

Matplotlib

Fernando Anselmo - Versão 1.0

Importar a Biblioteca

Importar a biblioteca:
`import matplotlib.pyplot as plt`

No Jupyter usar o comando:
`%matplotlib inline`

Pyplot é uma coleção de funções que fazem a biblioteca funcionar como o MATLAB. Cada função pyplot faz alguma alteração na plotagem do gráfico.

OBSERVAÇÃO: Para os exemplos importar a NumPy:
`import numpy as np`

Funções Padrões

Criar um gráfico:
`plt.plot(X, y)`

Mostrar o gráfico:
`plt.show()`

Modificar o tamanho da figura:
`plt.figure(figsize = [larg, alt])`

Principais tipos de gráficos:

<code>plot()</code>	Gráfico de Linhas X e y.
<code>bar()</code>	Gráfico de Barras Verticais (<code>barh()</code> para horizontais).
<code>hist()</code>	Histograma.
<code>polar()</code>	Gráfico de coordenadas cartesianas.
<code>pie()</code>	Gráfico de Pizza.
<code>boxplot()</code>	Modelo boxplot.

Comandos de Texto

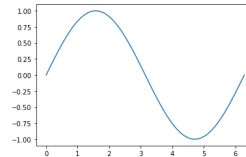
Comandos mais utilizados do Pyplot:

<code>text()</code>	Texto em um local específico dos eixos.
<code>xlabel()</code>	Legenda para o eixo X.
<code>ylabel()</code>	Legenda para o eixo y.
<code>title()</code>	Título para o gráfico.
<code>suptitle()</code>	Título da figura.
<code>figtext()</code>	Texto em um local específico da figura.
<code>annotate()</code>	Anotação ao eixos com uma seta opcional.
<code>legend()</code>	Opções de posicionamento do parâmetro <code>loc</code> : best right center [[upper, lower, center] [right, left, center]]

Exemplo de Gráfico com 1 elemento

```
x = np.linspace(0, 2 * np.pi, num = 100)
y = np.sin(x)
plt.plot(x, y)
plt.show()
```

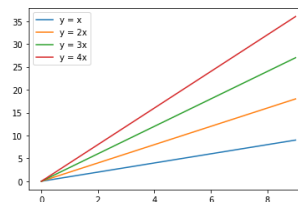
Geramos um gráfico que mostra a curva senóide de valores entre 0 e 2 vezes o **Pi**.



Exemplo de Gráfico com vários elementos

```
x = np.arange(10)
plt.plot(x, x)
plt.plot(x, 2 * x)
plt.plot(x, 3 * x)
plt.plot(x, 4 * x)
plt.legend(['y = x', 'y = 2x', 'y = 3x', 'y = 4x'], loc='upper left')
plt.show()
```

Neste gráfico temos uma amostra de como várias funções podem se combinar.



Tente este exemplo

```
plt.figure(figsize = [8,7])
t = np.arange(0,2 * np.pi, 0.1)
x = 16 * np.sin(t) ** 3
y = 13 * np.cos(t) - 5 * np.cos(2 * t) - 2 * np.cos(3 * t) - np.cos(4 * t)
plt.plot(x,y)
plt.title("Viva Python")
```

Matplotlib - SubGráficos na Figura

Inserir subgráfico:
`plt.subplot(nLin, nCol, indice)`

Por exemplo:
`ax1 = plt.subplot(2, 2, 1)`

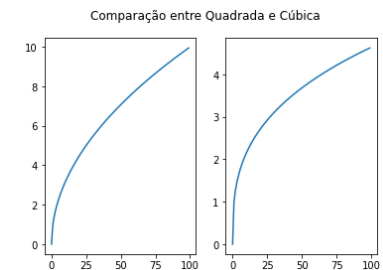
Retirar a borda:
`ax1 = plt.subplot(221, frameon=False)`

Adicionar o subgráfico a figura:
`plt.subplot(ax1)`

Excluir o subgráfico da figura:
`plt.delaxes(ax1)`

Exemplo de 2 SubGráficos

```
xvals = np.arange(0, 100, 1)
r2 = np.sqrt(xvals)
r3 = np.cbrt(xvals)
ax1 = plt.subplot(121)
ax1.plot(xvals, r2)
plt.subplot(ax1)
ax2 = plt.subplot(122)
ax2.plot(xvals, r3)
plt.suptitle('Comparação entre Raízes')
plt.subplot(ax2)
plt.show()
```



Ou com `axes`:

```
fig, axes = plt.subplots(1, 2,
figsize=(10,4))
axes[0].plot(xvals, r2)
axes[0].set_title("Quadrada")
axes[1].plot(xvals, r3)
axes[1].set_title("Cúbica")
```