TP 3
Statistiques bivariées

1 Objectif

Le but de ce TP est de donner quelques outils d'étude de séries statistiques à deux variables. La question principale est celle de la *dépendance* entre ces deux variables. On mettra à profit les outils de probabilité permettant de quantifier cela.

2 Les données et leur représentation

Une *série statistique double* est la donnée d'une liste de couples, qui représentent deux caractéristiques d'un individu. On peut par exemple envisager la superficie et la population de pays. Dans ce cas, on obtient par exemple la série :

Nom	Fra.	All.	U.K.	Suisse	Esp.	Portugal	Pologne	Suède	Grèce	Italie
Superficie (milliers de km^2)	552	357	243	41	510	92	313	450	132	301
Population (millions d'hab.)	67,6	82,8	65,3	8,3	46,4	10,3	38	10	10,8	61,3

On formalise cela en deux listes de même taille, contenant les valeurs de chaque caractère. Dans notre série, on obtiendra donc :

```
surf=[552,357,243,41,510,92,313,450,132,301]
pop=[67.6,82.8,65.3,8.3,46.4,10.3,38,10,10.8,61.3]
```

Une représentation naturelle de ces données consiste à associer à chaque « individu » (ici, pays), un point du plan dont l'abscisse est sa superficie et l'ordonnée sa population. On obtient ainsi un *nuage de points*. En Python, ceci se fait avec le package Matplotlib, importé traditionnellement par

```
import matplotlib.pyplot as plt
```

La commande `plt.scatter` prend en argument deux listes X = $[x_1,\dots,x_n]$ et Y = $[y_1,\dots,y_n]$, et renvoie le nuage des points $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$.

```
plt.scatter(X,Y)
plt.show()
```

Dans ce qui suit on fera figurer les données dans des objets `np.array` ; en effet le calcul d'indicateurs statistiques demande des opérations qui se font de manière bien plus aisée sur des `array` que sur des listes.

Point moyen On peut commencer par examiner la position des données, pour avoir un ordre de grandeur de ce qui est étudié. On définit le *point moyen* de la série comme le point ayant pour abscisse la moyenne des x_i, et pour ordonnée la moyenne des y_i.
Ainsi, si notre série statistique à deux variables est donnée par $\{(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)\}$, le point moyen a pour coordonnées $(\overline{x},\overline{y})$, avec

$$\overline{x}=\frac{1}{n}\sum_{i=1}^{n}x_i \quad \text{et} \quad \overline{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$$

($\overline{x}$ et $\overline{y}$ sont appelées *moyennes empiriques*).

La commande `np.mean` de Python permet un calcul immédiat du point moyen : il suffit de faire figurer le point de coordonnées `(np.mean(x),np.mean(y))`. Pour mettre en évidence ce point, on pourra le tracer en rouge, en passant l'option `c = 'red'`.

```
plt.scatter(np.mean(X),np.mean(Y),c = 'red')
plt.show()
```

3 Corrélation entre les deux quantités

On se pose maintenant la question de la dépendance entre les deux caractères : leurs monotonies sont-elles reliées ? Les points ont-ils tendance à s'arranger autour d'une droite (ce qui donnerait une dépendance affine) ? Ou selon une courbe plus compliquée (fonction puissance, etc.) ? S'arrangent-ils régulièrement, ou y a-t-il des « accidents » ?

3.1 Variance et covariance empiriques

On a vu que si X et Y sont deux variables aléatoires, on peut définir sous certaines conditions la variance de X et la covariance de X et Y : on a les formules

$$\mathrm{V}(\mathrm{X}) = \mathbb{E}\big((\mathrm{X}-\mathbb{E}(\mathrm{X}))^2\big) = \mathbb{E}\big(\mathrm{X}^2\big) - \mathbb{E}(\mathrm{X})^2$$

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = \mathbb{E}\big((\mathrm{X}-\mathbb{E}(\mathrm{X}))(\mathrm{Y}-\mathbb{E}(\mathrm{Y}))\big) = \mathbb{E}(\mathrm{XY}) - \mathbb{E}(\mathrm{X})\mathbb{E}(\mathrm{Y})$$

On a aussi défini le coefficient de corrélation linéaire de X et Y en divisant la covariance par le produit des écarts-type, ce qui permet une normalisation : ce coefficient sera toujours compris entre -1 et 1.

$$\rho(\mathrm{X},\mathrm{Y}) = \frac{\mathrm{Cov}(\mathrm{X},\mathrm{Y})}{\sigma(\mathrm{X})\sigma(\mathrm{Y})}$$

Ces quantités ont des équivalents *empiriques* dans le cas de nos jeux de données : si $\{(x_i, y_i)\}_{1\leqslant i\leqslant n}$ est la série à étudier, on a les indicateurs suivants :

$$s_x^2 = \overline{x^2} - \left(\overline{x}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 \qquad \text{(variance empirique)}$$

La racine carrée de la variance (notée s_x donc) est ici aussi appelée écart-type.

$$s_{x,y} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \frac{1}{n^2}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right) \qquad \text{(covariance empirique)}$$

$$r_{x,y} = \frac{s_{x,y}}{s_x s_y} \qquad \text{(coefficient de corrélation empirique)}$$

Des fonctions Python sont au programme pour effectuer ces calculs :

- La variance empirique d'une série statistique contenue dans un `np.array` est obtenue par `np.var(L)`
- L'écart-type empirique d'une série statistique contenue dans un `np.array` est obtenu par `np.std(L)`

Exercice 1. Programmer des fonctions Python :

- une fonction `cov_emp` qui calcule la covariance empirique de deux `np.array` X et Y.
- une fonction `corr_emp` qui calcule le coefficient de corrélation linéaire empirique de deux `np.array` X et Y (on pourra appeler la fonction précédente).

On fera en sorte de programmer efficacement : notamment nul besoin de boucle `for` pour calculer les quantités demandées !

3.2 Régression linéaire et méthode des moindres carrés

Si on suppose que la dépendance entre les caractères x et y est une équation de type $y \simeq ax+b$, où a et b sont fixés, on se ramène à un problème de *régression linéaire* : il s'agit de tracer une droite qui approxime le mieux possible le nuage de points précédent. La qualité de cette approximation sera donnée par le coefficient de corrélation linéaire discuté précédemment.

Il faut donc commencer par rechercher la meilleure droite. On utilise pour cela la *méthode des moindres carrés* : en notant encore $\{(x_i, y_i)\}_{1\leqslant i\leqslant n}$ la série à étudier, on cherche à trouver les réels a et b rendant la somme

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

minimale. Un théorème donne la valeur de ces réels.

Théorème 1. *Soit $\{(x_i, y_i)\}_{1\leqslant i\leqslant n}$ une série statistique double.*
La droite de régression linéaire de Y en X a pour équation $y = ax + b$, où :

$$a = \frac{s_{x,y}}{s_x^2} \ , \ b = \overline{y} - a\overline{x}$$

Démonstration. Admis. □

Remarque 1. La recherche d'une relation $y = ax + b$ sous-entend qu'on cherche à déterminer les valeurs de y à partir de celles de x. Dans ce contexte, X est appelée *variable explicative* et Y *variable à expliquer*.
Le choix de la variable explicative et de la variable à expliquer peut être délicat. De manière presque sûre vous n'aurez pas à vous en préoccuper.

Remarque 2. Il faut connaître la formule donnant a ; b s'obtient alors en se souvenant que :

La droite de régression linéaire passe par le point moyen.

4 Exemples de régression et utilisation

On souhaite effectuer une régression linéaire sur les données suivantes.
Les données se présentent sous la forme de deux listes `X` et `Y` (sur le modèle de `surf` et `pop` du début de TP).
On souhaite coder une fonction `regression(X,Y)` qui :

- représentera le nuage de points correspondant aux données ;
- dessinera le point moyen et la droite de régression linéaire ;
- *renverra* (par un `return`, donc) les valeurs de a et b définies dans le théorème 1, ainsi que le coefficient r de régression linéaire.

Compléter le code suivant :

```
def regression(X,Y):
    plt.scatter(... , ... , c='blue') # nuage de points en bleu
    plt.scatter(... , ... , c='red') # point moyen en rouge
    a = ...
    b = ...
    r = ...
    plt.plot(X,a*X+b,c='green') # droite de régression en vert
    plt.show()
    return ...
```

Taux de fertilité vs. taux d'urbanisation Pour divers pays d'un monde, on donne les taux de fertilité (nombre d'enfants par femme) et le pourcentage de population urbaine, pour l'année 1990 :

Pays	Taux de fertilité	% de pop. urbaine
Chine	2.51	26.4
Italie	1.33	66.7
Tadjikistan	5.34	31.7
Allemagne	1.45	73.1
Israël	2.82	90.3
Ghana	5.7	36.4
Côte d'Ivoire	6.7	39.4
Philippines	4.35	47
Australie	1.9	85.4
Colombie	3.08	69.5
Mongolie	4.23	57
France	1.77	74
Roumanie	1.83	53.2
Iran	4.86	56
Burkina Faso	7	13.87
Togo	6.1	28.6
Myanmar	3.54	25.2
Japon	1.54	77.3
Pakistan	6.4	30.6
Thaïlande	2.1	29.4
Afghanistan	7.6	21.8

Source : Banque Mondiale.

À l'aide de la pente de la droite de régression, estimer la baisse du taux de fertilité dans un pays si le taux de population urbaine augmente de 5%.

Densité de population vs. taux de criminalité On cherche à analyser s'il existe une relation linéaire entre la densité de population dans les régions métropolitaines et le taux de criminalité correspondant dans ces régions. Le taux de criminalité Y est indiqué en nombre de crimes par 10 000 habitants, et la densité de population X est mesurée en milliers d'habitants par km^2. On obtient le tableau suivant :

Région	1	2	3	4	5	6	7	8	9	10	11	12
X	7.7	5.8	11.5	2.1	3.7	3.6	7.5	4.2	3.8	10.3	8.6	7.2
Y	12	9	15	4	4	5	10	6	5	11	10	11

Par interpolation, estimer le taux de criminalité d'une ville d'une densité de 5 000 habitants par km^2.

5 Cas de dépendances non linéaires

Il peut arriver que la nature des données fasse qu'il est difficile d'envisager une relation entre les deux caractères de type $Y = aX + b$: c'est par exemple le cas si on considère la taille et le poids d'un individu[1]. Une méthode consiste, dans le cas de données à valeurs strictement positives, à passer au ln : une relation $Y = kX^a$, avec $k > 0$ et $a \in \mathbb{R}$, se réécrit aussi $\ln(Y) = a\ln(X) + \ln(k)$; soit en posant $U = \ln(X)$ et $V = \ln(Y)$, à $V = aU + \ln(k)$, qui est bien une équation de droite. La régression linéaire sur les variables U et V donnera la dépendance voulue pour X et Y.

Exercice 2. Considérons la série suivante :

X	3	13	15	6	5	2	30	6	19	14	26	6	1	7	30
Y	33	12	7	17	26	50	3	17	12	4	4	17	105	15	3

1. Tracer le nuage de points correspondant à ces données. La dépendance paraît-elle linéaire ? Calculer le coefficient de corrélation linéaire de X et Y.

[1] Le fameux indice de masse corporelle s'obtient d'ailleurs comme poids/taille^2.

2. En étudiant les listes `np.log(X)` et `np.log(Y)`, proposer des valeurs de α et β telles que $Y \simeq \alpha \times X^{\beta}$. Calculer le coefficient de corrélation linéaire de `np.log(X)` et `np.log(Y)` ; cette formule semble-t-elle plus performante ?

Exercice 3. Reprendre avec la série :

X	20	35	100	85	44	18	29	22	58	65	68	25	11	53
Y	601	1832	14996	10839	2899	492	1268	727	5047	6340	6947	929	179	4210