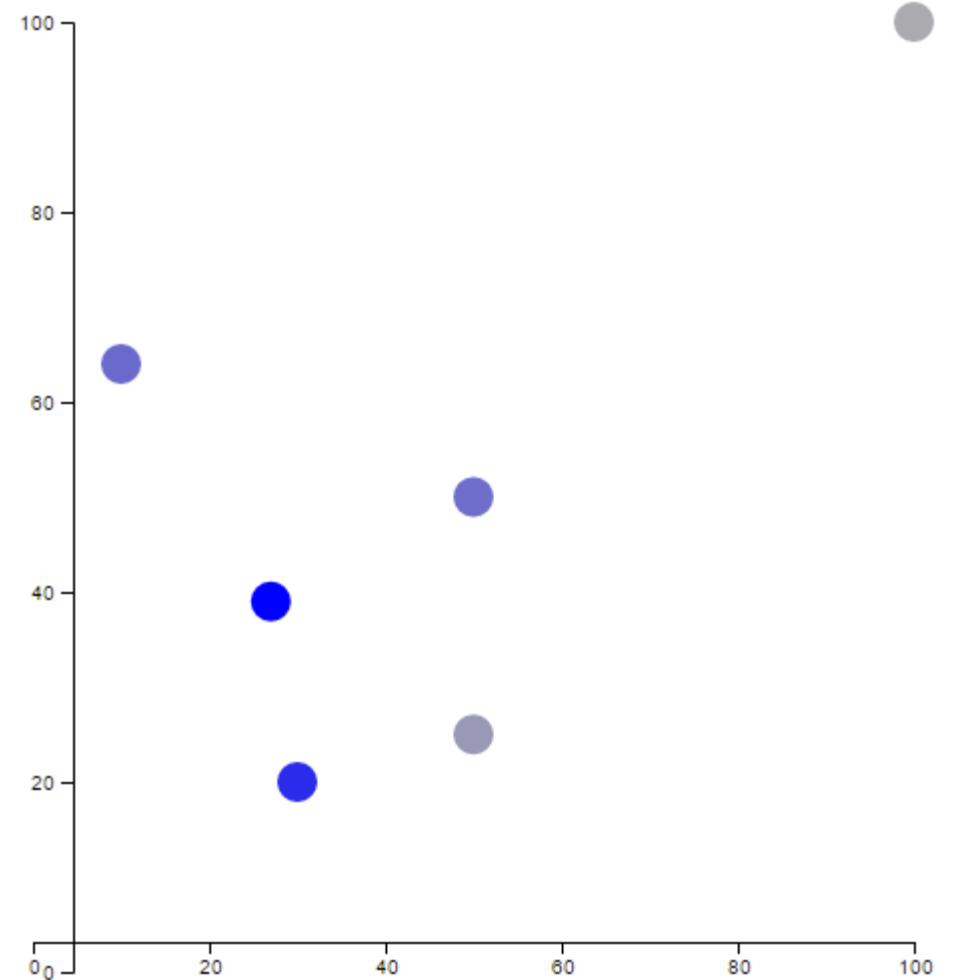
```
var colorScale = d3.scaleQuantize()
 .domain([70, 100])
 .range(colorset);
```



Eixos

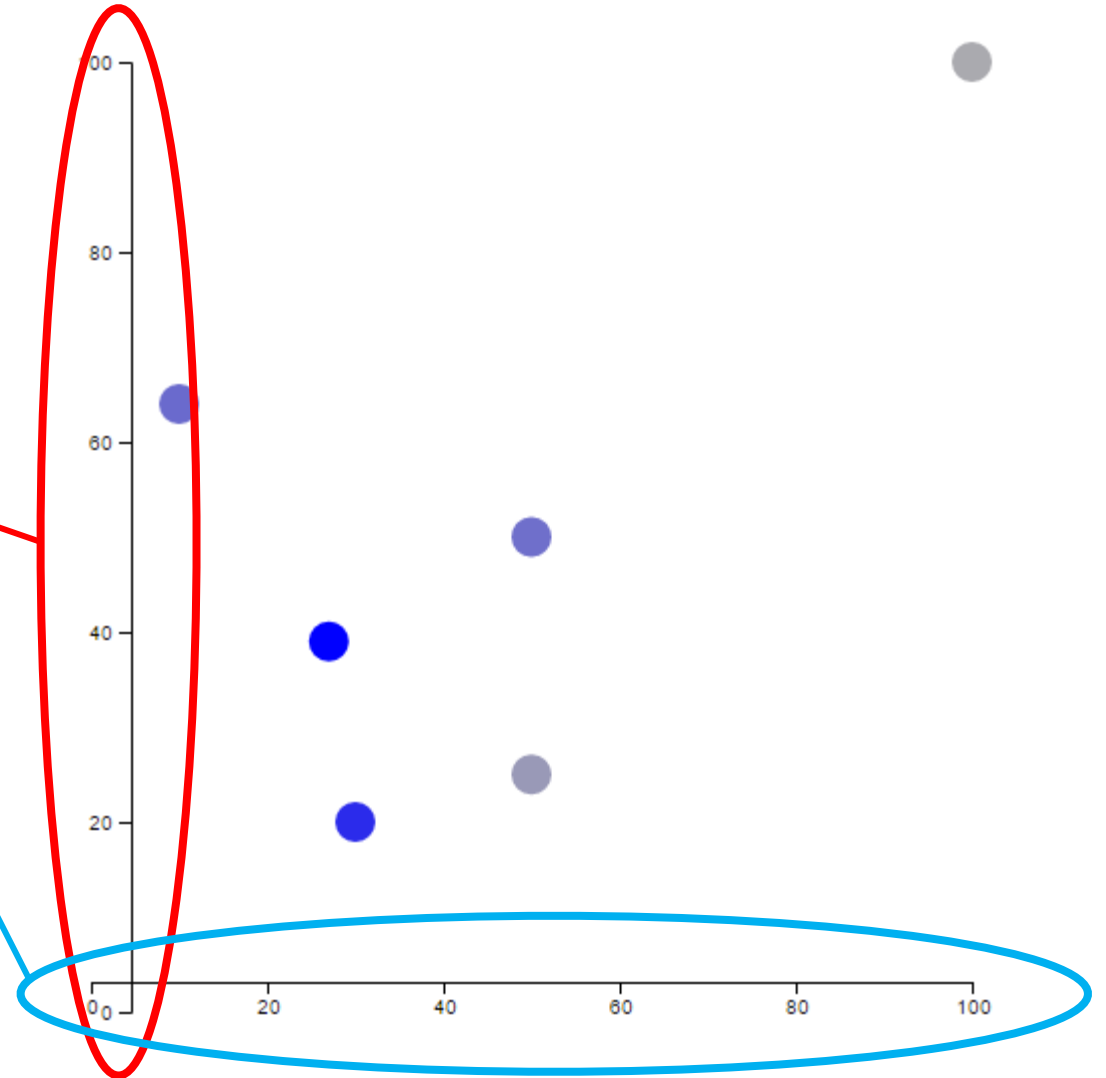
# Eixos

- O d3 possui funcionalidades para criar eixos
- Representações visuais de escalas



# Eixos

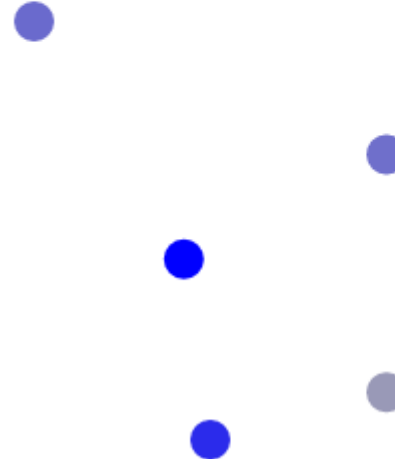
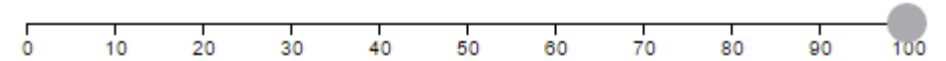
- `d3.axisBottom(xScale);`
- `d3.axisTop(xScale);`
- `d3.axisLeft(xScale);`
- `d3.axisRight(xScale);`



# Eixos

```
var xAxis = d3.axisBottom(xScale);
mySVGxAxisGroup.call(xAxis);
```

Função que passa a  
selection (chamador)  
para o objecto  
(parâmetro)  
equivalente à  
`xAxis(mySVGxAxisGroup)`



# Eixos

- Eixos são sempre desenhados na origem!
- Para colocar na posição desejada

```
var xAxisGroup = mySVG.append("g")
 .attr("class", "xAxis")
 .attr("transform",
 "translate(0, "+(height-margin.top)+")");
var xAxis = d3.axisBottom(xScale);
xAxisGroup.call(xAxis);
```

