

# Python pour les data scientistes

## Chapitre 6 : Le package Matplotlib

### 1. Introduction

Matplotlib est une bibliothèque permettant le traçage et la visualisation de graphiques en python. Matplotlib a été créée par John D. Hunter, c'est une librairie open source et gratuite.

La librairie Matplotlib est principalement écrite en python, quelques parties sont écrites en C, Objective-C et Javascript pour la compatibilité de la plateforme.

### 2. Installation et utilisation

L'installation de Matplotlib est très facile : **pip install matplotlib**

Si cette commande échoue, nous pouvons utiliser une distribution python dans laquelle Matplotlib est déjà installé, comme Anaconda, Spyder, etc.

Une fois la librairie installée, vous pouvez l'utiliser en l'important d'abord :

```
import matplotlib
```

### 3. Pyplot de Matplotlib

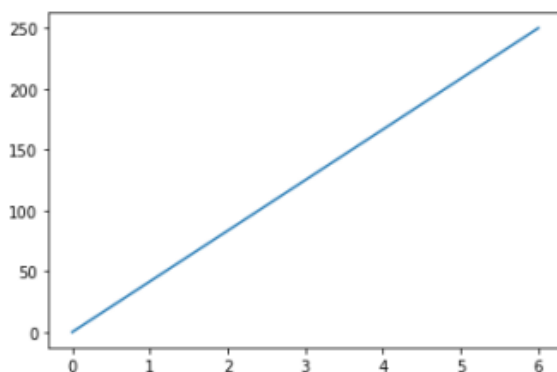
Matplotlib est une librairie riche en utilitaires, la plupart des utilitaires de Matplotlib se trouvent sous le sous-module pyplot, et sont généralement importés sous l'alias plt :

```
import matplotlib.pyplot as plt
```

### Exemple

Tracez une ligne dans un diagramme de la position (0,0) à la position (6,250) :

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array([0, 6])
y = np.array([0, 250])
plt.plot(x, y)
plt.show()
```



### 4. Tracer des points x et y

La fonction plot() est utilisée pour dessiner des points (marqueurs) dans un diagramme. Par défaut, la fonction plot() trace une ligne de point à point. Elle prend des paramètres pour spécifier les points dans le diagramme.

- Le paramètre 1 est un tableau contenant les points sur l'axe des x.
- Le paramètre 2 est un tableau contenant les points sur l'axe des ordonnées.

Pour dessiner une ligne de (1, 3) à (8, 10), nous devons passer deux tableaux [1, 8] et [3, 10] à la fonction plot.

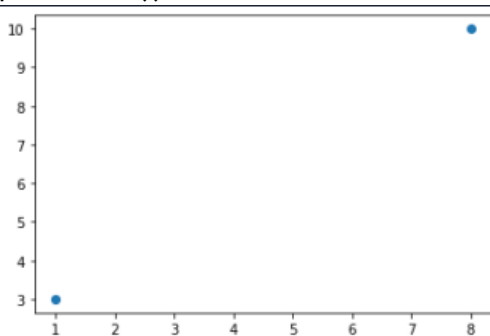
### Exemple

Ce code permet de tracer une ligne dans un diagramme de la position (1, 3) à la position (8, 10) :

```
import matplotlib.pyplot as plt
import numpy as np
# x l'axe horizontale
x = np.array([1, 8])
# y l'axe horizontale
y = np.array([3, 10])
plt.plot(x, y)
plt.show()
```

Pour ne tracer que les marqueurs, vous pouvez utiliser le paramètre de notation raccourci 'o', qui signifie 'anneaux' :

```
import matplotlib.pyplot as plt
import numpy as np
x= np.array([1, 8])
y= np.array([3, 10])
plt.plot(x, y, 'o')
plt.show()
```



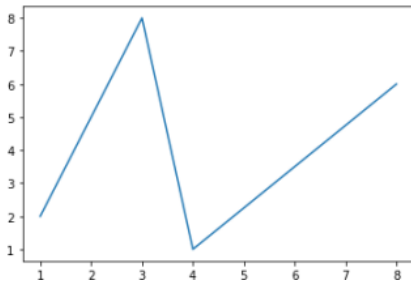
## 5. Points multiples

Il est possible de créer autant de point, seulement, il faut s'assurer qu'il y a le même nombre de points sur les deux axes.

### Exemple

Tracez une ligne dans un diagramme de la position (1, 3) à (2, 8) puis à (6, 1) et enfin à la position (8, 10) :

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array([1, 3, 4, 8])
y = np.array([2, 8, 1, 6])
plt.plot(x, y)
plt.show()
```



## 6. Les marqueurs

Vous pouvez utiliser le mot clé argument marqueur pour mettre en valeur chaque point avec un marqueur spécifié :

### Exemple

Marquer chaque point avec un cercle :

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([3, 8, 1, 10])
plt.plot(y, marker = 'o')
plt.show()
```

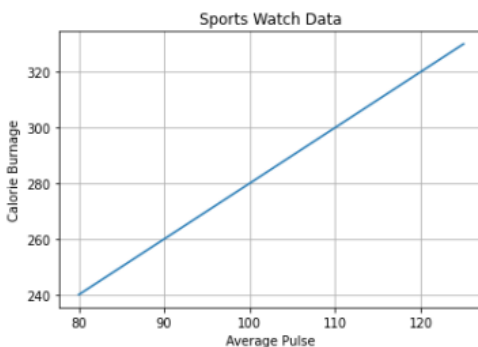
## 7. Ajout de lignes de grille

Avec Pyplot, vous pouvez utiliser la fonction `grid()` pour ajouter des lignes de grille au graphique.

### Exemple

Ajouter des lignes de grille au graphique :

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])
plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")
plt.plot(x, y)
plt.grid()
plt.show()
```



## 8. Dessiner plusieurs graphiques

Avec la fonction `subplot()`, vous pouvez dessiner plusieurs graphiques dans une seule figure.

### Exemple

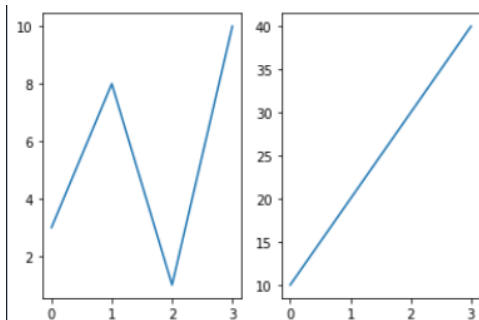
```
import matplotlib.pyplot as plt
import numpy as np
```

```
#plot 1:
```

```
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
plt.subplot(1, 2, 1)
plt.plot(x,y)
```

```
#plot 2:
```

```
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.show()
```



## 9. Les barres

La méthode `bar()` permet de dessiner des barres.

### Exemple

Dessiner 4 barres:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
plt.bar(x,y)
plt.show()
```

- Si vous souhaitez que les barres soient affichées horizontalement plutôt que verticalement, utilisez la fonction `barh()` : `plt.barh(x, y)`
- La méthode `barh()` prend l'argument mot-clé `height` pour définir la hauteur des barres : `plt.barh(x, y, height = 0.1)` la valeur de `height` par défaut : 0.8

## 10. Les graphiques circulaires

La méthode `pie()` permet de dessiner des diagrammes circulaires :

### Exemple

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
plt.pie(y)
plt.show()
```

Ajoutez des étiquettes au camembert à l'aide du paramètre label. Le paramètre label doit être un tableau avec un label pour chaque coin :

### Exemple

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
plt.pie(y, labels = mylabels)
plt.show()
```

Pour afficher la légende, ajouter avant la méthode show : `plt.legend()`

Pour afficher la légende avec titre : `plt.legend(title = "titre ici:")`