Python td 8 - Statistiques univariées et bivariées

On commencera notre programme par

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

I. Séries statistiques

Dans une population donnée, on souhaite étudier deux caractères X et Y . On peut alors s'intéresser aux propriétés de chacun des 2 caractères pris séparément (statistiques univariées), mais aussi au lien entre ces 2 caractères (statistiques bivariées) ; on étudie alors le couple de caractères $Z = (X; Y)$.
Ceci est intéressant notamment si on pense que l'une des variables, X par exemple, est une cause de l'autre, par exemple Y. On dit alors que X est la variable explicative et Y est la variable à expliquer. Dans ce cas, on tentera d'exprimer Y en fonction de X en commençant par tracer le nuage des points de Y en fonction de X pour deviner la relation entre ces données.

Dans ce TP, on s'intéresse au problème suivant : un chercheur en sociologie veut analyser s'il existe une relation linéaire entre la densité de population dans les départements français et le taux de criminalité correspondant dans ces départements. Que choisiriez-vous comme variable explicative X et comme variable à expliquer Y ?

On récupère alors les données un échantillon de départements.
Pour le département i, on note $M_i = (x_i; y_i)$ le couple de résultats obtenus.

Définition I.1
Soit un échantillon $\{\omega_1, \cdots, \omega_n\}$ de Ω. On appelle **série statistique** la donnée de la liste

$$x = [x_1, \cdots, x_n]$$

où chaque x_i est associé à une réalisation ω_i : $x_i = X(\omega_i)$.

Définition I.2
Soit un échantillon $\{\omega_1, \cdots, \omega_n\}$ de Ω et deux séries statistiques $x = [x_1, \cdots, x_n]$ et $y = [y_1, \cdots, y_n]$. On appelle série statistique double la donnée de la liste

$$[(x_1, y_1), \cdots, (x_n, y_n)]$$

où chaque couple (x_i, y_i) est associée à une réalisation ω_i : $(x_i, y_i) = (X, Y)(\omega_i)$.

Reprenons l'exemple sur la criminalité. On note X la densité de population mesurée en centaines d'habitants par km^2 et Y le taux de criminalité en nombre de crimes par 10 000 habitants.
On relève dans 13 départements les données suivantes :

Département numéro	1	2	3	4	5	6	7	8	9	10	11	12	13
x_i	11	7.2	5.9	4.8	3.8	2.7	2.0	1.7	1.5	1.4	1.1	0.7	0.2
y_i	6.4	6.3	5	4.3	7.6	4.3	3.8	4.6	3.6	3.6	3.9	3.5	2.6

II. Statistiques univariées

On s'intéresse à une série statistique $x = [x_1, \cdots, x_n]$.

La moyenne (empirique) de cette série statistique est donnée par

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

La variance empirique de cette série statistique est donnée par :

$$s^2(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

Il s'agit de la moyenne des carrés des écarts à la moyenne, comme en probabilités.
L'écart-type empirique $s(x)$ est égal à la racine carrée de la variance empirique.

Si x est la matrice (ligne) contenant la série statistique, ces données se calculent avec les instructions Python `np.mean(x)`, `np.var(x)`, `np.std(x)`

Exercice 1
Saisir la série `x` correspondant aux densités des départements (sous forme d'une matrice ligne).
Calculer la densité moyenne de ces 13 départements et donner la variance empirique des densités de population de ces départements.

III. Nuage de points, point moyen

On appelle **nuage de points** associé à la série des n couples $(x_i; y_i)$ l'ensemble des points M_i de coordonnées $(x_i; y_i)$ tracés dans un repère orthonormé du plan.
L'examen du nuage de points permet de faire des constatations qualitatives :
- est-il concentré ou dispersé ?
- relève-t-on une tendance ?
(variations dans le même sens (covariance positive) ? suit une courbe particulière ? ...)
- y a-t-il des valeurs aberrantes ?

Si l'on dispose des deux séries statistiques x et y (matrices lignes de même taille), la commande Python `plt.scatter(x,y)` permet de représenter le nuage de points correspondant.

Si on note $\overline{x}$ la moyenne de la série statistique x et $\overline{y}$ la moyenne de la série statistique y, alors le point de coordonnées $(\overline{x}, \overline{y})$ est le **point moyen du nuage**.

Exercice 2
Saisir les deux séries statistiques x et y correspondant à la densité de population et au taux de criminalité.
Représenter le nuage de points correspondant.
Calculer les coordonnées du point moyen et représenter ce point (on peut utiliser encore `plt.scatter` et le représenter en rouge).
Améliorez la présentation de votre graphique avec les commandes `plt.title`, `plt.xlabel`, `plt.ylabel`
Est-ce qu'une tendance générale semble se déduire de ce nuage de points ?

On rappelle que X est la variable explicative et Y est la variable à expliquer.
Nous nous intéresserons dans la suite au modèle de régression linéaire : on cherche une fonction affine, du type $f(x) = ax + b$, telle que $Y = f(X) + \epsilon = aX + b + \epsilon$, où ϵ est l'erreur d'ajustement.

IV. Covariance

Définition IV.1
On appelle **covariance empirique** de la série statistique double $(x, y) = (x_i, y_i)_{i\in[[1,n]]}$ le réel :

$$cov(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

Il s'agit de l'analogue en statistiques de la covariance que nous verrons en cours de probabilités.
L'instruction Python pour calculer la covariance est `np.cov(x,y)`. Attention cependant, cette instruction renvoie la matrice (dite de variance-covariance) suivante :

$$\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(x,y) & cov(y,y) \end{pmatrix}$$

et il faudra taper `np.cov(x,y)[0,1]` pour accéder à la covariance de x et y.

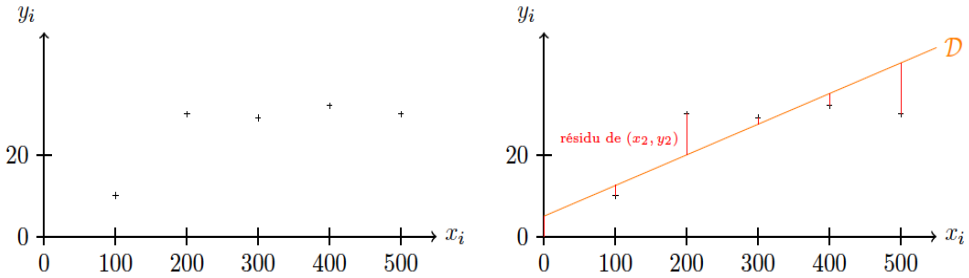
Exercice 3
Calculer la covariance de (x, y)

V. Méthode des moindres carrés

Si le nuage de points associé à une série statistique double possède une forme étirée, on peut avoir l'idée de chercher quelle droite approcherait au mieux les points de ce nuage. Le problème consiste donc à identifier une droite d'équation $y = ax + b$ qui ajuste bien le nuage de points. L'erreur que l'on commet en utilisant la droite de régression pour prédire y_i à partir de x_i est $y_i - (ax_i + b)$.

Pour déterminer la valeur des coefficients a et b, on utilise le principe des moindres carrés qui consiste à chercher la droite qui minimise la somme des carrés de ces erreurs :

$$\sum_{i=1}^{n}(y_i - (ax_i + b))^2$$



Théorème V.1
L'unique droite rendant minimale la quantité $\sum_{i=1}^{n}(y_i - (ax_i + b))^2$ est la droite d'équation

$$y = \frac{cov(x, y)}{s^2(x)}(x - \overline{x}) + \overline{y}$$

Cette droite passe toujours par le point moyen, on a donc $\overline{y} = a\overline{x} + b$.

Exercice 4
Calculer en Python les valeurs de a et b. Tracer la droite de régression correspondante sur le nuage de points.
Estimer le taux de criminalité le plus plausible pour un département ayant une densité de population de $1000h/km^2$.

VI. Coefficient de corrélation

Définition VI.1
On appelle coefficient de corrélation empirique de la série statistique double (x, y) le réel

$$\rho(x, y) = \frac{cov(x, y)}{s(x).s(y)}$$

$\rho(x, y) \in [-1, 1]$.
Plus $|\rho(x, y)|$ est proche de 1, plus les points sont proches de l'alignement et plus les prévisions données par la droite de régression sont pertinentes.
$|\rho(x, y)|$ ne vaut 1 que si les points du nuage sont alignés.
Si $\rho(x, y) > 0$ alors la droite de régression est de pente positive : X et Y évoluent dans le même sens.
Si $\rho(x, y) < 0$ alors la droite de régression est de pente négative : X et Y évoluent dans le sens opposé.

Exercice 5
Calculer le coefficient de corrélation des séries statistiques x et y étudiées précédemment.

VII. Exercice

On revient à un cadre probabiliste.
Soit a un réel strictement positif et $(X_1, \cdots, X_n)$ une famille de variables aléatoires indépendantes, suivant toutes la loi uniforme sur $[0, a]$. On pose $U = min(X_1, \cdots, X_n)$ et $V = Max(X_1, \cdots, X_n)$. On s'intéresse au coefficient de corrélation linéaire de U et V, noté $\rho_{a,n} = \rho(U, V)$.

On considère la fonction Python suivante :

```
import numpy as np
import numpy.random as rd
def rho(a,n):
    nbsim=100000
    A=a*rd.random([nbsim,n])
    U=A.max(axis=1)
    V=A.min(axis=1)
    cova=np.cov(U,V)[0,1]
    return cova/(np.std(U)*np.std(V))
```

1. Expliquer comment sont définis les matrices `U` et `V`. Dire quelles sont leurs tailles. Quel est le résultat renvoyé par cette fonction ? En quoi ceci peut-il être utile ?

2. On exécute plusieurs appels de cette fonction pour différentes valeurs de a et n. Les résultats sont donnés dans le tableau suivant :

	$a = 1$	$a = 5$	$a = 10$
$n = 2$	0.50073	0.50332	0.49990
$n = 3$	0.33465	0.33481	0.33391
$n = 4$	0.25237	0.250042	0.24936

   La corrélation $\rho_{a,n}$ semble-t-elle dépendre de a ?
   Quelle semble être, en fonction de n, la valeur de $\rho_{a,n}$?