NumPy - Matplotlib

Matplotlib é uma biblioteca de plotagem para Python. Ele é usado junto com o NumPy para fornecer um ambiente que seja uma alternativa eficaz de código aberto para MatLab. Também pode ser usado com kits de ferramentas gráficas como PyQt e wxPython.

O módulo Matplotlib foi escrito pela primeira vez por John D. Hunter. Desde 2012, Michael Droettboom é o principal desenvolvedor. Atualmente, Matplotlib versão. 1.5.1 é a versão estável disponível. O pacote está disponível em distribuição binária, bem como na forma de código-fonte em www.matplotlib.org .

Convencionalmente, o pacote é importado para o script Python adicionando a seguinte instrução -

```
from matplotlib import pyplot as plt
```

Aqui, **pyplot()** é a função mais importante na biblioteca matplotlib, que é usada para plotar dados 2D. O script a seguir traça a equação **y = 2x + 5**

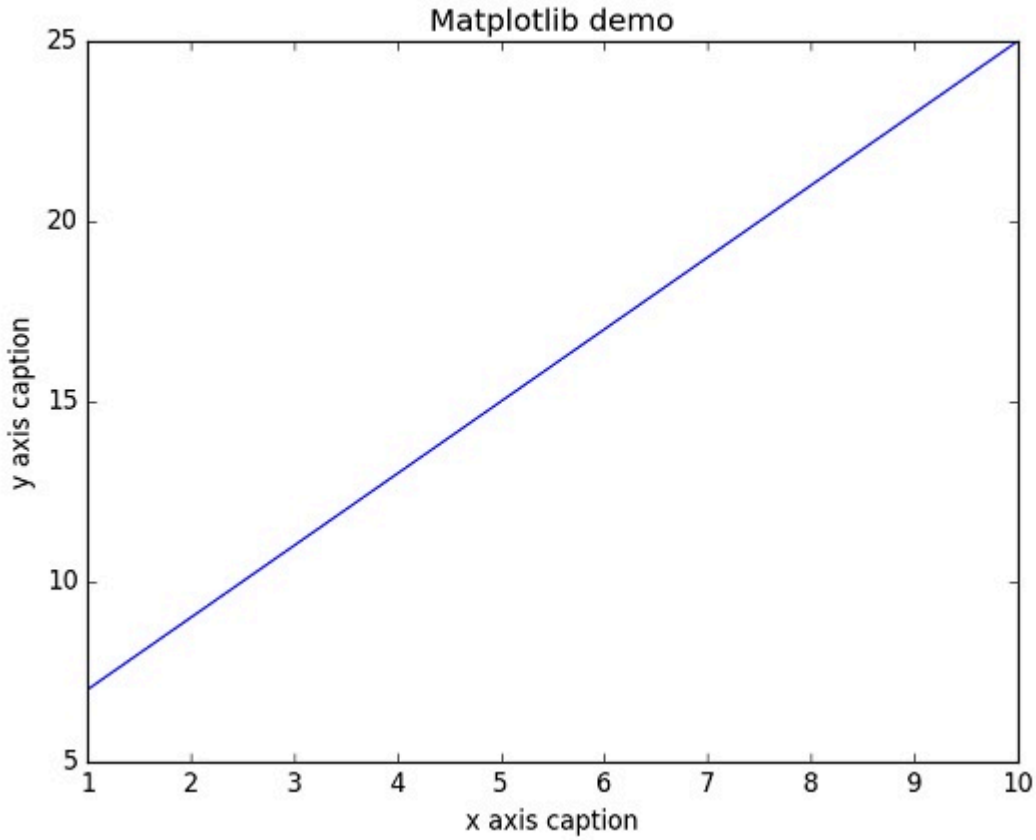
Exemplo

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

x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```

Um objeto ndarray x é criado a partir **da função np.arange()** como os valores no **eixo x** . Os valores correspondentes no **eixo y** são armazenados em outro **objeto ndarray y** . Esses valores são plotados usando a função **plot()** do submódulo pyplot do pacote matplotlib.

A representação gráfica é exibida pela função **show()** .

O código acima deve produzir a seguinte saída -



Em vez do gráfico linear, os valores podem ser exibidos discretamente adicionando uma string de formato à função **plot()** . Os seguintes caracteres de formatação podem ser usados.

Sr. Não.	Descrição do personagem
1	**'-'** Estilo de linha sólida
2	**'--'** Estilo de linha tracejada
3	**'-.'** Estilo de linha traço-ponto
4	**':'** Estilo de linha pontilhada
5	**'.'** Marcador de ponto
6	**','** Marcador de pixels
7	**'o'** Marcador de círculo

8	**'v'** Marcador triângulo_para baixo
9	**'^'** Marcador Triangle_up
10	**'<'** Marcador triângulo_esquerdo
11	**'>'** Marcador triângulo_direito
12	**'1'** Marcador tri_down
13	**'2'** Marcador Tri_up
14	**'3'** Marcador tri_esquerdo
15	**'4'** Marcador tri_right
16	**'é'** Marcador quadrado
17	**'p'** Marcador do Pentágono
18	**'*'** Marcador de estrela
19	**'h'** Marcador Hexágono1
20	**'H'** Marcador Hexágono2
21	**'+'** Mais marcador
22	**'x'** Marcador X
23	**'D'** Marcador de diamante
24	**'d'**

	Marcador Thin_diamond
25	'|' Marcador de linha V
26	'_' Marcador de linha H

As seguintes abreviações de cores também são definidas.

Personagem	Cor
'b'	Azul
'g'	Verde
'r'	Vermelho
'c'	Ciano
'eu'	Magenta
'você'	Amarelo
'k'	Preto
'c'	Branco

Para exibir os círculos que representam pontos, em vez da linha no exemplo acima, use **"ob"** como string de formato na função plot().

Exemplo

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

x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y,"ob")
plt.show()
```

O código acima deve produzir a seguinte saída -

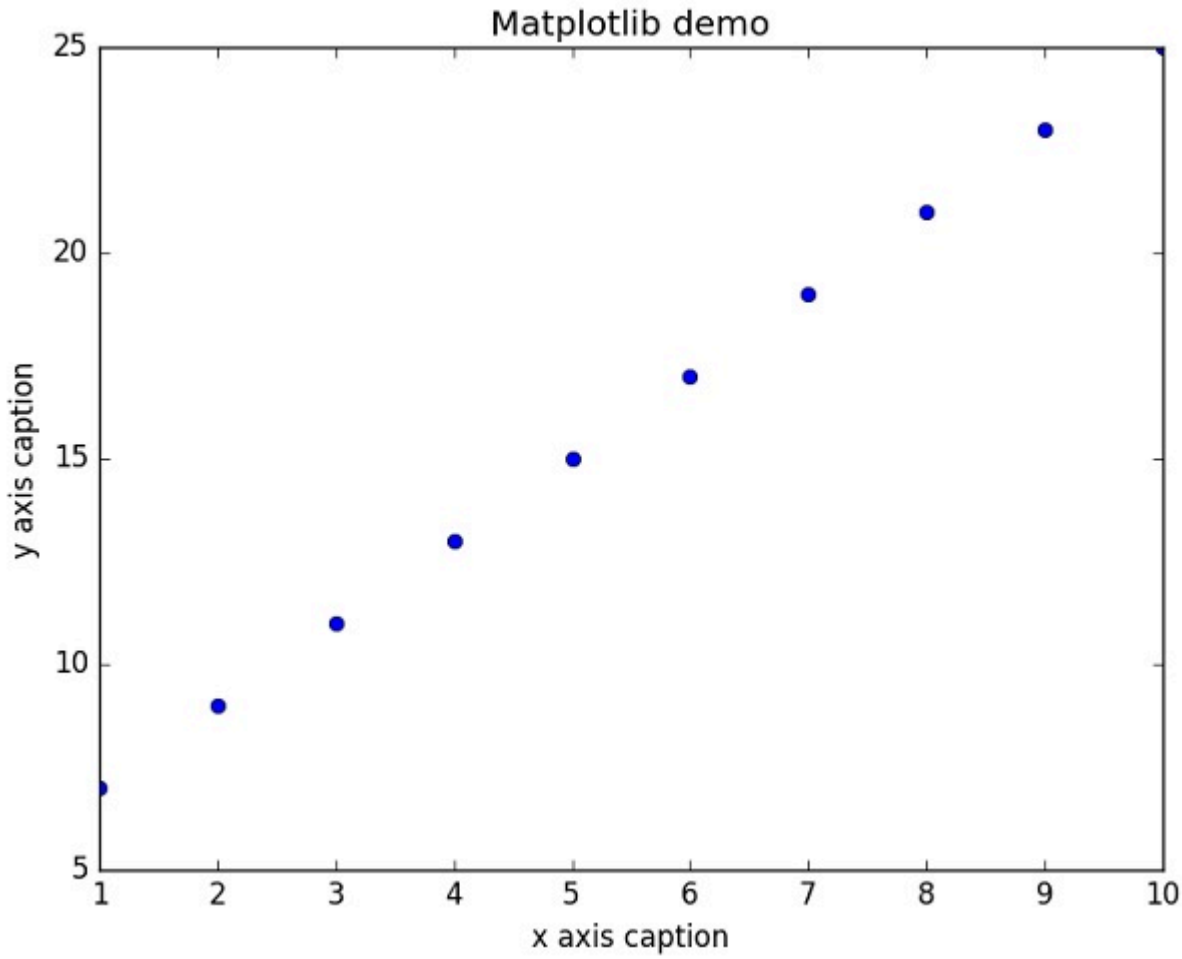
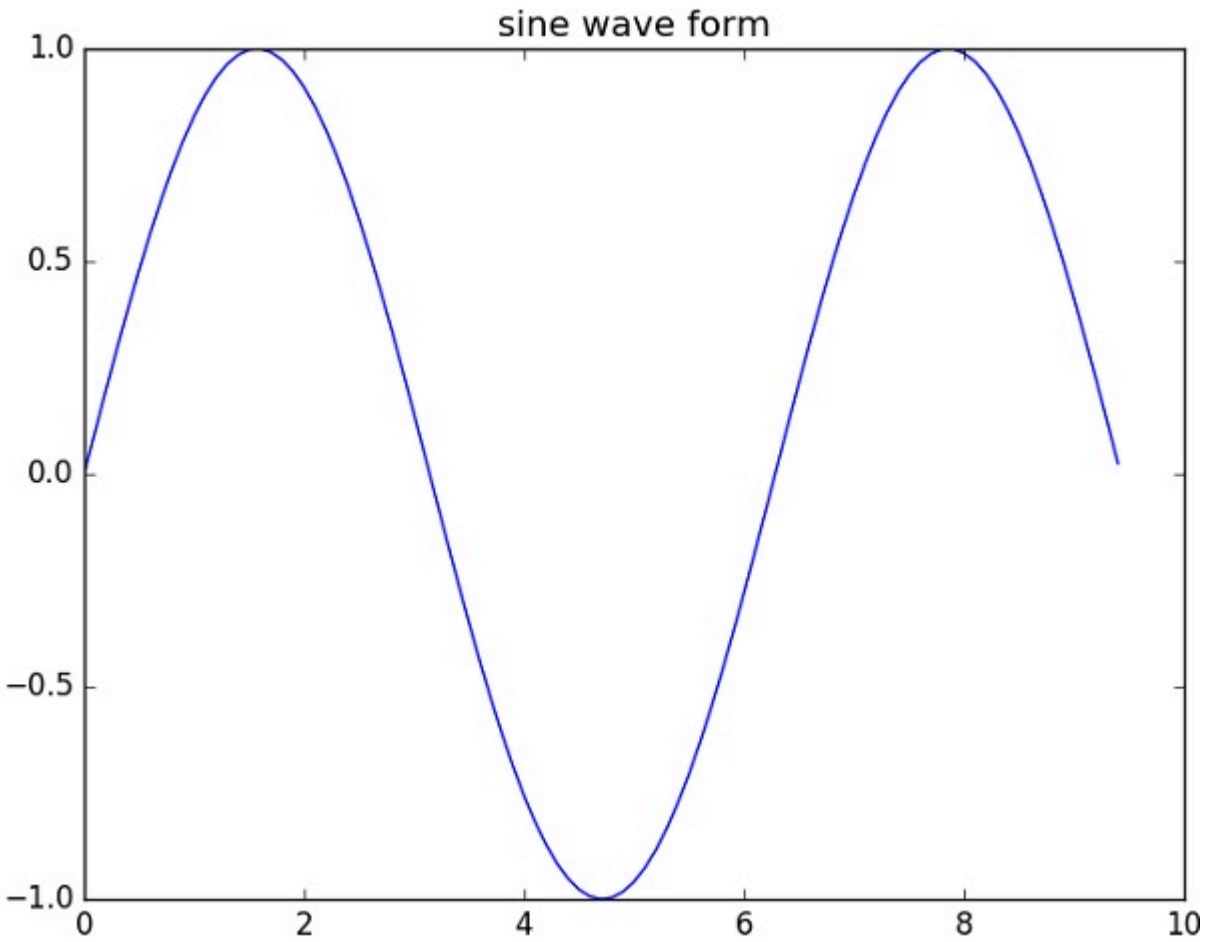


Gráfico de onda senoidal

O script a seguir produz o **gráfico de onda senoidal** usando matplotlib.

Exemplo

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

# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")

# Plot the points using matplotlib
plt.plot(x, y)
plt.show()
```



subtrama()

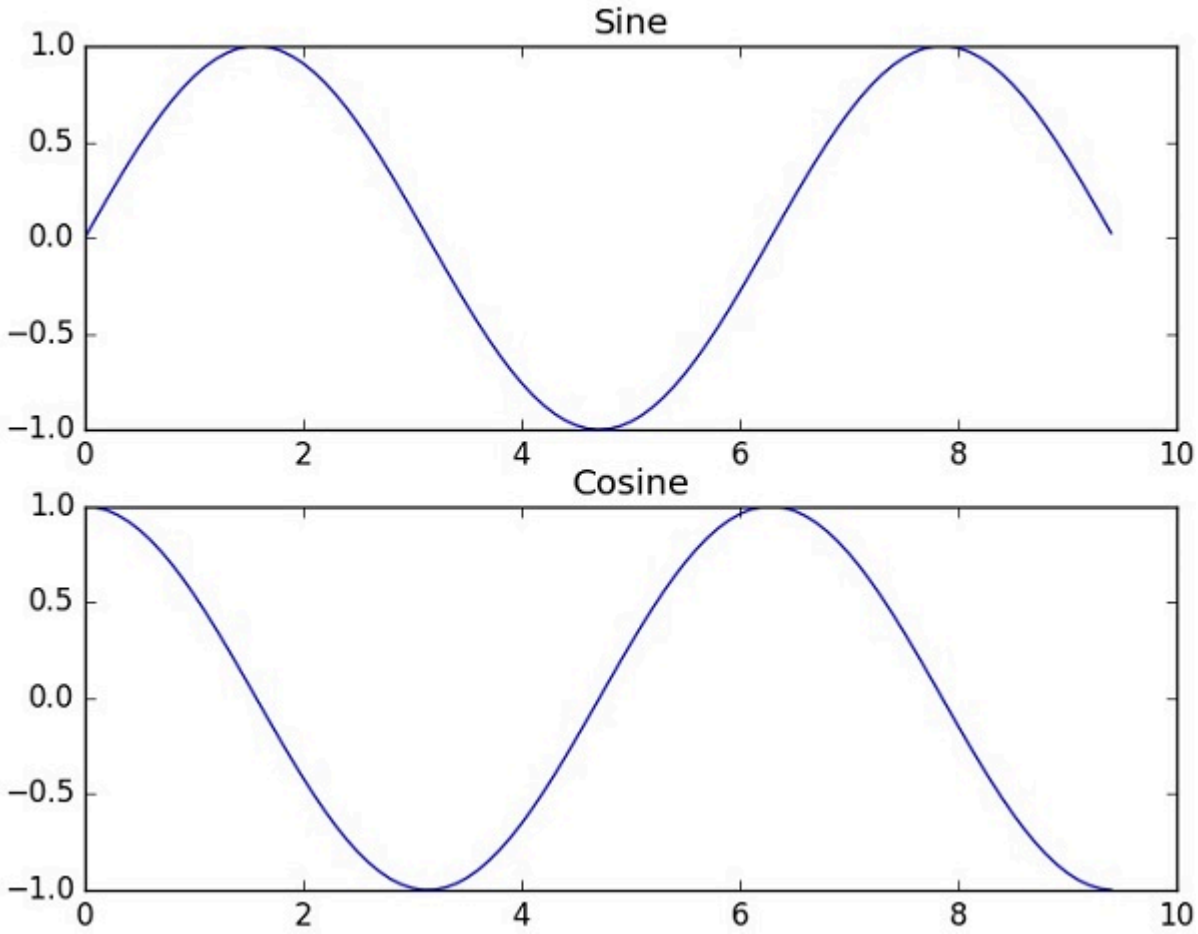
A função subplot() permite plotar coisas diferentes na mesma figura. No script a seguir, os valores de **seno** e **cosseno** são plotados.

Exemplo

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

# Compute the x and y coordinates for points on sine and cosine curves
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

# Set up a subplot grid that has height 2 and width 1,
# and set the first such subplot as active.
plt.subplot(2, 1, 1)

# Make the first plot
plt.plot(x, y_sin)
plt.title('Sine')
```

```
# Set the second subplot as active, and make the second plot.
plt.subplot(2, 1, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

# Show the figure.
plt.show()
```

O código acima deve produzir a seguinte saída -



bar()

O **submódulo pyplot** fornece a função **bar()** para gerar gráficos de barras. O exemplo a seguir produz o gráfico de barras de dois conjuntos **de** matrizes **xey** .

Exemplo

```
from matplotlib import pyplot as plt
x = [5,8,10]
y = [12,16,6]

x2 = [6,9,11]
y2 = [6,15,7]
```

```
plt.bar(x, y, align = 'center')
plt.bar(x2, y2, color = 'g', align = 'center')
plt.title('Bar graph')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.show()
```

Este código deve produzir a seguinte saída -