

## FÓRUM

SOLUCIONADO (ver solução)

### fig.add\_axes()

Data Science

Data Visualization

Data Visualization: criação de gráficos com o Matplotlib

Referente ao curso **Data Visualization: criação de gráficos com o Matplotlib**, no capítulo **Primeiras visualizações** e atividade **Funcionalidades básicas do matplotlib**

Publicado 6 meses atrás, em 16/07/2021

[Acompanhar tópico](#)



por **Júlio Ricardo Burlamaqui dos Santos** | 71.1k xp | 13 posts

Não compreendi bem a funcionalidade do `.add_axes()`. É um método do figure, certo? A lista de valores passados correspondem aos quadrantes que se quer visualizar, dessa forma variando entre -1 a 1? Mas não entendi porque são 4 valores na lista, ao invés de 2. São os máximos e mínimos? Quando o professor usa `add_axes([0, 0, 1, 1])` está falando que a visualização da figura vai de 0 a  $+\infty$  no x e 0 a  $+\infty$  no y, ou seja, o primeiro quadrante? Não sei se estou certo, alguém poderia me confirmar ou elucidar?

## 2 RESPOSTAS

SOLUÇÃO!



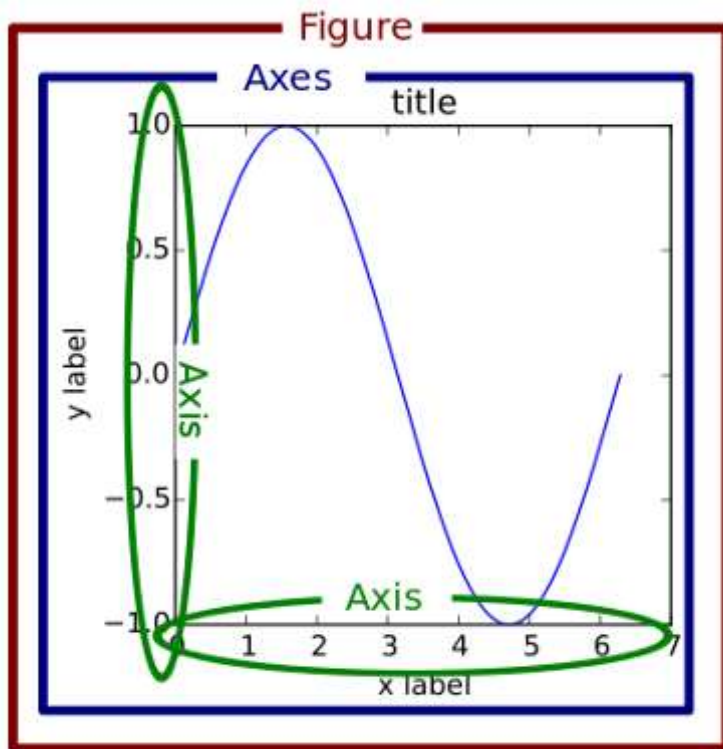
por **Millena Gená Pereira** | 580.8k xp | 954 posts

Alura Scuba Team

11/08/2021

Olá Júlio! Como você está?

Nos gráficos do matplotlib, `figure` e `axes` são camadas de uma figura. Observe essa imagem:



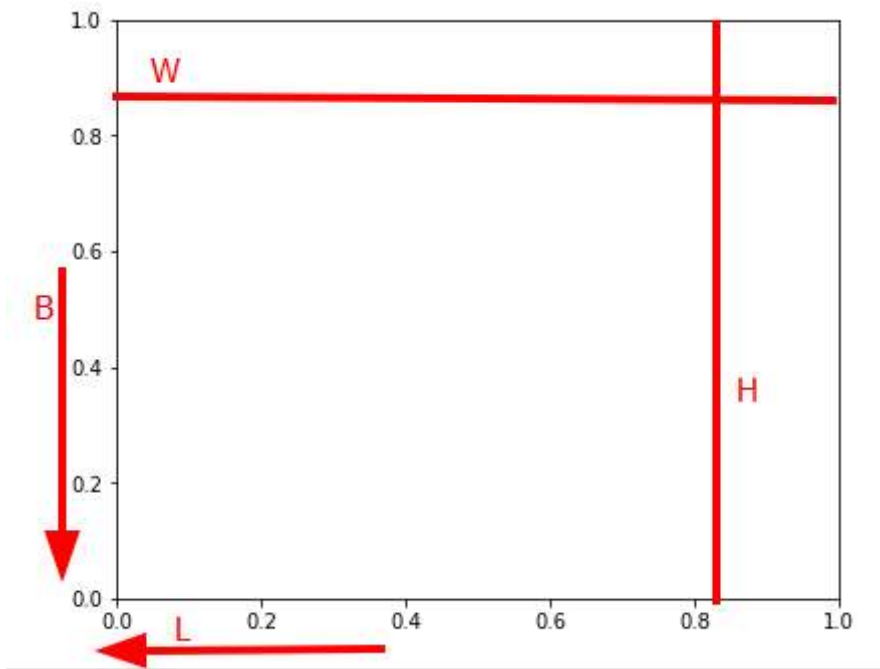
Nessa imagem, vemos as partes de uma figura do matplotlib. Onde temos a base da figura (`Figure`) como sendo a primeira camada, o eixo (`axes`) como sendo a segunda camada da figura e os eixos `x` e `y`. O `axes` é uma camada da figura onde um determinado gráfico irá ser colocado. Se temos uma figura com mais de um gráfico, essa figura consequentemente tem mais de um `axes` uma vez que temos que criar um `axes` específico para cada gráfico. Tudo bem até aqui?

Existem diferentes métodos para adicionarmos essa camada `axes` em uma figura. O professor utiliza o método `add_axes`. Para utilizarmos esse método, devemos passar pra ele as `dimensões` que desejamos que ele tenha. Analisando a documentação desse método, vamos ver que os parâmetros dimensionais a serem passados são:

[left, bottom, width, height]

COPIAR CÓDIGO

Que, traduzindo, significam, respectivamente [esquerda, parte inferior, largura, altura] . Na imagem abaixo, conseguimos visualizar um pouco melhor onde esses parâmetros estão localizados:



Sendo L: left, B: bottom, W: width e H: height.

Além disso, outra informação importante que a documentação nos trás é a seguinte: **"Todos os valores dessas dimensões estão em frações da largura e altura da figura"**. Ou seja, os valores que definirmos dessas dimensões vão ser proporcionais ao tamanho da figura que definirmos anteriormente utilizando o `plt.figure(figsize=(largura, altura))` .

Vamos fazer um exemplo para conseguirmos visualizar um pouco melhor. Suponhamos que tenhamos o seguinte gráfico:

```
x = list(range(1,9))
y = x

fig = plt.figure(figsize = (5,4))
eixo = fig.add_axes([0,0,1,1])
```

```
eixo.plot(x, y)
eixo.set_title('Título do gráfico', fontsize = 15)
plt.show()
```

[COPIAR CÓDIGO](#)

Quando criamos a figura, por meio do comando `plt.figure(figsize = (5,4))` definimos que a figura vai ter 5 de largura (eixo x) e 4 de altura (eixo y)

Quando criamos a figura, por meio do comando `fig = plt.figure(figsize = (5,4))` definimos que a figura vai ter 5 de largura (eixo x) e 4 de altura (eixo y). Ao acrescentarmos o eixo, na linha `eixo = fig.add_axes([0,0,1,1])`, estamos passando os seguintes valores para os parâmetros que vimos anteriormente:

- esquerda (left) = 0;
- parte inferior (bottom) = 0;
- largura (width) = 1;
- altura (height) = 1;

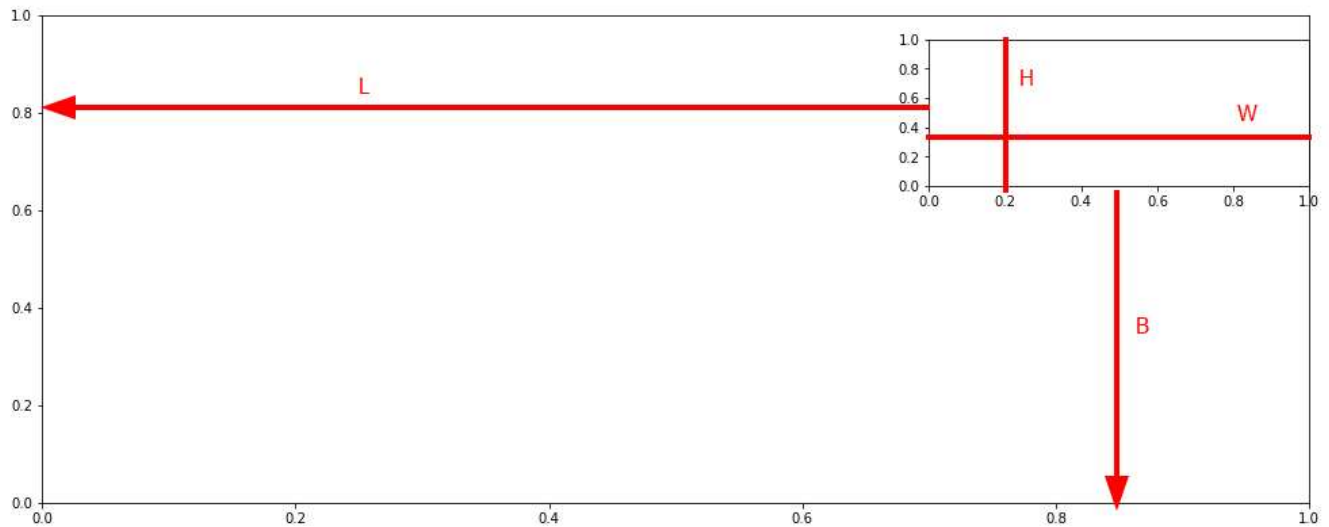
O que isso quer dizer? Bom, informa que, no espaço disponível que temos na nossa figura, queremos adicionar um `axes` de forma que sua parte esquerda (início do eixo x) e inferior (início do eixo y) comecem no ponto inicial da figura, ou seja, 0. E que sua largura e altura sejam do tamanho máximo da nossa figura, por isso o 1 (que proporcionalmente significaria 100%).

Agora, vamos criar dois axes vazios adicionados em uma mesma figura:

```
fig = plt.figure(figsize = (13,5))

eixo1 = fig.add_axes([0, 0, 1, 1])
eixo2 = fig.add_axes([0.7, 0.65, 0.3, 0.3])
```

[COPIAR CÓDIGO](#)



Observe que, o `eixo1` foi criado de forma que ocupasse todo o espaço da figura, enquanto que, para o `eixo2` já foram passados outros valores dos parâmetros, de forma que ele fosse um eixo menor e ficasse localizado em outra posição.

Espero que tenha conseguido te ajudar. Qualquer dúvida estou por aqui :)

Bons estudos!



por **Júlio Ricardo Burlamaqui dos Santos** | 71.1k xp | 13 posts

23/09/2021

Excelente explicação! Muito obrigado, entendi perfeitamente.

Esta dúvida já foi marcada como solucionada, se a dúvida ainda persiste, por favor abra outra discussão.

alura

SOBRE A ALURA

DÚVIDAS FREQUENTES

SUGIRA UM CURSO

SUGIRA UMA FUNCIONALIDADE

BLOG DA ALURA

ACOMPANHE-NOS



APPS



