

Guia básico para plotar gráficos usando o matplotlib do Python

[adminvooo0](#)



Por: [Ricardo Dorts Schönhofen](#)

Como plotar gráficos usando matplotlib.

matplotlib é uma biblioteca Python de plotagem 2d, que auxilia a biblioteca matemática NumPy. Pode ser usada em scripts Python, no shell Python e IPython, em servidores de aplicação web e outras ferramentas de interface gráfica.

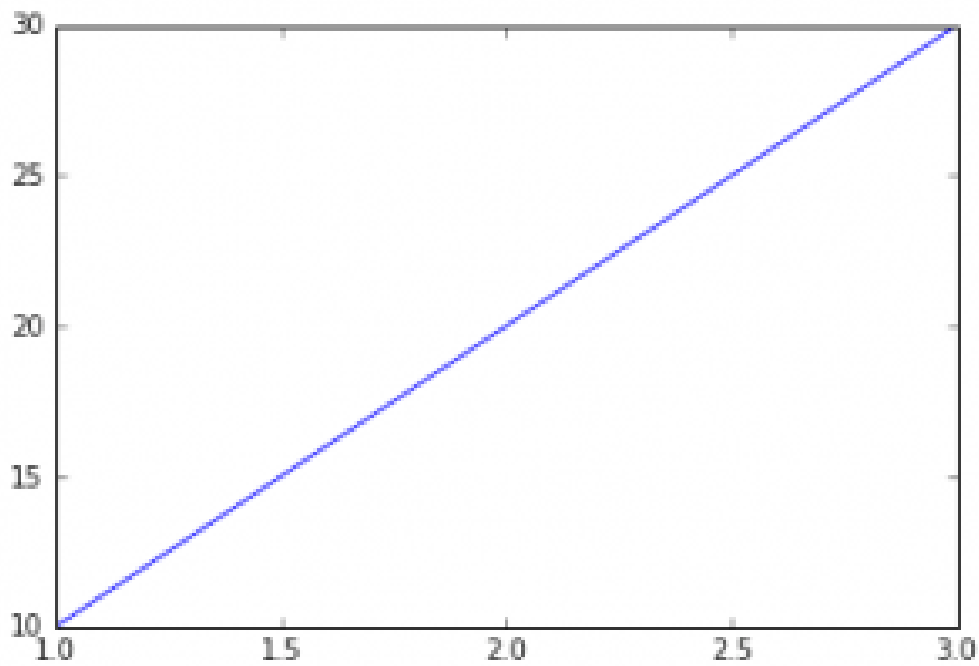
A seguir, mostramos como plotar alguns gráficos usando matplotlib.

```
#importando a classe, da forma mais usual
import matplotlib.pyplot as plt

#definindo variáveis que vamos usar nos exemplos
x = [1, 2, 3]
y = [10, 20, 30]
y2 = [15, 10, 40]
y3 = [20, 10, 35]
yBar = [3, 10, 7, 5, 3, 4.5, 6, 8.1]
z = [i * 1.5 for i in yBar]
xBar = range(len(yBar))
azul = "blue"
verde = "green"
preto = "black"
```

Desenhando uma linha:

```
def plota_linha_1():
    plt.plot(x, y)
```



Adicionando rótulos e título

```
def plota_linha_2():  
    plt.xlabel('X')  
    plt.ylabel('Y')  
    plt.title('Título do gráfico')  
    plt.plot(x, y)
```

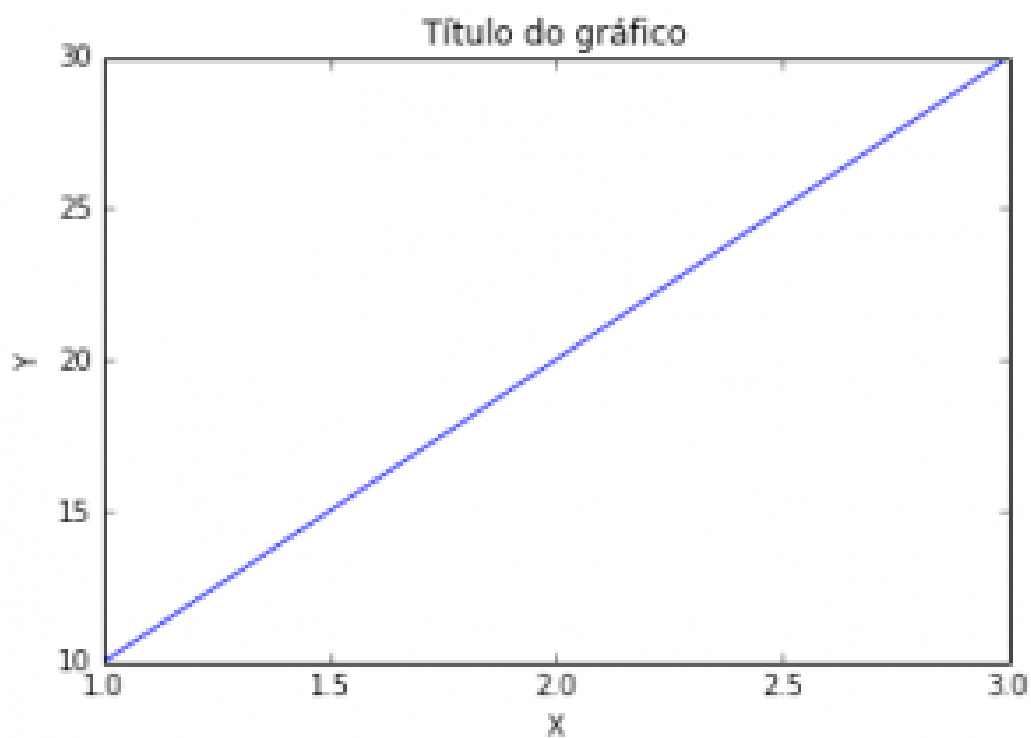


Gráfico com mais de uma linha

```
def plota_linhas_3():  
    plt.xlabel('X')  
    plt.ylabel('Y')  
    plt.title('Título do gráfico')  
    plt.plot(x, y)  
    plt.plot(x, y2)  
    plt.plot(x, y3)
```

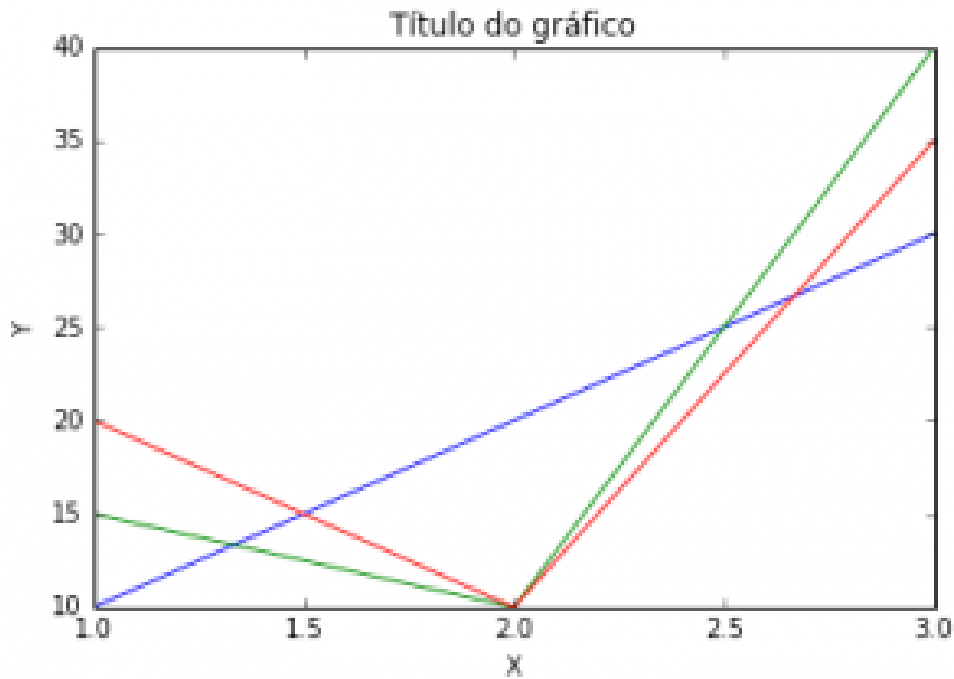


Gráfico de barras

```
def plota_barra_1():  
    plt.bar(xBar, yBar, color=azul)
```

Guia básico para plotar gráficos usando o matplotlib do Python parte 2

[adminvooo0](#)



Por: [Ricardo Dorts Schönhofen](#)

Como plotar gráficos usando matplotlib.

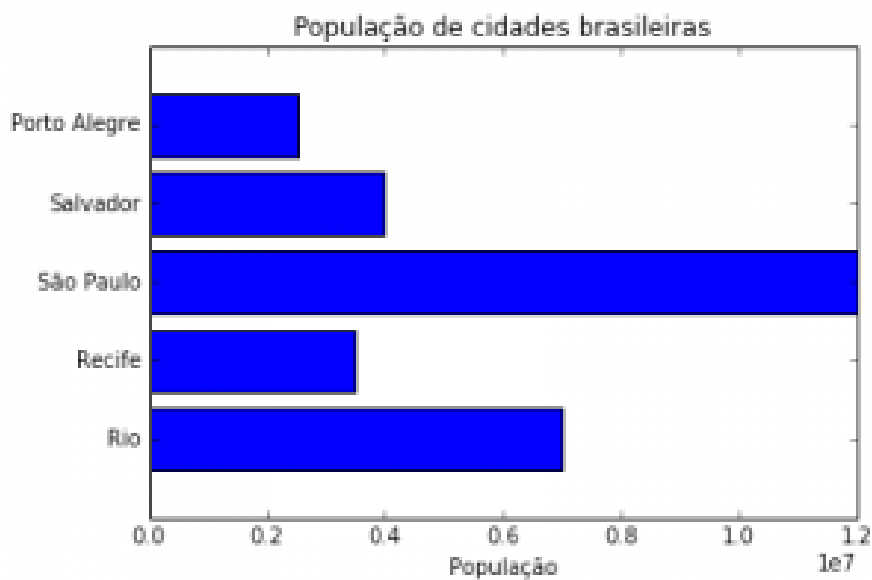
Continuação dos exemplos de como plotar gráficos usando matplotlib. Veja a [parte 1](#).

```
#importando as classes da forma mais usual
import matplotlib.pyplot as plt
import numpy as np

#definindo variáveis usadas nos exemplos
pop = {"São Paulo": 12000000, "Rio": 7000000, "Salvador": 4000000, "Recife":
3500000, "Porto Alegre": 2500000}
cidades = [i for i in pop.keys()]
populacao = [j for j in pop.values()]
popPos = np.arange(len(cidades))
azul = "blue"
verde = "green"
preto = "black"
```

Desenhando um gráfico de barras horizontal

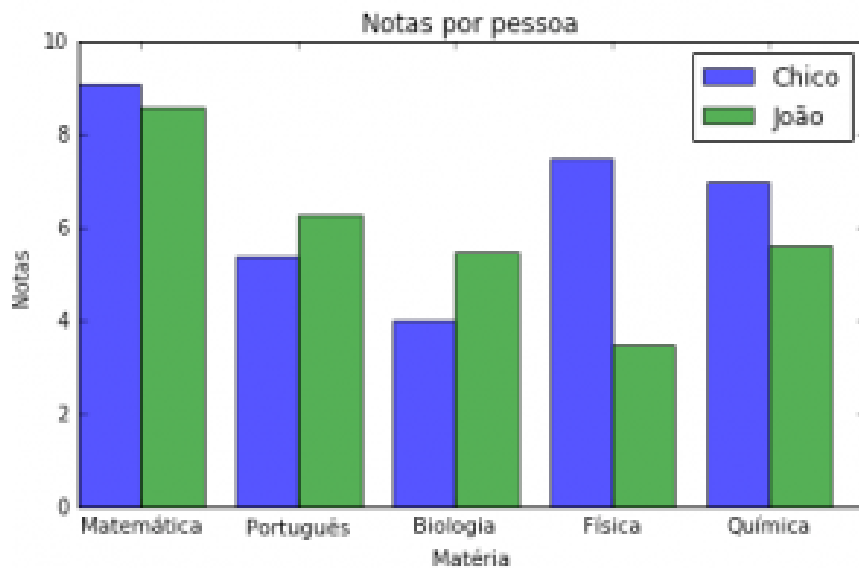
```
def plota_barra_h_1():
    plt.barh(popPos, populacao, align='center', color=azul)
    plt.yticks(popPos, cidades)
    plt.xlabel('População')
    plt.title('População de cidades brasileiras')
```



Desenhando um gráfico de duas barras

```
def plota_bar_dupla_1():
    grupos = 5
    media_chico = (9.1, 5.4, 4.0, 7.5, 7.0)
    media_joao = (8.6, 6.3, 5.5, 3.5, 5.6)
    fig, ax = plt.subplots()
    indice = np.arange(grupos)
    bar_larg = 0.4
    transp = 0.7
    plt.bar(indice, media_chico, bar_larg, alpha=transp, color=azul,
label='Chico')
    plt.bar(indice + bar_larg, media_joao, bar_larg, alpha=transp, color=verde,
label='João')

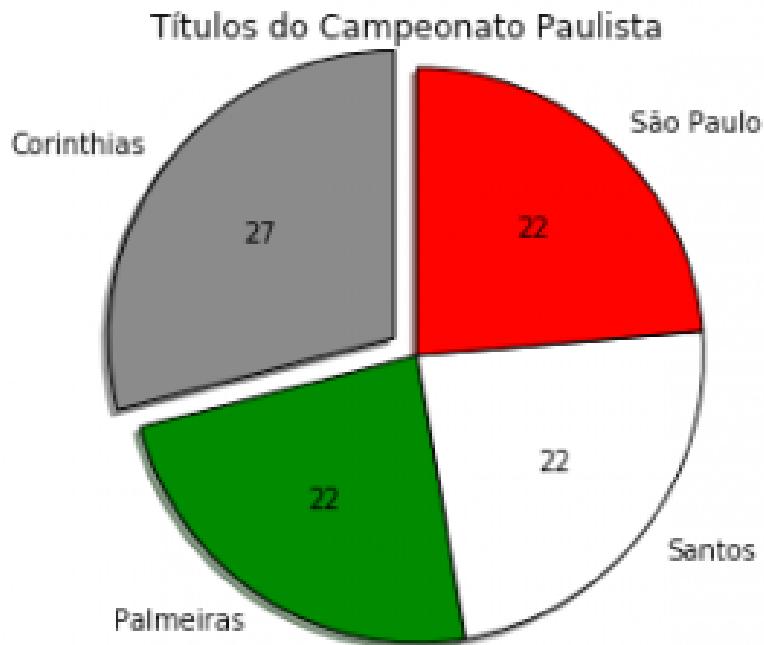
plt.xlabel('Matéria')
plt.ylabel('Notas')
plt.title('Notas por pessoa')
plt.xticks(indice + bar_larg, ('Matemática', 'Português', 'Biologia', 'Física',
'Química'))
plt.legend()
plt.tight_layout()
plt.show()
```



Desenhando um gráfico de pizza

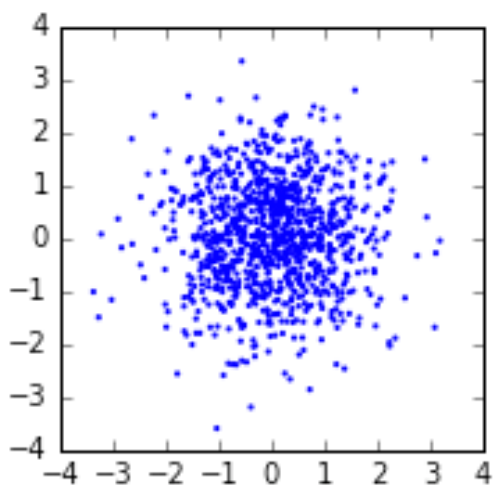
```
def plota_pizza_1():
    labels = ['Corinthians', 'Palmeiras', 'Santos', 'São Paulo']
    titulos = [27, 22, 22, 22]
    cores = ['lightblue', 'green', 'white', 'red']
    explode = (0.1, 0, 0, 0) # somente explode primeiro pedaço
    total = sum(titulos)
    plt.pie(titulos, explode=explode, labels=labels, colors=cores, autopct=lambda
p: '{:.0f}'.format(p * total / 100), shadow=True, startangle=90)

# Determina que as proporções sejam iguais ('equal') de modo a desenhar o
círculo
plt.axis('equal')
plt.show()
```



Desenhando um gráfico de dispersão (Scatter Plot)

```
def plota_dispersao_1():  
    fig = plt.figure()  
    ax1 = fig.add_subplot(121)  
  
    #gera dados aleatórios  
    N=1000  
    x = np.random.randn(N)  
    y = np.random.randn(N)  
    ax1.scatter(x,y,color=azul,s=5,edgecolor='none')  
    #deixa os eixos quadrados  
    ax1.set_aspect(1./ax1.get_data_ratio())  
    plt.show()
```



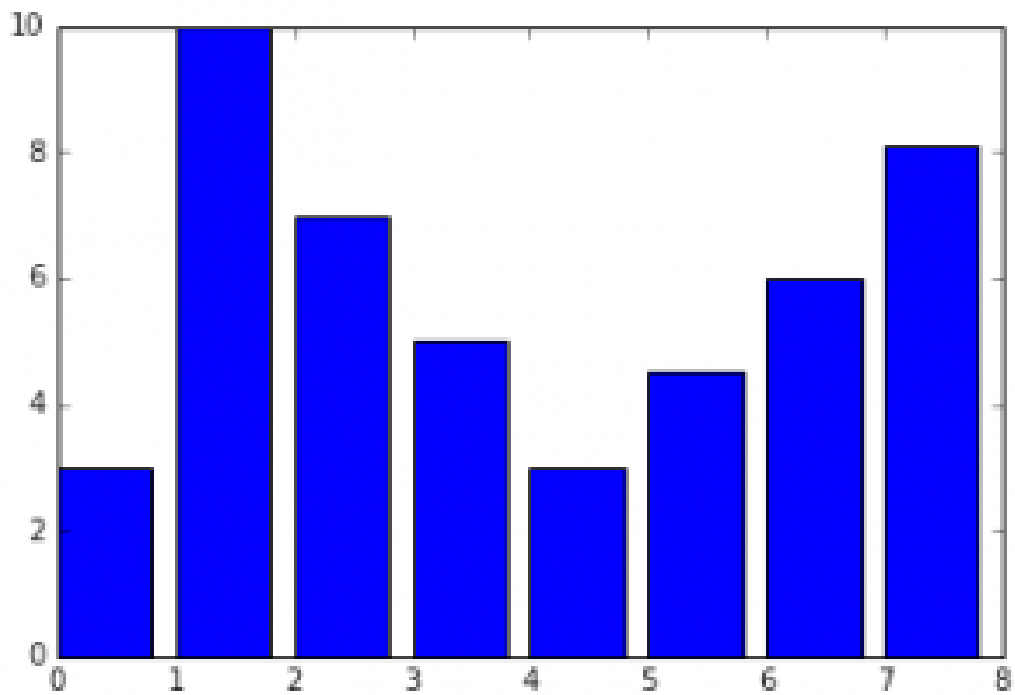


Gráfico de barras com linha

```
def plota_barra_linha_1():  
    plt.bar(xBar, yBar, color=preto, align='center')  
    plt.plot(xBar, z, color=verde, marker='o', linestyle='--', linewidth=2)
```

