

**ANALYSE DE DONNÉES AVEC PYTHON MATPLOTLIB**

Amara Dalila

Python pour l’analyse de données

Les packages Python pour l »analyse de données

Application

[dalilaa.amara@gmail.com](mailto:dalilaa.amara@gmail.com)

# Le package matplotlib:

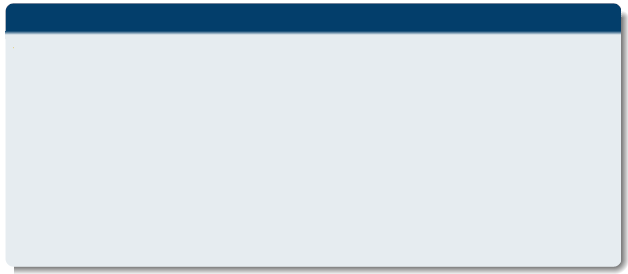
## **Matplotlib** est une bibliothèque de traçage de graphiques en python qui sert d'utilitaire de visualisation.

**Matplotlib** a été créé par John D. Hunter.

**Matplotlib** est open source et nous pouvons l'utiliser librement.

# Le package matplotlib: Installation et importation

Syntaxe d’utilisation



**Installation**

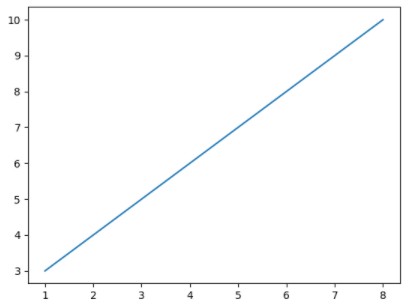
Si Python et PIP sont déjà installés sur un système, l'installation de matplotlib est très simple: Installez-le à l'aide de cette commande :

## C:\Users\Votre nom>pip install matplotlib

NB. Si cette commande échoue, utilisez une distribution python sur

laquelle matplotlib est déjà installé, comme Anaconda **Importation**

* La plupart des utilitaires Matplotlib se trouvent sous le sous-module **pyplot** et sont généralement importés sous l'alias plt :
* La fonction plot() est utilisée pour tracer des points (marqueurs) dans un diagramme.



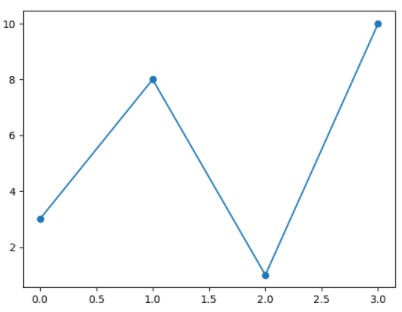
|  |  |
| --- | --- |
| **La fonction plot()** | **Exemple** |
| * Par défaut, la fonction plot() Dessiner une ligne de la position (1,3)   trace une ligne de point à point. à la position (8, 10)   * La fonction prend des import matplotlib.pyplot as plt   paramètres pour spécifier des import numpy as np  xpoints = np.array([1, 8])  points dans le diagramme. ypoints = np.array([3, 10])   * Le paramètre 1 est un tableau plt.plot(xpoints, ypoints)   contenant les points sur l'axe des plt.show()  x.   * Le paramètre 2 est un tableau contenant les points sur l'axe des y. * Si nous devons tracer une ligne de (1, 3) à (8, 10), nous devons passer deux tableaux [1, 8] et [3,   10] à la fonction plot. | |

|  |  |
| --- | --- |
| **La fonction plot()** | **Exemple** |
| * On peut utiliser l’argument import matplotlib.pyplot as plt   **marker** pour souligner chaque import numpy as np  point avec un marqueur spécifié : | |

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')

plt.show()



**La fonction scatter()**

**Exemple**

* + La fonction **scatter()** permet de dessiner un nuage de points.
  + La fonction scatter() trace un point pour chaque observation.
  + Il a besoin de deux tableaux de même longueur, un pour les valeurs de l'axe des x et un pour les valeurs de l'axe des y :

import matplotlib.pyplot as plt

import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])

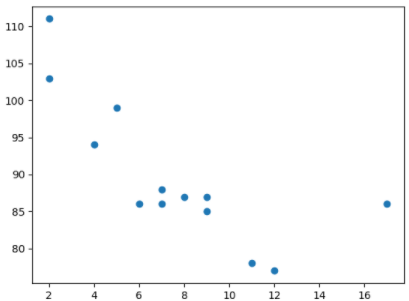
y =

np.array([99,86,87,88,111,86,103,87,94,78,77

,85,86])

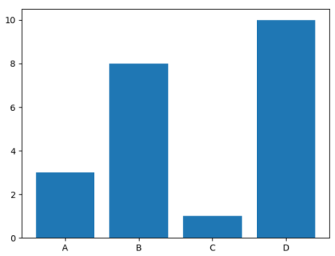
plt.scatter(x, y)

plt.show()



|  |  |
| --- | --- |
| **La fonction bar()** | **Exemple** |
| * La fonction **bar()** permet de import matplotlib.pyplot as plt   dessiner des graphiques à import numpy as np  barres : x = np.array(["A", "B", "C", "D"]) | |

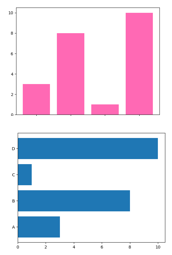
y = np.array([3, 8, 1, 10])

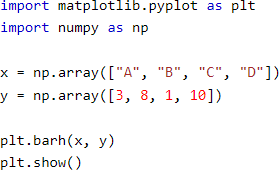
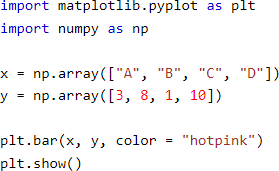


plt.bar(x,y)

plt.show()

**La fonction bar() Exemple**





|  |  |
| --- | --- |
| **La fonction pie()** | **Exemple** |
| * La fonction **pie()** permet de import matplotlib.pyplot as plt   dessiner des camemberts : import numpy as np | |

y = np.array([35, 25, 25, 15])



plt.pie(y)

plt.show()