

Departamento de Engenharia de Sistemas e Computação

Visualização de Dados com Matplotlib

Gráficos de linha, dispersão, histograma e caixa

João Araujo

Algoritmos Computacionais I

Aula 27

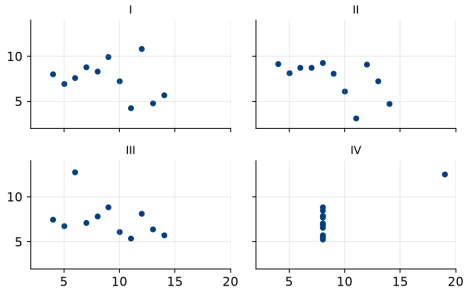
Roteiro

- Por que visualizar dados? O quarteto de Anscombe
- O básico: `pyplot`, `plot`, `show`
- Figuras e eixos (`Figure` e `Axes`); *subplots*
- Gráficos estatísticos: histograma, dispersão, *boxplot*, barras
- Rótulos, legendas e como salvar a figura

Material: notebook `notebooks/tutorial_de_matplotlib.ipynb`. Todas as figuras desta aula são geradas por `slides/figuras/gera_figuras.py`.

Por que visualizar? O quarteto de Anscombe

Quarteto de Anscombe: mesma média, mesma reta, dados diferentes



No notebook: `sns.load_dataset("anscombe")`.

- Quatro conjuntos de dados com **estatísticas praticamente idênticas**: mesma média de x e y, mesma variância, mesma correlação ($\approx 0{,}816$) e a mesma reta de regressão
- Mas os dados são completamente diferentes!
- Moral: só números resumidos não bastam; **olhe os dados**

Primeiro gráfico

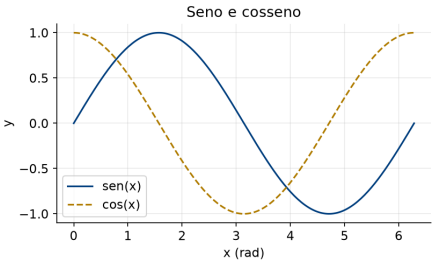
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 200)
plt.plot(x, np.sin(x))
plt.show()              # necessário para exibir
```

- `matplotlib.pyplot` (apelido `plt`) concentra as funções de desenho
- `plt.plot(x, y)` desenha **linhas**; `plt.plot(x, y, 'o')` desenha **pontos**
- Em scripts, `plt.show()` abre a janela; em Jupyter o gráfico aparece sozinho
- Instalação: `pip install matplotlib`

Figura e eixos: a forma recomendada

```
fig, ax = plt.subplots()
ax.plot(x, np.sin(x), label="sen(x)")
ax.plot(x, np.cos(x), "--", label="cos(x)")
ax.set_title("Seno e cosseno")
ax.set_xlabel("x (rad)")
ax.set_ylabel("y")
ax.legend()
fig.savefig("fig_linha.png", dpi=150)
```

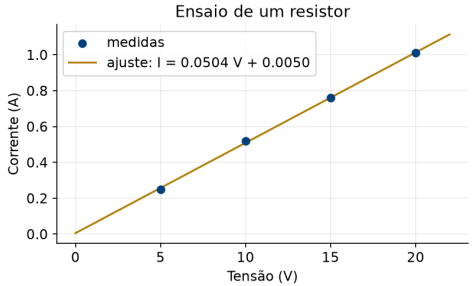


- **Figure**: a tela inteira; **Axes**: um gráfico dentro dela
- Título, rótulos com unidades e legenda: sempre
- `savefig` grava PNG, PDF ou SVG

Dispersão e ajuste: o ensaio do resistor

```
V = np.array([5.0, 10.0, 15.0, 20.0])
I = np.array([0.25, 0.52, 0.76, 1.01])
a, b = np.polyfit(V, I, 1)

fig, ax = plt.subplots()
ax.scatter(V, I, label="medidas")
xx = np.linspace(0, 22, 50)
ax.plot(xx, a * xx + b,
        label=f"I = {a:.4f} V + {b:.4f}")
ax.set_xlabel("Tensão (V)")
ax.set_ylabel("Corrente (A)")
ax.legend()
plt.show()
```



Dados da Aula 19, ajuste da Aula 21; agora *visualizados*.

Vários gráficos: *subplots*

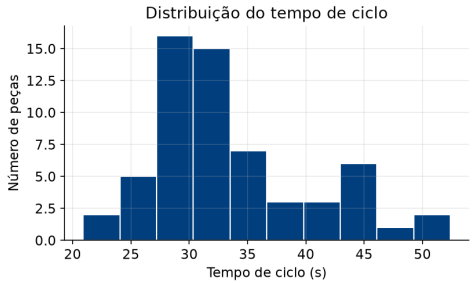
```
fig, axs = plt.subplots(2, 2, figsize=(6, 4), sharex=True, sharey=True)
for ax, (xi, yi) in zip(axs.ravel(), conjuntos):
    ax.scatter(xi, yi)
fig.suptitle("Quarteto de Anscombe")
fig.tight_layout()
```

- `plt.subplots(linhas, colunas)` devolve a figura e uma *matriz* de eixos
- `sharex`/`sharey` compartilham as escalas: facilita comparar
- `figsize=(largura, altura)` em polegadas; `tight_layout()` evita sobreposição
- No notebook usa-se `fig.add_subplot(2, 2, k)`: mesmo resultado

Histograma: a distribuição de uma variável

```python
import pandas as pd
df = pd.read_csv("dadosproducao.csv")

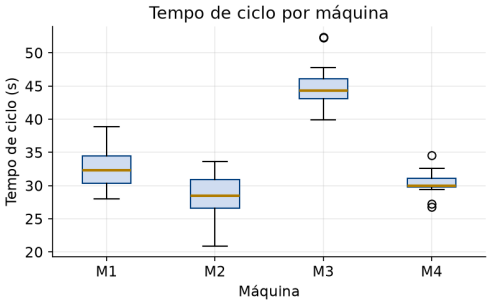
fig, ax = plt.subplots()
ax.hist(df["tempo_ciclo_seg"], bins=10)
ax.set_title("Distribuição do tempo de ciclo")
ax.set_xlabel("Tempo de ciclo (s)")
ax.set_ylabel("Número de peças")
plt.show()
```



- Os valores são agrupados em faixas (*bins*); a altura é a contagem
- Dois "montes" sugerem dois grupos: investigue (veja o *boxplot*)

Boxplot: comparando grupos

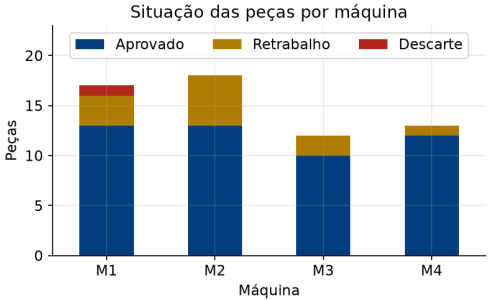
```
maqs = sorted(df["maquina"].unique())
dados = [df.loc[df.maquina == m,
                "tempo_ciclo_seg"]
         for m in maqs]
fig, ax = plt.subplots()
ax.boxplot(dados, tick_labels=maqs)
ax.set_xlabel("Máquina")
ax.set_ylabel("Tempo de ciclo (s)")
plt.show()
```



- Caixa = quartis (25%–75%); linha = mediana; pontos = *outliers*
- A máquina M3 é claramente mais lenta: explica o segundo "monte" do histograma

Barras: contagens por categoria

```
tab = (df.groupby("maquina")["status"]
         .value_counts()
         .unstack(fill_value=0))
tab.plot(kind="bar", stacked=True)
plt.ylabel("Peças")
plt.show()
```



- Barras para **categorias**; histograma para **variáveis contínuas**
- O Pandas desenha direto com `.plot()` (usa o Matplotlib por baixo): Aulas 28 e 29

Dados multivariados: cor, tamanho e transparência

```
cores = {"Female": "#f1a340", "Male": "#998ec3"}
fig, ax = plt.subplots()
ax.scatter(data=tips, x="total_bill", y="tip",
           c=tips["sex"].map(cores),        # cor = variável categórica
           s=tips["size"] * 10,             # tamanho = outra variável
           alpha=0.5)                       # transparência
ax.set_xlabel("Conta total")
ax.set_ylabel("Gorjeta")
```

- Exemplo do notebook (conjunto `tips`, gorjetas em restaurante)
- Use cor para *categorias*, tamanho com moderação, `alpha` para evitar sobreposição

Boas práticas de gráficos

Faça

- Título, rótulos **com unidades**, legenda
- Escolha o gráfico certo para o tipo de dado
- Cores com bom contraste (e que funcionem em preto e branco)
- Salve em alta resolução (`dpi=150` ou mais) ou em PDF/SVG

Evite

- Eixos truncados que exageram diferenças
- Gráficos de pizza com muitas fatias
- Efeitos 3D desnecessários
- Poluição visual (grades pesadas, muitas cores)

Qual gráfico usar?

Quero mostrar...	Gráfico	Função
evolução ao longo do tempo / função	linha	`ax.plot`
relação entre duas variáveis	dispersão	`ax.scatter`
distribuição de uma variável	histograma	`ax.hist`
comparar distribuições entre grupos	*boxplot*	`ax.boxplot`
comparar valores entre categorias	barras	`ax.bar`

Atividade no laboratório

1. Plote $f(x) = e^{-x/5}\cos(2x)$ para $x \in [0, 20]$, com título, rótulos e grade.
2. Com os dados do ensaio (Aula 19), faça a dispersão e a reta de ajuste; informe a resistência estimada no título.
3. Com `code/Projeto1/dadosproducao.csv`, faça o histograma do tempo de ciclo e o *boxplot* por turno.
4. Faça uma figura com 2 gráficos lado a lado (`plt.subplots(1, 2)`): seno e cosseno em eixos separados.
5. Salve a figura em PDF e insira-a em um documento.

Para praticar e consultar

- Notebook da aula: `notebooks/tutorial_de_matplotlib.ipynb`
- Galeria de exemplos: `matplotlib.org/stable/gallery/`
- Guia rápido: *cheat sheets* do Matplotlib
- Exercícios adicionais do livro: `https://www.araujo.eng.uerj.br/livro/extras/`

Conexão com o projeto

Todo projeto final deve apresentar pelo menos dois gráficos que ajudem a responder a pergunta do problema.

Resumo da aula

- Visualizar é parte da análise: estatísticas iguais podem esconder dados diferentes
- Use `fig, ax = plt.subplots()` e configure título, rótulos e legenda
- Escolha o gráfico pelo tipo de dado: linha, dispersão, histograma, *boxplot*, barras
- Salve com `fig.savefig`
- Próxima aula: Pandas para carregar e manipular tabelas de dados