



Unité 4 : Utiliser la librairie ti_plotlib

Compétence 3 : Nuage de points et régression

Dans cette troisième leçon de l'unité 4, vous allez découvrir comment représenter graphiquement une série de données, puis rechercher un modèle mathématique s'ajustant au mieux au nuage de points.

Objectifs :

- Réaliser la représentation d'un nuage de points.
- Rechercher et utiliser un modèle de régression.
- Utiliser les listes en Python.

Position du problème : Lors d'une inondation, afin de fournir à la population des informations pratiques, les autorités enregistrent, chaque heure à partir du début de la décrue, la hauteur d'eau maximale par rapport à un point de référence.

Les données sont fournies dans le tableau ci-dessous. On souhaite représenter le nuage de points en utilisant la bibliothèque **TI PlotLib**, puis rechercher un modèle mathématique permettant de faire une extrapolation, afin de prévoir le temps total de la décrue.

T(h)	0	1	2	3	4	5	6	7	8	9	10	11	12
H(cm)	130	127	123	118	116	111	105	103	101	95	86	80	71



Mise en œuvre : Commencer un nouveau script et le nomme U4SB3.

- 1) Importer le module **TI PlotLib**.
- 2) Enregistrer les données dans deux listes « **lt** » et « **lh** ».

```

1.1 *Classeur RAD 5/5
*U4SB3.py
import tiplotlib as plt
lt=[0,1,2,3,4,5,6,7,8,9,10,11,12]
lh=[130,127,123,118,116,111,105,103,101,95,86,80,71]
# Représentation graphique

```

- 3) Préparer ensuite la représentation graphique :
 - o Effacer l'écran **plt.cls()**.
 - o Régler les paramètres de la fenêtre graphique : $x_{min} = -2$; $x_{max} = 20$; $y_{min} = -20$; $y_{max} = 200$ en écrivant **plt.window(-2, 20, -20, 200)**.
 - o Afficher les axes **plt.axes()**.
 - o Afficher la représentation graphique sous forme d'un nuage de points **plt.scatter()**.

```

1.1 *Classeur RAD 9/9
*U4SB3.py
import tiplotlib as plt
lt=[0,1,2,3,4,5,6,7,8,9,10,11,12]
lh=[130,127,123,118,116,111,105,103,101,95,86,80,71]
# Représentation graphique
plt.cls()
plt.window(-2,20,-20,200)
plt.axes("on")
plt.scatter(lt,lh,"x")
plt.show_plot()

```

Les instructions de paramétrage de la fenêtre graphique se trouvent dans le module **TI PlotLib** dans le menu **Setup**. La syntaxe de la représentation du nuage de points se situe quant à elle dans le menu **Draw**.

Conseil à l'enseignant : la liste des données du temps, en heures, peut être complétée à l'aide d'une boucle fermée.

```

1.1 1.2 *Classeur RAD 4/4
Shell Python
>>>lt=[i for i in range(13)]
>>>lt

```

Exécuter le script et observer la représentation graphique. Si tout se passe correctement, vous devriez obtenir l'écran ci-contre.

L'ensemble des points semble-t-il aligné ?

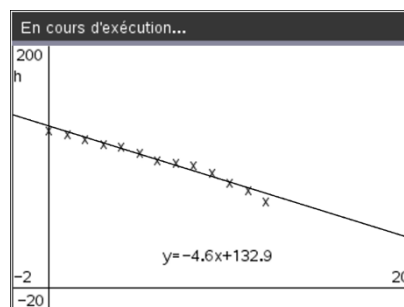
Vous allez maintenant rechercher le modèle mathématique (affine) passant au mieux par l'ensemble des points du nuage.

Pour cela, vous devez rajouter la ligne `plt.lin_reg(xliste,yliste, « center »)` dans votre script afin que la droite de régression soit calculée à partir des listes de données `lt` et `lh`, puis affichée et centrée sur l'écran.

```

1.1 1.2 *Classeur RAD 11/12
U4SB3.py
lt=[0,1,2,3,4,5,6,7,8,9,10,11,12]
lh=[130,127,123,118,116,111,105,103,101,95,86,
# Représentation graphique
plt.cls()
plt.window(-2,20,-20,200)
plt.color(0,0,0)
plt.labels("t","h",12,2)
plt.axes("on")
plt.scatter(lt,lh,"x")
plt.lin_reg(lt,lh,"center")
plt.show_plot()
    
```

Afin de connaître l'heure de la décrue totale, il vous reste à résoudre l'équation du premier degré $-4.64x + 132.90 = 0$.



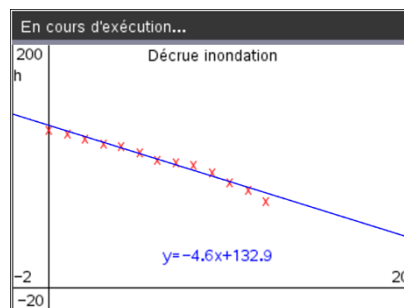
Améliorons notre représentation graphique :

Inclure dans votre script des instructions permettant de conserver les axes en noir `plt.color(0,0,0)`, puis une représentation graphique du nuage de point en rouge `plt.color(255,0,0)`, et enfin une représentation en bleu de la droite de régression `plt.color(0,0,255)`.

```

1.1 1.2 *Classeur RAD 14/14
U4SB3.py
# Représentation graphique
plt.cls()
plt.window(-2,20,-20,200)
plt.color(0,0,0)
plt.labels("t","h",12,2)
plt.axes("on")
plt.color(255,0,0)
plt.scatter(lt,lh,"x")
plt.color(0,0,255)
plt.lin_reg(lt,lh,"center")
plt.show_plot()
    
```

Il est aussi intéressant de rajouter un titre et des étiquettes aux axes.





Conseils à l'enseignant : Afin d'éviter des problèmes de superposition pour l'affichage, il convient de choisir des noms courts pour les étiquettes des axes.

L'étiquette des axes « t » et « h » peut être configurée selon vos souhaits en utilisant l'instruction `labels(« x-étiq », « y-étiq » », x, y)`, **x** et **y** représentant la ligne où sont écrits ces labels. **Par défaut ces lignes sont respectivement 12 et 2 pour x et y et respectivement justifiées à gauche et à droite.**

De même, l'équation de régression peut être affichée à l'emplacement souhaité, en utilisant la commande `lin_reg(xlist, ylist, « disp », row)`. Par défaut, cette équation est affichée à la ligne 11.

