

Data-Driven Documents

Lucas Pagliosa

Aula de Visualização - Parte Prática

9 de setembro de 2014

1 Configurando

2 SVG

3 Escalas

4 Histogramas

5 Scatterplots

Navegadores recentes como Safari ou **Chrome** oferecem um sistema de depuração

Navegadores recentes como Safari ou **Chrome** oferecem um sistema de depuração

F12 abre a WebKit do navegador Chrome (utilizado neste aula)

Navegadores recentes como Safari ou **Chrome** oferecem um sistema de depuração

F12 abre a WebKit do navegador Chrome (utilizado neste aula)

A aba rotulada *Elements* mostra a estrutura HTML da página

Navegadores recentes como Safari ou **Chrome** oferecem um sistema de depuração

F12 abre a WebKit do navegador Chrome (utilizado neste aula)

A aba rotulada *Elements* mostra a estrutura HTML da página

A aba rotulada *Sources* abre os arquivos que influenciam no layout da página. Depuração ocorre nesse ambiente.

Navegadores recentes como Safari ou **Chrome** oferecem um sistema de depuração

F12 abre a WebKit do navegador Chrome (utilizado neste aula)

A aba rotulada *Elements* mostra a estrutura HTML da página

A aba rotulada *Sources* abre os arquivos que influenciam no layout da página. Depuração ocorre nesse ambiente.

A aba rotulada *Console* oferece um sistema de saída ao usuário ou desenvolvedor
`console.log("mensagem");`

Random values

I'm number 4

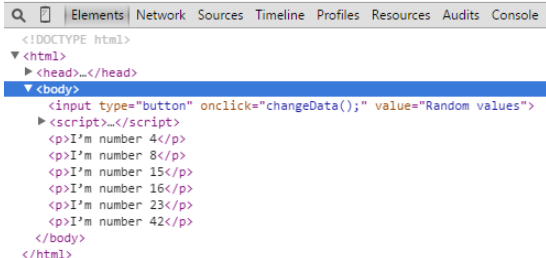
I'm number 8

I'm number 15

I'm number 16

I'm number 23

I'm number 42



The screenshot shows the Chrome DevTools interface. At the top, there is a search bar and a navigation menu with tabs for Elements, Network, Sources, Timeline, Profiles, Resources, Audits, and Console. The 'Elements' tab is active, displaying the DOM tree. The tree is expanded to show the <body> element, which contains an <input type="button" onclick="changeData();" value="Random values"> and a <script>...</script> tag. Below the script tag, there are six <p>I'm number [value]</p> elements with values 4, 8, 15, 16, 23, and 42. The <body> element is highlighted with a blue background.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <input type="button" onclick="changeData();" value="Random values">
    <script>...</script>
    <p>I'm number 4</p>
    <p>I'm number 8</p>
    <p>I'm number 15</p>
    <p>I'm number 16</p>
    <p>I'm number 23</p>
    <p>I'm number 42</p>
  </body>
</html>
```

Figura: WebKit do Chrome

Referenciando D3

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3 Test</title>
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    // OU
    // <script src="http://d3js.org/d3.v3.min.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      // Todo
    </script>
  </body>
</html>
```

1º caso: (a)

```
<script type="text/javascript">
  var dataset = ["Lucas", "your father"];

  //d3.select("body").append("p").text("Hello world!");
  d3.select("body")
    .data(dataset)
    .append("p")
    .text(function(d) {
      return "I am " + d);
</script>
```

1º caso: (b)

```
<script type="text/javascript">
  var dataset = ["Lucas", "your father"];

  d3.select("body").append("p").text("Hello world!");
  d3.select("body")
    .data(dataset)
    .append("p")
    .text(function(d) {
      return "I am " + d);
</script>
```

2º caso:

```
<script type="text/javascript">
  var dataset = ["Lucas", "Your father"];

  d3.select("body").append("p").text("Hello world!");

  d3.select("body").selectAll("p").data(dataset)
    .enter().append("p")
    .html(function(d) { return "I am " + d ; });

  d3.select("body").selectAll("p").data(dataset)
    .transition().delay(500).duration(3000)
    .text(function(d) { return "I am " + d ; })
    .style("color","red");
```

1 Configurando

2 SVG

3 Escalas

4 Histogramas

5 Scatterplots

SVG são entidades gráficas vetoriais

SVG são entidades gráficas vetoriais

Ao contrário do elemento *canvas* do HTML, elementos desse tipo são "ajustáveis" à interação do usuário (zoom in/zoom out)

SVG são entidades gráficas vetoriais

Ao contrário do elemento *canvas* do HTML, elementos desse tipo são "ajustáveis" à interação do usuário (zoom in/zoom out)

Criando um elemento SVG

```
var svg = d3.select("body").append("svg")  
  .attr("width", 500).attr("height", 50);
```

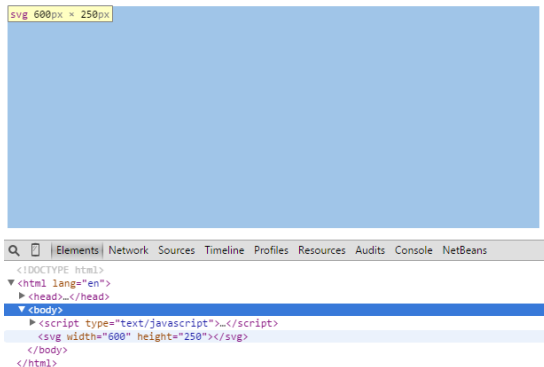
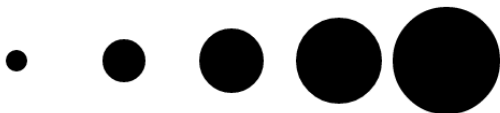



Figura: Elemento SVG

```
var dataset = [ 5, 10, 15, 20, 25 ];

svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle");

circles.attr("cx", function(d, i) {
  return (i * 50) + 25;
})
.attr("cy", h/2)
.attr("r", function(d) {
  return d;
});
```



```
Elements Network Sources Timeline Profiles Resources Audits Console NetB
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <script type="text/javascript">...</script>
    <svg width="600" height="250">
      <circle cx="25" cy="125" r="5"></circle>
      <circle cx="75" cy="125" r="10"></circle>
      <circle cx="125" cy="125" r="15"></circle>
      <circle cx="175" cy="125" r="20"></circle>
      <circle cx="225" cy="125" r="25"></circle>
    </svg>
  </body>
</html>
```

Figura: Circulos

1 Configurando

2 SVG

3 Escalas

4 Histogramas

5 Scatterplots

Escalas são essenciais quando é necessário mapear um espaço de instâncias em um espaço visual

Suponha a área SVG :

```
// Create SVG element
var w = 600;
var h = 250;
var svg = d3.select("body")
  .append("svg")
  .attr("width", w)
  .attr("height", h);
```

E os seguintes dados:

```
var dataset = [5, 10, 13, 19, 21, 28, 22, 18, 15, 13, \\
  11, 12, 15, 20, 18, 17, 16, 18, 23, 25];
```

Tarefa

Criar um histograma com os valores

Tarefa

Criar um histograma com os valores

Problema

A área SVG possui altura de 250 unidades, e o maior histograma terá altura de 28 unidades.

Tarefa

Criar um histograma com os valores

Problema

A área SVG possui altura de 250 unidades, e o maior histograma terá altura de 28 unidades.

Resultado:

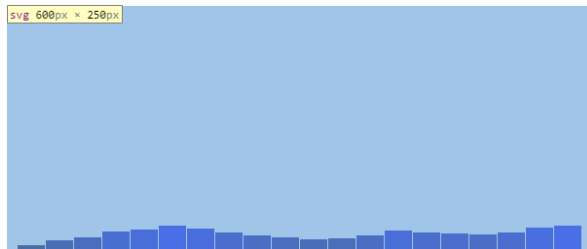


Figura: Histograma sem escala

Normalização

Para resolver o problema, é necessário mapear os valores de entrada para os valores de saída

Normalização

Para resolver o problema, é necessário mapear os valores de entrada para os valores de saída



Figura: Normalização

Escalas no D3

A função `d3.scale()` cria uma escala em D3

Escalas no D3

A função `d3.scale()` cria uma escala em D3

Métodos essenciais:

- Tipo da escala:
 - Linear: `d3.scale.linear()`
 - Logarítmica: `d3.scale.log()`
 - Quantitativo: `d3.scale.quantize()`
 - Quantitativo: `d3.scale.ordinal()`

Escalas no D3

A função `d3.scale()` cria uma escala em D3

Métodos essenciais:

- Tipo da escala:
 - Linear: `d3.scale.linear()`
 - Logarítmica: `d3.scale.log()`
 - Quantitativo: `d3.scale.quantize()`
 - Quantitativo: `d3.scale.ordinal()`
- Entrada: `d3.scale.TIPO.domain(VETOR)`

Escalas no D3

A função `d3.scale()` cria uma escala em D3

Métodos essenciais:

- Tipo da escala:
 - Linear: `d3.scale.linear()`
 - Logarítmica: `d3.scale.log()`
 - Quantitativo: `d3.scale.quantize()`
 - Quantitativo: `d3.scale.ordinal()`
- Entrada: `d3.scale.TIPO.domain(VETOR)`
- Saída: `d3.scale.TIPO.range(VETOR)`

Exemplos:

Linear

```
var q = d3.scale.linear().domain([0, 28]).range([0, 250]);  
q(5) === 44, q(10) === 89, q(13) === 116, q(19) === 169;
```

Exemplos:

Linear

```
var q = d3.scale.linear().domain([0, 28]).range([0, 250]);  
q(5) === 44, q(10) === 89, q(13) === 116, q(19) === 169;
```

Ordinal

```
var q = d3.scale.ordinal().domain([1, 2, 4, 3])  
  .range([10, 50, 60, 100]);  
q(1.9) === 10, q(2.1) === 50, q(3) === 100, q(4) === 60;
```


Logarítmico

```
var q = d3.scale.log().domain([2, 16]).range([1, 6]).base(2);  
q(2) === 1, q(4) === 2.66, q(8) === 4.33, q(16) === 6;
```

Logarítmico

```
var q = d3.scale.log().domain([2, 16]).range([1, 6]).base(2);  
q(2) === 1, q(4) === 2.66, q(8) === 4.33, q(16) === 6;
```

Quantitativo

```
var q = d3.scale.quantize().domain([0, 1])  
  .range(['a', 'b', 'c']);  
q(0.3) === 'a', q(0.4) === 'b', q(0.6) === 'b',  
q(0.7) === 'c';
```

1 Configurando

2 SVG

3 Escalas

4 Histogramas

5 Scatterplots

Objetivo: criar um histograma com os dados

Passo 1: Criação da área SVG

```
// Width and height
var w = 600;
var h = 250;

// Create SVG element
var svg = d3.select("body").append("svg")
    .attr("width", w).attr("height", h);
```

Passo 2: Criação das escalas e obtenção dos dados

```
var dataset = [5, 10, 13, 19, 21, 28, 22, 18, 15, 13,  
  11, 12, 15, 20, 18, 17, 16, 18, 23, 25];  
var xScale = d3.scale.ordinal()  
  .domain(d3.range(dataset.length))  
  .rangeRoundBands([0, w], 0.05);  
var yScale = d3.scale.linear()  
  .domain([0, d3.max(dataset)])  
  .range([0, h - 30]);
```

Passo 3: Criação dos histogramas

```
// Create bars
svg.selectAll("rect")
  .data(dataset)
  .enter()
  .append("rect")
  .attr("x", function(d, i){return xScale(i);})
  .attr("y", function(d) { return h - yScale(d); })
  .attr("width", xScale.rangeBand())
  .attr("height", function(d){return yScale(d);})
  .attr("fill", function(d) {
    return "rgb(0, 0, " + (d * 10) + ")";
  });
```

\\ xScale.rangeBand() retorna o tamanho da largura da banda de
\\ cada intervalo, no caso a largura de cada histograma

Passo 3: Criação dos histogramas

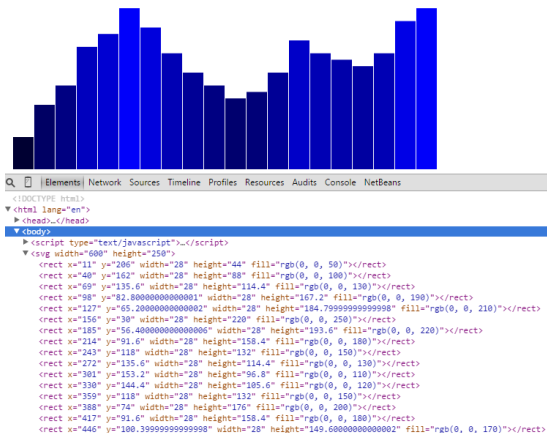


Figura: Visualização até o momento

Passo 4: Adicionando labels

```
// Create labels
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function(d){return d;})
  .attr("text-anchor", "middle")
  .attr("x", function(d, i) {
    return xScale(i) + xScale.rangeBand() / 2;
  })
  .attr("y", function(d) { return h - yScale(d) - 10; })
  .attr("font-family", "sans-serif")
  .attr("font-size", "15px")
  .attr("fill", "red");
```


Passo 4: Adicionando labels

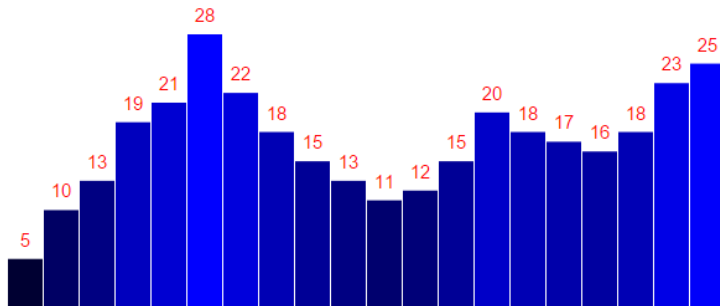


Figura: Visualização até o momento

No próprio HTML vamos adicionar o seguinte parágrafo: `<p>Click on this text to update the chart with new data values as many times as you like!</p>`

No próprio HTML vamos adicionar o seguinte parágrafo: `<p>Click on this text to update the chart with new data values as many times as you like!</p>`

e. `OnClick(f)`

A função `onClick` é um *listening* que invoca a função f quando o elemento é clicado

Passo 5: Iteração e animação

```
// On click, update with new data
d3.select("p").on("click", function()
{
  //New values for dataset
  var numValues = dataset.length;
  var maxValue = 100;

  dataset = []; // Initialize empty array
  for (var i = 0; i < numValues; i++)
  {
    var newNumber = Math.floor(Math.random() * maxValue);
    dataset.push(newNumber);
  }
}
```

Passo 5: Iteração e animação (cont)

```
// Recalibrate the scale domain, given the new max value in dataset
yScale.domain([0, d3.max(dataset)]);
```

```
// Update all rects
svg.selectAll("rect")
  .data(dataset)
  .transition().delay(function(d, i) {
    return i / dataset.length * 1000;
  })
  .duration(500)
  .attr("y", function(d) { return h - yScale(d); })
  .attr("height", function(d){return yScale(d);})
  .attr("fill", function(d) {
    return "rgb(0, 0, " + (d * 10) + ")";
  });
```

Passo 5: Iteração e animação (cont)

```
// Update all labels
svg.selectAll("text")
  .data(dataset)
  .transition()
  .delay(function(d, i) {
    return i / dataset.length * 1000;
  })
  .duration(500)
  .text(function(d){return d;})
  .attr("x", function(d, i) {
    return xScale(i) + xScale.rangeBand() / 2;
  })
  .attr("y", function(d) {
    return h - yScale(d) - 10;
  });
});
```

1 Configurando

2 SVG

3 Escalas

4 Histogramas

5 Scatterplots

Scatterplots

Click on this text to update the chart with new data values as many times as you like!

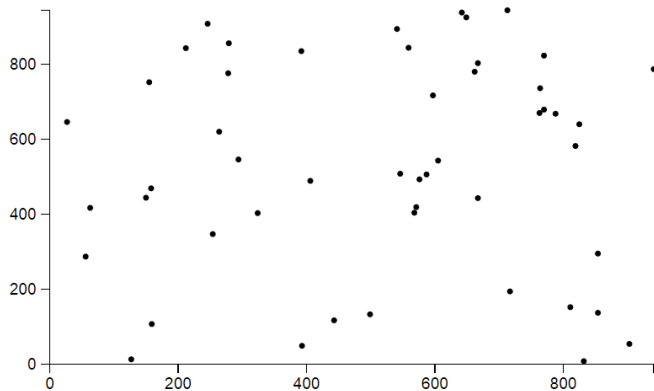


Figura: Scatterplot

Trabalho

`file:///D:/Dropbox/Work/ICMC/Semin%20rio%20D3/Pr%20etica/
Exemplos/nice%20scatterchart%20transaction.html`

The End

The End

Perguntas??