Régression linéaire

I Explication

La régression linéaire consiste à établir une relation linéaire entre une variable dépendante y et une ou plusieurs variables indépendantes $x_1, ..., x_n$.

Pour cela, on utilise Python et les bibliothèques `numpy` et `matplotlib`.

II Comment faire?

II.1 Importer les bibliothèques

Pour importer les bibliothèques, on utilise la commande `import`.

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

II.2 Créer les données

On considère les listes X et Y suivantes (ces données sont fictives et sont normalement issues d'une expérience réelle) :

```
X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Y = [1, 2.4, 3.6, 4.8, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5]
```

II.3 Tracer le nuage de points

En physique on ne relie jamais des points expérimentaux par des segments, mais on trace un nuage de points. Pour cela, on utilise la commande `plt.plot` avec `o` comme forme.

```
plt.plot(X, Y, "o")
plt.label("X (unité)")
plt.ylabel("Y (unité)")

plt.show()
```

II.4 Réaliser la régression linéaire

Pour réaliser la régression linéaire, on utilise la commande `np.polyfit` qui prend en argument les listes X et Y ainsi que le degré du polynôme (ici 1 car on veut une droite).

```
a, b = np.polyfit(X, Y, 1)
```

II.5 Tracer la droite de régression

Pour tracer la droite de régression, on utilise la commande `plt.plot` avec `--` comme forme.

Pour avoir des valeurs régulières en abscisse, on utilise la commande `np.linspace` qui prend en argument la valeur minimale, la valeur maximale et le nombre de valeurs voulues dans l'intervalle.

```
# Si on veut laisser les points expérimentaux on utilise la commande suivante
plt.plot(X, Y, "o")

# Tracé de la droite de régression
list_x = np.linspace(min(X), max(X), 100) # 100 valeurs entre min(X) et max(X)
plt.plot(list_x, a * np.array(X) + b, "--")
plt.xlabel("X (unité)")
plt.ylabel("Y (unité)")

plt.show()
```

Il est ensuite possible de récupérer les coefficients de la droite de régression avec `a` et `b` et de les afficher.

```
print("a = ", a)
print("b = ", b)
```

Il est bien sûr aussi possible de les récupérer de manière géométrique avec une règle.