

TP4 Numpy - Régression linéaire

2024 - 2025

1 Régression linéaire

On cherche à effectuer une régression linéaire sur 4 sets de données (notés I, II, III et IV) qui sont représentés sur le tableau 1.

Set I		Set II		Set III		Set IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	10.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	13.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	9.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	11.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	14.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	6.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	4.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	12.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	7.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	5.0	6.89

TABLE 1 – Données à traiter.

Pour chaque set on cherche à obtenir la droite de régression (coefficients) et les statistiques associées. Dans un premier lieu vous tracerez sur 4 **subplots** les 4 jeu de données et vous ferez en sorte que la présentation de la figure soit parfaite. Pour chaque set vous devrez calculer et afficher la droite de régression ainsi que l'équation sur le graphique (via la légende ou en texte placé sur le graphique).

Les outils statistiques dans Python sont dans `scipy.stats` et l'outil `linregress` vous permettre de réaliser les régressions. Attention les données doivent être triées selon l'axe des x et vous pouvez obtenir les indices qui trient x à l'aide de la commande suivante :

```
1 indsort=np.argsort(x) # on obtient les indices qui trient x
```

Le graphique produit doit ressembler à celui de la figure 1.

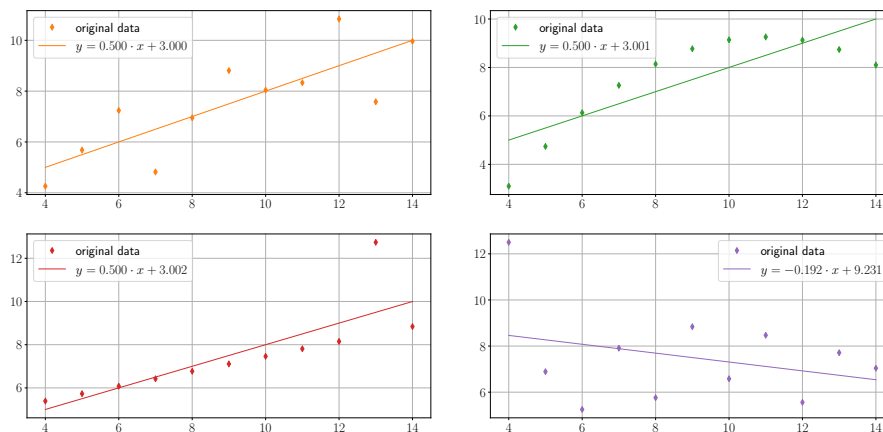


FIGURE 1 – Régression linéaire

2 Numpy

Numpy est une bibliothèque très puissante pour la manipulation des tableaux et des matrices. Nous allons traiter un exemple complet qui permettra de comprendre l'architecture d'un fichier Python (pas de cellules `###`).

On voit donc réaliser une fonction qui permettra d'effectuer une rotation d'un angle θ (en degré) d'un point à l'aide de la matrice de rotation :

$$M = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Si le point a pour coordonnées (x, y) alors le point (x^*, y^*) sera déplacé selon une rotation θ

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = M \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Reprenez le `fichier_resultat.txt` et faites tourner les données d'un angle de 10° . Refaites ce processus en traçant les données pour chaque angle.

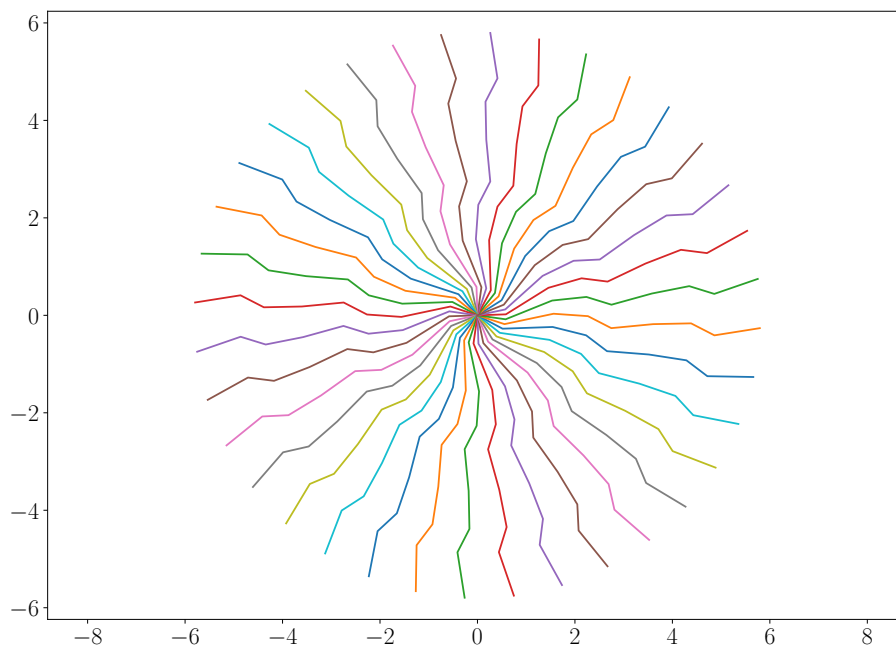


FIGURE 2 – Rotation de données.

Pour cet exemple faites un fichier Python unique qui aura la structure suivante :

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def mafonction(param1,param2) :
5     ...
6     ...
7     return sortie1
8
9 def mafonction2(param) :
10    ...
11    ...
12    return sortie1 ,sortie2
13
```

```
14 if __name__ == '__main__' : # programme principal
15     chaine1='Tête'
16     chaine2='Toto'
17     Totohead=mafonction(chaine1 , chaine2)
18     Totobonus , Totosuperbonus=mafonction2(Totohead)
19     ...
```