



**SUJET ZÉRO n°2**

**MATHÉMATIQUES APPLIQUÉES  
VOIE ECG**

**CONCOURS ECRICOME PRÉPA 2023**

# Mathématiques appliquées - Sujet zéro 2

Dans les questions faisant intervenir des instructions en langage Python, on prendra soin d'importer les bibliothèques nécessaires lors de leur première utilisation.

Pour traiter les questions d'informatique, les candidats sont invités à se référer à l'annexe fournie en fin de sujet. Ils ne sont pas limités à l'utilisation des seules fonctions mentionnées dans cette annexe.

## EXERCICE 1

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  $X_2$ .
2. Déterminer la loi conjointe du couple  $(X_1, X_2)$ .
3. En déduire que les variables aléatoires  $X_1$  et  $X_2$  sont indépendantes.
4. Montrer que, pour tout entier  $n \geq 2$  :

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

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  $Y$ .

```
import numpy.random as rd
def simul_Y(p):
    Y = 0
    nb_pile = 0
    while ..... :
        if ..... :
            nb_pile = nb_pile + 1
        else :
            .....
    return Y
```

6. Exprimer  $Y$  en fonction de  $X_1$  et  $X_2$ .
7. En déduire l'espérance de  $Y$ , la variance de  $Y$  et la loi de  $Y$ .

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 pourra faire appel à la fonction `simul_Y` définie à la question 5.
9. Justifier que  $U(\Omega) = \mathbf{N}$  et préciser  $V(\Omega)$ .
10. Pour tout  $k \in \mathbf{N}$ , déterminer  $P(U = k)$ .
11. Montrer que la variable aléatoire  $U + 1$  suit une loi usuelle que l'on reconnaîtra.  
En déduire l'espérance et la variance de  $U$ .
12. Montrer que  $V$  suit la même loi que  $U$ .
13. (a) Montrer que  $U$  et  $V$  sont indépendantes.  
(b) En déduire la covariance du couple  $(Y, U)$ .  
(c) Montrer que le coefficient de corrélation linéaire du couple  $(Y, U)$  est égal à  $\frac{\sqrt{2}}{2}$ .

## EXERCICE 2

### Partie 1

Soit  $f$  la fonction définie sur  $\mathbf{R}$  par :

$$\forall x \in \mathbf{R}, \quad f(x) = x^3 - 3x.$$

On note  $\mathcal{C}$  la courbe représentative de  $f$ .

1. Étudier la parité de  $f$  sur  $\mathbf{R}$ .
2. Déterminer les variations de  $f$  sur  $[0, +\infty[$ , et calculer  $\lim_{x \rightarrow +\infty} f(x)$ .
3. Montrer que  $f$  est convexe sur  $[0, +\infty[$  et justifier que le point d'abscisse 0 est un point d'inflexion de la courbe  $\mathcal{C}$ .
4. On note  $T$  la tangente à la courbe  $\mathcal{C}$  au point d'abscisse 0.  
Déterminer une équation de  $T$  et préciser la position relative de  $\mathcal{C}$  par rapport à  $T$ .
5. Tracer, sur une même figure, l'allure de la courbe  $\mathcal{C}$  (sur  $\mathbf{R}$ ) et de la droite  $T$ .
6. Soit  $a$  un réel.
  - (a) Montrer que, si  $|a| < 2$ , alors l'équation  $x^3 - 3x + a = 0$ , d'inconnue  $x$ , possède exactement trois solutions réelles.
  - (b) Montrer que, si  $|a| > 2$ , alors l'équation  $x^3 - 3x + a = 0$ , d'inconnue  $x$ , possède une unique solution réelle.

### Partie 2

Pour tout réel  $a$ , on considère la matrice :

$$A_a = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a & 3 & 0 \end{pmatrix}.$$

7. (a) Calculer  $A_a^3 - 3A_a + aI_3$ .
- (b) Soient  $\lambda$  un réel tel que  $\lambda^3 - 3\lambda + a = 0$ , et soit  $X = \begin{pmatrix} 1 \\ \lambda \\ \lambda^2 \end{pmatrix}$ .  
Exprimer  $A_a X$  en fonction de  $\lambda$  et de  $X$ .
- (c) Dédurre des deux questions précédentes que pour tout réel  $\lambda$  :

$$\lambda \text{ est valeur propre de } A_a \iff \lambda^3 - 3\lambda + a = 0$$

8. **Dans cette question uniquement**, on suppose  $a = 2$ .

- (a) Déterminer les valeurs propres et une base de chaque sous-espace propre de  $A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 3 & 0 \end{pmatrix}$ .
- (b) La matrice  $A_2$  est-elle diagonalisable ?

9. **Dans cette question uniquement**, on suppose  $|a| > 2$ .

La matrice  $A_a$  est-elle diagonalisable ?

*Indication:* On pourra utiliser le résultat de la question 6.

10. **Dans cette question uniquement**, on suppose  $a \in ]-2, 2[$ .

- (a) Montrer que  $A_a$  est diagonalisable.  
*Indication:* On pourra utiliser le résultat de la question 6.

- (b) On note  $\lambda, \mu, \nu$  les valeurs propres de  $A_a$ . On considère les matrices  $D = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \nu \end{pmatrix}$  et  $P = \begin{pmatrix} 1 & 1 & 1 \\ \lambda & \mu & \nu \\ \lambda^2 & \mu^2 & \nu^2 \end{pmatrix}$ .

Justifier que  $P$  est inversible.

Exprimer  $A_a$  en fonction de  $D$  et  $P$ .

### Partie 3

Pour tout réel  $a$ , on considère l'équation différentielle  $(\mathcal{E}_a)$  suivante, d'inconnue  $y : \mathbf{R} \rightarrow \mathbf{R}$  trois fois dérivable sur  $\mathbf{R}$  :

$$(\mathcal{E}_a) : \quad y''' - 3y' + ay = 0.$$

11. Dans cette question uniquement, on suppose que  $a = 0$ .

- (a) Soit  $y$  une solution de l'équation  $(\mathcal{E}_0) : y''' - 3y' = 0$ . Déterminer la forme générale de la fonction  $y'$ .
- (b) En déduire l'ensemble des solutions de l'équation  $(\mathcal{E}_0)$ .

Dans la suite de l'exercice, pour toute fonction  $y$  trois fois dérivable sur  $\mathbf{R}$ , on note  $Y = \begin{pmatrix} y \\ y' \\ y'' \end{pmatrix}$  et  $Y' = \begin{pmatrix} y' \\ y'' \\ y''' \end{pmatrix}$ .

12. Montrer que, pour tout réel  $a$ ,  $y$  est solution de l'équation différentielle  $(\mathcal{E}_a)$  si et seulement si  $Y' = A_a Y$ .

13. Dans cette question uniquement, on suppose  $a \in ]-2, 2[$ .

- (a) Montrer qu'une fonction  $y : \mathbf{R} \rightarrow \mathbf{R}$  est solution de l'équation différentielle  $(\mathcal{E}_a)$  si et seulement si  $Z' = DZ$ , où  $Z = P^{-1}Y$ , où  $D$  et  $P$  sont définies à la question 10.(b).
- (b) Montrer que les solutions de l'équation différentielle  $(\mathcal{E}_a)$  sont les fonctions  $y$  définies par une expression de la forme :

$$\forall x \in \mathbf{R}, \quad y(x) = \alpha_1 e^{\lambda x} + \alpha_2 e^{\mu x} + \alpha_3 e^{\nu x},$$

où  $\alpha_1, \alpha_2, \alpha_3$  sont des réels.

- (c) Vérifier en particulier que les résultats de 11(b) et 13(b) sont cohérents.

## EXERCICE 3

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

```
UPDATE annonce
SET prix_occasion = prix_neuf
FROM vehicule
WHERE vehicule.id_vehicule = annonce.id_vehicule
AND vehicule.prix_neuf < annonce.prix_occasion
```

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`.

## 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.

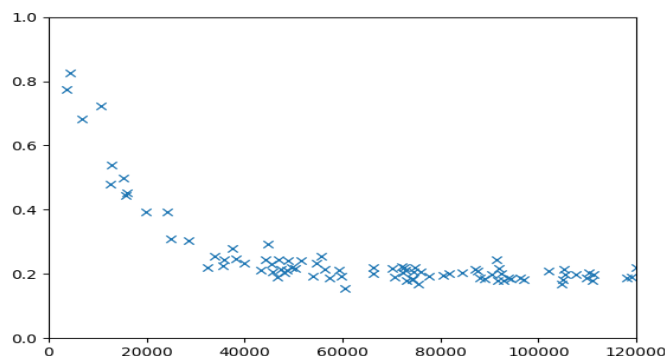
Ces informations sont stockées sous forme de tableaux Numpy à une dimension et à  $n$  éléments :

- 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).
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  $n - 1$ .
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 ?

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)  $\rho = 0.917$ .

**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  $c$ .

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

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

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 (\*).

```
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 = ..... # variance de x
    c = .....
    a = .....
    return a, c
```

## Annexe - Fonctions Python utiles

### La bibliothèque `numpy`.

- Exemple d'importation : `import numpy as np`.
- Les opérations `+`, `-`, `*`, `/`, `**`, lorsqu'elles sont possibles, peuvent être réalisées entre deux tableaux Numpy de dimensions compatibles et agissent alors **coefficient par coefficient**.
- Les fonctions `np.log` (logarithme népérien) et `np.exp` (fonction exponentielle) s'appliquent à une quantité numérique ou à un tableau Numpy de nombres. Dans ce dernier cas, les fonctions sont appliquées à chaque élément du tableau donné en argument d'entrée.
- La fonction `np.mean`, prend en argument d'entrée un tableau Numpy de nombres, et renvoie la moyenne des éléments du tableau.

### La bibliothèque `matplotlib.pyplot`.

- Exemple d'importation : `import matplotlib.pyplot as plt`.
- La fonction `plt.plot` prend en arguments d'entrée deux tableaux Numpy `x` et `y` à une ligne et de même longueur, et renvoie une figure constituée de la ligne brisée joignant les points du plan de coordonnées  $(x_i, y_i)$ , où  $x_i$  et  $y_i$  sont respectivement les coefficients des tableaux `x` et `y`.
- La fonction `plt.scatter` prend en arguments d'entrée deux tableaux Numpy `x` et `y` à une dimension et de même longueur, et produit un nuage de points composé de tous les points du plan de coordonnées  $(x_i, y_i)$ , où  $x_i$  et  $y_i$  sont respectivement les coefficients des tableaux `x` et `y`.
- La fonction `plt.show`, employée sans argument d'entrée, permet l'affichage d'une figure préalablement tracée, par exemple avec les fonctions `plt.plot` ou `plt.scatter`.

### Le module `numpy.random`.

- Exemple d'importation : `import numpy.random as rd`.
- La fonction `rd.random`, appelée sans argument d'entrée, renvoie une réalisation aléatoire de la loi uniforme sur l'intervalle  $[0, 1[$ . Il est également possible de spécifier les dimensions d'un tableau Numpy en argument d'entrée pour obtenir un tableau dont les coefficients sont des réalisations indépendantes de la loi uniforme sur  $[0, 1[$ .
- La fonction `rd.randint` prend deux entiers  $n$  et  $p$  (avec  $p > n$ ) en arguments d'entrée et renvoie une réalisation aléatoire de la loi uniforme discrète sur  $\llbracket n, p - 1 \rrbracket$ .