Nota 0,1

Matplotlib

Plotar dados em gráficos.
Importar a biblioteca $matplotlib$ como mpl, as funções de plotagem $pyplot$ como plt e a biblioteca $numpy$ como np.

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

Componentes principais

Artist é uma classe abstrata que é base para os objetos que são renderizados em uma Figure.

Figure é a instância de nível mais alto da classe $Artist$. Todos os elementos visíveis em uma $Figure$ (figura) são subclasses da classe $Artist$.

Axes é um $Artist$ anexado a uma Figure que contém uma região para plotagem de dados. Inclui dois ou três (no caso de 3D) objetos $Axis$.

Axis é uma classe para os eixos de umm subplot. Fornece ticks e rótulos de ticks para as escalas dos dados nos eixos da figura. Cada Eixo tem um título (definido por meio de set_title()), um rótulo x (definido por meio de set_xlabel()) e um rótulo y definido por set_ylabel()).

Função $pyplot.subplots$: Cria uma figura e um conjunto de áreas de plotagens (subplot). Cada subplot é um objeto $Axes$ que contém dois ou tês eixos de coordenadas ($Axis$).

Função $Axes.plot$: Recebe um conjunto de dados para cada eixo ($Axis$) e plota os dados recebidos.

Tipos de entradas para funções de plotagem

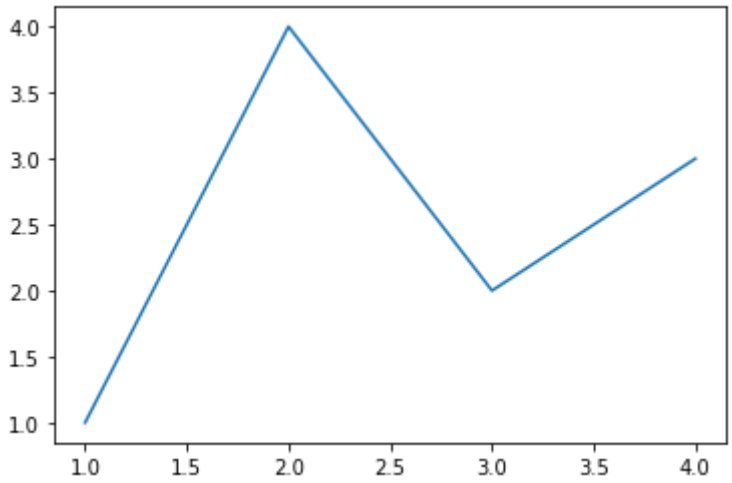
- numpy.array
- numpy.ma.masked_array
- objetos que podem ser convertidos pela função numpy.asarray

A maioria dos métodos poderá receber um objeto endereçável: $dict$, $numpy.matrix$ ou $pandas.DataFrame$.
Classes semelhantes a arrays, como $pandas.DataFrame$ e $numpy.matrix$, podem não funcionar conforme o esperado e devem ser convertidos para numpy.array antes da plotagem.

Exemplo

- Criar uma figura com uma única área de plotagem (ax)
- Usar ax.plot para desenhar dados

```
fig, ax = plt.subplots()  # Cria figura com um único subplot (eixo ax)
ax.plot([1, 2, 3, 4], [1, 4, 2, 3]);  # Plotar os dados
```

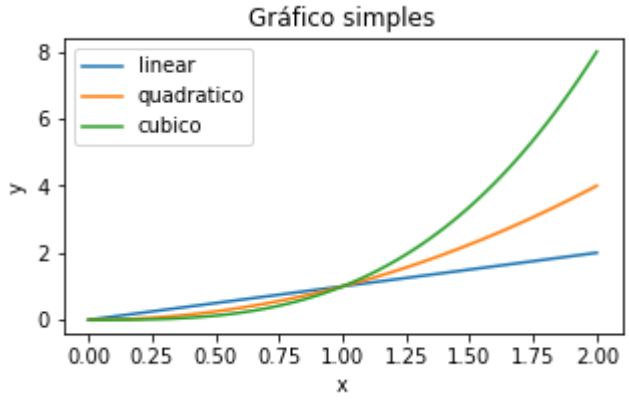


Estilos de codificação

- orientado a objetos: criar Figures e Axes e chamar seus métodos
- automático: utilizar no pyplot para criar e gerenciar automaticamente as Figures e Axes

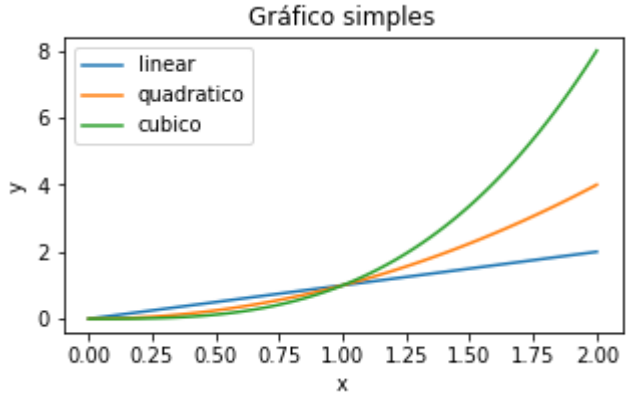
Orientado a objetos

```
x = np.linspace(0, 2, 100)  # Dados

# Nos dois estilos usamos pyplot.figure para criar a figura
fig, ax = plt.subplots(figsize=(5, 2.7))
ax.plot(x, x, label='linear')  # Plotar y = x
ax.plot(x, x**2, label='quadratico')  # Plotar y = x**2
ax.plot(x, x**3, label='cubico')  # Plotar y = x**3
ax.set_xlabel('x')  # Rótulo do eixo x (Axis)
ax.set_ylabel('y')  # Rótulo do eixo y (Axis)
ax.set_title("Gráfico simples")  # Título do subplot (Axe)
ax.legend();  # Legenda
```



Automático

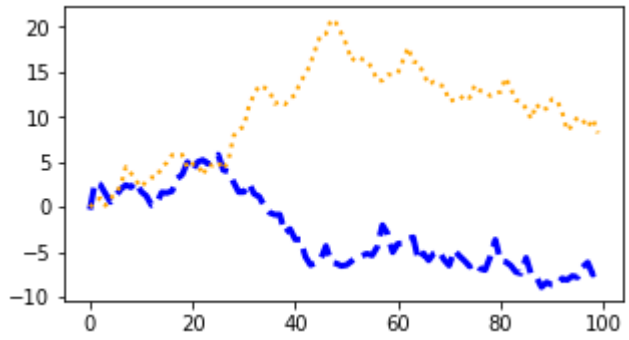
```
x = np.linspace(0, 2, 100)  # Dados

plt.figure(figsize=(5, 2.7))
plt.plot(x, x, label='linear')  # Plotar y = x
plt.plot(x, x**2, label='quadratico')  # Plotar y = x**2
plt.plot(x, x**3, label='cubico') # Plotar y = x***3
plt.xlabel('x')
plt.ylabel('y')
plt.title("Gráfico simples")
plt.legend();
```



Estilos

A maioria dos métodos de plotagem tem opções de estilo para os objetos da classe $Artist$.
As linhas podem ter estilos diferentes.
Exemplo: definição manual da cor, da largura de linha e do estilo de linha.

```
dados1, dados2, dados3, dados4 = np.random.randn(4, 100)  # Quatro datasets aleatórios

fig, ax = plt.subplots(figsize=(5, 2.7))
x = np.arange(len(dados1))
ax.plot(x, np.cumsum(dados1), color='blue', linewidth=3, linestyle='--')
ax.plot(x, np.cumsum(dados2), color='orange', linewidth=2, linestyle=':');
```



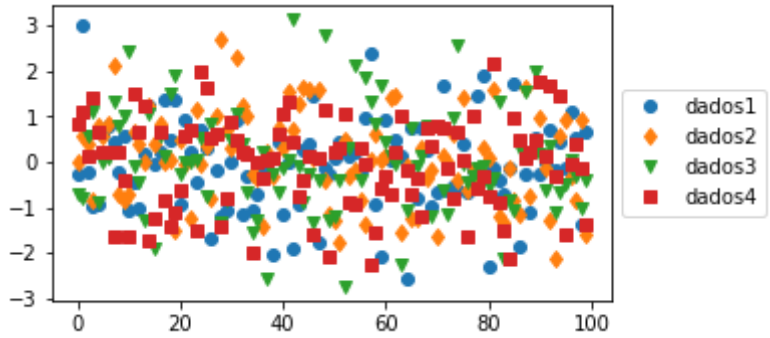
Marcadores

O tamanho do marcador depende do método que está sendo usado:

- plot especifica o tamanho do marcador em pontos e geralmente é o "diâmetro" ou a largura do marcador
- scatter especifica o tamanho do marcador como aproximadamente proporcional à área visual do marcador.

Há vários estilos de marcador disponíveis.
Legendas podem ser colocada fora do quadro da figura.

```
fig, ax = plt.subplots(figsize=(5, 2.7))
ax.plot(dados1, 'o', label='dados1')
ax.plot(dados2, 'd', label='dados2')
ax.plot(dados3, 'v', label='dados3')
ax.plot(dados4, 's', label='dados4')
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5));
```

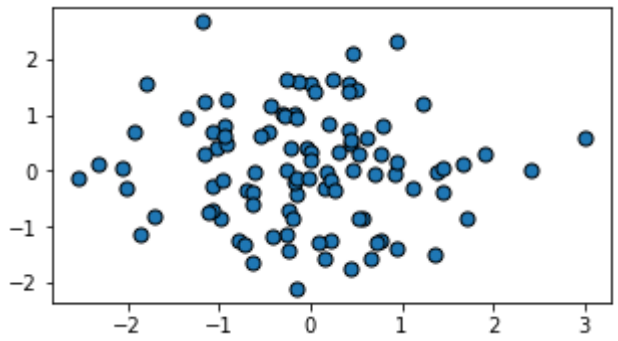


Cores

Um $Artist$ pode ter mais de uma cor.
No diagrama de dispersão (scatter) a seguir:

- a borda dos marcadores possue cor diferente do seu interior
- o tamanho do marcador é modificado pelo parâmetro s

```
fig, ax = plt.subplots(figsize=(5, 2.7))
ax.scatter(dados1, dados2, s=50, facecolor='C0', edgecolor='k');
```



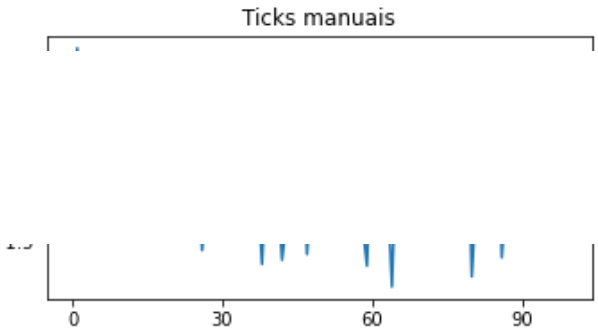
Rótulos e intervalos de eixos

```
mu, sigma = 115, 15
x = mu + sigma * np.random.randn(10000)
fig, ax = plt.subplots(figsize=(5, 2.7))
# Histograma
n, bins, patches = ax.hist(x, 50, density=1, facecolor='C0', alpha=0.75)

ax.set_xlabel('Tamanho [cm]')
ax.set_ylabel('Probabilidade')
ax.set_title('Vamos estudar o que é um histograma')
ax.axis([55, 175, 0, 0.03])
ax.grid(True);
```

Múltiplos Axes

A função subplots pode criar múltiplos $Axes$ (informar quantidade de linhas e colunas).
Os ticks podem ser atribuídos automaticamente ou manualmente.

```
x = np.arange(len(dados1))  # dados eixo x

fig, axes = plt.subplots(1, 2, figsize=(12, 2.7))
axes[0].plot(x, dados1)
axes[0].set_title('Ticks automáticos')

axes[1].plot(x, dados1)
axes[1].set_xticks(np.arange(0, 100, 30))
axes[1].set_yticks([-1.5, 0, 1.5])
axes[1].set_title('Ticks manuais');

plt.savefig('figura.png')
```



Entrega

Imprimir o notebook para pfd e entregar no AVA