

## Régression linéaire avec matplotlib.pyplot

Rappelons le programme officiel :

|                      |  |
|----------------------|--|
| Régression linéaire. | Utiliser la fonction <b>polyfit</b> de la bibliothèque <b>numpy</b> (sa spécification étant fournie) pour exploiter des données. |
|----------------------|--|

On importe la bibliothèque **numpy** avec l'alias **np**. On appellera alors les différentes fonctions de cette bibliothèque avec le préfixe **np**. (avec un point séparant np du nom de la fonction).

On entre les valeurs des entrées x et y sous la forme de deux colonnes de valeurs avec numpy.

```
import numpy as np
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 3, 4, 5, 6])
```

Vérifier qu'il y a bien autant de valeurs de x que de y – par exemple avec :

```
print( len (x))           affiche le nombre de valeurs dans la colonne x.
```

Il est important avant de vouloir faire une régression linéaire de vérifier que celle-ci se justifie, autrement dit que la représentation graphique  $y = f(x)$  donne bien des points correctement alignés, aux erreurs de mesure près.

Dans la bibliothèque **numpy** se trouve la fonction **polyfit**, qui permet de trouver le polynôme qui colle le mieux à la courbe expérimentale – ici on prendra un polynôme de degré 1.

Exemple :

```
import numpy as np
x = np.array([0, 1, 2, 3, 4])           # pour pouvoir utiliser polyfit, on ne peut pas utiliser un simple
ensemble de valeurs, comme on a fait ci-dessus, mais on crée une ligne de tableau.
y = np.array([1, 2, 3, 4, 5])         # Vérifier que vous avez autant de valeurs de x que de y.
[a, b] = np.polyfit ( x, y, 1)        #on crée le polynôme de degré 1 qui colle le plus à la courbe y fonction de x,
donc y = a * x +b
print(a, b)                           #Cela affiche les coefficients a et b du polynôme précédent
```

On peut judicieusement superposer les points de coordonnées (x, y) précédents et la droite de régression linéaire :

```
import matplotlib.pyplot as plt
plt.plot(x, y, 'ro')                   # pointe les points de coordonnées (x,y) par des ronds rouges
ymodel = a * x + b
plt.plot(x, ymodel, 'b:')              # trace la courbe de régression linéaire en pointillés bleus
plt.show()
```