TP INFORMATIQUE

Utilisation de modules sur l'exemple du traitement de données

Dans ce TP, on présente quelques méthodes permettant de lire des informations contenues dans un fichier externe, de les traiter pour in fine enregistrer un résultat (sous forme d'un fichier texte ou d'une image).

Lecture des données

1. Créer un fichier python pour le TP dans votre répertoire de travail. **Copiez** le fichier `IntensiteTension.csv` disponible dans le dossier d'échange de la classe dans le même répertoire que celui contenant votre script.
2. Ouvrir le fichier `IntensiteTension.csv` avec un éditeur de texte (par exemple Notepad) et observez sa structure. Fermez ensuite le fichier. Il contient les données obtenues pour mesurer la valeur d'un résistor de résistance R par la méthode de longue dérivation.
3. Pour lire les données contenues dans le fichier on utilise les commandes suivantes :

```
# ouverture du fichier en lecture
file = open("IntensiteTension.csv", "r" )

data = file.readlines() # lecture des donnees

file.close() # fermeture du fichier
```

Observez le contenu de `data[0]` et de `data[1]`.

4. Observez le résultat de la commande suivante :

```
print(data[1].split(";"))
```

5. Compléter alors votre script python pour créer deux listes `lesU`, `lesI` contenant les valeurs des tensions, des intensités. Chaque donnée sera convertie en nombre flottant avec la commande `float`.

Tracé des données

On veut afficher graphiquement les tensions mesurées en fonction des intensités. Pour cela on utilise la librairie de tracé `matplotlib.pyplot`.

1. Compléter votre script en ajoutant les commandes suivantes. *On prendra soin de mettre les différentes commandes au bon endroit dans le script...*

```
import matplotlib.pyplot as plt # librairie de trace

plt.plot(lesI[0:5], lesU[0:5], marker = "o" , label="donnees")
plt.legend()
plt.savefig("MonImage.png")
```

2. Observer le contenu du fichier `MonImage.png` qui a du se créer dans votre répertoire courant. À quoi correspondent les différents traits?

| On peut afficher l'image sans l'enregistrer avec la commande `plt.show()`

3. Modifier la ligne `plt.plot` comme suit et observez le résultat :

```
plt.plot(lesI, lesU, linewidth=0, marker = "o" , label="donnees")
```

4. Compléter votre image à l'aide des commandes `plt.xlabel()`, `plt.ylabel()`, `plt.title()` dont vous aurez pris soin de lire l'aide.

On constate visuellement que les données suivent une droite de tendance. Il existe plusieurs fonctions permettant de réaliser une regression linéaire. Par exemple :

- la librairie `scipy.stats` contient la fonction `linregress` ;
- la fonction `polyfit` de `numpy` permet de réaliser des modélisations polynomiales (donc a fortiori affines).

1. En utilisant au choix une des deux fonctions précédentes, déterminer la pente `a` et l'ordonnée à l'origine `b` de la régression linéaire modèle.
2. Ajouter à l'aide de la commande `plt.plot` la droite de régression linéaire à l'image précédente. Pour ce faire la liste des abscisses ne devra contenir que deux points (et donc celle des ordonnées également).

Écriture de données

Dans cette section, on souhaite créer le fichier `Resistance.txt` contenant les différentes valeurs de résistance mesurées. Chaque valeur i est obtenue par l'opération `lesU[i]/lesI[i]`.

1. Exécutez le code suivant (écrit à la suite de votre script) :

```
file2 = open("Resistance.txt" , "w" ) # Création du fichier en écriture

for i in range(len(lesU)):
    file2.write(str(lesU[i]/lesI[i]) + "\n")
file2.close()
```

2. Editez le fichier ainsi obtenu avec un éditeur de texte pour comprendre le détail de l'exécution.

Attention lors de la création du fichier : si un fichier du répertoire courant a le même nom il est écrasé !
On peut ajouter des données à la fin d'un fichier existant en utilisant l'argument `"a"` au lieu de `"w"`.

Un peu d'incertitude

On cherche à évaluer l'incertitude type $u(R)$ de l'ensemble des mesures de R . On note $\bar{R}$ la valeur moyenne des N mesures de R :

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N R[i]$$

On rappelle que $u(R) = \frac{s_R}{\sqrt{N}}$ avec $s_R = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (R[i] - \bar{R})^2}$

1. Créer un script python vierge `Incertainite.py`. Complétez le pour stocker les données contenues dans le fichier `Resistance.txt` sous forme d'une liste de flottants `lesR`.
2. Écrire une fonction `moyenne(L)` d'argument une liste de flottants qui retourne la valeur moyenne de ses éléments.
3. Écrire une fonction `Incertainite(L)` d'argument une liste de flottants qui retourne l'incertitude type associée.
4. Donnez alors le résultat expérimental de la mesure.