Corrigé Code de partage avec Capytale : 645d-9007032

On utilisera les bibliothèques suivantes :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Exercice 1 - PIB et population urbaine

On va étudier ici l'évolution de deux variables de 1960 à 2021 en Norvège : le pourcentage de population urbaine et le PIB par habitant (données issues du World Bank Group), et leur possible corrélation.
Dans un premier temps, il faut importer le jeu de données. On commence donc par exécuter la commande suivante :

```
import pandas as pd
donnees=pd.read_csv('tp11_norvege.csv', delimiter=';')
```

1. Représenter le nuage de points des deux variables étudiées (pourcentage de population urbaine en fonction du PIB par habitant) et commenter le résultat.

   Il suffit d'extraire les données des deux colonnes qui nous intéresse et de représenter :

   ```
   x=donnees['PIB_capita']
   y=donnees['Pop_urbaine']
   plt.plot(x,y,'.')
   plt.show()
   ```

   Les points ne semblent pas s'aligner, mais font plutot penser à l'allure d'une courbe logarithmique.

2. Que fait le programme suivant ? interpréter le résultat.

   ```
   np.mean([(x[i]-x.mean())*(y[i]-y.mean()) for i in x.index])/(x.std()*y.std()
       )
   ```

   Il calcule le coefficient de corrélation linéaire $r_{xy} = \dfrac{\mathrm{Cov}(x,y)}{s_x s_y}$

   Le résultat est proche de $0,81$, ce que l'on va considérer comme insuffisant pour modéliser la relation entre x et y par une fonction affine.

3. Représenter le nuage de points $(\ln(x), y)$

   La commande suivante suffit, car `np.log(x)` transforme la « colonne » `x` en la « colonne » de ses logarithme.

   ```
   plt.close()
   plt.plot(np.log(x),y,'.')
   plt.show()
   ```

   On peut aussi définir une nouvelle variable pour y voir plus clair.

   ```
   x2=[np.log(x[i]) for i in x.index]
   plt.plot(x2,y,'.')
   plt.show()
   ```

   Les données présentent alors une allure plus rectiligne.

4. Calculer le coefficient de corrélation linéaire de y et $\ln(x)$

   On reprend le programme de la question 2. en modifiant `x` en `x2` (ou `np.log(x)`)

```
np.mean([(x2[i]-x2.mean())*(y[i]-y.mean()) for i in x2.index])/(x2.std()*y.
    std())
```

On trouve alors une valeur approchée de $0,95$ ce que l'on va considérer comme suffisant pour modéliser une relation entre y et $\ln(x)$ par une fonction affine.

5. Déterminer l'équation de la droite de régression de y en fonction de $\ln(x)$

   D'après les résultats du cours, on cherche les coefficients a et b de la droite $t = au + b$, où $a = \dfrac{\text{Cov}(u,t)}{s_u^2}$ et $b = \overline{t} - a * \overline{u}$ d'où les commandes :

```
a=np.mean([(x2[i]-x2.mean())*(y[i]-y.mean()) for i in x2.index])/(x2.std()
    **2)
b=y.mean()-a*x2.mean()
```

   on obtient $a \simeq 5,87$ et $b \simeq 13,69$

6. En déduire qu'on peut supposer que la dépendance entre y et x est de la forme $y = a\ln(x) + b$

   C'est l'équation de la droite de régression linéaire que l'on juge pertinente pour représenter une relation entre y et $\ln(x)$ d'après la valeur du coefficient de corrélation linéaire.
   Pour mieux visualiser le résultat, on peut représenter la droite de régression avec le nuage avec la commande suivante :

```
plt.close()
plt.plot(x2,y,'.')
plt.plot(x2,a*x2+b)
plt.show()
```

7. Représenter le nuage de points initial avec lequel on fera apparaître la courbe d'équation $y = a\ln(t) + b$

   Enfin avec les valeurs de a et b trouvées, on représente à nouveau le nuage de point initial (x, y) en superposant la courbe $y = a\ln(x) + b$

```
plt.close()
plt.plot(x,y,'.')
plt.plot(x,a*x2+b)
plt.show()
```

   Le résultat semble plutôt intéressant avec une « bonne superposition » (visuellement) entre le nuage de points et la courbe.