DS5 (version A)

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies.
*Les candidat·es sont invité·es à **encadrer** dans la mesure du possible les résultats de leurs calculs.*
*Aucun document n'est autorisé. **L'utilisation de toute calculatrice et de tout matériel électronique** est **interdite**. Seule l'utilisation d'une règle graduée est autorisée.*
Si au cours de l'épreuve, un candidat ou une candidate repère ce qui lui semble être une erreur d'énoncé, il la signalera sur sa copie et poursuivra sa composition en expliquant les raisons des initiatives qu'il sera amené à prendre.

On suppose, et c'est valable pour toute l'épreuve, que les librairies suivantes sont importées sous leurs alias habituels :

- `import numpy as np`
- `import numpy.linalg as al`
- `import numpy.random as rd`
- `import matplotlib.pyplot as plt`

Problème 1

Partie I : Etude d'une matrice A

On considère la matrice carrée $A = \begin{pmatrix} 2 & -2 & 2 \\ 1 & 1 & 2 \\ -2 & 0 & -3 \end{pmatrix}$.

1. ***a)*** On exécute le code **Python** suivant :

```
import numpy as np
import numpy.linalg as al
A = np.array([[2,-2,2],[1,1,2],[-2,0,-3]])
print(al.matrix_power(A,3))
```

et on obtient l'affichage :

```
[[ 2 -2  2]
 [ 1  1  2]
 [-2  0 -3]]
```

Traduire ce résultat par une égalité entre deux matrices.

b) En déduire les valeurs propres possibles de A.

2. ***a)*** Déterminer $\mathrm{Sp}(A)$ et une base de chacun des sous-espaces propres de A.

b) Démontrer qu'il existe une matrice $P \in \mathscr{M}_3(\mathbb{R})$ inversible, dont la première ligne est $\begin{pmatrix} 2 & 3 & -2 \end{pmatrix}$, et une matrice $D \in \mathscr{M}_3(\mathbb{R})$ diagonale, dont les coefficients diagonaux sont dans l'ordre croissant, qui vérifient $A = PDP^{-1}$. On explicitera les matrices P et D.

Partie II : Exponentielle d'une matrice carrée

Si $(a_n), (b_n), (c_n), (d_n), (e_n), (f_n), (g_n), (h_n), (i_n)$ désignent neuf suites convergentes, de limites respectives $a, b, c, d, e, f, g, h, i$, et si $(M_n)_{n\in\mathbb{N}}$ est une suite de matrices de $\mathscr{M}_3(\mathbb{R})$ définie par

$$\forall n \in \mathbb{N}, \quad M_n = \begin{pmatrix} a_n & b_n & c_n \\ d_n & e_n & f_n \\ g_n & h_n & i_n \end{pmatrix},$$

on dit que la suite de matrices $(M_n)_{n\in\mathbb{N}}$ *admet une limite coefficient par coefficient*, et on le note

$$\lim_{n\to+\infty} M_n = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \qquad \text{ou} \qquad M_n \underset{n\to+\infty}{\longrightarrow} \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}.$$

Si $M \in \mathscr{M}_3(\mathbb{R})$, on pose, pour tout entier naturel n,

$$S_n(M) = \sum_{k=0}^{n} \frac{1}{k!} M^k \in \mathscr{M}_3(\mathbb{R}).$$

Lorsque $(S_n(M))_{n\in\mathbb{N}}$ admet une limite coefficient par coefficient, on note e^M cette limite.

3. *Deux résultats théoriques.* On utilisera les notations du préambule de la partie II pour les preuves.

a) Soit $M \in \mathscr{M}_3(\mathbb{R})$ et soit (α_n) une suite réelle convergente, de limite ℓ. Montrer que la suite de matrices $(\alpha_n M)$ admet une limite coefficient par coefficient et que

$$\lim_{n\to+\infty} \alpha_n M = \ell M$$

b) Soient (M_n) et (M'_n) deux suites de matrices de $\mathscr{M}_3(\mathbb{R})$ qui admettent chacune une limite coefficient par coefficient. On note $\lim\limits_{n\to+\infty} M_n = M$ et $\lim\limits_{n\to+\infty} M'_n = M'$. Montrer que les suites de matrices $(M_n + M'_n)$ et $(M_n M'_n)$ admettent chacune une limite coefficient par coefficient et que

$$\lim_{n\to+\infty}(M_n + M'_n) = M + M' \qquad \text{et} \qquad \lim_{n\to+\infty}(M_n M'_n) = MM'$$

Les candidat·es devront référer précisément à ces questions lorsque ces résultats seront utilisés.

4. Montrer que, si $D = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$, alors e^D existe et vaut $\mathrm{e}^D = \begin{pmatrix} \mathrm{e}^a & 0 & 0 \\ 0 & \mathrm{e}^b & 0 \\ 0 & 0 & \mathrm{e}^c \end{pmatrix}$.

Ainsi, on a montré que l'exponentielle d'une matrice diagonale est une matrice diagonale.

5. Dans cette question uniquement, la matrice M est donnée par $M = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$.

a) Calculer M^2 et M^3. En déduire la matrice M^k pour tout entier naturel k.

b) Donner, pour tout entier n supérieur ou égal à 2, l'expression de $S_n(M)$. En déduire l'existence et l'expression de la matrice e^M.

6. Dans cette question uniquement, la matrice M est donnée par $M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$.

a) Calculer M^2 et M^3 en fonction de M.

b) Soit $k \in \mathbb{N}^*$. Conjecturer une formule simple pour l'expression de M^k puis la démontrer par récurrence.

c) Soit $n \in \mathbb{N}$. Montrer que :

$$S_n(M) = I + \frac{1}{3}\left(\sum_{k=0}^{n} \frac{3^k}{k!} - 1\right) M$$

d) En déduire que e^M existe et que :

$$\mathrm{e}^M = I + \frac{\mathrm{e}^3 - 1}{3} M$$

7. Dans cette question, on considère la matrice A de la Partie I et on réutilise les notations de la question ***2.b)***. On fixe un réel t.

a) Montrer que, pour tout $n \in \mathbb{N}$:

$$S_n(tA) = P S_n(tD) P^{-1}$$

b) Conclure que e^{tA} existe et en donner une expression sous la forme $\mathrm{e}^{tA} = P\Delta(t)P^{-1}$.
On explicitera la matrice $\Delta(t)$ sous forme de tableau matriciel en fonction de t.

En généralisant ce résultat, on montre alors que l'exponentielle d'une matrice diagonalisable est une matrice diagonalisable (on ne demande pas de le faire).

Partie III : Etude d'un système différentiel linéaire

On considère le système différentiel linéaire suivant :

$$(S) : \left\{ \begin{array}{rcrcrcr} x'(t) & = & 2x(t) & - & 2y(t) & + & 2z(t) \\ y'(t) & = & x(t) & + & y(t) & + & 2z(t) \\ z'(t) & = & -2x(t) & & & - & 3z(t) \end{array} \right.$$

où les inconnues x, y, z sont des fonctions de classe $\mathcal{C}^1$ sur $\mathbb{R}$. Pour tout $t \in \mathbb{R}$, on pose $X(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$.

On a alors :

$$(S) \iff X' = AX$$

où A est la matrice étudiée dans la partie I.

8. Déterminer l'ensemble des états d'équilibre du système différentiel linéaire (S).

9. Soient X et Y deux solutions de (S). On suppose qu'il existe un instant $t_0 \in \mathbb{R}$ tel que $X(t_0) = Y(t_0)$. Que peut-on en déduire sur X et Y ?

10. Justifier que l'ensemble des solutions de (S) est :

$$\left\{ t \mapsto \alpha \mathrm{e}^{-t} U_{-1} + \beta U_0 + \gamma \mathrm{e}^t U_1 \mid (\alpha, \beta, \gamma) \in \mathbb{R}^3 \right\}$$

où $U_{-1} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix}$, $U_0 = \begin{pmatrix} 3 \\ 1 \\ -2 \end{pmatrix}$ et $U_1 = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$.

11. On considère dans cette question deux problèmes de Cauchy :

$$(\mathcal{P}_1) : \begin{cases} X' = AX \\ X(0) = \begin{pmatrix} 9 \\ 4 \\ -8 \end{pmatrix} \end{cases} \qquad \text{et} \qquad (\mathcal{P}_2) : \begin{cases} X' = AX \\ X(0) = \begin{pmatrix} 3 \\ 2 \\ -3 \end{pmatrix} \end{cases}$$

a) ***i)*** Déterminer l'unique solution du problème de Cauchy $(\mathcal{P}_1)$, que l'on notera X_1.

ii) Montrer que la trajectoire associée à la solution X_1 est convergente. Expliciter le point limite (ℓ_1, ℓ_2, ℓ_3). Quelle propriété possède ce point limite vis-à-vis du système différentiel linéaire (S) ?

b) ***i)*** Déterminer l'unique solution du problème de Cauchy $(\mathcal{P}_2)$, que l'on notera X_2.

ii) Montrer que la trajectoire associée à la solution X_2 est divergente.

c) On a représenté page suivante les tracés de 4 solutions du système différentiel linéaire (S). Dire quels sont les tracés associés aux solutions X_1 et X_2 étudiées ci-dessus. On justifiera les réponses.

12. Dans cette question, on souhaite faire le lien entre la résolution d'un système différentiel linéaire (homogène) et l'exponentielle de matrice introduite à la partie II.

a) On fixe $(\alpha, \beta, \gamma) \in \mathbb{R}^3$ et on considère la solution de (S) :

$$X : t \mapsto \alpha \mathrm{e}^{-t} U_{-1} + \beta U_0 + \gamma \mathrm{e}^t U_1$$

On pose $C = P \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \in \mathscr{M}_{3,1}(\mathbb{R})$ (où P est définie à la question ***2.b)***).

Montrer que, pour tout $t \in \mathbb{R}$, $X(t) = \mathrm{e}^{tA} C$.

b) Commenter le résultat obtenu à la question précédente, au regard des résultats du cours sur les équations différentielles linéaires du premier ordre à coefficient constant.

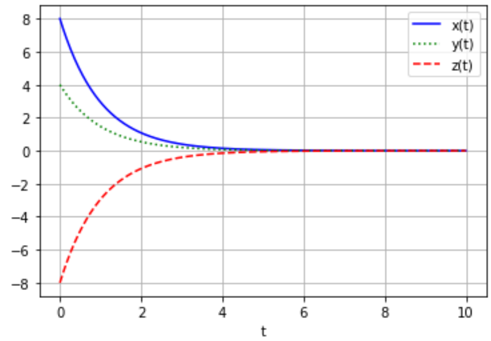

Fig. 1 Tracé 1

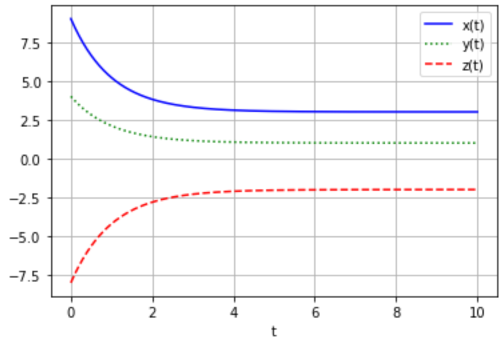

Fig. 2 Tracé 2

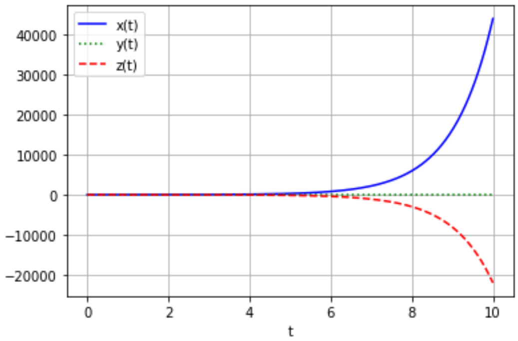

Fig. 3 Tracé 3

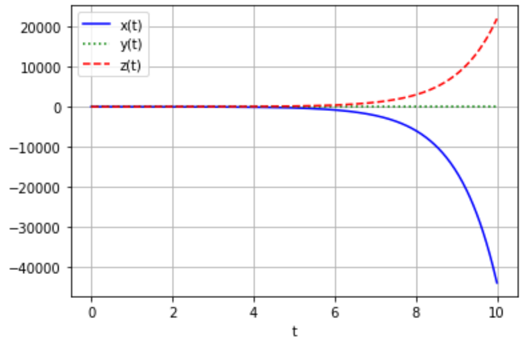

Fig. 4 Tracé 4

Problème 2

Toutes les variables aléatoires de cet exercice sont supposées définies sur le même espace probabilisé $(\Omega, \mathscr{A}, \mathbb{P})$.
Soit n un entier naturel non nul.
La population active d'un territoire est divisée en n catégories socioprofessionnelles, numérotées de 1 à n.
Pour tout entier i compris entre 1 et n, on note X_i la variable aléatoire égale au revenu mensuel, en milliers d'euros, d'un individu choisi au hasard avec équiprobabilité au sein de la catégorie socioprofessionnelle numéro i. On suppose que la variable aléatoire X_i admet pour densité la fonction f_i définie sur $\mathbb{R}$ par :

$$\forall x \in \mathbb{R},\ f_i(x) = \begin{cases} \dfrac{i}{x^{i+1}} & \text{si } x \geqslant 1, \\ 0 & \text{si } x < 1. \end{cases}$$

On note F_i la fonction de répartition de X_i.

Partie I.

1. Vérifier que, pour tout entier i compris entre 1 et n, la fonction f_i est une densité de probabilité.

2. *a)* Déterminer les entiers naturels i, compris entre 1 et n, tels que X_i admet une espérance, et déterminer alors l'espérance de X_i.

b) En justifiant la réponse, classer les numéros de catégorie socioprofessionnelle dans l'ordre de leur revenu mensuel moyen, du moins élevé au plus élevé.

On ne considérera que les entiers i pour lesquels l'espérance de X_i est bien définie.

3. Montrer que, pour tout entier i compris entre 1 et n et pour tout réel x :

$$F_i(x) = \begin{cases} 1 - \dfrac{1}{x^i} & \text{si } x \geqslant 1, \\ 0 & \text{si } x < 1. \end{cases} \tag{1}$$

4. Soit U une variable aléatoire à densité de loi uniforme sur $]0,1[$.

Soit i un entier compris entre 1 et n. On pose $V_i = \dfrac{1}{U^{1/i}}$.

a) Montrer que V_i suit la même loi que X_i.

b) Écrire une fonction en langage **Python** nommée `simulX`, prenant en argument d'entrée l'entier i, et renvoyant une simulation de la variable aléatoire X_i.

Partie II.

Soit p un réel de $]0,1[$. On choisit un individu au hasard dans la population et on note Y la variable aléatoire égale au numéro de la catégorie socioprofessionnelle à laquelle cet individu appartient. On suppose que la variable aléatoire $Y-1$ suit la loi binomiale $\mathcal{B}(n-1,p)$.

5. En utilisant uniquement la fonction `random` du module `numpy.random`, écrire une fonction en langage **Python** nommée `simulY`, prenant en arguments d'entrée les paramètres n et p, et renvoyant une simulation de la variable aléatoire Y.

6. Recopier et compléter la fonction, en langage **Python**, nommée `loiY`, prenant en arguments d'entrée les paramètres n et p, et renvoyant une liste $(p_1, \ldots, p_n)$ où pour i compris entre 1 et n, p_i est une valeur approchée de $\mathbb{P}(Y = i)$.

```
def loiY(n,p):
    N = 10000
    loi = [0] * n
    for k in ------ :
        y = simulY(n,p)
        loi[--] --
    return loi
```

7. Écrire une fonction **Python**, prenant en arguments d'entrée les paramètres n et p, permettant d'afficher un diagramme en bâtons représentant approximativement la loi de Y. On représentera les valeurs de Y en abscisses et les probabilités correspondantes en ordonnées. On pourra utiliser les fonctions définies dans les questions précédentes, et l'annexe fournie en fin de sujet.

8. Dans cette question, on suppose que chaque profession est identifiée de manière unique par un numéro appelé <u>code PCS</u>.

Par ailleurs, les différentes professions sont regroupées dans six grandes catégories socioprofessionnelles, que l'on identifie par un entier de 1 à 6.

On dispose d'une base de données comportant trois tables nommées `individu`, `departement` et `profession`, décrites ci-dessous.

- La table `individu` contient des informations sur tous les individus de la population active française. Chaque entrée correspond donc à un individu. La table comporte les attributs suivants.
  - ◇ `i_nom` (de type `TEXT`) : le nom de l'individu.
  - ◇ `i_prenom` (de type `TEXT`) : le prénom de l'individu.
  - ◇ `i_departement` (de type `INTEGER`) : le numéro de département où réside l'individu.
  - ◇ `i_insee` (de type `INTEGER`) : le numéro INSEE (ou <u>numéro de sécurité sociale</u>) de l'individu.
  - ◇ `i_code_profession` (de type `INTEGER`) : le code PCS identifiant la profession actuelle de l'individu.

- La table `departement` contient des informations sur les départements français. Chaque entrée correspond donc à un département. La table comporte les attributs suivants.
  - ◇ `d_numero` (de type `INTEGER`) : le numéro du département.
  - ◇ `d_nom` (de type `TEXT`) : le nom du département.
  - ◇ `d_population` (de type `INTEGER`) : le nombre d'habitants vivant dans le département.
- La table `profession` contient des informations sur toutes les professions recensées dans la base de données. Chaque entrée correspond donc à une profession différente. La table comporte les attributs suivants.
  - ◇ `p_pcs` (de type `INTEGER`) : le code PCS permettant d'identifier la profession.
  - ◇ `p_categorie` (de type `INTEGER`) : le numéro de la catégorie socioprofessionnelle (de 1 à 6) à laquelle la profession se trouve rattachée.
  - ◇ `p_intitule` (de type `TEXT`) : l'intitulé de la profession (par exemple, `chirurgien dentiste`).

a) Que doit vérifier la clé primaire d'une table dans une base de données ?

b) Pour chacune des trois tables de la base de données de cet exemple, indiquer sans justification un attribut pouvant jouer le rôle de clé primaire.

c) Dresser un schéma relationnel de la base de données décrite ci-dessus, en mettant en évidence les relations qui existent entre les tables et les attributs permettant d'établir ces relations.
On s'assurera que chaque table est reliée à au moins l'une des deux autres tables.

d) Écrire une requête SQL renvoyant tous les codes PCS des professions exercées dans le département de l'Eure-et-Loir (numéro 28). Chaque code PCS ne pourra apparaître qu'une seule fois.
On pourra utiliser la commande `DISTINCT` décrite dans l'annexe fournie en fin de sujet.

e) Écrire une requête SQL permettant d'obtenir le numéro de catégorie socioprofessionnelle (entre 1 et 6) de chaque individu.
La requête devra renvoyer deux attributs pour chaque individu : son numéro INSEE, et la catégorie socioprofessionnelle à laquelle est rattachée sa profession.

Partie III.

Soit p un réel de $]0,1[$.
Un institut réalise un sondage selon le protocole suivant :

- On choisit une catégorie socioprofessionnelle de manière aléatoire (mais sans équiprobabilité), et on note Y la variable aléatoire égale au numéro de la catégorie choisie.
  Comme dans la partie **II**, on suppose que $Y-1$ suit la loi binomiale $\mathcal{B}(n-1,p)$.
- On sélectionne alors un individu au hasard (avec équiprobabilité) dans la catégorie socioprofessionnelle choisie à l'étape précédente, et on note Z_n la variable aléatoire égale à son revenu mensuel, en milliers d'euros.

Enfin, on rappelle que, pour tout entier i compris entre 1 et n, la fonction de répartition F_i de la variable aléatoire X_i est donnée par l'égalité (1) montrée à la question ***3***.
On note G_n la fonction de répartition de la variable aléatoire Z_n.

9. Expliciter $G_n(x)$ pour tout réel x strictement inférieur à 1.

10. Soit x un réel supérieur ou égal à 1.

a) Justifier que pour tout entier i compris entre 1 et n :

$$\mathbb{P}_{[Y=i]}(Z_n \leqslant x) = F_i(x).$$

b) Montrer que :

$$G_n(x) = \sum_{k=0}^{n-1} F_{k+1}(x)\binom{n-1}{k}p^k(1-p)^{n-1-k}.$$

c) En déduire que :

$$G_n(x) = 1 - \frac{(p+(1-p)x)^{n-1}}{x^n}.$$

11. Justifier que Z_n est une variable aléatoire à densité.

12. En utilisant les fonctions `simulX` et `simulY` définies aux questions ***4.b)*** et ***5***, écrire une fonction en langage **Python** nommée `sondage`, prenant en arguments d'entrée les paramètres n et p, et renvoyant une simulation de la variable aléatoire Z_n.

13. Dans cette question uniquement, on suppose que $p = \dfrac{1}{n}$.

a) Montrer que, pour tout x réel :

$$G_n(x) = \begin{cases} 1 - \dfrac{1}{x}\left(1 - \dfrac{x-1}{nx}\right)^{n-1} & \text{si } x \geqslant 1, \\ 0 & \text{si } x < 1. \end{cases}$$

b) Calculer, pour tout $x \in \mathbb{R}$, $\lim\limits_{n \to +\infty} G_n(x)$.

Annexe A - Fonctions Python utiles

Manipulation de listes

On suppose que `L` désigne une liste à n éléments.

- L'opérateur de concaténation `+`, appliqué entre deux listes, renvoie la liste obtenue en plaçant les éléments de la seconde liste à la suite de ceux de la première liste.
  Par exemple, `[1, 2, 5]+[4, 3]` renvoie la liste `[1, 2, 5, 4, 3]`.
- L'opérateur `*`, appliqué entre une liste `L` et un entier `n`, renvoie la liste obtenue en concaténant `n` fois la liste `L` avec elle-même.
  Par exemple, `[1, 4, 2]*3` renvoie la liste `[1, 4, 2, 1, 4, 2, 1, 4, 2]`.
- La fonction `len` prend en argument d'entrée une liste et renvoie le nombre d'éléments dans cette liste.
- La commande `L.append(x)` permet d'inclure l'élément `x` à la fin de la liste `L`.

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.sqrt` (racine carrée), `np.abs` (valeur absolue), `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.linspace`, prenant en arguments d'entrée deux flottants `a` et `b` et un entier `n`, renvoie un tableau Numpy contenant `n` éléments régulièrement espacés entre `a` et `b` inclus.
  Exemple :

```
tableau = np.linspace(1.5, 4.5, 7)
print(tableau)
```

`> > > array([1.5, 2., 2.5, 3., 3.5, 4., 4.5])`

La bibliothèque matplotlib.pyplot

- Exemple d'importation : `import matplotlib.pyplot as plt`.
- La fonction `plt.plot` prend en arguments d'entrée deux listes `x` et `y` de même longueur ou 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.bar` prend en arguments d'entrée deux listes `x` et `y` de même longueur ou deux tableaux Numpy `x` et `y` à une ligne et de même longueur, et produit un diagramme en bâtons, les coefficients de `x` indiquant les abscisses auxquels sont centrés les bâtons, et les coefficients de `y` déterminant la hauteur de chaque bâton en ordonnée.
- 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.bar`.

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 en entrée deux entiers n et p (avec $p > n$) et renvoie une réalisation aléatoire de la loi uniforme discrète sur $[\![n, p-1]\!]$

Annexe B - Commande SQL utile

La commande `SELECT DISTINCT`.

La commande SQL `SELECT DISTINCT`, utilisée à la place de la commande `SELECT`, permet de ne retenir qu'une occurrence de chaque valeur dans une colonne donnée, même si cette colonne comporte des valeurs qui se répètent.
Exemple : Une table `etablissement` contient les données suivantes concernant plusieurs établissements scolaires et la ville où ils se situent.

id	nom	ville
1	Edouart Herriot	Livry-Gargan
2	Diderot	Lyon
3	Edouard Herriot	Lyon
4	Louise Michel	Champigny-sur-Marne
5	Diderot	Paris

La requête

```
SELECT DISTINCT nom
FROM etablissement;
```

permet d'obtenir les différents noms d'établissements, sans répétition :

```
Edouard Herriot
Diderot
Louise Michel
```

La commande `SELECT DISTINCT` peut éventuellement être combinée avec la commande `WHERE` pour filtrer une partie des entrées.