



# Visualização de dados em Python

Parte 2/3 do curso de visualização computacional

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

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



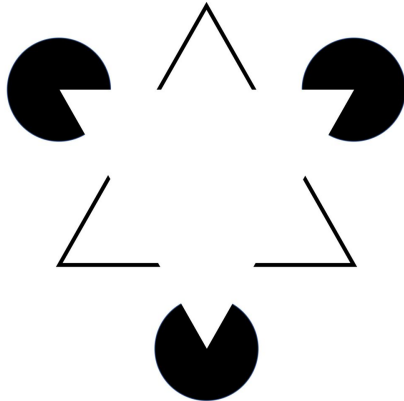
# Motivação

- A visualização de dados no formato tabular pode exigir muito esforço
  - Falhas de atenção
  - Sobrecarga de informação
- Capacidade cognitiva do ser humano
  - “O sistema visual humano tem um canal amplo para os nossos cérebros” <sup>1</sup>
- Informações são perdidas durante o processo de sumarização estatística <sup>1</sup>

1. Munzner, T. (2014). Visualization Analysis and Design. A K Peters/CRC Press

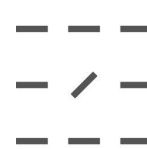
# Motivação

## Processamento pré-atentivo e Princípios de Gestalt



Fonte: [The 7 Gestalt Principles of Visual Perception \(Pt 2\)](#)

Fonte: [Lies, Damn Lies, and Data Visualisation \(2019\)](#)



Orientation



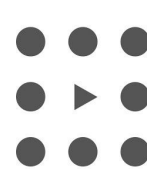
Length



Width



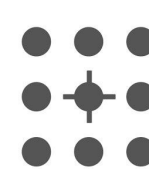
Size



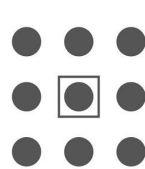
Shape



Curvature



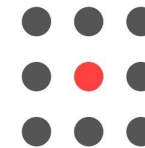
Added Marks



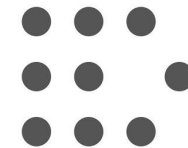
Enclosure



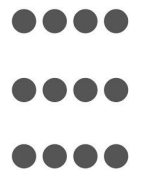
Contrast



Colour



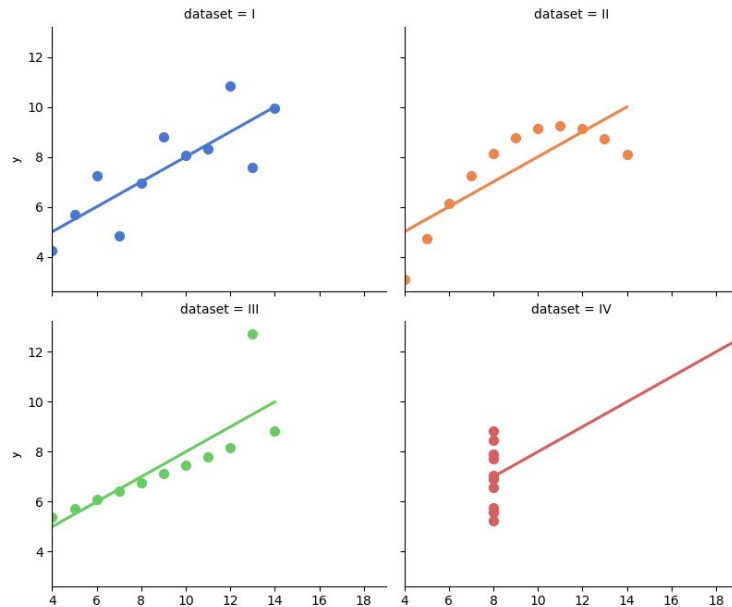
Position



Spatial Grouping

# Motivação

## Anscombe Quartet



I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.7	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.5
12.0	10.8	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

---

`exemplos/00.ipynb`





# O que veremos neste módulo?

- Bibliotecas gratuitas de visualização de dados
  - Plotly
  - Matplotlib
  - Seaborn
- Transformação de dados para posteriormente visualizá-los
- Como mapear dados em representações visuais abstratas

# Considerações Iniciais

## Taxonomia

- A técnicas de visualização vistas nesta aula estão catalogadas seguindo um critério de dados
- Qual técnica de visualização é mais comumente utilizada em uma determinada natureza de dados

Natureza dos dados → Visualização

<https://www.data-to-viz.com/>



from Data to Viz

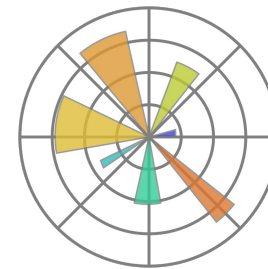
# Considerações Iniciais

## Dependências

- Dependências:

```
pip install plotly matplotlib "ipywidgets>=7.5"
jupyter labextension install @jupyter-widgets/jupyterlab-manager \
plotlywidget@4.11.0 jupyterlab-plotly@4.11.0
```

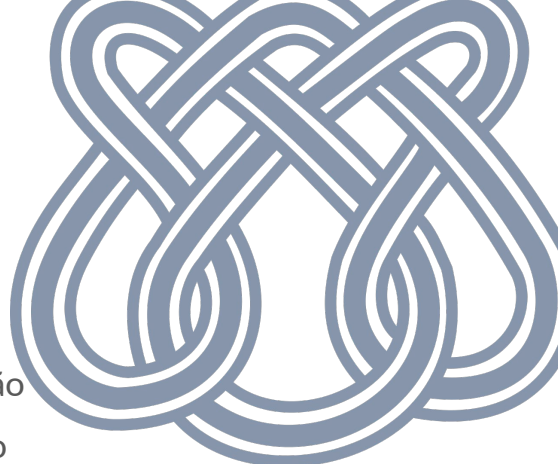
- [Plotly](#)
- [Matplotlib](#)
- [ipywidgets](#)





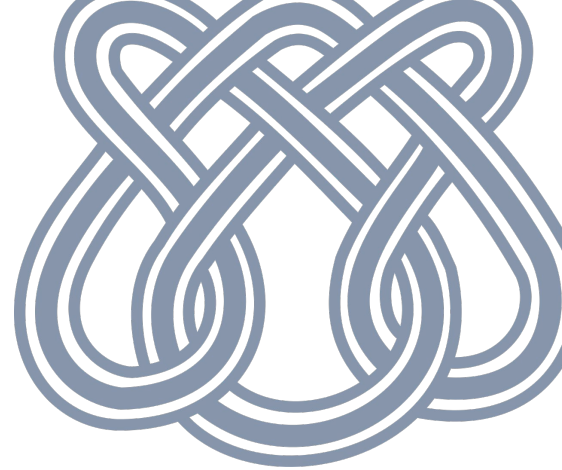


# Sumário

- 
1. Distribuição
  2. Correlação
  3. Ranqueamento
  4. Parte de um todo (Hierarquias)
  5. Evolução
  6. Geográficos
  7. Fluxos



# 1 Distribuição



1. Histograma
2. Boxplot
3. Violin plot

# Distribuição



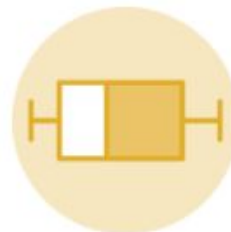
Violin



Density



Histogram



Boxplot



Ridgeline

# Histograma

- Distribuição de **dados numéricos**
- Bins
  - Explore valores de Bins
- Não é o mesmo que um gráfico de barras

<https://www.data-to-viz.com/graph/histogram.html>

<https://plotly.com/python/histograms/>

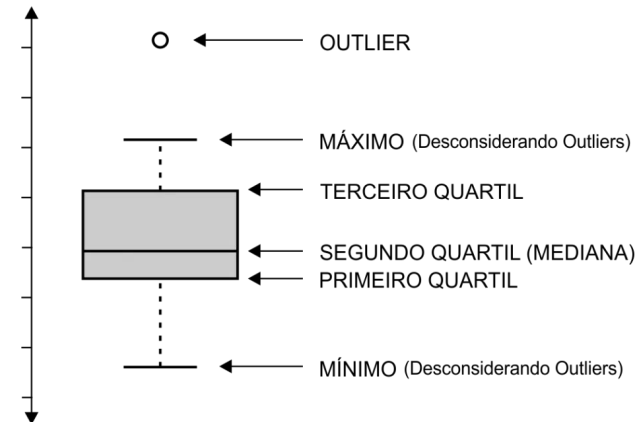
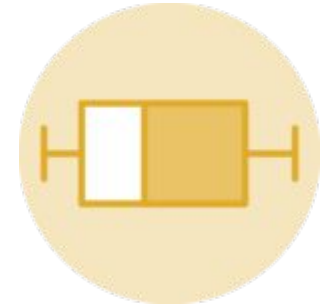


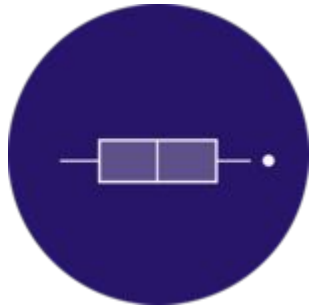
# Boxplot

- Representação estatística de variáveis através de seus quartis
  - IQR
- Não representa quantidade de observações
- Outliers

<https://www.data-to-viz.com/caveat/boxplot.html>

<https://plotly.com/python/box-plots/>

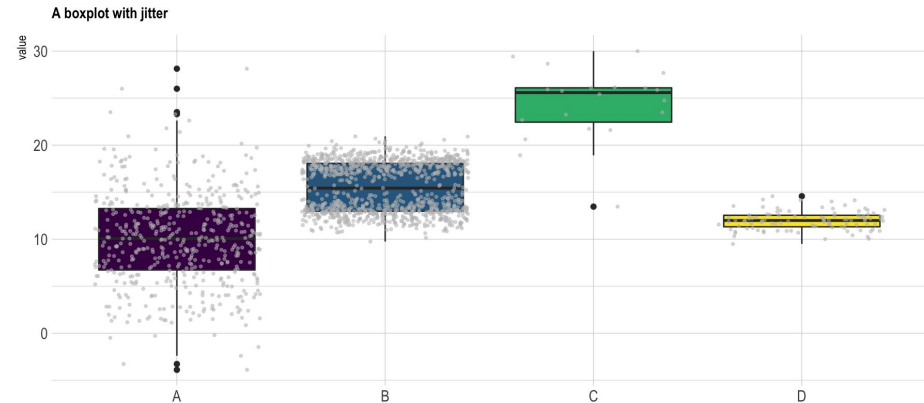




## Caveat

### Desvantagens do Boxplot

- Sumarização implica em perda de informação
- Plotar os pontos de dados pode gerar *overplotting*
- Em muito dos casos, Violin plot resolve



<https://www.data-to-viz.com/caveat/boxplot.html>

# Violin plot

- Representação estatística de dados numéricos
- Densidade de kernel
  - Quantidade de observações

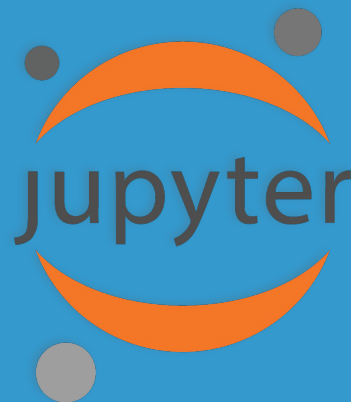
<https://www.data-to-viz.com/graph/violin.html>

<https://plotly.com/python/violin/>



---

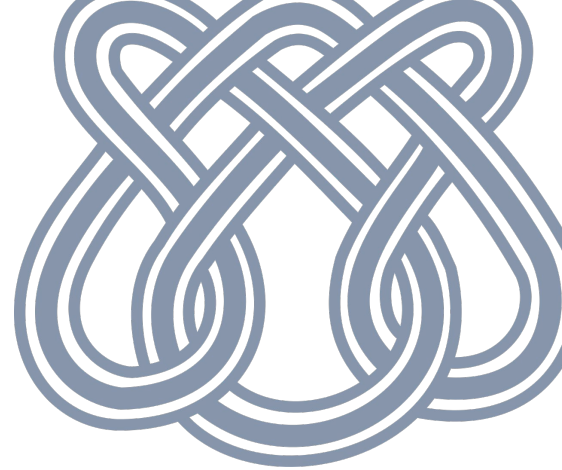
`exemplos/01.ipynb`







## 2 Correlação



1. Gráficos de dispersão
  - a. Scatter plot
  - b. Matriz de Scatter Plot
2. Mapa de calor (Heatmap)

# Correlação



Scatter



Heatmap



Correlogram



Bubble



Connected scatter



Density 2d

# Gráficos de dispersão

## Scatter plot

- Distribuição entre 2 variáveis numéricas
- Pode ser enriquecido por distribuições marginais
- Evite Overplotting

<https://www.data-to-viz.com/graph/scatter.html>

<https://plotly.com/python/line-and-scatter/>





# Caveat

## O paradoxo de Simpson

- Correlações (aparentemente) positivas
- Correlações espúrias

<https://www.tylervigen.com/spurious-correlations>

<https://www.data-to-viz.com/caveat/simpson.html>

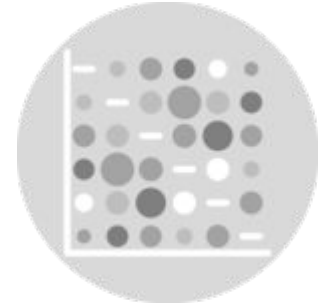
# Gráficos de dispersão

## Matriz de Scatter plot (Correlograma)

- Útil para análise exploratória
- Visualiza as relações entre as diversas variáveis do conjunto de dados
- Evite mais do que 9 variáveis

<https://www.data-to-viz.com/graph/correlogram.html>

<https://plotly.com/python/splom/>

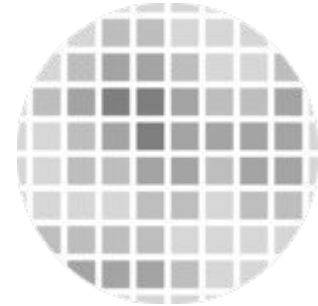


# Mapa de calor (*Heatmap*)

- Representa valores numa tabela por intensidades de cores
- Dados normalizados
- Séries temporais

<https://www.data-to-viz.com/graph/heatmap.html>

<https://plotly.com/python/heatmaps/>



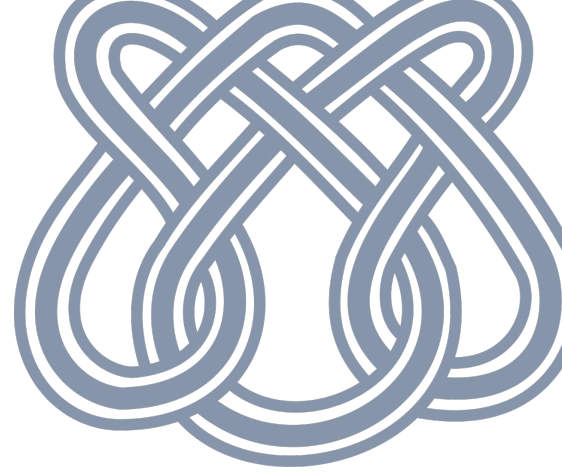
---

`exemplos/02.ipynb`





## 3 Ranqueamento



1. Gráfico de barras
2. Coordenadas paralelas
3. Nuvem de palavras



# Ranqueamento



Barplot



Spider / Radar



Wordcloud



Parallel



Lollipop



Circular Barplot

# Gráfico de barras

- Relação entre uma variável categórica e uma métrica numérica
- Barras ordenadas são mais intuitivas
- Não é um histograma

<https://www.data-to-viz.com/graph/barplot.html>

<https://plotly.com/python/bar-charts/>



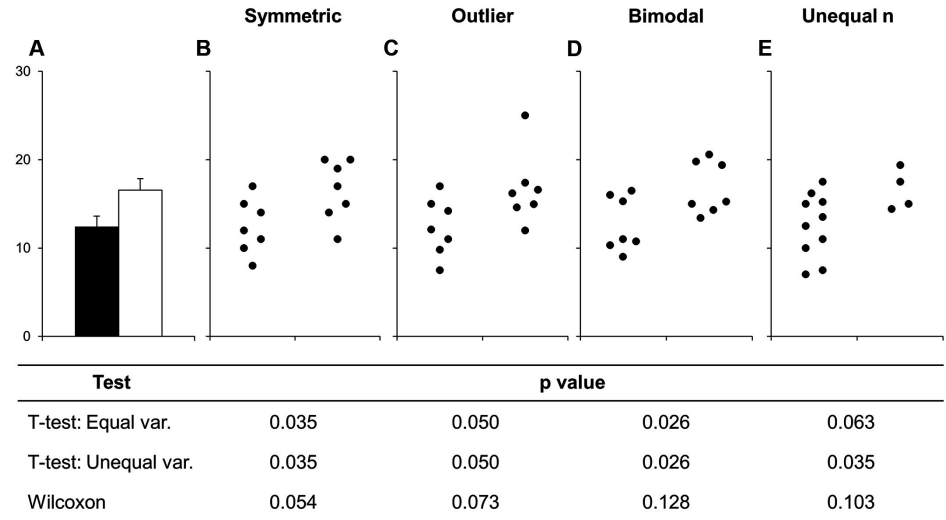


# Caveat

## O problema das barras de erros

- Barras de erros são úteis, mas escondem informações, como no Boxplot
- Use gráficos de distribuição
  - Boxplot (em alguns casos)
  - Violin

[https://www.data-to-viz.com/caveat/error\\_bar.html](https://www.data-to-viz.com/caveat/error_bar.html)



# Coordenadas paralelas

- Comparação entre múltiplas variáveis
  - Podem ser heterogêneas
- Eixos verticais
- Relações entre variáveis
- Evite *overplotting* (*Spaghetti plot*)



<https://www.data-to-viz.com/graph/parallel.html>

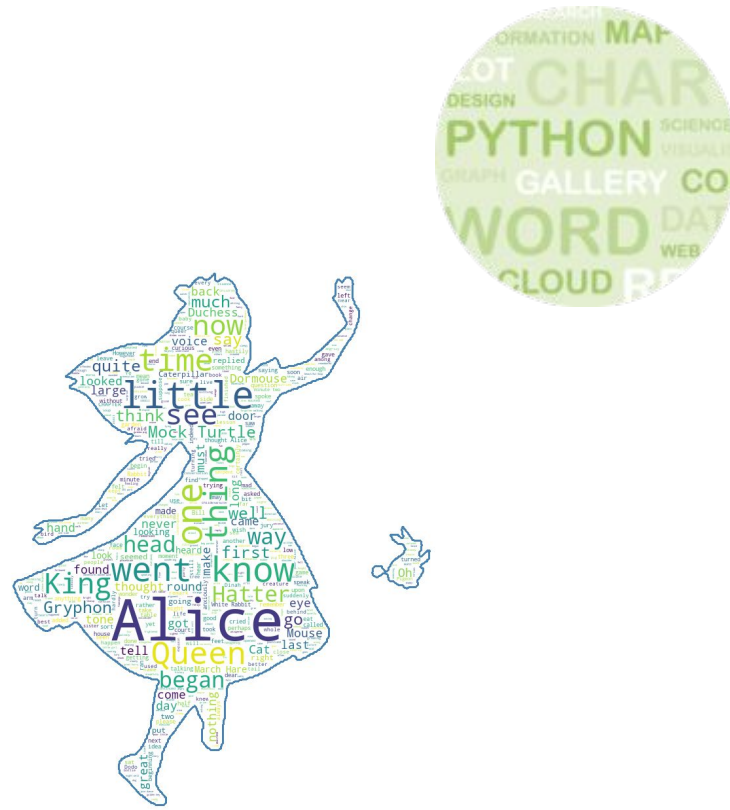
<https://plotly.com/python/parallel-coordinates-plot/>

# Nuvem de palavras

- Representação da relevância de palavras
  - **Cor**
  - **Tamanho**
- Máscaras (formas)

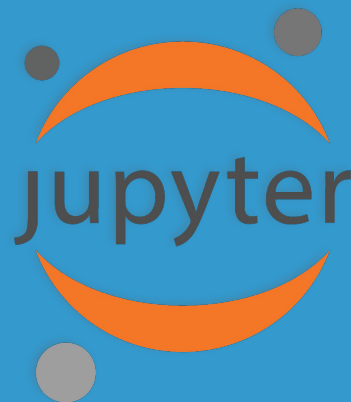
<https://www.data-to-viz.com/graph/wordcloud.html>

[https://amueller.github.io/word\\_cloud/](https://amueller.github.io/word_cloud/)



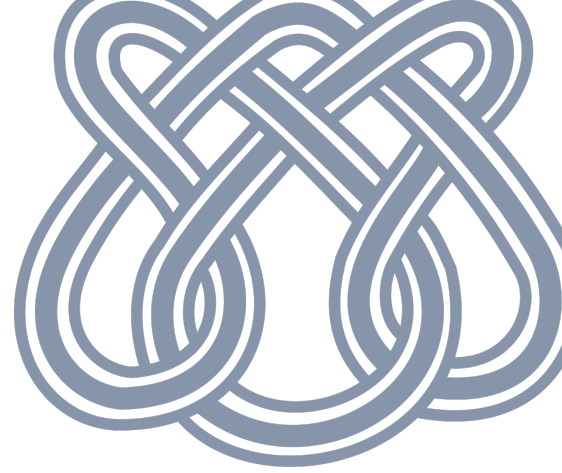
---

`exemplos/03.ipynb`





## 4 Parte de Um Todo (Hierarquias)



1. Gráfico de setores
2. Treemap
3. Sunburst

## Parte de Um Todo (Hierarquias)



Treemap



Venn diagram



Doughnut



Pie chart



Dendrogram



Circular packing

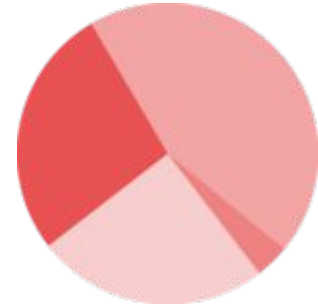


Sunburst



# Gráfico de setores

- Gráfico de “pizza”
- Representa porções de um todo (%)
  - Soma de todos os setores = 100%
- Não confundir com o *Sunburst Plot*



<https://www.data-to-viz.com/caveat/pie.html>

<https://plotly.com/python/pie-charts/>

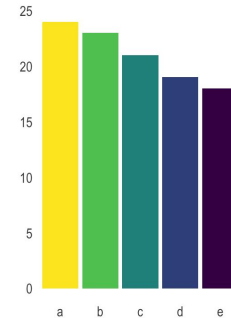
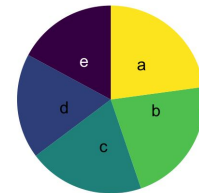
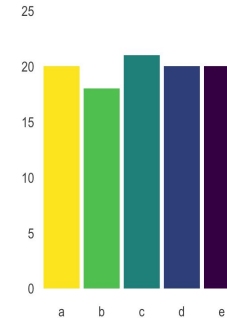
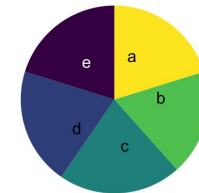
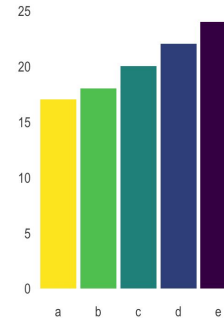
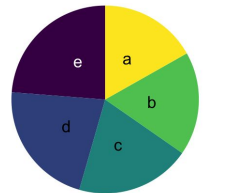


# Caveat

## O problema dos gráficos de setores

- É difícil para os usuários medir áreas e ângulos

What to consider when creating pie charts



# Treemap

- Dados hierárquicos
  - Grupos
  - Retângulos
- Área proporcional ao valor do grupo
- Uso eficiente de espaço



<https://www.data-to-viz.com/graph/treemap.html>

<https://plotly.com/python/treemaps/>

# Sunburst

- Mistura características do Treemap e do gráfico de setores
- Porém, também herda boa parte de suas desvantagens

<https://www.data-to-viz.com/graph/sunburst.html>

<https://plotly.com/python/sunburst-charts/>



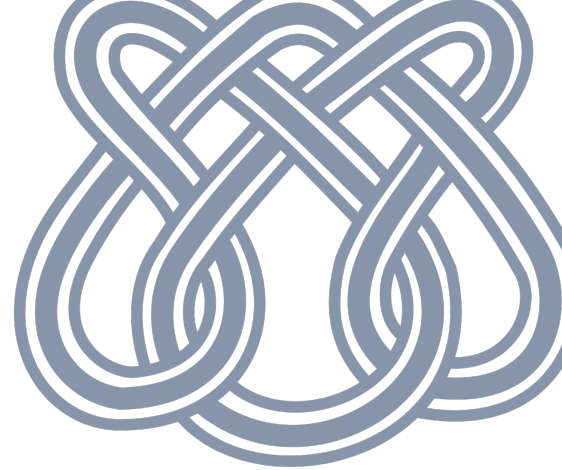
---

`exemplos/04.ipynb`





## 5 Evolução



1. Gráfico de linhas
2. Gráfico de área

# Evolução



Line plot



Area



Stacked area



Streamchart

# Gráfico de linhas

- Representa a evolução de uma ou várias variáveis numéricas
- Também utilizado em Scatter plots para representar tendências e padrões
  - p.e. Linha de regressão
- Também sofre do problema de Spaghetti plot



<https://www.data-to-viz.com/graph/line.html>

<https://plotly.com/python/line-charts/>



# Gráfico de área

- Representa a evolução de um **conjunto** de dados todo
- Grupos
  - **Proporções relativas**
- Valor relativo representado pela largura da “onda” no ponto x



<https://www.data-to-viz.com/graph/stackedarea.html>

<https://plotly.com/python/filled-area-plots/>

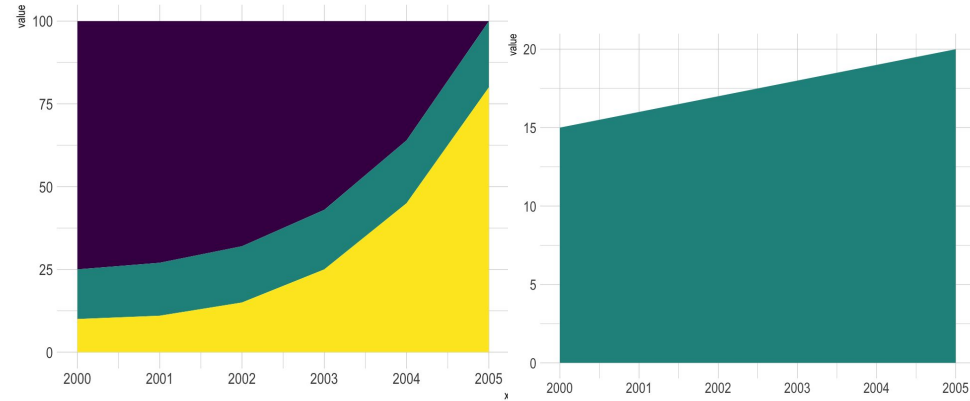


# Caveat

## O problema com empilhamento

- Empilhamento de áreas pode representar bem a evolução do todo
- Mas pode levar a interpretações errôneas sobre as partes

<https://www.data-to-viz.com/caveat/stacking.html>



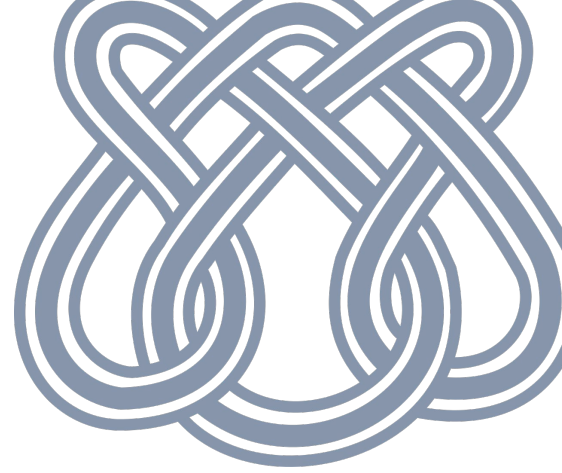
---

`exemplos/05.ipynb`





## 6 Geográficos



1. Mapa de bolhas
2. Mapas de Choropleth

# Evolução



Map



Choropleth



Hexbin map



Cartogram



Connection



Bubble map

# Mapa de bolhas



- Representação geográfica de valores numéricos
  - Os valores devem ser codificado na área do círculo, não em seu raio

<https://www.data-to-viz.com/graph/bubblemap.html>

<https://plotly.com/python/bubble-maps/>

# Mapas de Choropleth

- Uma espécie de mapa de calor representado em dados geográficos
- Os dados devem estar normalizados
- Regiões com áreas maiores tendem a tirar a atenção de outras menores
- Não se chama chLoropleth



<https://www.data-to-viz.com/graph/choropleth.html>

<https://plotly.com/python/choropleth-maps/>

---

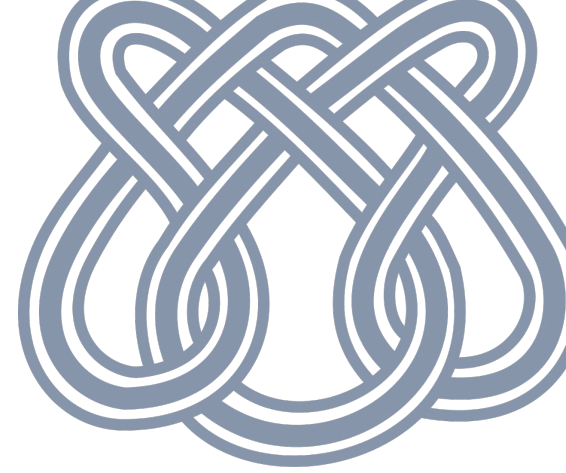
`exemplos/06.ipynb`







## 7 Fluxos



1. Diagrama de Sankey

# Fluxos



Chord diagram



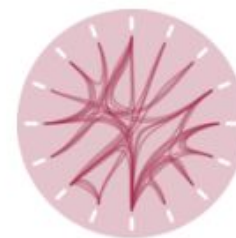
Network



Sankey



Arc diagram



Edge bundling

# Diagrama de Sankey

- Também conhecido como “Diagrama Aluvial”
  - É uma forma específica de diagrama de Sankey
- Entidades são representadas por retângulos
- Fluxos são representados por arestas entre entidades
- Representa a evolução de dados e a interação entre entidades



<https://www.data-to-viz.com/graph/sankey.html>

<https://plotly.com/python/sankey-diagram/>

---

`exemplos/07.ipynb`



# Projeto etapa 2

## Descrição

1. Com seus dados pré-processados, identifique em qual grupo taxonômico seu conjunto de dados se encaixa (p.e. categórico, numérico, híbrido, etc...). Justifique com uma explicação dos seus dados.
    - a. Vide o catálogo [From Data to Viz](#)
  2. Identifique qual mapeamento visual é o mais indicado para os seus dados e, se você julgar que aquele paradigma visual realmente é a melhor escolha, apresente seus dados com a visualização
    - a. Caso decida utilizar outra visualização, justifique a escolha
  3. Descreva os insights que a visualização proporcionou para os seus dados
-

# Projeto etapa 2

## Organização

Arquivo ZIP contendo:

- Jupyter notebook (Python Versão 3.\*) - Código e documentação
- Arquivos externos necessários (.csv, .py, .json, etc...)

Aproveite as funcionalidades do Jupyter para enriquecer e organizar a documentação com fórmulas, tabelas e figuras. Lembre-se que você está entregando um relatório!

---

# Projeto etapa 2

## Entrega

- Até 01/11/2020 às 23:55
  - No eDisciplinas
  - Apenas um membro do grupo
  - Mesmo grupo da etapa anterior

