

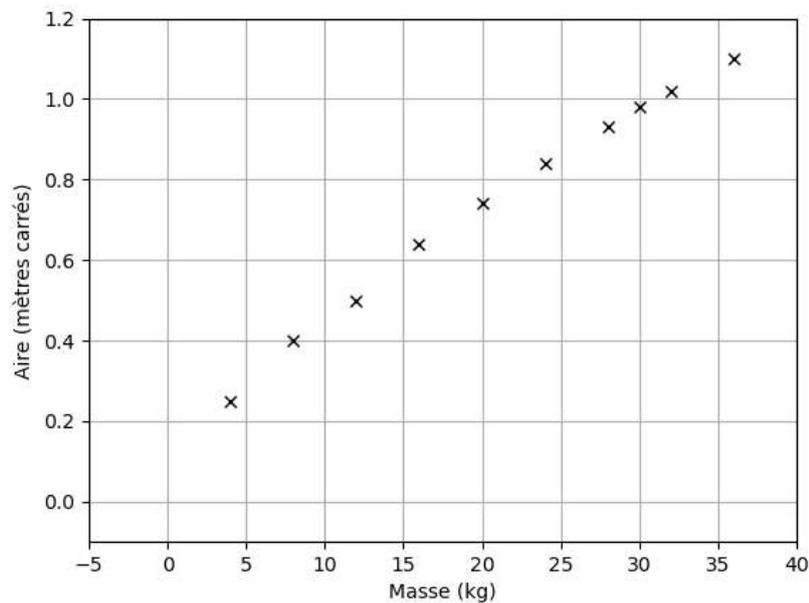
# Python pour les statistiques

## I/ Représentations graphiques

Python dispose d'un module permettant de tracer des représentations graphiques en deux et en trois dimensions : **matplotlib**. Ce module sera toujours importé sous la forme : `import matplotlib.pyplot as plt`.

### 1. Premier exemple

```
1 import matplotlib.pyplot as plt
2
3 exemple_X = [4 , 8 , 12 , 16 , 20 , 24 , 28 , 30 , 32 , 36]
4 exemple_Y = [0.25 , 0.4 , 0.5 , 0.64 , 0.74 , 0.84 , 0.93 , 0.98 , 1.02 , 1.1]
5
6 def nuage(X,Y):
7     plt.plot(X,Y, 'kx')
8     plt.grid()
9     plt.xlim(-5,40)
10    plt.ylim(-0.1,1.2)
11    plt.xlabel("Masse (kg)")
12    plt.ylabel("Aire (metres carrés)")
13    plt.show()
```



## 2. Commandes usuelles

Les commandes les plus utiles sont regroupées dans la tableau ci-dessous.

Dans tout ce tableau, **X** et **Y** désignent deux listes de nombres de même longueur représentant respectivement les abscisses et les ordonnées des points que l'on souhaite représenter.

Python	Interprétation
<code>plt.plot(X,Y)</code>	Cette commande place les points dont les abscisses sont contenues dans X et les ordonnées dans Y. Par défaut, ces points sont reliés. Cette commande dispose de nombreux arguments optionnels qui sont présentés à la page suivante.
<code>plt.xlim()</code>	Cette commande prend en argument deux nombres $x_{\min}$ et $x_{\max}$ et adapte la fenêtre en conséquence.
<code>plt.ylim()</code>	Cette commande prend en argument deux nombres $y_{\min}$ et $y_{\max}$ et adapte la fenêtre en conséquence.
<code>plt.grid( )</code>	Cette commande ne prend pas d'argument. Elle permet d'afficher un quadrillage.
<code>plt.axis('equal')</code>	Cette commande rend le repère orthonormé.
<code>plt.xlabel( )</code>	Cette commande prend en argument une chaîne de caractères. Elle permet de donner un titre à l'axe des abscisses.
<code>plt.ylabel( )</code>	Cette commande prend en argument une chaîne de caractères. Elle permet de donner un titre à l'axe des ordonnées.
<code>plt.title( )</code>	Cette commande permet de donner un titre au graphique.
<code>plt.show ( )</code>	Cette commande ne prend pas d'argument. Elle affiche tout ce qui a été créé depuis le début de la figure.

La commande `plt.plot()` possède des arguments optionnels. Les plus utilisés sont rassemblés dans les tableaux suivants :

Symbole	Chaîne
points non reliés	'.'
ronds non reliés	'o'
hexagones non reliés	'h'
plus non reliés	'+'
croix non reliées	'x'
étoiles non reliées	'*'
carrés non reliés	's'

Ligne	Chaîne
trait plein	'_'
pointillés	'--'
alterné	'-.'

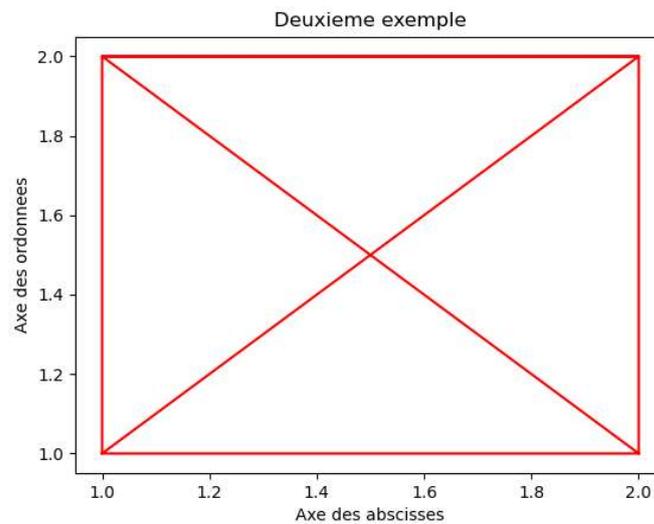
Couleur	Chaîne
bleu	'b'
rouge	'r'
vert	'g'
cyan	'c'
magenta	'm'
noir	'k'

### Remarques

- **Attention** : X et Y doivent être de type **list** (et pas autre chose).
- Pour connaître la liste complète des arguments optionnels de `plt.plot`, on peut taper directement dans le shell `help(plt.plot)`. De manière générale, la commande `help( )` peut s'avérer très utile !

### 3. Deuxième exemple

Ecrire une fonction Python **exemple2** permettant de créer la figure suivante :



#### 4. Troisième exemple

On considère la suite  $(u_n)$  définie par :  $\forall n \in \mathbb{N}, u_n = \cos(n) + \sin(n)$  .

Ecrire une fonction Python **nuage\_u** qui prend en argument un entier naturel **n** et qui affiche le nuage de points associé à  $u_0, \dots, u_n$  .

#### 5. Quatrième exemple

On considère la suite  $(v_n)$  définie par :  $v_0 = 1$  et  $\forall n \in \mathbb{N}, v_{n+1} = \frac{v_n}{n+1}$  .

Ecrire une fonction Python **nuage\_v** qui prend en argument un entier naturel **n** et qui affiche le nuage de points associé à  $v_0, \dots, v_n$  .