DM4

À rendre pour le 06/01

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

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 s'efforcera de faire appel à la fonction `simul_Y` définie à la question 5.
9. Justifier que $U(\Omega) = \mathbb{N}$ et préciser $V(\Omega)$.
10. Pour tout $k \in \mathbb{N}$, déterminer $\mathbf{P}(U = k)$.
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 $\dfrac{\sqrt{2}}{2}$.

Exercice 2 : commutant d'une matrice

On note I_3 la matrice identité de $\mathscr{M}_3(\mathbb{R})$, 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 λ de A sont les solutions de l'équation : $\lambda^3 - 9\lambda^2 - 27\lambda + 53 = 0$.

   (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).

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

   (d) Montrer que A admet trois valeurs propres, que l'on ne cherchera pas à calculer et que l'on notera λ_1, λ_2 et λ_3 avec $\lambda_1 < \lambda_2 < \lambda_3$.

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

2. On s'intéresse ici au *commutant* de A, c'est-à-dire l'ensemble des matrices qui commutent avec A.
   On pose $\mathscr{C}_A = \{M \in \mathscr{M}_3(\mathbb{R}) \mid AM = MA\}$; et de même $\mathscr{C}_D = \{M \in \mathscr{M}_3(\mathbb{R}) \mid DM = MD\}$

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

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

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

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

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

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

   (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_3, A, A^2) est une famille libre, puis que c'est une base de $\mathscr{C}_A$.

Exercice 3 : bases de données et statistiques

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.*

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.
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`.

```sql
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.

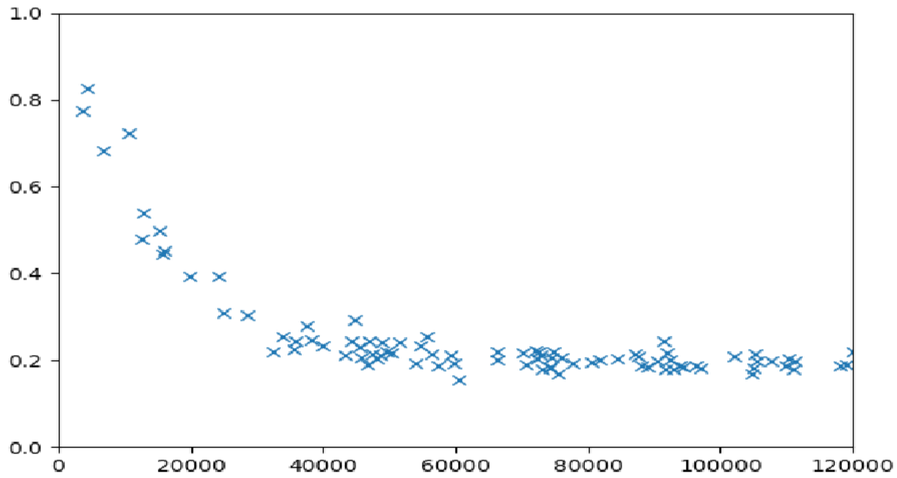
Soit $n \geqslant 1$ un entier. On suppose dans cette partie qu'on dispose des informations suivantes concernant n annonces de véhicules d'occasion.
Ces informations sont stockées sous forme de tableaux `Numpy` à une dimension et à n éléments :

- 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 $n \geqslant 1$ 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 :

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

où a et c sont des réels que l'on cherche à déterminer, avec $a > 0$. On dispose pour cela de deux séries statistiques $(x_1, x_2, \ldots, x_n)$ et $(r_1, r_2, \ldots, r_n)$ 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 α et β sont des paramètres réels que l'on exprimera en fonction de a et c.

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

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

où :

- $s_{x,y}$ désigne la covariance empirique de la série statistique double (x, y);
- $\overline{x}$ et $\overline{y}$ 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 $\overline{x}, s_x^2$ et $s_{x,y}$ en fonction de n, $x_1, \ldots, x_n$, et $y_1, \ldots, y_n$. 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 $\overline{x}, \overline{y}, s_x^2$ 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 $(x_1, x_2, \ldots, x_n)$ et $(r_1, r_2, \ldots, r_n)$ (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 $[\![n, p-1]\!]$.