

2 Étude informatique de données expérimentales

Consignes

- Les programmes sont à rédiger en langage Python.
- Les extraits de codes matérialisés par _____ correspondent à des portions à compléter.
- L'annexe 2 page 8 comporte des rappels sur les commandes utiles.

Dans cette partie, on suppose que le fichier Python débute par l'importation du module `matplotlib.pyplot` et par la définition des deux listes `Ls` et `Lv`, de la façon suivante :

```
import matplotlib.pyplot as plt
Ls = [0.333, 0.167, 0.0833, 0.0416, 0.0208, 0.0104, 0.0052]
Lv = [3.636, 3.636, 3.236, 2.666, 2.114, 1.466, 0.866]
```

- (a) Écrire une fonction `inv` qui prend en entrée une liste de nombres (supposés non nuls) `L` et qui renvoie la liste composée des inverses de ces nombres. Exemple : `inv([0.25, 2, 1, 8])` renvoie `[4.0, 0.5, 1.0, 0.125]`.
- (b) Écrire une version améliorée `inv_ex` de la fonction `inv` qui prend en entrée une liste de nombres `L`, puis : si un de ces nombres est nul, alors elle renvoie le booléen `False`, sinon elle renvoie la liste composée des inverses de ces nombres.
- (c) Compléter les lignes de codes suivantes afin qu'elles effectuent le tracé demandé en question 11 de la partie 1 (les points seront représentés par des petits cercles et ne seront pas reliés entre-eux) :

```
plt.plot(_____)
plt.show()
```

- Écriture de fonctions préliminaires. *Pour cette question, on s'interdit d'utiliser les commandes préprogrammées de Python qui renvoient la somme, la moyenne ou la variance. Avant chaque fonction, on écrira brièvement le raisonnement suivi et la formule qu'elle est censée calculer.*
 - Écrire une fonction `moyenne` qui prend en entrée une liste `X` (non vide) de nombres réels et qui renvoie la moyenne des éléments de la liste.
 - Écrire une fonction `variance` qui prend en entrée une liste `X` (non vide) de nombres réels et qui renvoie la variance des éléments de la liste.
- (a) Compléter le programme suivant afin qu'il renvoie la valeur de la covariance de `X` et `Y` si elle existe et le booléen `False` sinon. (*Même consigne qu'à la question 2.*)

```
1 def cov(X, Y):
2     """Entrée: X,Y (liste)."""
3     nx = len(X); ny = len(Y)
4     if _____ or nx == 0:
5         return(False)
6     else:
7         _____
8         for k in range(_____):
9             S = _____
10            y = 1/nx*S
11            return(y)
```

- Parmi les quatre valeurs suivantes, lesquelles ne peuvent pas être renvoyées par la fonction `cov`? (*On justifiera les réponses.*)
 - `True`
 - `[1,2]`
 - `"False"`
 - `-0.5`
- On considère les fonctions `Coef` et `Trace` suivantes :

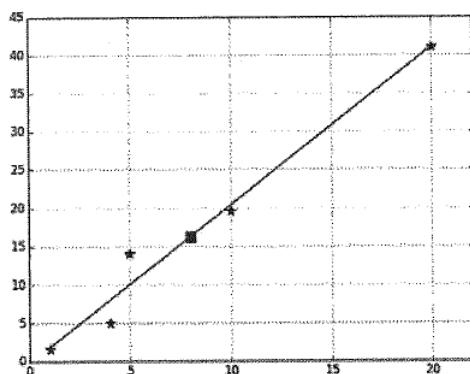
```
def Coef(X, Y):
    a = cov(X, Y)/variance(X)
    b = moyenne(Y)-cov(X, Y)/variance(X)*moyenne(X)
    return([a, b])
```

```

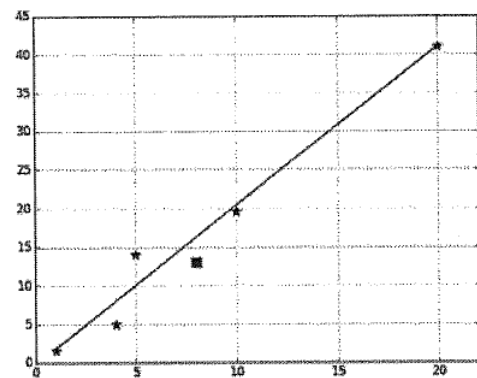
1 def Trace(X, Y):
2     [a, b] = Coef(X,Y)
3     xmin = min(X); xmax = max(X)
4     plt.plot(X, Y, "*")
5     plt.plot(_____)
6     plt.plot([moyenne(X)], [moyenne(Y)], "s")
7     plt.grid()
8     plt.show()

```

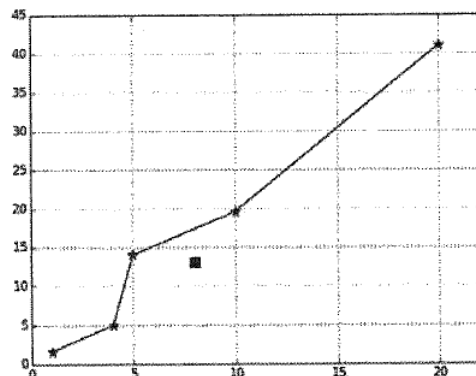
- i. Compléter la ligne 5 de la fonction Trace afin de tracer le segment d'extrémités $(x_{\min}, a \cdot x_{\min} + b)$ et $(x_{\max}, a \cdot x_{\max} + b)$.
- ii. Donner l'équation de la droite qui passe par ces deux points.
- iii. Quel nom porte cette droite ?
- iv. On exécute la fonction Trace pour des listes X et Y quelconques de taille 5. Pour chacun des trois tracés suivants, indiquer avec justification s'il peut être ou non le résultat de Trace ?



(a) Tracé n° 1



(b) Tracé n° 2



(c) Tracé n° 3

4. (a) En utilisant les fonctions et variables qui précèdent, proposer un code qui calcule les valeurs de K_M et v_{\max} en suivant la démarche de la Partie 1.3.
- (b) Pour ces données, le coefficient de corrélation linéaire vaut 0.9995 : qu'en déduire ?

3 Analyse de l'équation de Michaelis-Menten par Schnell et Mendoza

Les Parties 1 et 2 ont permis de déterminer expérimentalement les constantes K_M et v_{\max} qui interviennent dans l'équation de Michaelis-Menten (MM). Dans cette partie, on s'intéresse à la dépendance de s par rapport au temps, sous l'hypothèse de l'AEQS (cf. Partie 1).

Annexe 2 : Rappels Python pour la partie 2

On suppose que le module `matplotlib.pyplot`, qui permet de tracer des graphiques, est importé via `import matplotlib.pyplot as plt`.

Les variables `X` et `Y` sont ici deux listes de réels, de même longueur.

Python	Interprétation
<code>plt.plot(X,Y)</code>	Place les points dont les abscisses sont contenues dans <code>X</code> et les ordonnées dans <code>Y</code> et les relie entre eux par des segments. Si cette fonction n'est pas suivie de <code>plt.show()</code> , le graphique n'est pas affiché.
<code>plt.grid()</code>	Dessine en arrière plan du graphique un quadrillage.
<code>plt.show()</code>	Affiche le(s) tracé(s) précédemment créé(s) par <code>plt.plot</code>
<code>plt.plot(X,Y,"o")</code>	Même effet que <code>plt.plot(X,Y)</code> à la différence près que les points sont représentés par un symbole en forme de cercle et ne sont pas reliés. En remplaçant <code>o</code> par <code>s</code> (respectivement par <code>*</code>), le symbole est un carré (respectivement une étoile)
<code>min(X)</code> (resp. <code>max(X)</code>)	Renvoie le minimum (resp. le maximum) de <code>X</code>