Statistique descriptive bivariée

En statistique descriptive bivariée (on parle aussi de statistique double), on se propose d'étudier le comportement d'une population donnée à partir de deux types d'observations effectuées sur cette population. Plus précisément, l'objectif principal sera de définir et de calculer à partir de deux données observées de nouvelles données permettant de décrire rapidement les tendances générales de la population étudiée.

1 Notion de série statistique double

Définition 1.1. *Soient X et Y deux variables. Une série statistique double (d'un échantillon) est la liste $((x_i, y_i))_{1\leq i\leq N}$ des couples de modalités (ou valeurs) prises par X et Y (sur cet échantillon).*

En général, une série statistique double se présentera sous la forme d'un tableau de deux lignes et N colonnes, la première ligne rassemblant les valeurs prises par X et la deuxième ligne les valeurs correspondantes prises par Y. On peut aussi la représenter géométriquement, à l'aide de la :

Définition 1.2. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double. Le nuage de points de cette série statistique est l'ensemble des points $M_i(x_i, y_i)$, où $i \in \{1, ..., N\}$.*

Pour obtenir ce nuage de points en Python, on utilisera la commande `plt.plot` avec le style `'.'` ou `'x'`, ceci afin de représenter uniquement les points, sans les relier par une ligne brisée. Bien entendu, on aura importé la bibliothèque `matplotlib.pyplot` au préalable.

Exemple 1.3. *Considérons la série statistique double suivante :*

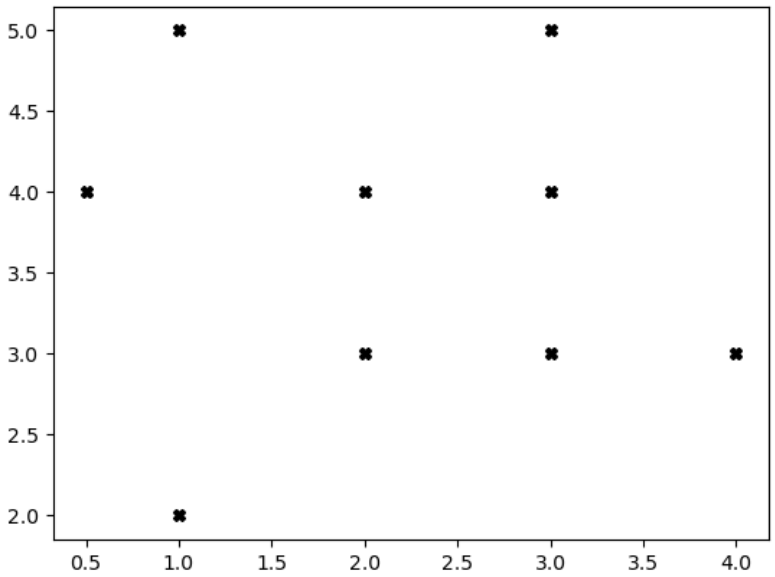
x_i	1	0.5	1	2	3	3	2	4	3
y_i	5	4	2	3	4	5	4	3	3

Pour tracer le nuage de points associé, on pourra utiliser la liste de commandes suivante :

```
import mathplotlib.pyplot as plt

x=np.array([1,0.5,1,2,3,3,2,4,3])
y=np.array([5,4,2,3,4,5,4,3,3])
plt.plot(x,y,'X',color='k')
plt.show()
```

Le résultat donné par Python (avec les styles `'X'` et `color='k'`) est le suivant :



2 Covariance et coefficient de corrélation linéaire

Définition 2.1. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y). La covariance de $((x_i, y_i))_{1\leq i\leq N}$ est définie par :*

$$\mathrm{cov}(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{X}\right)\left(y_i - \overline{Y}\right).$$

Pour calculer la covariance d'une série statistique double (donnée sous forme de deux listes x, y), on dispose de la formule de Koenig-Huygens pour les séries statistiques, c'est-à-dire :

$$\boxed{\mathrm{cov}(X,Y) = \frac{1}{N}\left(\sum_{i=1}^{N} x_i y_i\right) - \left(\overline{X}\times\overline{Y}\right).}$$

La covariance se calculera donc à l'aide de la commande `c=np.mean(X*Y)-np.mean(X)*np.mean(Y)`. En effet, la commande `X*Y` donne le vecteur dont les composantes sont les produits $x_1y_1, ..., x_ny_n$, où $x_1, ..., x_n$ et $y_1, ..., y_n$ sont les composantes respectives de X et Y. A noter qu'il existe une commande spécifique en Python pour calculer la covariance de deux séries statistiques simples X et Y, à savoir la commande `np.cov(X,Y)`, mais que celle-ci n'est pas au programme !

Définition 2.2. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y). Le coefficient de corrélation linéaire de $((x_i, y_i))_{1\leq i\leq N}$ est défini par :*

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}.$$

A noter que, comme pour les variables aléatoires discrètes, le coefficient de corrélation linéaire est toujours compris entre -1 et 1. De plus, ce coefficient est d'autant plus proche de ± 1 que les points du nuage sont proches d'une droite. C'est ce que nous verrons dans le paragraphe suivant. A noter aussi que, comme pour la covariance, il existe une commande spécifique en Python pour calculer le coefficient de corrélation linéaire de deux séries statistiques simples X et Y, mais que celle-ci n'est pas au programme ! Pour le calculer, on utilisera donc la commande :

```
r=(np.mean(x*y)-np.mean(x)*np.mean(y))/(np.std(x)*np.std(y))
```

Exemple 2.3. *Considérons la série statistique double suivante :*

x_i	1	0.5	1	2	3	3	2	4	3
y_i	5	4	2	3	4	5	4	3	3

Pour calculer la covariance et le coefficient de corrélation linéaire associé à cette série statistique double, on pourra utiliser la fonction suivante :

```
import numpy as np

def calculstats();
    x=np.array([1,0.5,1,2,3,3,2,4,3])
    y=np.array([5,4,2,3,4,5,4,3,3])
    cov=np.mean(x*y)-np.mean(x)*np.mean(y)
    r=cov/(np.std(x)*np.std(y))
    return cov,r
```

Avec la fonction Python ci-dessus, Python affichera le résultat suivant :

$$(-0.0555555555,\ \ -0.05330017908)$$

3 Régression linéaire

Etant donnée une série statistique double de la forme $((x_i, y_i))_{1\leq i\leq N}$, l'objectif de la régression linéaire est d'une part, de déterminer la droite qui s'ajuste le mieux par rapport à son nuage de points, et d'autre part de vérifier si l'ensemble des points du nuage est "proche" de cette droite. En effet, en statistique, on peut chercher à déterminer s'il existe une relation de dépendance (on parle de corrélation) entre deux types de données observées sur le terrain (par exemple, le nombre de ventes de glaces et celui des crèmes solaires en période estivale). Pour ce faire, on s'intéressera en priorité aux relations de dépendance de type linéaire, ce qui nous amène à la question suivante. Existe-t-il une relation de dépendance linéaire entre les

données $(x_i)_{1\leq i\leq N}$ et $(y_i)_{1\leq i\leq N}$? En d'autres termes, existe-t-il des réels a, b tels que $y_i \simeq ax_i + b$ pour tout $i \in \{1, ..., N\}$? D'un point de vue géométrique, cela revient à se demander si les points du nuage sont "proches" d'une certaine droite et si oui, laquelle ? Pour répondre à cette question, on utilise la méthode dite "des moindres carrés ordinaires" (MCO en abrégé) afin de déterminer la droite qui s'ajuste le mieux par rapport au nuage de points. Plus précisément, étant donnée une droite du plan d'équation cartésienne $y = ax + b$, on cherche à minimiser la somme des carrés des écarts verticaux entre les points du nuage et la droite en question, c'est-à-dire à minimiser l'expression $Q(a, b) = \sum_{i=1}^{N}(y_i - ax_i - b)^2$ en fonction de a et b. On dispose alors du résultat suivant :

Théorème - définition 3.1. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y), telle que les x_i ne sont pas tous égaux. Alors il existe une unique droite notée $\mathcal{D}_{Y/X}$ d'équation $y = ax + b$ qui minimise l'expression $Q(a, b) = \sum_{i=1}^{N}(y_i - ax_i - b)^2$. Cette droite est appelée la droite de régression de Y par rapport à X, et elle admet pour équation :*

$$y = \overline{Y} + \frac{\mathrm{cov}(X, Y)}{V(X)}(x - \overline{X}).$$

Cette droite peut se trouver à l'aide des résultats sur le problème des moindres carrés que l'on a vu en Algèbre bilinéaire. Plus précisément, considérons une série statistique double $((x_i, y_i))_{1\leq i\leq N}$ telle que les x_i ne sont pas tous égaux, et posons pour tout $(a, b) \in \mathbb{R}^2$:

$$U = \begin{pmatrix} a \\ b \end{pmatrix} \quad , \quad A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix} \quad , \quad V = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}.$$

Si l'on munit $\mathcal{M}_{N,1}(\mathbb{R})$ du produit scalaire canonique, alors on constate que :

$$Q(a, b) = \sum_{i=1}^{N}(y_i - ax_i - b)^2 = \|V - AU\|^2.$$

Comme les x_i ne sont pas tous égaux, les vecteurs colonnes de A ne sont pas colinéaires, et donc la matrice A est de rang 2. D'après le problème des moindres carrés, il existe un unique vecteur colonne U_0, de composantes a_0, b_0, qui minimise $Q(a, b)$. De plus, ce vecteur colonne est donné par :

$$U_0 = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = ({}^tAA)^{-1}({}^tAV) = \begin{pmatrix} \sum_{k=1}^{N} x_k^2 & \sum_{k=1}^{N} x_k \\ \sum_{k=1}^{N} x_k & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{k=1}^{N} x_k y_k \\ \sum_{k=1}^{N} y_k \end{pmatrix}.$$

Après calculs, on trouve alors que :

$$\begin{cases} a_0 &= \dfrac{\mathrm{cov}(X, Y)}{V(X)} \\ \\ b_0 &= \overline{Y} - \dfrac{\mathrm{cov}(X, Y)}{V(X)}\overline{X} \end{cases},$$

lesquels sont les coefficients de l'équation de la droite de régression de Y par rapport à X. On verra en exercice comment retrouver cette droite à l'aide du théorème sur la projection orthogonale. A noter enfin que les coefficients a, b sont donnés en Python par la liste de commandes suivante (après avoir importé la bibliothèque `numpy` au préalable) :

```
c=np.mean(x*y)-np.mean(x)*np.mean(y)
a=c/np.var(x)
b=np.mean(y)-a*np.mean(x)
```

De la même façon, on peut utiliser la méthode MCO pour déterminer la droite du plan d'équation cartésienne $x = ay + b$ qui s'ajuste le mieux au nuage de points, au sens où la somme des carrés des écarts horizontaux entre les points du nuage et la droite en question est minimale, ce qui revient à minimiser l'expression $R(a, b) = \sum_{i=1}^{N}(x_i - ay_i - b)^2$ en fonction de a et b. Comme précédemment, on montre (et on admettra) le résultat suivant :

Théorème - définition 3.2. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y), telle que les y_i ne sont pas tous égaux. Alors il existe une unique droite notée $\mathcal{D}_{X/Y}$ d'équation $x = ay + b$ qui minimise l'expression $R(a, b) = \sum_{i=1}^{N}(x_i - ay_i - b)^2$. Cette droite est appelée la droite de régression de X par rapport à Y, et elle admet pour équation :*

$$x = \overline{X} + \frac{\mathrm{cov}(X, Y)}{V(Y)}(y - \overline{Y}).$$

A noter que les coefficients a, b sont donnés en Python par les instructions :

```
c=np.mean(x*y)-np.mean(x)*np.mean(y)
a=c/np.var(y)
b=np.mean(x)-a*np.mean(y)
```

Définition 3.3. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y). Le point moyen du nuage est le point de coordonnées $(\overline{X}, \overline{Y})$.*

A noter qu'une propriété remarquable des droites de régression est donnée par le :

Théorème 3.4. *Soit $((x_i, y_i))_{1\leq i\leq N}$ une série statistique double associée à un couple de variables (X, Y), telle que les x_i (resp. y_i) ne sont pas tous égaux. Alors les droites de régression $\mathcal{D}_{Y/X}$ et $\mathcal{D}_{X/Y}$ passent par le point moyen du nuage.*

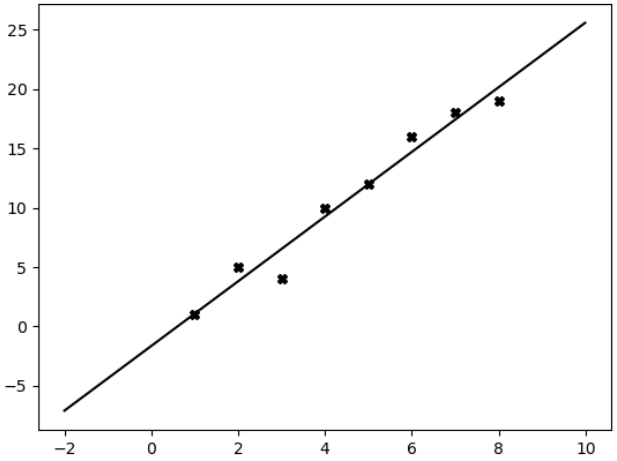
Une fois que l'on a déterminé les droites qui s'ajustent le mieux au nuage de points, on peut se demander si les points du nuage sont proches des droites en question. Pour ce faire, on utilisera le coefficient de corrélation linéaire. Plus précisément, on retiendra que, ***plus*** $\rho(X, Y)$ ***est proche de*** 1 ***ou*** -1***, plus le nuage de points est proche des deux droites de régression***, la situation limite correspondant au cas où tous les points du nuage sont situées sur les deux droites de régression (et où celles-ci sont confondues). En outre, le signe de $\rho(X, Y)$ nous indique si les variables X, Y en question ont tendance à varier dans le même sens (c'est-à-dire si Y tend à croître quand X croît) ou l'inverse, vu que ce signe est celui des coefficients directeurs (ou pentes) des droites de régression.

Exemple 3.5. *Considérons la série statistique double suivante :*

x_i	1	2	3	4	5	6	7	8
y_i	1	5	4	10	12	16	18	19

Pour tracer le nuage de points correspondant, la droite de régression de Y par rapport à X et enfin calculer et afficher le coefficient de corrélation linéaire, on utilisera la fonction suivante (après avoir importé les bilbiothèques `numpy` *et* `matplotlib.pyplot`*) :*

```
def regression():
    x=np.array([1,2,3,4,5,6,7,8])
    y=np.array([1,5,4,10,12,16,18,19])
    r=np.mean(x*y)-np.mean(x)*np.mean(y)
    a=r/np.var(x)
    c=r/(np.std(x)*np.std(y))
    u=np.arange(-2,10,0.01)
    v=a*(u-np.mean(x))+np.mean(y)
    plt.plot(x,y,'x',color='k')
    plt.plot(u,v,color='k')
    plt.show()
    return c
```



Dans ce cas, Python affiche un coefficient de corrélation linéaire égal à $0,981730149$ et nous donne le graphe ci-dessus.