

Visualização de dados em Python

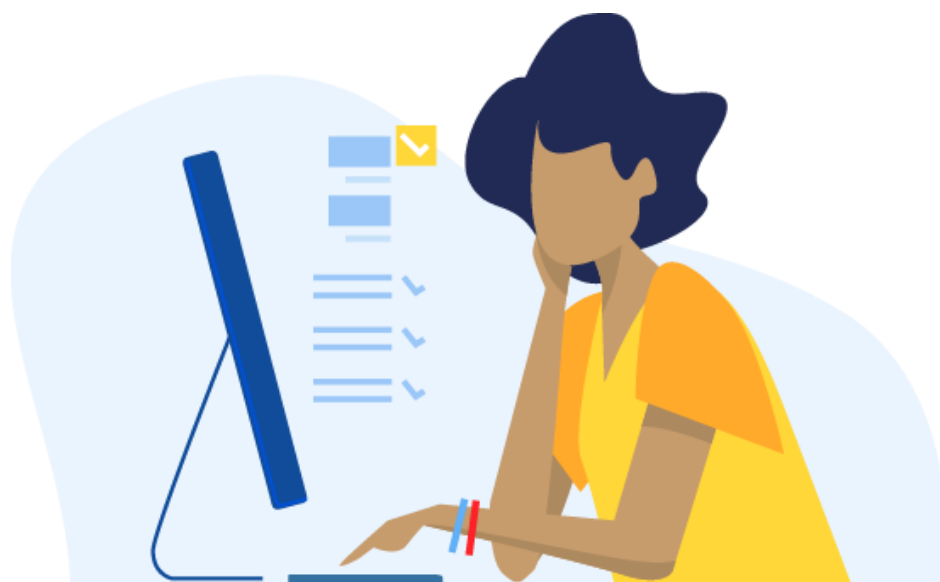
Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar.

Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados.

Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Biblioteca matplotlib

Ao se falar em criação de gráficos em Python, o profissional precisa conhecer a biblioteca matplotlib, pois diversas outras são construídas a partir desta.



Fonte: Shutterstock.

A criação e grande parte da evolução dessa biblioteca se deve a John Hunter, que a desenvolveu como uma opção ao uso dos softwares gnuplot e MATLAB (MCGREGGOR, 2015). Com a utilização da linguagem Python na área científica para trabalhar com dados, após a extração dos resultados, o cientista precisava criar seus gráficos nos outros softwares mencionados, o que se tornava inconveniente, motivando a criação da biblioteca em Python.

Figura e eixo

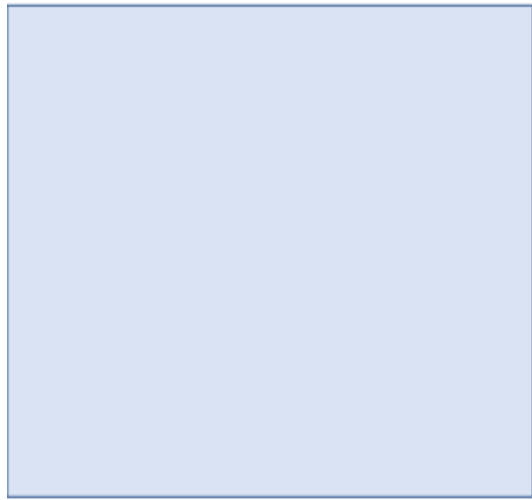
Ao utilizar a biblioteca matplotlib é preciso entender o conceito de figura e eixo (axes).

Ao criar uma figura estamos criando um espaço (tipo uma tela em branco) para se plotar o gráfico nesse espaço. Naturalmente, uma figura pode ter “subfiguras”, ou seja, uma figura pode ser dividida.

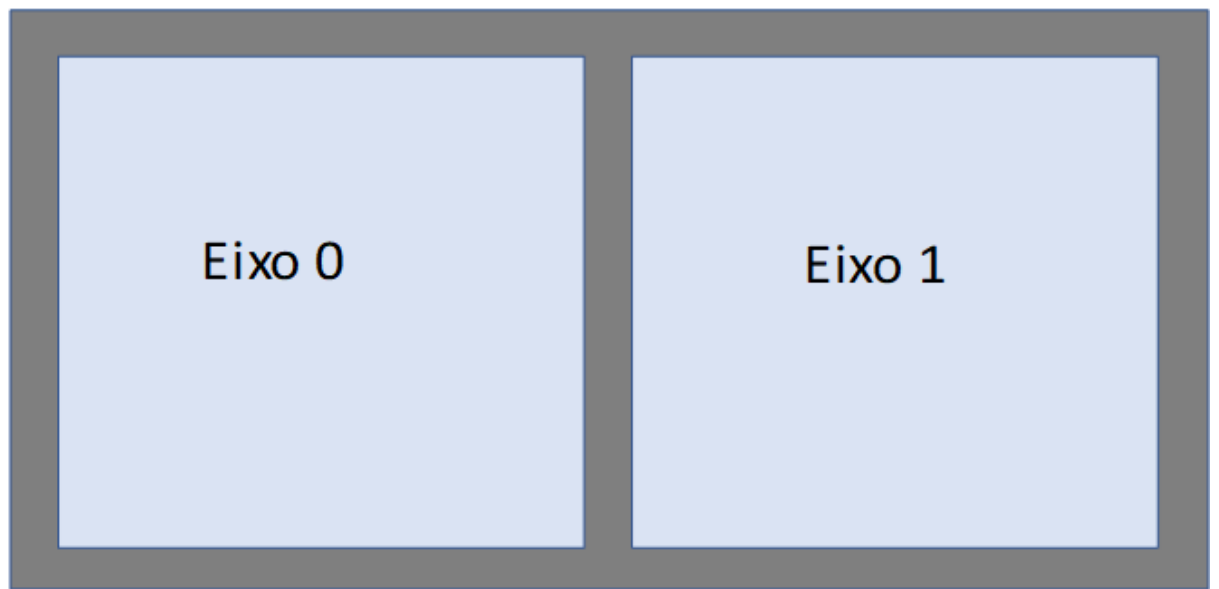
- » Quando uma figura possui um único espaço, ela também possui um único eixo para plotagem.
- » Quando uma figura possui duas divisões, ela possui dois eixos; quando possui três divisões, três eixos, e assim por diante.

Figura e eixo matplotlib

1 figura
1 eixo



1 figura, com 2 subplots
2 eixos



Fonte: elaborada pela autora.

Método `plt.plot(dados)`

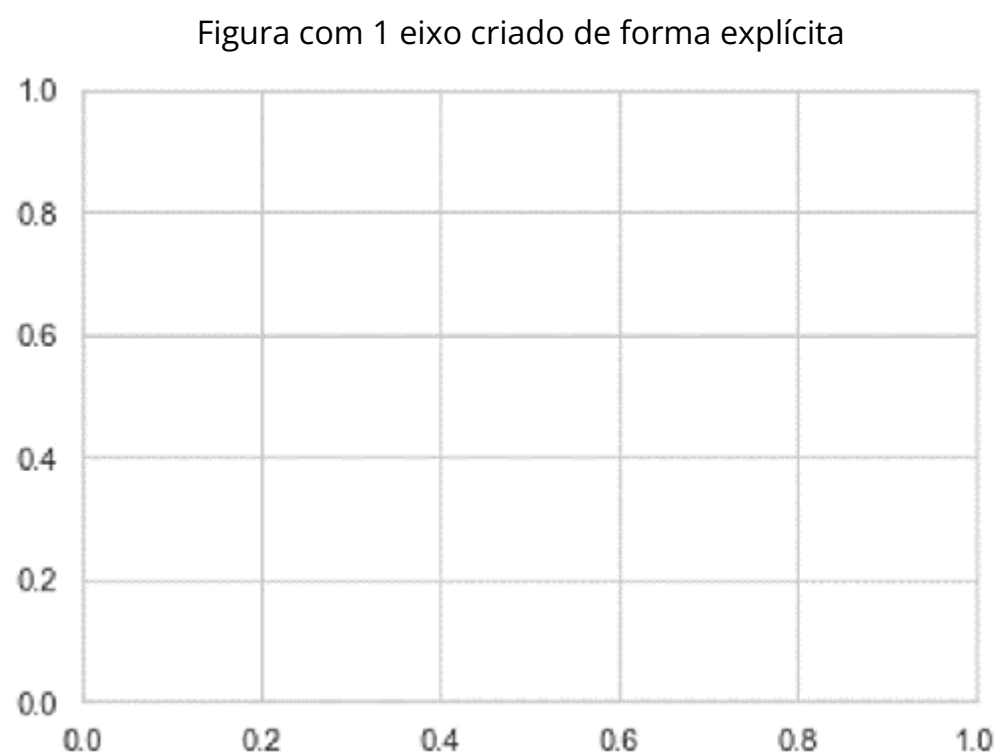
Existem algumas opções de sintaxe para se criar figuras e eixos. Podemos deixar que a própria biblioteca gerencie para nós, dessa forma basta utilizar o método `plt.plot(dados)` que o gráfico será construído sobre uma figura e um eixo criados automaticamente.

A forma automática, embora prática, não permite que o desenvolvedor tenha controle sobre o eixo que gostaria de plotar, nesse caso o ideal é criar, explicitamente, uma figura com os eixos, usando a função `plt.subplots()`.

Para criar uma figura com um eixo, de forma explícita, usamos:

```
fig, ax = plt. Subplots (1, 1)
```

Resultado:



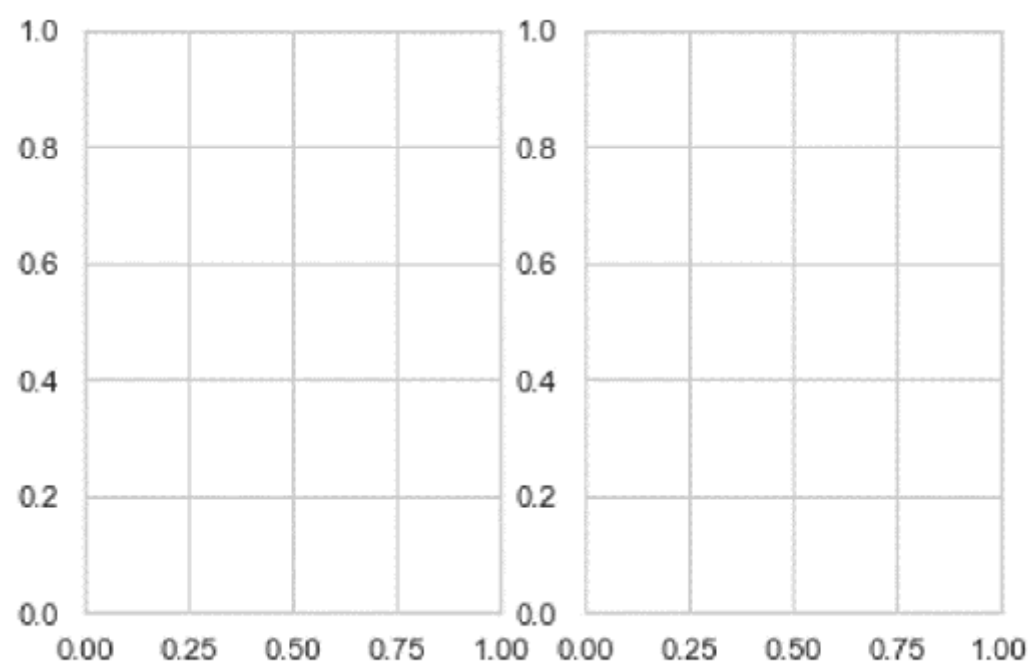
Fonte: elaborada pela autora.

Para criar uma figura com dois eixos (1 linha, 2 colunas), de forma explícita, usamos:

```
fig, ax = plt. Subplots (1, 2)
```

Resultado:

Figura com 2 eixos criados de forma explícita



Fonte: elaborada pela autora.

Ao criar uma figura com mais de um eixo, temos que informar em qual vamos criar um gráfico. A variável “ax”, criada no último exemplo, é um vetor de eixos, ou seja, podemos acessar cada eixo pelo índice, começando por 0.

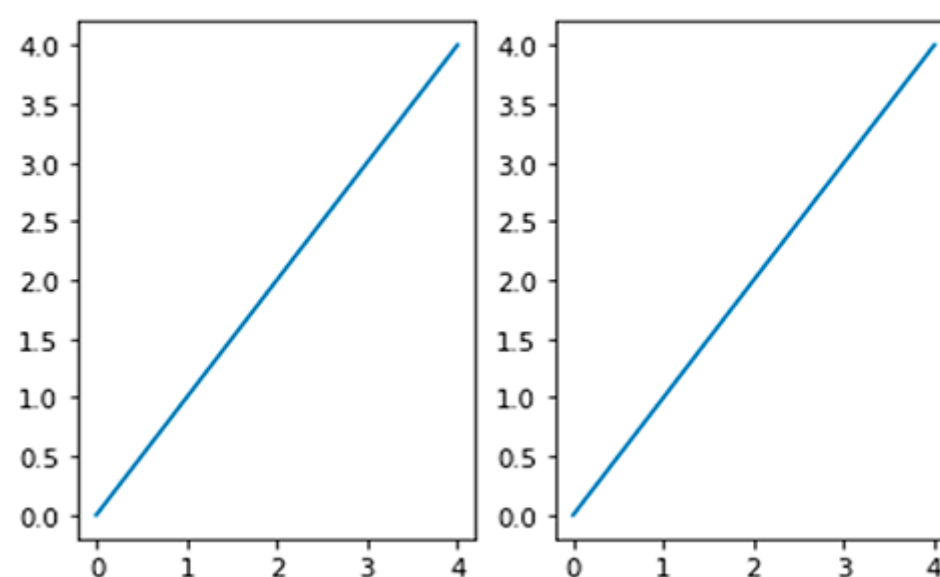
Portanto, para plotar na figura mais à esquerda escolhemos `ax[0].plot()` e na da direita, `ax[1].plot()`.

Veja a seguir:

```
1 fig, ax = plt. Subplots (1, 2)
2
3 Dados = range(5)
4
5 ax[0].plot(dados)
6 ax[1].plot(dados)
```

Plotagem sobre eixos

[<matplotlib.lines.Line2D at 0x2843cbfca20>]



Fonte: elaborada pela autora.

As figuras podem ser criadas sem nenhum eixo, com o comando:

```
fig = plt.subplots()
```

Dessa forma, para se plotar teremos que usar a função `plt.subplot()`, que adiciona um `subplot` em um figura existente.

Na documentação oficial da biblioteca matplotlib, você encontrará o método `add_subplot()` que funciona como o `subplot()`.

Veja um exemplo no portal [matplotlib.org](https://matplotlib.org/gallery/lines_bars_and_markers/eventcollection_demo.html#sphx-glr-gallery-lines-bars-and-markers-eventcollection-demo-py), disponível em: https://matplotlib.org/gallery/lines_bars_and_markers/eventcollection_demo.html#sphx-glr-gallery-lines-bars-and-markers-eventcollection-demo-py. Acesso em: 1 jul. 2020.



Fonte: Shutterstock.

Para visualizar o vídeo, acesse seu material digital.