

Introdução ao Matplotlib



CRONOGRAMA DA AULA

1

O que é Matplotlib e
Por que Usar?

2

Instalação e
Configuração

3

API Básica: Figuras e
Subplotagens

4

Plotando Dados:
Estilos e Elementos

5

Adicionando Texto e
Formas

6

Importância dos
Gráficos na Análise

7

Tipos de Gráficos
Comuns

8

Integração com
Pandas e
Personalização

O QUE É MATPLOTLIB E POR QUE USAR?

Biblioteca de Visualização



Matplotlib é a biblioteca mais amplamente utilizada em Python para criar plotagens em duas dimensões.

Integração Perfeita



Funciona perfeitamente com outras bibliotecas do ecossistema de dados como pandas e NumPy.

Comunicação Visual



Permite resumir conjuntos de dados complexos, destacando tendências e padrões não óbvios.

INSTALAÇÃO E CONFIGURAÇÃO

Instalação

Matplotlib pode já estar incluída em distribuições como Anaconda. Caso contrário, instale via pip: `pip install matplotlib`.

Configuração em Ambientes

Use `%matplotlib inline` em notebooks Jupyter para exibir gráficos estaticamente no notebook.

Importação Padrão

A convenção recomendada é: `import matplotlib.pyplot as plt`

INDENTAÇÃO EM PYTHON



Elemento Estrutural

Define blocos de código



Herança da ABC

Conceito herdado da linguagem ABC

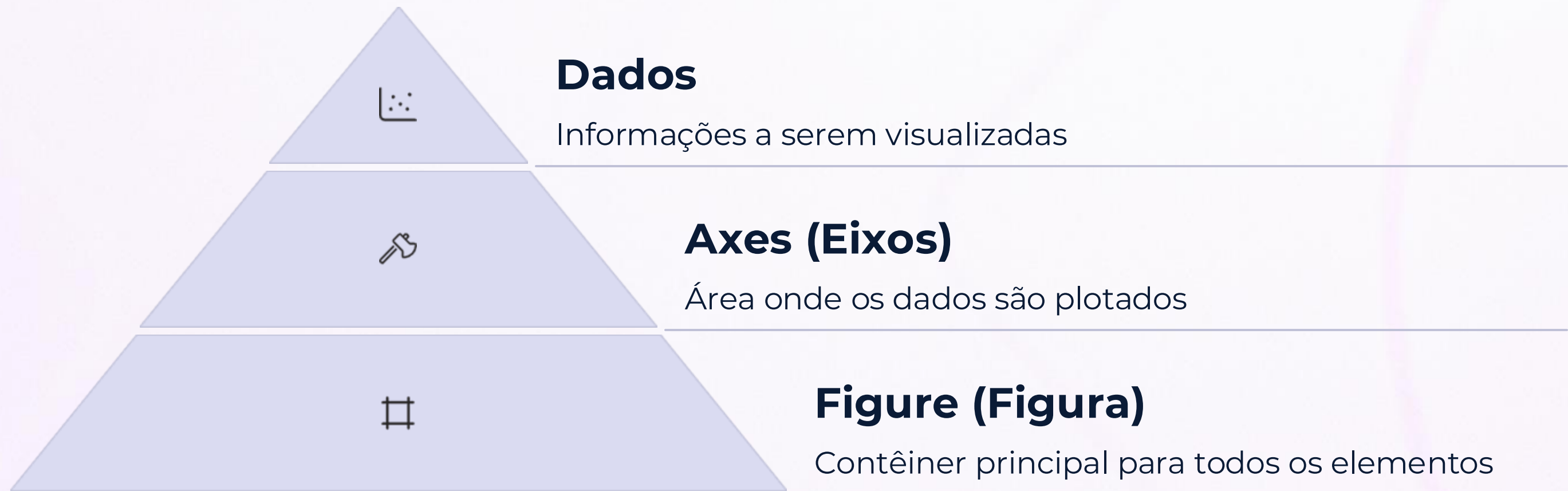


Cuidados Necessários

Evite misturar espaços e tabulações

A indentação não é apenas estilo em Python.
É parte fundamental da sintaxe que define a estrutura do código.

API BÁSICA: FIGURAS E SUBPLOTAGENS



Todas as plotagens são criadas dentro de um objeto Figure. Use `plt.figure()` para criar uma figura vazia ou `plt.subplots()` para criar figura e subplots simultaneamente.

PLOTANDO DADOS: ESTILOS E ELEMENTOS

Função Principal

`plt.plot()` ou `ax.plot()` para adicionar dados à área de plotagem.

Aceita arrays para coordenadas x e y dos pontos.

Estilização

Controle cores com `'g--'` (linha verde tracejada) ou argumentos explícitos como `color='#674ea7'`.

Adicione marcadores para destacar pontos individuais.

Elementos Contextuais

Use `ax.set()` para definir título, rótulos de eixos e limites.

Adicione legendas com `plt.legend()` para identificar séries.

ADICIONANDO TEXTO E FORMAS



Texto

Use `text()` para adicionar texto em coordenadas específicas na plotagem.



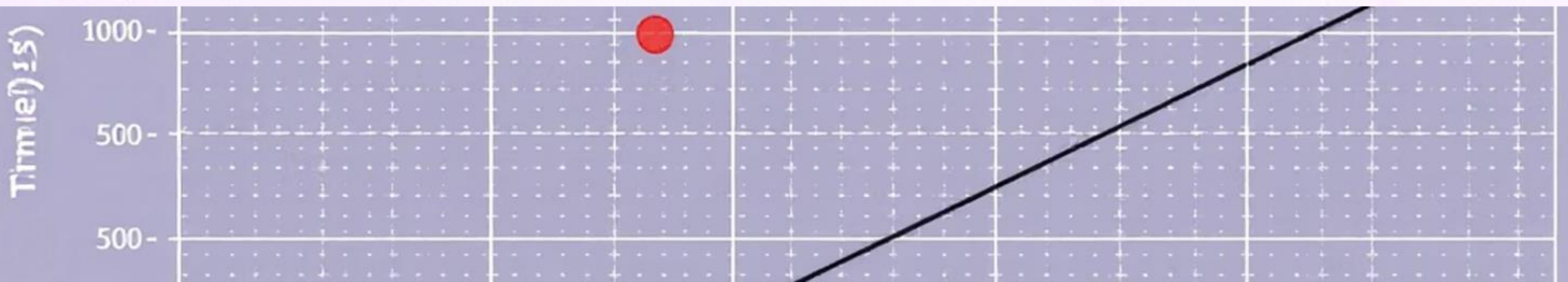
Anotações

Use `annotate()` para incluir texto com setas apontando para pontos específicos.



Formas

Adicione retângulos, círculos e outras formas com `matplotlib.patches`.



IMPORTÂNCIA DOS GRÁFICOS NA ANÁLISE

Exploração

Visualize distribuições e identifique padrões nos dados.

Insights

Extraia conclusões que tabelas numéricas não revelam facilmente.



Análise

Entenda relações entre variáveis e detecte valores atípicos.

Comunicação

Transmita resultados de forma clara e impactante.

TIPOS DE GRÁFICOS COMUNS



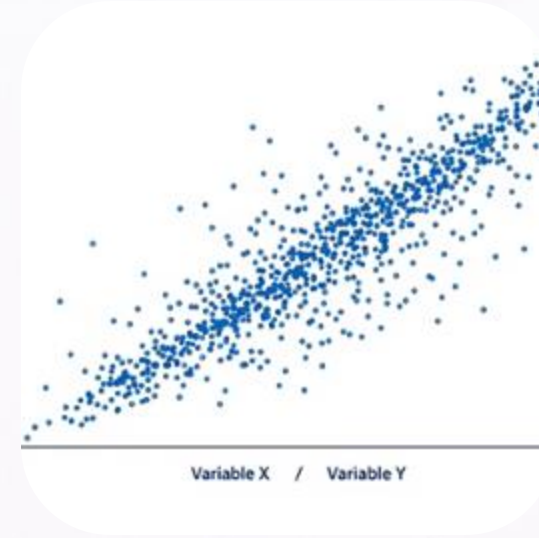
Gráficos de Linha

Ideais para mostrar tendências ao longo do tempo. Conectam pontos com linhas.



Gráficos de Barras

Comparam quantidades entre categorias diferentes. Úteis para variáveis qualitativas.



Gráficos de Dispersão

Mostram relações entre duas variáveis quantitativas. Essenciais antes de regressões.



Gráficos de Pizza

Mostram a composição percentual de um todo. Úteis só para poucas categorias.

INTEGRAÇÃO COM PANDAS E PERSONALIZAÇÃO



Integração com Pandas

Use `DataFrame.plot()` para criar gráficos diretamente dos seus dados.



Personalização

Modifique parâmetros globais com `plt.rc()` ou `matplotlib.rcParams`.



Configuração Persistente

Use o arquivo `matplotlibrc` para definir estilos padrão entre sessões.



Vamos
avaliar o
encontro?