

# *Introdução a Ciência de Dados*



Professor: Alex Pereira

# *Scrolltelling*

- É uma maneira dinâmica de compor uma narrativa
  - E seus elementos visuais
- Com a simplicidade da interação baseada em rolagem
  - os leitores não precisam adivinhar o que pressionar, clicar ou deslizar para ver a história completa.
- Fornece novas oportunidades para eventos multimídia acionados por posição,
  - como reprodução de vídeo, animação e transições de imagem.
- Possibilita uma interação dinâmica de texto, imagens e som
  - que dá à história um ritmo envolvente e fornece elementos de surpresa quando os componentes dinâmicos assumem novas formas.

# *Scrolltelling: Alguns exemplos*

- Scroll telling com mapa
- Scroll telling Garantias Concedidas pela União
  - do Tesouro Nacional
- Detentos de Chicago
  - pelo The Guardian
- Outros exemplos

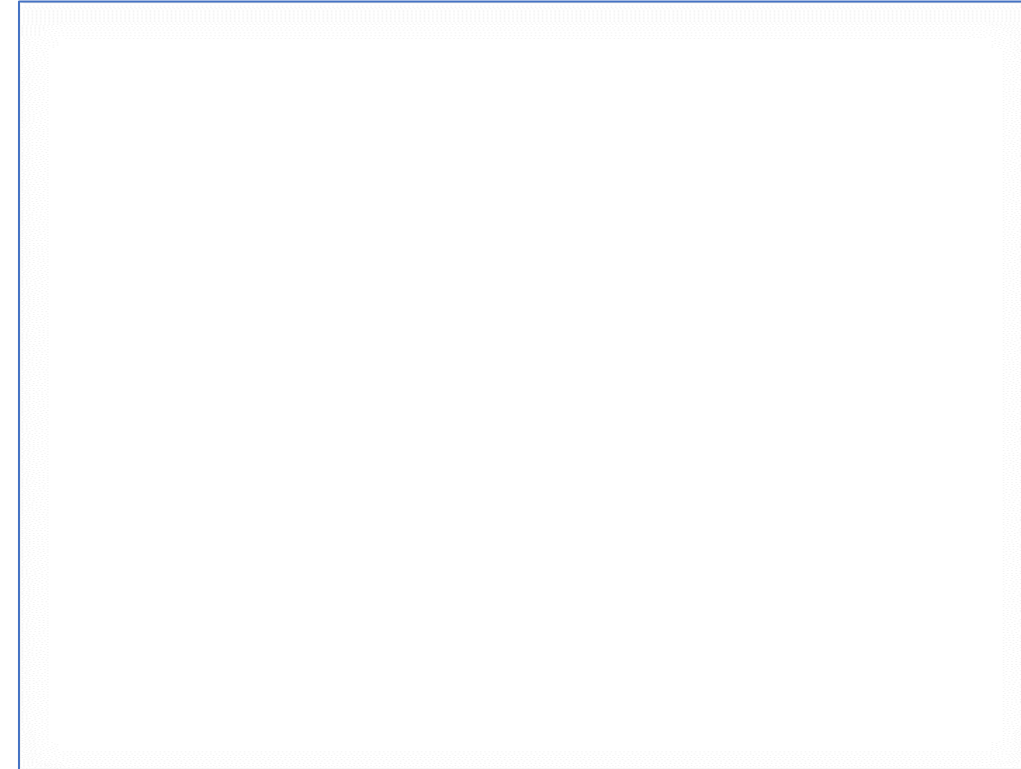
# *Scrolltelling: Boas práticas*

- Pensar e testar a experiência na interface mobile
  - Algumas dicas [aqui](#)
- Manter uma experiência coerente
  - Se precisar de uma ação diferente do scroll, que ela fique intuitiva
- Adquirir um bom domínio de HTML, CSS e JavaScript \*
  - Introdução ao HTML (em inglês)
    - ✓ [Parte 1](#) (HTML), [Parte 2](#) (CSS), [Parte 3](#) (CSS)

\* Esta não será uma aula de desenvolvimento web.

# *Scrolltelling: Funcionamento*

- Começa com uma parte estática
- O evento "scroll-to-trigger" serve de um gatilho
  - E mudança de estado nos gráficos
    - ✓ O elemento de gatilho é o texto, quando ele atingir um certo ponto
      - é hora de alterar o estado do gráfico
- Pode terminar com uma parte estática também





# *Como adicionar o Veja-Lite a uma Página HTML*

- Adicionar as tags script do veja-lite ao HTML

```
<script src="https://cdn.jsdelivr.net/npm/vega@5.21.0"></script>  
<script src="https://cdn.jsdelivr.net/npm/vega-lite@5.2.0"></script>  
<script src="https://cdn.jsdelivr.net/npm/vega-embed@6.20.2"></script>
```

- Adicionar uma div com um id. Exemplo: id="vis"

```
<div id="vis"></div>
```

- Chamar a função vegaEmbed num trecho de Javascript
  - `yourVlSpec` é o nome da variável contendo o seu gráfico

```
vegaEmbed('#vis', yourVlSpec);
```

# *Como adicionar o Vega-Lite a uma Página HTML*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Embedding Vega-Lite</title>
    <script src="https://cdn.jsdelivr.net/npm/vega@5.21.0"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-lite@5.2.0"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-embed@6.20.2"></script>
  </head>
  <body>
    <div id="vis"></div>
    <script type="text/javascript">
      var yourVLSpec = {
        $schema: 'https://vega.github.io/schema/vega-lite/v5.json',
        data: {...}, mark:'bar', encoding:{...}
      };
      vegaEmbed('#vis', yourVLSpec);
    </script>
  </body>
</html>
```

# Como converter um json para um javascript

- Use uma ferramenta online
  - Teste a opção minify

```
{ "$schema": "https://vega.github.io/schema/vega-lite/v5.json",  
  "data": {...},  
  "mark": "line",  
  "encoding": {  
    "x": {"timeUnit": "month", "field": "data1"},  
    "y": {"field": "Anomaly", "type": "quantitative"},  
    "color": {"field": "ano", "type": "ordinal", "scale": {"scheme": "turbo"}}  
  }  
}
```

```
{$schema: "https://vega.github.io/schema/vega-lite/v5.json",  
 data: {...},  
 mark: "line",  
 encoding: {  
   x: {timeUnit: "month", field: "data1"},  
   y: {field: "Anomaly", type: "quantitative"},  
   color: {field: "ano", type: "ordinal", scale: {scheme: "turbo"}}  
 }}
```





## Atividade 10.1 (5 min)

- Crie um arquivo HTML com um gráfico vega-lite **seu**
  - Use o Pycharm, VS Code ou o Notepad. Teste no navegador com o Pycharm.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Embedding Vega-Lite</title>
    <script src="https://cdn.jsdelivr.net/npm/vega@5.21.0"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-lite@5.2.0"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-embed@6.20.2"></script>
  </head>
  <body>
    <div id="vis"></div>
    <script type="text/javascript">
      var yourVlSpec = {
        $schema: 'https://vega.github.io/schema/vega-lite/v5.json',
        data: {...}, mark:'bar', encoding:{...}
      };
      vegaEmbed('#vis', yourVlSpec);
    </script>
  </body>
</html>
```

# Scrollama

- Russell Goldenberg ([The Pudding](#))
  - Fez a biblioteca javascript Scrollama, para implementar o scrolltelling
    - ✓ <https://github.com/russellgoldenbergs/scrollama/>
- Jonathan Soma
  - Fez um modelo simplificado de página HTML usando o Scrollama
    - ✓ <https://github.com/jsoma/simplified-scrollama-scrolltelling/>
      - E fez um tutorial básico e detalhado neste [vídeo](#)
- Alex Pereira
  - Fez um fork do repositório do Jonathan Soma
    - ✓ E adicionou gráficos do vega-lite (entre outras melhorias)
      - <https://github.com/alexlopespereira/scrollama-with-vegalite>

# *Como contar uma história com o Scrollama e o vega-lite*

- Fazer um clone do repositório
  - Pode ser no Pycharm
    - ✓ Gif de como fazer o clone no pycharm
- Editar o arquivo sticky-side.html ou sticky-overlay.html
  - Alterar os gráficos que ficam nas variáveis
    - ✓ yourSPec1, yourSPec2, yourSPec3, yourSPec4
- É recomendável que carregue os dados localmente
  - Utilize a pasta data e reference esse caminho na propriedade data do vega-lite. Exemplo:
    - ✓ data: { url: "data/temperature\_anomalies\_co2.csv", format: {type: "csv"}} }
- Altere os textos das tags p (linhas 148 a 157)

## *Atividade 10.2 (5 min)*

- Faça um clone do repositório
  - Gif da clonagem com o pycharm
- Faça uma página HTML com Scrolltelling e gráficos veja-lite
  - Substitua os gráficos dos arquivos
    - ✓ sticky-side.html ou sticky-overlay.html
      - por gráficos seus
- Gif de como testar no navegador usando o pycharm

# *Pequenos ajustes no Scrollteling com vega-lite*

- Para mover a linha de referência
  - Altere a propriedade offset da função init (linha 212)
- Para esconder a linha
  - Altere a propriedade debug da função init (linha 212)

```
function init() { // linha 212
  scroller.setup({
    step: "#scrolly article .step",
    offset: 0.6,
    debug: true
  })
  .onStepEnter(handleStepEnter);
  window.addEventListener("resize", scroller.resize);
}
```

## *Alterar a cor do fundo (cinza escuro)*

- Altere a propriedade background color da classe .sticky-thing

```
24
25  .sticky-thing {
26    position: -webkit-sticky;
27    position: sticky;
28    left: 0;
29    width: 100%;
30    margin: 0;
31    background-color: #8a8a8a;
32    z-index: 0;
```

Altere, por exemplo,  
para white



## *Alterar a cor do fundo (cinza claro)*

- Altere a propriedade background color do id #scrolly

```
11      #scrolly {  
12          position: relative;  
13          background-color: #f3f3f3;  
14          padding: 1rem;  
15          margin-bottom: 10rem;  
16      }
```

Altere, por exemplo,  
para white

# Alterar a cor e a transparência do fundo dos textos (Amarelo)

- Inspecione o retângulo colorido (div) no navegador
  - Edite e escolha a cor pelo navegador
    - ✓ Copie o código e altere da propriedade background-color
      - do identificador .step.is-active-p

```
57
58     .step.is-active p {
59         background-color: #daa520a8;
60         color: #3b3b3b;
61     }
```

Os últimos 2 dígitos  
se referem a  
transparência

# *Alterar a cor e a transparência do fundo dos textos (Cinza Escuro)*

- Inspecione o retângulo colorido (div) no navegador
  - Edite e escolha a cor pelo navegador
    - ✓ Copie o código e altere da propriedade background-color
      - do identificador .step p

```
62
63     .step p {
64         text-align: center;
65         padding: 1rem;
66         font-size: 1.5rem;
67         background-color: #3b3b3bc4;
68     }
```

Os últimos 2 dígitos  
se referem a  
transparência

# Otimização de Dados

- Usar dados inline e testar o desempenho
  - Código disponibilizado no arquivo [sticky-overlay\\_otimizado.html](#)
- Faltou medir a diferença de desempenho

```
232
233
234     var yourVlSpec1 = {$schema:"https://vega.github.io/schema/vega-lite/v5.json",width: 600,height: 400,
    $data:{values:[{Anomaly:-2.7,ano:1880,mes:"jan",mes_num:1,
    $data1:"1880-1-01",year:1880,co2:3852.3},{Anomaly:-2.33,ano:1880,mes:"feb",
    $mes_num:2,data1:"1880-2-01",year:1880,co2:3852.3},{Anomaly:-1.58,ano:1880,
    $mes:"mar",mes_num:3,data1:"1880-3-01",year:1880,co2:3852.3},{Anomaly:-0.
    $67,ano:1880,mes:"apr",mes_num:4,data1:"1880-4-01",year:1880,co2:3852.3},
```

# Gráfico com mapa no Vega-lite (Choropleth)

- Inspecione o retângulo colorido (div) no navegador

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v4.json",
  "width": 600, "height": 500,
  "mark": "geoshape",
  "data": {
    "url": "https://.../topojson_brasil_uf.json",
    "format": {"type": "topojson", "feature": "BRUF"}
  },
  "projection": {"type": "mercator"},
  "encoding": {"color": {"field": "properties.codarea",
    "type": "quantitative"}}
}
```



# Topojson (Dados Geográficos)

- Elimina redundância
  - Fronteiras são representadas apenas uma vez
- Topojson do Brasil (IBGE)
  - <https://servicodados.ibge.gov.br/api/docs/malhas?versao=3#api-Malhas-paisesIdGet>
- Topojson das UFs em formato json
  - <https://servicodados.ibge.gov.br/api/v3/malhas/paises/BR?intrarregiao=UF&formato=application/json>

```
{
  "type": "Topology",
  "objects": {
    "example": {
      "type": "GeometryCollection",
      "geometries": [
        {
          "type": "Point",
          "properties": {
            "prop0": "value0"
          },
          "coordinates": [102, 0.5]
        },
        {
          "type": "LineString",
          "properties": {
            "prop0": "value0",
            "prop1": "value1"
          },
          "coordinates": [[
            [100, 0],
            [101, 0],
            [102, 0.5]
          ]]
        }
      ]
    }
  }
}
```



# Gráfico com mapa no Vega-lite (Choropleth)

- Join (lookup) com dados de [consumo de energia](#)

```
{  "$schema": "https://vega.github.io/schema/vega-lite/v4.json",
  "width": 600, "height": 500, "mark": "geoshape",
  "data": {
    "url": "https://servicodados.ibge.gov.br/api/v3/malhas/paises/BR?intrarregiao=UF&formato=application/json",
    "format": {"type": "topojson", "feature": "BRUF"}},
  "transform": [{
    "lookup": "properties.codarea",
    "from": {
      "data": {"url": "https://raw.githubusercontent.com/alexlopespereira/scrollama-with-vegalite/main/data/consumo_2021_uf_copy.json"},
      "key": "codarea",
      "fields": ["rate"]
    }
  }],
  "projection": {"type": "mercator"},
  "encoding": {"color": {"field": "rate", "type": "quantitative"}}
}
```

## *Enquete para a próxima semana*

- Na última semana haverá tempo para abordar um novo assunto
  - ou reforçar/aprofundar um assunto já visto
- Vote aqui qual conteúdo você gostaria de reforçar/aprofundar
  - Na próxima semana

## *Exercício 10.1 (Deadline 13/12, 23:59)*

- Escolha um tema e seus respectivos dados, à sua conveniência
  - A história deve ser diferente da história do exercício 8.1
- Faça um relatório usando a técnica de Scrolltelling e Vega-lite
  - No formato de uma história
- Seu relatório/história deve conter
  - Pelo menos 3 gráficos
    - ✓ E para cada gráfico pelo 1 comentário
- Use a metodologia de ETL, DW e Scrolltelling (Scrollama)
  - apresentada no curso
    - ✓ Converta os dados para csv/json e importe no seu código
- Submeta [aqui](#) um arquivo zip contendo todo o projeto (dados, html, css e javascript)