

TP8 - STATISTIQUES BIVARIÉES

Dans tout le TP, on importe les modules suivants :

```
1 import numpy as np
import numpy.random as rd
import numpy.linalg as al
import matplotlib.pyplot as plt
```

1 Généralités

Dans ce TP, nous reprenons les statistiques. Comme le nom du TP le suggère, nous allons nous concentrer sur les statistiques bivariées, c'est-à-dire à l'étude simultanée de deux caractères d'une population et de leur relation. Avant de nous lancer dans l'étude informatique et mathématique à proprement parler, reprenons les commandes Python qui permettent le tracer de graphes et qui vont nous être utiles pour visualiser.

Exercice 1

★

Nous allons représenter les données suivantes : on étudie le lien entre la consommation de viande (en grammes par jour et par personne) et l'incidence du cancer du colon (pour 100 000 femmes par an) dans certains pays industrialisés. On a le tableau suivant :

Pays	Consommation de viande	Taux de cancer
Japon	26	7.5
Israël	124	16.4
États-Unis	284	34
Finlande	101	9.8
Grande-Bretagne	205	23.3

1. Créer deux tableaux `numpy` nommés `X` et `Y` contenant respectivement la consommation de viande et le taux de cancer de chaque pays.
2. Utiliser les commandes :

```
1 plt.plot(X, Y)
plt.show()
```

Commenter. Cette représentation présente-t-elle un défaut ?

3. On améliore le tracé avec :

```
1 plt.plot(X, Y, '+')
plt.show()
```

Le '+' est une chaîne de formatage propre à la bibliothèque `matplotlib`. Inutile de retenir tout à ce sujet mais garder peut-être en tête ce format pour faire un nuage de points.

Commenter désormais l'allure du nuage de points ? Les points sont-ils sur une droite ? Si non, pourrait-on approcher ce nuage par une droite ?

4. Tout nuage de points peut-il être bien représenté par une droite ? À titre d'exemple regarder ce qu'il se passe si on prend :

```
1 X = rd.random(1000)
Y = rd.random(1000)
```

2 Méthode des moindres carrés

Étant donnée un ensemble de points dans le plan, on aimerait trouver la *meilleure* droite approchant ces points. Il y a bien des manières de définir la *meilleure* droite mais aujourd'hui nous allons nous concentrer sur la méthode des moindres carrés.

Ainsi, si on considère une collection de points $((x_1, y_1), \dots, (x_n, y_n))$ on va considérer que la meilleure droite approchant ces points est la droite d'équation $y = ax + b$ minimisant $\sum_{k=1}^n (y_k - (ax_k + b))^2$ pour peu qu'elle existe et qu'elle soit unique. Dans ce cas, on l'appellera **droite des moindres carrés**.

Exercice 2

★★

On considère une collection de points $((x_1, y_1), \dots, (x_n, y_n))$.

1. **Version Appro** : On pose :

$$A = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \in M_{n,2}(\mathbb{R}) \quad \text{et} \quad B = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in M_{n,1}(\mathbb{R}).$$

Définir les tableaux `numpy` correspondants avec les données de l'exercice 1.

On **admet** que les coefficients de la droite des moindres carrés sont données par :

$$\begin{pmatrix} a \\ b \end{pmatrix} = ({}^tAA)^{-1} {}^tAB.$$

Vérifier que les tailles des matrices sont cohérentes puis en utilisant cette formule, calculer a et b pour le tableau de données initiales et les stocker dans des variables éponymes.

Version Appli : Calculer a et b avec la formule de cours.

2. Utiliser le code suivant afin de représenter de manière superposée le nuage de points et la droite des moindres carrés.

```
1 Z = a*X + b
plt.plot(X, Y, '+')
plt.plot(X, Z)
plt.show()
```

Commenter.

3. On pose :

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k \quad \text{et} \quad \bar{y} = \frac{1}{n} \sum_{k=1}^n y_k.$$

Le point (\bar{x}, \bar{y}) est appelé **point moyen** du nuage de points. Le placer sur le graphe en utilisant le format `'*'`. Que constate-t-on ?

3 Covariance, coefficients de corrélations

Exercice 3

★★

1. Rappeler la définition de covariance et de coefficients de corrélation linéaire pour une variable aléatoire.

Que peut-on dire si $\rho(X, Y) = \pm 1$?

2. On adapte la définition de la manière suivante :

$$V(X) = \sigma_X^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \bar{x})^2 \quad \text{et} \quad \text{Cov}(X, Y) = \frac{1}{n} \sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y}).$$

On pose alors $\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$.

Calculer le coefficient de corrélation linéaires avec les données de l'exercice 1. Calculer ensuite le coefficient de corrélation linéaire pour les vecteurs aléatoires utilisés en fin de l'exercice 1. Commenter.