

DM4  
Corrigé

**Exercice 1 : tirages aléatoires**

Soit  $p \in ]0, 1[$ . On considère une pièce donnant Pile avec probabilité  $p$ . On effectue une succession de lancers indépendants de cette pièce jusqu'à l'obtention du deuxième Pile.

On note alors  $X_1$  le rang d'apparition du premier Pile et  $X_2$  le nombre de lancers supplémentaires effectués après le premier Pile jusqu'à l'apparition du deuxième Pile.

Par exemple, si les lancers donnent dans cet ordre :

**Face, Pile, Face, Face, Face, Pile**

alors  $[X_1 = 2]$  et  $[X_2 = 4]$  se réalisent.

1. **Reconnaître la loi de  $X_1$  et la loi de  $X_2$ .**

Par indépendance des lancers successifs,  $X_1$  et  $X_2$  sont des temps d'attente du 1er succès (obtenir Pile) dans une succession d'expériences indépendantes. Ainsi  $X_1$  et  $X_2$  suivent  $\mathcal{G}(p)$ .

2. **Déterminer la loi conjointe du couple  $(X_1, X_2)$ .**

On note les événements usuels  $P_k$  (resp.  $F_k$ ) : « obtenir Pile (resp. Face) au lancer  $k$  » .  
 $X_1(\Omega) = X_2(\Omega) = \mathbb{N}^*$  ; et  $\forall (i, j) \in (\mathbb{N}^*)^2$ ,

$$\mathbf{P}(X_1 = i \cap X_2 = j) = \mathbf{P}(F_1 \cap \dots \cap F_{i-1} \cap P_i \cap F_{i+1} \cap \dots \cap F_{i+j-1} \cap P_{i+j}) = (1-p)^{i+j-2} p^2$$

la dernière égalité résultant de l'indépendance des lancers successifs.

3. **En déduire que les variables aléatoires  $X_1$  et  $X_2$  sont indépendantes.**

D'après la première question :  $\forall (i, j) \in (\mathbb{N}^*)^2$ ,  $\mathbf{P}(X_1 = i) = (1-p)^{i-1} p$  et  $\mathbf{P}(X_2 = j) = (1-p)^{j-1} p$ .  
 En comparant les résultats des deux premières questions on voit que :

$$\forall (i, j) \in (\mathbb{N}^*)^2, \mathbf{P}(X_1 = i \cap X_2 = j) = \mathbf{P}(X_1 = i) \mathbf{P}(X_2 = j)$$

et on a bien l'indépendance demandée.

4. **Montrer que, pour tout entier  $n \geq 2$  :**

$$\mathbf{P}(X_1 + X_2 = n) = (n-1)p^2(1-p)^{n-2}$$

On écrit par exemple (avec  $X_1(\Omega) = \mathbb{N}^*$ )

$$\begin{aligned} \mathbf{P}(X_1 + X_2 = n) &= \sum_{k=1}^{n-1} \mathbf{P}(X_1 = k \cap X_2 = n-k) \\ &= \sum_{k=1}^{n-1} \mathbf{P}(X_1 = k) \mathbf{P}(X_2 = n-k) \quad \text{indép de } X_1 \text{ et } X_2 \\ &= \sum_{k=1}^{n-1} (1-p)^{k-1} p \times (1-p)^{n-k-1} p \\ &= p^2 \sum_{k=1}^{n-1} (1-p)^{n-2} \\ &= (n-1)p^2(1-p)^{n-2} \end{aligned}$$

On note  $Y$  la variable aléatoire égale au nombre de Face obtenus sur toute l'expérience.

5. Recopier et compléter la fonction Python suivante, prenant en argument d'entrée le réel  $p$  de  $]0, 1[$ , et renvoyant une simulation de la variable aléatoire  $Y$ .

On simule des lancers jusqu'à ce qu'on ait obtenu 2 Pile, et on renvoie le nombre de Face qu'on aura pris soin de compter au fur et à mesure de l'expérience :

```
import numpy.random as rd
def simul_Y(p):
    Y = 0 # compte les Face
    nb_pile = 0 # compte les Pile
    while nb_pile < 2:
        if rd.random() < p : # obtenir Pile
            nb_pile = nb_pile + 1
        else : # obtenir Face
            Y = Y+1
    return Y
```

6. Exprimer  $Y$  en fonction de  $X_1$  et  $X_2$ .

Au moment du 2ème Pile on a effectué  $X_1 + X_2$  lancers ; dont 2 ont donné Pile. Ainsi

$$Y = X_1 + X_2 - 2$$

7. En déduire l'espérance de  $Y$ , la variance de  $Y$  et la loi de  $Y$ .

Par linéarité de l'espérance,  $\mathbb{E}(Y)$  existe et  $\mathbb{E}(Y) = \mathbb{E}(X_1) + \mathbb{E}(X_2) - 2 = \frac{2}{p} - 2$ .

Par indépendance de  $X_1$  et  $X_2$ , et invariance par translation de la variance,  $\mathbb{V}(Y)$  existe et  $\mathbb{V}(Y) = \mathbb{V}(X_1) + \mathbb{V}(X_2) = \frac{2(1-p)}{p^2}$ .

Enfin la loi de  $Y$  se déduit de celle de  $X_1 + X_2$  :  $Y(\Omega) = \mathbb{N}$  et

$$\forall k \in \mathbb{N}, \mathbf{P}(Y = k) = \mathbf{P}(X_1 + X_2 = k + 2) = (k + 1)p^2(1 - p)^k$$

Une fois le second Pile obtenu, si l'on a obtenu un nombre  $n$  de Face, alors on place  $n + 1$  boules numérotées de 0 à  $n$  dans une urne. On effectue alors un unique tirage dans cette urne et on note  $U$  la variable aléatoire égale au numéro obtenu. On note également  $V = Y - U$ .

8. Écrire une fonction Python `simul_UV`, prenant en argument d'entrée la valeur du réel  $p$  de  $]0, 1[$ , et renvoyant une simulation du couple  $(U, V)$ . On s'efforcera de faire appel à la fonction `simul_Y` définie à la question 5.

Il s'agit donc de tirer une valeur de  $Y$  par la fonction `simul_Y`, puis de renvoyer un entier aléatoire de  $\{0, 1, \dots, Y\}$ . On propose :

```
def simul_UV(p):
    Y=simul_Y(p)
    U=rd.randint(Y+1)
    return U, Y-U
```

9. Justifier que  $U(\Omega) = \mathbb{N}$  et préciser  $V(\Omega)$ .

$Y(\Omega) = \mathbb{N}$  ; et si  $Y = n$ ,  $U$  prend une valeur arbitraire de  $\llbracket 0, n \rrbracket$ . On en déduit donc  $U(\Omega) = \mathbb{N}$ .

Il est aussi clair que si  $Y = n$ ,  $V$  prend une valeur arbitraire de  $\llbracket 0, n \rrbracket$  ;  $V(\Omega) = \mathbb{N}$ .

10. Pour tout  $k \in \mathbb{N}$ , déterminer  $\mathbf{P}(U = k)$ .

On va bien sûr passer par une FPT avec le SCE associé à  $Y$ .

D'après l'énoncé on a les probas conditionnelles :

$$\forall (n, k) \in \mathbb{N}^2, \mathbf{P}_{[Y=n]}(U = k) = \begin{cases} \frac{1}{n+1} & \text{si } 0 \leq k \leq n \\ 0 & \text{si } k > n \end{cases}$$

et on en déduit, avec le SCE  $((Y = n))_{n \in \mathbb{N}}$  :

$$\begin{aligned} \forall k \in \mathbb{N}, \mathbf{P}(U = k) &= \sum_{n=0}^{+\infty} \mathbf{P}_{[Y=n]}(U = k) \mathbf{P}(Y = n) \\ &= \sum_{n=k}^{+\infty} \frac{1}{n+1} (n+1) p^2 (1-p)^n \\ &= p^2 \sum_{n=k}^{+\infty} (1-p)^n \\ &= p^2 \frac{(1-p)^k}{1-(1-p)} \quad \text{avec } |1-p| < 1 \\ \forall k \in \mathbb{N}, \mathbf{P}(U = k) &= p(1-p)^k \end{aligned}$$

11. **Montrer que la variable aléatoire  $U+1$  suit une loi usuelle que l'on reconnaîtra. En déduire l'espérance et la variance de  $U$ .**

On a  $(U+1)(\Omega) = \mathbb{N}^*$ , et

$$\forall k \in \mathbb{N}^*, \mathbf{P}(U+1 = k) = \mathbf{P}(U = k-1) = (1-p)^{k-1} p$$

et on reconnaît donc  $U+1 \hookrightarrow \mathcal{G}(p)$ .

$U+1$  admet une espérance donc  $U$  aussi ; et  $\mathbb{E}(U) = \mathbb{E}(U+1-1) = \mathbb{E}(U+1) - 1 = \frac{1}{p} - 1 = \frac{1-p}{p}$ .

$U+1$  admet une variance donc  $U$  aussi ; et  $\mathbb{V}(U) = \mathbb{V}(U+1) = \frac{1-p}{p^2}$ .

12. **Montrer que  $V$  suit la même loi que  $U$ .**

On reprend le calcul des probas conditionnelles :

$$\forall (n, k) \in \mathbb{N}^2, \mathbf{P}_{[Y=n]}(V = k) = \mathbf{P}_{[Y=n]}(U = n-k) = \begin{cases} \frac{1}{n+1} & \text{si } 0 \leq n-k \leq n \\ 0 & \text{sinon} \end{cases} = \begin{cases} \frac{1}{n+1} & \text{si } 0 \leq k \leq n \\ 0 & \text{sinon} \end{cases}$$

On voit qu'on va effectuer les mêmes calculs que pour  $U$  :  $U$  et  $V$  suivent donc la même loi.

13. (a) **Montrer que  $U$  et  $V$  sont indépendantes.**

Assez contre-intuitif !

Soient  $(i, j) \in \mathbb{N}^2$  :

$\mathbf{P}(U = i) \mathbf{P}(V = j) = p^2 (1-p)^{i+j}$  d'après les lois obtenues ; et

$$\begin{aligned} \mathbf{P}(U = i \cap V = j) &= \mathbf{P}(U = i \cap Y = i+j) \\ &= \mathbf{P}_{[Y=i+j]}(U = i) \times \mathbf{P}(Y = i+j) \\ &= \frac{1}{i+j+1} \times (i+j+1) p^2 (1-p)^{i+j} \quad (\text{on a bien } 0 \leq i \leq i+j \text{ pour l'expression de la conditionnelle}) \end{aligned}$$

$$\mathbf{P}(U = i \cap V = j) = p^2 (1-p)^{i+j}$$

On observe bien que :

$$\mathbf{P}(U = i \cap V = j) = \mathbf{P}(U = i) \mathbf{P}(V = j)$$

et on a donc l'indépendance de  $U$  et  $V$ .

- (b) **En déduire la covariance du couple  $(Y, U)$ .**

Cette covariance existe car  $Y$  et  $U$  admettent des variances.

On utilise la bilinéarité, en remarquant que  $Y = U + V$ .

Dès lors, avec l'indépendance de  $U$  et  $V$  :

$$\text{Cov}(Y, U) = \text{Cov}(U + V, U) = \text{Cov}(U, U) + \text{Cov}(U, V) = \mathbb{V}(U) + 0 = \frac{1-p}{p^2}$$

- (c) **Montrer que le coefficient de corrélation linéaire du couple (Y, U) est égal à  $\frac{\sqrt{2}}{2}$ .**

Il suffit d'appliquer la formule :

$$\rho = \frac{\text{Cov}(Y, U)}{\sqrt{V(U)}\sqrt{V(Y)}} = \frac{\frac{1-p}{p^2}}{\sqrt{\frac{1-p}{p^2}}\sqrt{\frac{2(1-p)}{p^2}}} = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}$$

## Exercice 2 : commutant d'une matrice

On note  $I_3$  la matrice identité de  $\mathcal{M}_3(\mathbb{R})$ , et on considère la matrice  $A = \begin{pmatrix} 7 & 5 & 1 \\ 6 & -1 & 2 \\ 6 & 1 & 3 \end{pmatrix}$

1. (a) **Calculatoire ; repenser au moyen de gérer le « pivot délicat » de la feuille de TD**  
**Montrer, grâce à la méthode du pivot de Gauss, que les valeurs propres  $\lambda$  de A sont les solutions de l'équation :  $\lambda^3 - 9\lambda^2 - 27\lambda + 53 = 0$ .**

On effectue un pivot sur  $A - \lambda I_3 = \begin{pmatrix} 7-\lambda & 5 & 1 \\ 6 & -1-\lambda & 2 \\ 6 & 1 & 3-\lambda \end{pmatrix}$ .

Avec  $L_1 \leftrightarrow L_3$  :

$$\begin{pmatrix} 6 & 1 & 3-\lambda \\ 6 & -1-\lambda & 2 \\ 7-\lambda & 5 & 1 \end{pmatrix}$$

puis :  $L_2 \leftarrow L_2 - L_1$ ,  $L_3 \leftarrow 6L_3 - (7-\lambda)L_1$  :

$$\begin{pmatrix} 6 & 1 & 3-\lambda \\ 0 & -2-\lambda & -1+\lambda \\ 0 & 23+\lambda & -15+10\lambda-\lambda^2 \end{pmatrix}$$

$L_2 \leftarrow L_2 + L_3$  :

$$\begin{pmatrix} 6 & 1 & 3-\lambda \\ 0 & 21 & -16+11\lambda-\lambda^2 \\ 0 & 23+\lambda & -15+10\lambda-\lambda^2 \end{pmatrix}$$

$L_3 \leftarrow 21L_3 - (23+\lambda)L_2$  :

$$\begin{pmatrix} 6 & 1 & 3-\lambda \\ 0 & 21 & -16+11\lambda-\lambda^2 \\ 0 & 0 & 53-27\lambda-9\lambda^2+\lambda^3 \end{pmatrix} = T$$

$\lambda$  est valeur propre de A ssi  $A - \lambda I_3$  n'est pas inversible, donc ssi T n'est pas inversible, donc ssi  $53 - 27\lambda - 9\lambda^2 + \lambda^3 = 0$ .

- (b) **Étudier la fonction  $f$  qui, à tout réel  $x$  associe  $f(x) = x^3 - 9x^2 - 27x + 53$ , puis dresser son tableau de variation (on précisera les limites de  $f$  en  $+\infty$  et en  $-\infty$ , on notera  $m$  le minimum local de  $f$  sur  $\mathbb{R}$ ,  $M$  le maximum local de  $f$  sur  $\mathbb{R}$  et on ne cherchera à calculer ni  $m$ , ni  $M$ ).**

$f'(x) = 3x^2 - 18x - 27 = 3(x^2 - 6x - 9)$ . On résout  $x^2 - 6x - 9 = 0$  ; on trouve  $\Delta = 72 = 2.6^2$  puis les racines  $3 \pm 3\sqrt{2}$ .

On note que  $3 - 3\sqrt{2} < 0 < 3 < 3 + 3\sqrt{2}$ .

Le tableau de variations de  $f$  est :

|         |           |  |                 |           |   |
|---------|-----------|--|-----------------|-----------|---|
| $x$     | $-\infty$ | $3 - 3\sqrt{2}$                                      | $3 + 3\sqrt{2}$ | $+\infty$ |   |
| $f'(x)$ | +         | 0  | -               | 0         | + |
| $f(x)$  | $-\infty$ | $\swarrow$ M $\searrow$<br>$\nearrow$ $m$ $\nwarrow$ |                 | $+\infty$ |   |

(c) Calculer  $f(0)$  et  $f(3)$  puis déterminer les signes de  $m$  et  $M$ .

$f(0) = 53 > 0$  ; par décroissance stricte de  $f$  sur  $[3 - 3\sqrt{2}, 0]$  on en déduit que  $M > 0$ .  
 $f(3) = -82 < 0$  ; par décroissance stricte de  $f$  sur  $[3, 3 + 3\sqrt{2}]$  on en déduit que  $m < 0$ .

(d) Montrer que  $A$  admet trois valeurs propres, que l'on ne cherchera pas à calculer et que l'on notera  $\lambda_1, \lambda_2$  et  $\lambda_3$  avec  $\lambda_1 < \lambda_2 < \lambda_3$ .

On reprend le tableau de variation de  $f$  en se souvenant que  $M > 0$  et  $m < 0$ . Par théorèmes de la bijection ( $f$  continue, et strictement monotone sur les intervalles qui suivent) successivement sur les intervalles  $]-\infty, 3 - 3\sqrt{2}[$ ,  $[3 - 3\sqrt{2}, 3 + 3\sqrt{2}[$  et  $]3 + 3\sqrt{2}, +\infty[$  on obtient l'existence de  $\lambda_1, \lambda_2, \lambda_3$  réels tels que  $\lambda_1 < 3 - 3\sqrt{2} < \lambda_2 < 3 + 3\sqrt{2} < \lambda_3$  et  $f(\lambda_1) = f(\lambda_2) = f(\lambda_3) = 0$ . Ce sont les trois seules solutions de  $f(x) = 0$  sur  $\mathbb{R}$ .

(e) En déduire qu'il existe une matrice  $P$  inversible telle que  $A = PDP^{-1}$ , avec  $D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$ .

La matrice  $A \in \mathcal{M}_3(\mathbb{R})$  a donc 3 valeurs propres  $\lambda_1, \lambda_2, \lambda_3$ . Les trois sep associés sont de dimension 1, car la somme des dimensions de tous les sep ne peut pas excéder 3.  
Si  $E_{\lambda_i}(A) = \text{Vect}(C_i)$ , la famille  $(C_1, C_2, C_3)$  est libre (vep pour des vap 2 à 2 distinctes) donc forme une base de  $\mathcal{M}_{3,1}(\mathbb{R})$  ; si on définit  $P$  comme la matrice donc les colonnes sont  $C_1, C_2, C_3$  on sait alors que  $P$  est inversible et que  $A = PDP^{-1}$ , avec  $D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$ .

2. On s'intéresse ici au **commutant** de  $A$ , c'est-à-dire l'ensemble des matrices qui commutent avec  $A$ .

On pose  $\mathcal{C}_A = \{M \in \mathcal{M}_3(\mathbb{R}) \mid AM = MA\}$  ; et de même  $\mathcal{C}_D = \{M \in \mathcal{M}_3(\mathbb{R}) \mid DM = MD\}$ .

(a) Montrer que  $\mathcal{C}_A$  est un sous-espace vectoriel de  $\mathcal{M}_3(\mathbb{R})$ . On admet que  $\mathcal{C}_D$  est aussi un sev de  $\mathcal{M}_3(\mathbb{R})$ .

$\mathcal{C}_A$  est non vide car la matrice nulle commute avec  $A$ .

Si  $(M_1, M_2) \in \mathcal{C}_A^2$ , et  $\lambda \in \mathbb{R}$ , alors

$$A(\lambda M_1 + M_2) = \lambda AM_1 + AM_2 = \lambda M_1 A + M_2 A = (\lambda M_1 + M_2)A$$

et donc  $\lambda M_1 + M_2 \in \mathcal{C}_A$ .

$\mathcal{C}_A$  est donc bien un sev de  $\mathcal{M}_3(\mathbb{R})$ .

(b) Montrer que les matrices de  $\mathcal{C}_D$  sont exactement les matrices diagonales.

On cherche à quelle condition  $N = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$  commute avec  $D$ .

On a  $ND = \begin{pmatrix} \lambda_1 a & \lambda_2 b & \lambda_3 c \\ \lambda_1 d & \lambda_2 e & \lambda_3 f \\ \lambda_1 g & \lambda_2 h & \lambda_3 i \end{pmatrix}$  et  $DN = \begin{pmatrix} \lambda_1 a & \lambda_1 b & \lambda_1 c \\ \lambda_2 d & \lambda_2 e & \lambda_2 f \\ \lambda_3 g & \lambda_3 h & \lambda_3 i \end{pmatrix}$  d'où  $ND = DN$  ssi

$$\begin{cases} \lambda_2 b = \lambda_1 b \\ \lambda_3 c = \lambda_1 c \\ \lambda_1 d = \lambda_2 d \\ \lambda_3 f = \lambda_2 f \\ \lambda_1 g = \lambda_3 g \\ \lambda_2 h = \lambda_3 h \end{cases} \Leftrightarrow \begin{cases} (\lambda_2 - \lambda_1)b = 0 \\ (\lambda_3 - \lambda_1)c = 0 \\ (\lambda_1 - \lambda_2)d = 0 \\ (\lambda_3 - \lambda_2)f = 0 \\ (\lambda_1 - \lambda_3)g = 0 \\ (\lambda_2 - \lambda_3)h = 0 \end{cases} \Leftrightarrow b = c = d = f = g = h = 0$$

CAR  $\lambda_1 - \lambda_2$ ,  $\lambda_1 - \lambda_3$  et  $\lambda_2 - \lambda_3$  sont non nuls !!

On obtient bien que tous les coefficients non diagonaux de  $N$  sont nuls : les matrices qui commutent avec  $D$  sont exactement les matrices diagonales.

(c) **Montrer :**  $M \in \mathcal{C}_A \Leftrightarrow P^{-1}MP \in \mathcal{C}_D$ .

Les équivalences étant justifiées car  $P$  et  $P^{-1}$  sont inversibles :

$$\begin{aligned} M \in \mathcal{C}_A &\Leftrightarrow AM = MA \\ &\Leftrightarrow PDP^{-1}M = MPDP^{-1} \\ &\Leftrightarrow DP^{-1}M = P^{-1}MPDP^{-1} \quad (\text{mult. par } P^{-1} \text{ à gauche}) \\ &\Leftrightarrow D(P^{-1}MP) = (P^{-1}MP)D \quad (\text{mult. par } P \text{ à droite}) \\ &\Leftrightarrow P^{-1}MP \in \mathcal{C}_D \end{aligned}$$

(d) **Établir que  $\mathcal{C}_A$  est engendré par les trois matrices suivantes :**

$$P \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} P^{-1}, \quad P \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} P^{-1}, \quad P \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} P^{-1}$$

Notons  $D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ ,  $D_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ ,  $D_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ .

On a

$$\begin{aligned} M \in \mathcal{C}_A &\Leftrightarrow P^{-1}MP \in \mathcal{C}_D \\ &\Leftrightarrow P^{-1}MP \text{ est diagonale} \\ &\Leftrightarrow P^{-1}MP = aD_1 + bD_2 + cD_3, \quad (a, b, c) \in \mathbb{R}^3 \\ &\Leftrightarrow M = aPD_1P^{-1} + bPD_2P^{-1} + cPD_3P^{-1}, \quad (a, b, c) \in \mathbb{R}^3 \end{aligned}$$

ce qui montre bien que  $\mathcal{C}_A = \text{Vect}(PD_1P^{-1}, PD_2P^{-1}, PD_3P^{-1})$ .

(e) **En déduire une base de  $\mathcal{C}_A$ , et donner sa dimension.**

La famille de la question précédente est donc génératrice de  $\mathcal{C}_A$  ; montrons qu'elle est libre.

Si on a

$$\alpha P \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} P^{-1} + \beta P \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} P^{-1} + \gamma P \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} P^{-1} = 0$$

alors  $P \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} P^{-1} = 0$  en multipliant par  $P$  à droite et  $P^{-1}$  à gauche on trouve  $\alpha = \beta = \gamma = 0$ .

D'où la liberté, et on a donc bien une base de  $\mathcal{C}_A$ .

Cette base contenant 3 matrices,  $\dim(\mathcal{C}_A) = 3$ .

- (f) **Montrer, en raisonnant sur les valeurs propres de  $A$ , qu'il n'existe aucun polynôme annulateur non nul de  $A$  qui soit de degré inférieur ou égal à 2. En déduire que  $(I, A, A^2)$  est une famille libre, puis que c'est une base de  $\mathcal{C}_A$ .**

D'après le cours, si  $Q$  est un polynôme annulateur de  $A$ ,  $\lambda_1, \lambda_2$  et  $\lambda_3$  sont des racines de  $Q$ . Comme ces 3 vap sont 2 à 2 distinctes, et que  $Q$  est un polynôme non nul, il faut forcément que  $\deg(Q) \geq 3$ . Ainsi il n'existe pas de polynôme annulateur de  $A$  non nul de degré  $\leq 2$ .  
Supposons alors qu'il existe une combi linéaire nulle de  $I, A, A^2$  :

$$\alpha I + \beta A + \gamma A^2 = 0$$

Ceci montre que  $\alpha + \beta X + \gamma X^2$  est annulateur de  $A$  : d'après ce qu'on vient de dire cela impose que c'est le polynôme nul, donc  $\alpha = \beta = \gamma = 0$ .  
 $(I, A, A^2)$  est donc libre.

Donc  $\dim(\text{Vect}(I, A, A^2)) = 3$ . Mais comme  $I, A, A^2$  commutent avec  $A$  (c'est évident) on a  $\text{Vect}(I, A, A^2) \subset \mathcal{C}$  et on vient de voir que  $\dim(\text{Vect}(I, A, A^2)) = \dim(\mathcal{C})$ .  
Ceci montre que  $\text{Vect}(I, A, A^2) = \mathcal{C}$ , et donc que  $(I, A, A^2)$  est une base de  $\mathcal{C}$ .

### Exercice 3 : bases de données et statistiques

Sur le marché des véhicules d'occasion, on observe en général une baisse du prix de revente (ou *décote*) d'un véhicule lorsque le nombre de kilomètres parcourus augmente. Une bonne estimation de cette baisse de prix permet au vendeur de fixer avec précision le prix de revente d'un véhicule.  
Les trois parties de cet exercice sont indépendantes.

#### Partie I. Étude d'une base de données.

On dispose d'une base de données comportant deux tables `vehicule` et `annonce` décrites ci-dessous.

La table `vehicule` recense des informations sur les modèles de véhicules en vente sur le marché. Elle est composée des attributs suivants.

- `id_vehicule` (de type `INTEGER`) : un code permettant d'identifier de façon unique chaque référence de véhicule (marque et modèle).
- `marque` (de type `TEXT`) : le nom du constructeur du véhicule.
- `modele` (de type `TEXT`) : le modèle du véhicule, un constructeur proposant en général plusieurs modèles de véhicules à la vente.
- `prix_neuf` (de type `INTEGER`) : prix de vente du véhicule neuf.

La table `annonce` regroupe des informations sur un grand nombre d'annonces de véhicules d'occasion. Chaque enregistrement correspond à une annonce et possède les attributs suivants.

- `id_annonce` (de type `INTEGER`) : un code permettant d'identifier chaque annonce de façon unique.
- `id_vehicule` (de type `INTEGER`) : l'identifiant du modèle de véhicule vendu, qui correspond à l'identifiant utilisé dans la table `vehicules`.
- `annee` (de type `INTEGER`) : année de première mise en circulation du véhicule.
- `km` (de type `INTEGER`) : nombre de kilomètres parcourus par le véhicule au moment de la revente.
- `prix_occasion` (de type `INTEGER`) : prix de vente du véhicule d'occasion.

1. En justifiant brièvement, identifier une clef primaire dans chacune des tables `vehicule` et `annonce`, ainsi qu'une clef étrangère dans la table `annonce`. Préciser à quel champ fait référence cette clé étrangère.

`id_vehicule` identifie de manière unique un véhicule dans la table `vehicule` ; de même pour `id_annonce` dans la table `annonce`)

Ce sont donc deux clés primaires.

Par ailleurs, le champ `id_vehicule` de la table `id_annonce` renvoie à l'identifiant d'un véhicule dans la table `vehicule` :

donc le champ `id_vehicule` de la table `id_annonce` est une clé étrangère, qui fait référence au champ `id_vehicule` de la table `vehicule`.

2. **Écrire une requête SQL permettant d'extraire les noms de tous les modèles de véhicules mis en vente par le constructeur Dubreuil Motors.**

Ces informations se trouvent toutes dans la table `vehicule` : pas de jointure nécessaire.

```
1 SELECT modele
2 FROM vehicule
3 WHERE marque='Dubreuil Motors'
```

3. **Expliquer le fonctionnement de la requête SQL suivante et préciser l'effet éventuel de cette requête sur chacune des tables `vehicule` et `annonce`.**

```
1 UPDATE annonce
2 SET prix_occasion = prix_neuf
3 FROM vehicule
4 WHERE vehicule.id_vehicule = annonce.id_vehicule
5 AND vehicule.prix_neuf < annonce.prix_occasion
```

Pas évident ! Lignes 1 et 2 : dans la table `annonce`, on remplace, dans une annonce le prix d'occasion par le prix neuf. Lignes 3 et 4 : on fait la jointure « clé étrangère = clé primaire » .

Ligne 5 : pour récupérer les annonces pour lesquelles le prix de vente du véhicule est supérieur au prix du véhicule neuf.

Conclusion : cette requête cherche les annonces pour lesquelles le prix de vente est supérieur au prix du véhicule neuf ; et donne alors à ce prix de vente le prix du véhicule neuf.

C'est la moindre des choses, mais ça ne reste pas très attractif...

4. **À l'aide d'une jointure, écrire une requête SQL permettant d'obtenir, sur une même table, la liste de toutes les annonces de la table `annonce` avec les attributs suivants :**

- l'identifiant de l'annonce `id_annonce`,
- le kilométrage `km`,
- le prix de vente du véhicule neuf `prix_neuf`,
- le prix de l'annonce d'occasion `prix_occasion`.

On fait ce qu'on nous dit ; la jointure a lieu en identifiant les valeurs de la clé primaire à celles de la clé étrangère.

```
1 SELECT id_annonce , km , prix_neuf , prix_occasion
2 FROM vehicule INNER JOIN annonce ON vehicule.id_vehicule = annonce.id_vehicule
```

## Partie II. Étude de la décote.

Soit  $n \geq 1$  un entier. On suppose dans cette partie qu'on dispose des informations suivantes concernant  $n$  annonces de véhicules d'occasion.

Ces informations sont stockées sous forme de tableaux Numpy à une dimension et à  $n$  éléments :



- le tableau Numpy nommé `km` contient, pour chaque annonce, le nombre de kilomètres parcourus par le véhicule vendu depuis sa mise en circulation.
- le tableau Numpy nommé `rapport` contient, pour chaque annonce, le quotient du prix de revente d'occasion du véhicule par son prix de vente neuf

Ainsi, les éléments du tableau `rapport` sont des flottants compris entre 0 et 1.

Les données contenues dans ces variables sont classés dans le même ordre. Ainsi, pour tout indice  $i$  entre 0 et  $n-1$ , `km[i]` et `rapport[i]` correspondent respectivement au kilométrage et au rapport de prix occasion/neuf d'une même annonce.

5. Écrire une suite d'instructions Python permettant d'afficher le rapport de prix occasion/neuf du véhicule ayant le plus grand kilométrage parmi toutes les annonces (ou de l'un de ces véhicules si plusieurs annonces possèdent un kilométrage maximal).

Ça ressemble à une recherche de max dans le tableau des kilométrages, mais il faut conserver l'indice où apparaît le kilométrage maximal car on renvoie un élément de l'autre tableau !

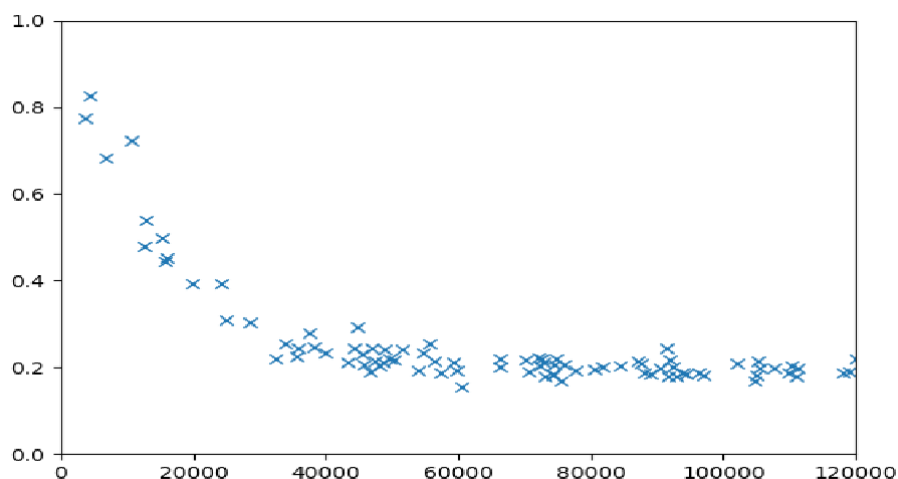
```
maxi = km[0]
indice_maxi = 0
for k in range(1,n-1):
    if km[k]>maxi:
        maxi = km[k]
        indice_maxi = k
# à ce stade k est un indice où on a un véhicule de kilométrage maximal
print(rapport[k])
# et on renvoie le rapport associé à ce véhicule
```

6. Écrire une suite d'instructions Python permettant de représenter graphiquement, sous forme de nuage de points, les points du plan de coordonnées `(km[i], rapport[i])`, où  $i$  décrit l'ensemble des indices de 0 à  $n-1$ .

Un nuage de points s'obtient avec `plt.scatter`. Conformément aux exigences du sujet il faut écrire la ligne d'import ; ensuite l'énoncé fournit directement les abscisses et les ordonnées :

```
import matplotlib.pyplot as plt
plt.scatter(km, rapport)
plt.show()
```

7. Les instructions données à la question précédente permettent d'obtenir la figure représentée ci-dessous.



Une approximation par régression linéaire vous semble-t-elle pertinente pour modéliser l'évolution du prix des véhicules d'occasion (rapporté au prix neuf) en fonction de leur kilométrage ?

Bof bof... ça ne ressemble pas trop à une droite ! On pense plutôt à du  $1/x$  ou du  $e^{-x}$ ... Une approximation par régression linéaire ne semble pas pertinente ici.

8. On calcule le coefficient de corrélation linéaire rho de la série statistique double (km, rapport). Parmi les quantités suivantes, en justifiant votre réponse, indiquer la valeur obtenue.

- (a) rho = 0.113
- (b) rho = -0.406
- (c) rho = -0.985
- (d) rho = 0.917

Globalement, le rapport décroît quand le kilométrage augmente : on aura donc rho négatif. De plus on vient de voir que la régression linéaire ne sera pas trop performante donc rho ne sera pas proche de -1.

La seule solution est donc rho = -0.406.

### Partie III. Un modèle de décroissance exponentielle.

Soit  $n \geq 1$  un entier. Dans cette partie, on étudie deux variables statistiques quantitatives  $x$  et  $r$ . On souhaite expliquer la dépendance de  $r$  par rapport à  $x$  à l'aide d'une relation exponentielle de la forme :

$$r = ae^{-cx} \quad (*)$$

où  $a$  et  $c$  sont des réels que l'on cherche à déterminer, avec  $a > 0$ . On dispose pour cela de deux séries statistiques  $(x_1, x_2, \dots, x_n)$  et  $(r_1, r_2, \dots, r_n)$  représentant des mesures des variables  $x$  et  $r$  respectivement.

9. On pose  $y = \ln(r)$ .

Montrer que les variables  $x$  et  $r$  vérifient la relation (\*) si et seulement si  $x$  et  $y$  vérifient une relation de la forme :

$$y = \alpha x + \beta$$

où  $\alpha$  et  $\beta$  sont des paramètres réels que l'on exprimera en fonction de  $a$  et  $c$ .

En passant au ln dans la relation (\*) :

$$r = ae^{-cx} \Leftrightarrow \ln(r) = \ln(a) - cx \Leftrightarrow y = \alpha x + \beta \quad \text{avec } \alpha = -c \text{ et } \beta = \ln(a).$$

On rappelle que la droite de régression linéaire de  $y = (y_1, \dots, y_n)$  par rapport à  $x = (x_1, \dots, x_n)$  a pour équation :

$$y = \frac{s_{x,y}}{s_x^2}(x - \bar{x}) + \bar{y}$$

où :

- $s_{x,y}$  désigne la covariance empirique de la série statistique double  $(x, y)$ ;
  - $\bar{x}$  et  $\bar{y}$  désignent les moyennes des séries statistiques  $x$  et  $y$  respectivement;
  - $s_x^2$  désigne la variance de la série statistique  $x$ .
10. Rappeler les formules mathématiques définissant  $\bar{x}$ ,  $s_x^2$  et  $s_{x,y}$  en fonction de  $n$ ,  $x_1, \dots, x_n$ , et  $y_1, \dots, y_n$ . Rappeler les formules de Koenig et de Koenig-Huygens permettant de reformuler  $s_x^2$  et  $s_{x,y}$ .

Ce sont les indicateurs empiriques :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad ; \quad s_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad ; \quad s_{x,y} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Koenig / Koenig-Huygens donnent :

$$s_x^2 = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - (\bar{x})^2 \quad ; \quad s_{x,y} = \left( \frac{1}{n} \sum_{i=1}^n x_i y_i \right) - \bar{x} \bar{y}$$

NB : après relecture des programmes, Koenig est celle pour la variance et Koenig-Huygens celle pour la covariance ; pour être honnête je ne m'en souvenais plus...

11. **Exprimer  $a$  et  $c$  en fonction de  $\bar{x}$ ,  $\bar{y}$ ,  $s_x^2$  et  $s_{x,y}$ .**

La formule de régression linéaire rappelée par l'énoncé, identifiée à  $y = \alpha x + \beta$ , donne

$$\alpha = \frac{s_{x,y}}{s_x^2} ; \quad \beta = -\bar{x} \frac{s_{x,y}}{s_x^2} + \bar{y}$$

On rappelle par ailleurs que  $\alpha = -c$  et  $\beta = \ln(a)$ , donc  $a = \exp(\beta)$  et  $c = -\alpha$ .

Finalement :

$$a = \exp\left(-\bar{x} \frac{s_{x,y}}{s_x^2} + \bar{y}\right) \quad \text{et} \quad c = -\frac{s_{x,y}}{s_x^2}$$

12. **Recopier et compléter la fonction suivante pour qu'elle renvoie la covariance empirique des séries statistiques  $x$  et  $y$  fournies en argument d'entrée sous forme de tableaux Numpy à une dimension.**

On se souvient (et c'est rappelé dans l'annexe) que les opérations sur les tableaux Numpy agissent composantes par composante. En reprenant l'expression de la covariance de Koenig-Huygens,  $s_{x,y} = \overline{xy} - \bar{x}\bar{y}$ , ce qui s'exprime tout simplement comme :

```
import numpy as np
def covariance(x,y):
    prod = x*y
    return np.mean(prod)-np.mean(x)*np.mean(y)
```

(et pour l'amour du ciel ne calculez pas une moyenne en sommant les éléments par une boucle for et en divisant par n... qui n'est d'ailleurs pas défini dans ce code.)

13. **Recopier et compléter la fonction suivante, prenant en arguments d'entrée les séries statistiques  $(x_1, x_2, \dots, x_n)$  et  $(r_1, r_2, \dots, r_n)$  (sous forme de tableaux Numpy à une dimension  $x$  et  $r$  respectivement), pour qu'elle renvoie des valeurs approchées des paramètres  $a$  et  $c$  qui interviennent dans la relation (\*).**

On compile tout ce qui précède, et on UTILISE LES FONCTIONS DÉJÀ PROGRAMMÉES :

```
import numpy as np
def ajustement_exp(x, r):
    y = np.log(r)
    moy_x = np.mean(x)
    moy_y = np.mean(y)
    cov_xy = covariance(x,y)
    var_x = covariance(x,x) # variance de x ; on peut aussi utiliser np.var(x)
    c = -cov_xy/var_x
    a = np.exp(-moy_x*cov_xy/var_x+moy_y)
    return a, c
```