



# LPSARI UE3 TP 2

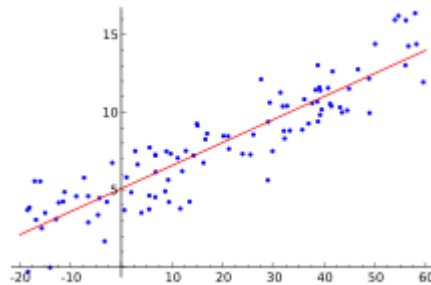
## Curve fitting

**Vous devez rendre des programmes commentés et expliquer vos choix de conception : choix des structures de données, enchaînement des fonctions,....**

### 1. Ajustement affine

**(voir article de wikipedia)**

Lorsque le nuage des points  $(x,y)$  de deux variables aléatoires  $X,Y$  semble s'organiser autour d'une droite, on peut envisager de trouver une relation linéaire qui permet de déduire approximativement une valeur de  $Y$  en fonction d'une valeur de  $X$ .



Il s'agit de trouver deux nombres réels  $a$  et  $b$  tel qu'on ait approximativement :

$$Y = a \cdot X + b$$

Plus précisément, l'objectif est de minimiser la somme des écarts entre les points réels ( $Y$ ) et les points recalculés ( $a \cdot X + b$ ). Il faut donc trouver  $a$  et  $b$  qui rendent minimale  $S$  :

$$S = \sum_{i=0}^N (y_i - a \cdot x_i - b)^2$$

La méthode dite des moindres carrés (régression linéaire) permet de déterminer  $a, b$  en calculant :

- les moyennes :

$$\bar{x} = \frac{1}{n} \sum_i x_i, \quad \bar{y} = \frac{1}{n} \sum_i y_i$$



## LPSARI UE3 TP 2

### Curve fitting

-les variances :

$$\text{var}(x) = \frac{1}{n} \sum_i (x_i - \bar{x})^2 = \frac{1}{n} \sum_i x_i^2 - \bar{x}^2$$

$$\text{var}(y) = \frac{1}{n} \sum_i (y_i - \bar{y})^2 = \frac{1}{n} \sum_i y_i^2 - \bar{y}^2$$

-la covariance :

$$\text{cov}(x, y) = \frac{1}{n} \sum_i (x_i - \bar{x}) \cdot (y_i - \bar{y}) = \frac{1}{n} \sum_i x_i y_i - \bar{x} \bar{y}$$

a est alors donné par :

$$a = \frac{\text{cov}(x, y)}{\text{var}(x)}$$

et enfin b par :

$$b = \bar{y} - a \cdot \bar{x}$$

On se propose d'essayer de retrouver la température extérieure (*extTemp*) à partir de la mesure de la température interne du microcontrôleur (*microTemp*) pour palier à une défaillance éventuelle du capteur extérieur.

- À partir d'un fichier JASON\_PC\_FORTEM-xxxx tracer l'évolution temporelle de *extTemp* et *microTemp* ainsi que le nuage de points (*extTemp, microTemp*), et montrer qu'un ajustement affine est envisageable.

- Écrire une fonction python qui effectue la régression linéaire des deux séries x,y (array numpy) et retourne les coefficients a et b.

```
def regression_lineaire(x,y) :
```

```
...
```

```
return a,b
```

- Tracer la valeur recalculée de *extTemp* en plus sur le graphique précédent.

- Tracer un graphique montrant l'écart entre la valeur mesurée et la valeur calculée.

- Conclure.



## LPSARI UE3 TP 2

### Curve fitting

## 2. Curve Fitting

La méthode précédente peut être étendue au cas où la fonction à retrouver n'est pas une simple droite mais une fonction quelconque. Dans ce cas les calculs sont beaucoup plus compliqués et l'unicité de la solution n'est pas toujours garantie. Les algorithmes peuvent être assez complexes car il faut éviter de s'arrêter sur des minimums locaux. Heureusement, la bibliothèque python `scipy` fournit une fonction d'optimisation très puissante utilisable pour ces cas : `curve_fit`

Voir :

[docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve\\_fit.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)

(et bien comprendre l'exemple)

Nous allons l'utiliser pour retrouver les paramètres de la loi mathématique d'une thermistance en fonction de valeurs mesurées.

En première approximation, la résistance d'une thermistance suit la loi :

$$R(T) = R_{T_0} \cdot e^{(B(1/T - 1/T_0))}$$

où  $T$  est la température absolue en kelvin :  $T_{(K)} = t_{(°C)} + 273,15 K$

et  $R_{T_0}$  est la résistance à la température absolue  $T_0$

L'objectif est de trouver la "meilleure" valeur pour les paramètres  $R_{T_0}$  et  $B$  en fonction des valeurs mesurées. On cherche la valeur des paramètres qui minimisent l'erreur entre la courbe réelle et la courbe estimée par la formule.

Les fonctions à optimiser doivent être écrites en python avec les fonctions mathématiques de numpy car elles doivent pouvoir être utilisées avec des listes (et des array de numpy). En particulier utilisez `np.add` pour les additions.

Estimer les paramètres  $R_{T_0}$  et  $B$  pour les deux thermistances en fixant  $t_0$  à  $25°C$ .

Faire un graphique montrant les points de mesures, la courbe estimée obtenue ainsi que l'écart (en Ohm) entre les valeurs mesurées et les valeurs calculées.

Pour améliorer l'approximation on peut utiliser la formule :

$$R(T) = R_{T_0} \cdot e^{(A + \frac{B}{T} + \frac{C}{T^2} + \frac{D}{T^3})}$$



## LPSARI UE3 TP 2

### Curve fitting

Estimer les paramètres  $R_{T_0}$ , A,B,C,D pour les deux thermistances.  
Faire un graphique montrant les points de mesures, la courbe estimée obtenue ainsi que l'écart (en Ohm) entre les valeurs mesurées et les valeurs calculées.