Visualização Interativa

Parte 3/3 do curso de visualização computacional

Estagiário PAE: Eric Macedo Cabral
cabral.eric@usp.br

Docente: Maria Cristina Ferreira de Oliveira
cristina@icmc.usp.br

Motivação

- Permitir ao usuário interagir com os dados sendo visualizados, pode ampliar as oportunidades de obter novos insights
- O usuário pode explorar várias perspectivas sobre os dados
  - Alternar entre diferentes mapeamentos
- O usuário pode decidir o que é mais importante na visualização
  - Filtrar ou ressaltar dados
- Explorar os dispositivos de interação (teclado, mouse, tela, etc)

O que veremos neste módulo?

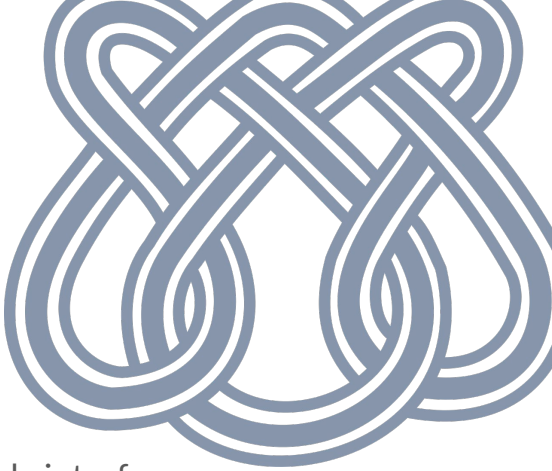
- Incorporação de elementos de interface ao Jupyter Notebook
- Como fazer a comunicação entre elementos de interface
- Controlar parâmetros de visualização por meio da interação
- Técnicas de interação

Considerações iniciais
Dependências

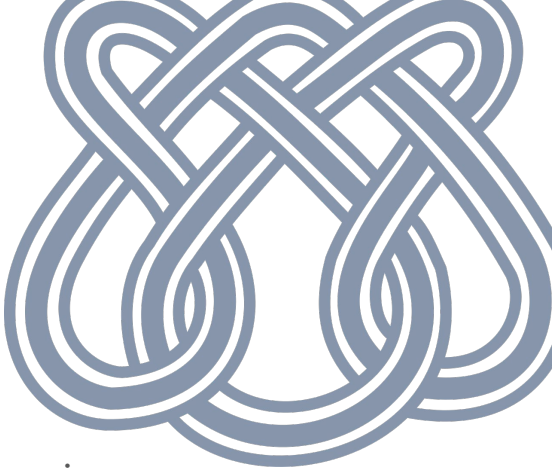
```
pip install plotly voila "ipywidgets>=7.5"
jupyter labextension install @jupyter-widgets/jupyterlab-manager
plotlywidget@4.12.0 jupyterlab-plotly@4.12.0 @jupyter-voila/jupyterlab-preview
```

- Plotly
- ipywidgets
- Voila

Sumário



1 Elementos de interface



1. Upload de arquivos
2. Caixa de texto + Botões
3. Dropdown
4. Sliders

ipywidgets

- Elementos HTML incorporados ao Jupyter Notebook
- Interativos e programáveis
- Baseado em eventos
  - Interação do usuário
  - Mudança do valor de alguma variável
  - Término do processamento de um algoritmo

Lista de elementos de interface

Upload de arquivos

- Formato do arquivo
  - “.csv”, “.pdf”, “*”...
- Múltiplos arquivos
- Formato binário

```
from ipywidgets import widgets

widgets.FileUpload(
      accept='*',
      multiple=False)
```

Caixa de texto + Botões

- Entrada do usuário
- String
- Query
  - Ações
  - Eventos

```
from ipywidgets import widgets

widgets.Text(
      value="Default",
      placeholder="Dica",
      description="Tooltip")

widgets.Button(
      description=”Label”,
      button_style='',
      icon=”search”,
      tooltip=””)
```

Dropdown

- Permite que o usuário selecione um valor dentre um conjunto de opções pré-definidas.

```
from ipywidgets import widgets

widgets.Dropdown(
      options=[Opções],
      value="Default",
      description="Label")
```

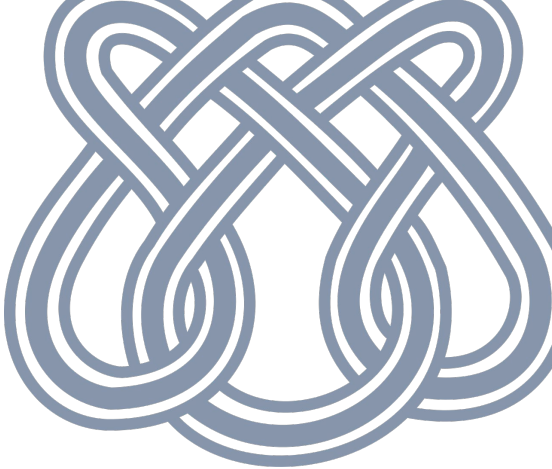
Sliders

- Valor flutuante dentro de um intervalo pré-definido

```
from ipywidgets import widgets

widgets.IntSlider(
      value=5,
      min=0,
      max=10,
      step=1)
```

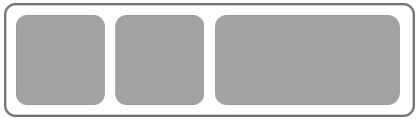
exemplos/01.ipynb

2 Layout



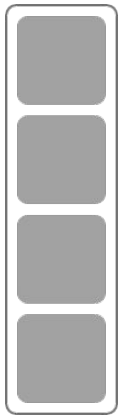
1. Caixas
2. Grade
3. Accordion
4. Abas
5. AppLayout

Caixas



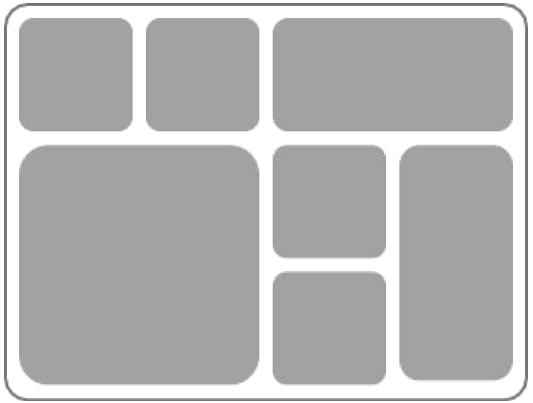
- Distribuição de elementos sequencialmente
- Horizontal ou verticalmente
- Aninhamento
    - Hierarquias

```
from ipywidgets import widgets

# Vertical
widgets.VBox([widgets])

# Horizontal
widgets.HBox([widgets])
```



Grade

- Disposição dos elementos numa matriz
- O tamanho de cada elemento é especificado por quantidade de células
  - Linhas + Colunas

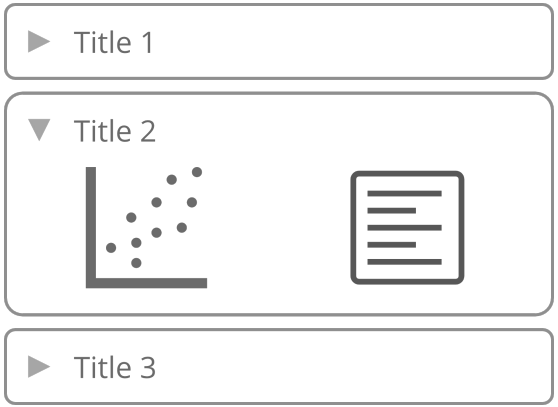


```
from ipywidgets import \
GridspecLayout, widgets

grid = GridspecLayout(rows, cols)

# Adiciona Button na primeira célula
grid[0, 0] = widgets.Button()
```

Accordion

- Contêiner com sistema de seções
- Mostra uma seção por vez
  - Collapse

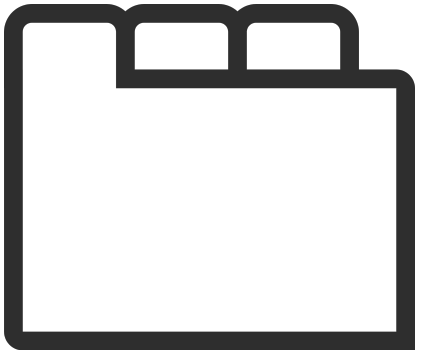
```
from ipywidgets import widgets

widgets.Accordion([widgets])
```



Abas

- Estrutura de conteiners não ordenados
- Mostra uma aba por vez

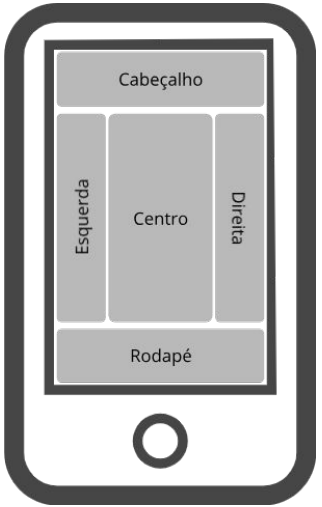
```
from ipywidgets import widgets

widgets.Tab([widgets])
```



AppLayout

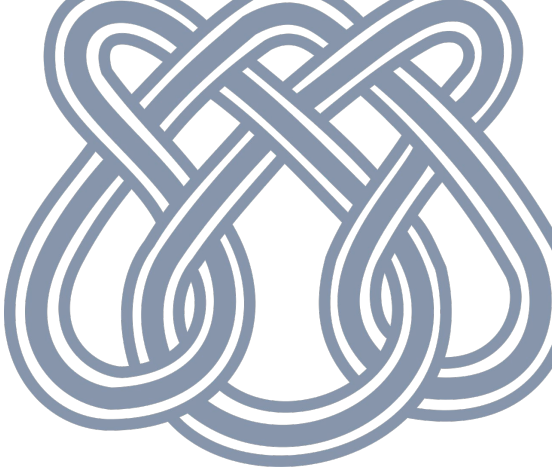
- Template com elementos pré-definidos
  - Geralmente utilizados em aplicações modernas
- Responsivo

```
from ipywidgets import AppLayout

AppLayout(
      header=header_widget,
      left_sidebar=left_widget,
      center=center_widget,
      right_sidebar=right_widget,
      footer=footer_widget)
```



exemplos/02.ipynb

3 Técnicas de interação



1. Hovering
2. Faceting
3. Linking and Brushing
4. Dynamic query

Hovering

- É uma forma de **Overview+Detail**
  - Não sobrecarrega o usuário com informações
  - Detalhes sob demanda
- Informações contextuais
- Pode complementar os insights obtidos por processos pré-atentivos

InfoVis Wiki

Faceting

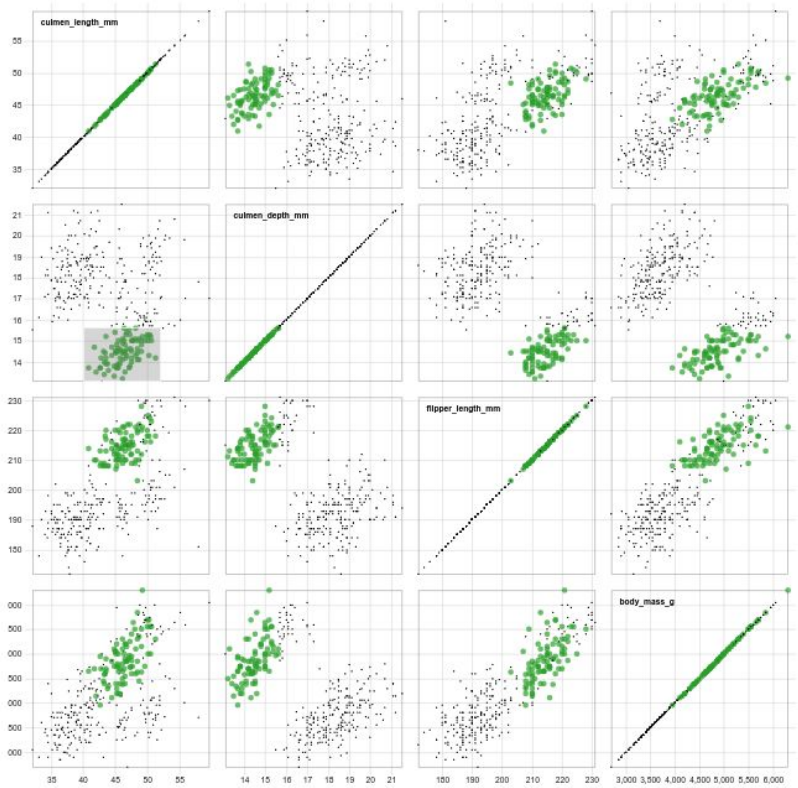
- Facet: cortar em partes, dividir (facetas = diferentes maneiras de olhar, perspectivas)
- Organiza os dados segundo múltiplas perspectivas
  - Apresenta perspectivas (propriedades, atributos) distintas dos dados
  - p.ex.: variação temporal, distribuição geográfica, etc.
- Camadas

InfoVis Wiki

Linking and Brushing

- Utiliza ferramentas de seleção de dados
- Permite que a seleção de dados em um mapeamento seja refletido em outros mapeamentos do mesmo conjunto de dados
- Auxiliar a compreensão sobre os dados pela combinação de elementos visuais

InfoVis Wiki



Fonte: Brushable Scatterplot Matrix

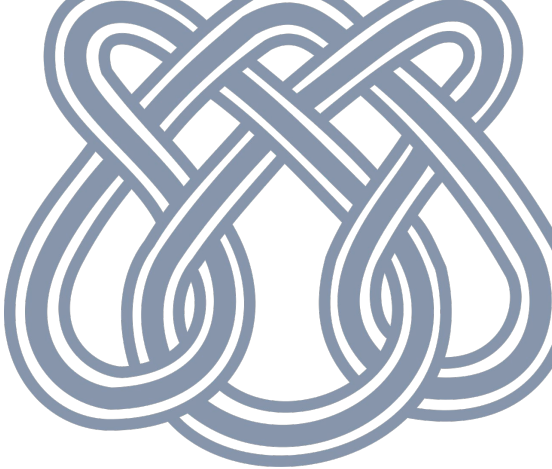
Dynamic query

- Permite a filtragem de parâmetros com feedback visual imediato
- Os dados entre as iterações devem ser relacionados
  - A mudança não pode ser drástica
  - Caso contrário, o usuário irá perder o mapa mental dos dados criado na iteração anterior
  - Incremental

InfoVis Wiki

exemplos/03.ipynb

4 Prototipação com Jupyter Notebooks



1. Voilà
2. Dash

Voilà

- Converte Jupyter Notebooks em Dashboards interativos
- Renderiza os *outputs* dos notebooks
  - Inclusive os *outputs* interativos
- Aceita temas e templates

Voilà

Dash

- Framework de criação de Dashboards interativos em Python
- Aceita os gráficos do Plotly nativamente
- Criar páginas web do zero
- Altamente configurável
- Freemium

Plotly Dash

Projeto etapa 3

Descrição

1. Nesta etapa final, você deve entregar um sistema de Visualização Analítica (obtido com Voilà, p.e.)
2. Seguindo os feedbacks recebidos nas etapas anteriores, aplique uma ou mais técnicas de interação nas suas visualizações de dados (Etapa 2)
3. Caso seja necessário (e possível), permita ao usuário final do seu sistema interagir com os algoritmos de processamento, de forma que ele/ela possa decidir qual parâmetro funciona melhora para o seu caso. Por exemplo: número de clusters,, alternar entre técnicas de normalização, etc.

Projeto etapa 3

Organização

Arquivo ZIP contendo:

- Jupyter notebook (Python Versão 3.*) - Código e sistema
- Documentação do sistema completo em PDF (Pode aproveitar documentação das etapas 1 e 2)
- Arquivos externos necessários (.csv, .py, .json, etc...)

A documentação deve estar em um arquivo separado para que o arquivo .ipynb seja dedicado ao sistema.

Projeto etapa 3

Entrega

- Até 30/11/2020 às 23:55
  - **No eDisciplinas**
  - Apenas um membro do grupo
  - Mesmo grupo das etapas anteriores

Seminário final

Instruções

Você deve produzir um seminário apresentando seu sistema (um `vídeo relatório´), que consiste em introduzir os **dados**, as técnicas de **processamento**, **visualização** e **interação** adotadas. Você deve motivar a escolha dos dados (relevância, desafios de análise), justificar a escolha das técnicas e mapeamentos visuais e recursos de interação, e apresentar como cada parte compõe seu sistema.

Especificações:

- Tempo: 10-12 minutos
- Todos membros devem participar da apresentação
- O seminário será gravado, mas os membros devem estar na aula para responder perguntas
- Datas dos seminários: 01/12 e 08/12

Seminário final

Entrega

- O vídeo deve ser enviado no **eDisciplinas**
  - Até 30/11/2020 às 23:55
  - Enviar link do vídeo na submissão (Google Drive)