Concours Blanc n°2 - Sujet B

Mercredi 28 Février
Durée : 4 heures

Dans tout le sujet, on suppose déjà importées sous leur alias habituels les bibliothèques Python usuelles.

```python
import numpy as np
import numpy.random as rd
import numpy.linalg as al
import matplotlib.pyplot as plt
import pandas as pd
```

Exercice 1

Soient n et m deux entiers supérieurs ou égaux à 2 fixés. On considère deux variables aléatoires X et Y telles que $X(\Omega) \subset [\![0, n]\!]$ et $Y(\Omega) \subset [\![0, m]\!]$ et le couple $Z = (X, Y)$.
Pour $(i, j) \in [\![0, n]\!] \times [\![0, m]\!]$, on note $p_{i,j} = P([X = i] \cap [Y = j])$ et on introduit les fonctions G_X, G_Y définies sur $\mathbb{R}$ par

$$G_X(x) = \sum_{i=0}^{n} P(X = i)x^i, \quad G_Y(x) = \sum_{i=0}^{m} P(Y = i)x^i,$$

et la fonction G_Z définie sur $\mathbb{R}^2$ par

$$G_Z(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{i,j} x^i y^j.$$

(1) Calculer $G_Z(1, 1)$. Exprimer $E(X)$, $E(Y)$, $E(XY)$ et $\mathrm{cov}(X, Y)$ à l'aide des dérivées partielles d'ordres 1 et 2 de G_Z évaluées en $(1, 1)$.

(2) On considère une fonction polynomiale $f : (x, y) \longmapsto \displaystyle\sum_{i=0}^{n} \sum_{j=0}^{m} a_{i,j} x^i y^j$. Montrer que,

$$\left(\forall (x, y) \in \mathbb{R}^2, \quad f(x, y) = 0\right) \Longleftrightarrow \left(\forall (i, j) \in [\![0, n]\!] \times [\![0, m]\!], \quad a_{i,j} = 0\right).$$

(3) Déduire de la question précédente que X et Y sont indépendantes si et seulement si, pour tout $(x, y) \in \mathbb{R}^2$, on a $G_Z(x, y) = G_X(x)G_Y(y)$.

(4) Une urne contient des jetons portant chacun une des trois lettres A, B ou C. La proportion des jetons A (resp. B, C) dans l'urne est égale à $p \in]0, 1[$ (resp. $q, r \in]0, 1[$) avec $p + q + r = 1$. On effectue n tirages (avec $n \in \mathbb{N}^*$) avec remise dans cette urne et on note X (resp. Y) la variable aléatoire correspondant au nombre de jetons A (resp. B) piochés.

(a) Déterminer la loi de X et celle de Y puis les expressions de G_X et G_Y.
(b) Déterminer la loi de Z puis l'expression de G_Z.
(c) Les variables X et Y sont-elles indépendantes ?
(d) Déterminer $\mathrm{cov}(X, Y)$. Le résultat était-il prévisible ?

Exercice 2

On admet le résultat suivant, noté $(\star)$:

Résultat admis

Si $(a_n)_{n\in\mathbb{N}^*}$ est une suite convergente telle que $\lim\limits_{n\to+\infty} a_n = \ell$, alors $\lim\limits_{n\to+\infty} \frac{1}{n}\sum\limits_{k=1}^{n} a_k = \ell.$ $(\star)$

On considère désormais la suite $(u_n)_{n\in\mathbb{N}^*}$ telle que : $u_1 = 1$ et, pour tout $n \in \mathbb{N}^*$, $u_{n+1} = u_n + \frac{1}{u_n^2}$.

(1) (a) Montrer que pour tout $n \in \mathbb{N}^*$, u_n est bien défini et $u_n > 0$.
(b) Étudier la monotonie de la suite (u_n).
(c) Montrer que $\lim\limits_{n\to+\infty} u_n = +\infty$.

(2) Soit β un réel non nul.

(a) Établir l'égalité suivante : $\forall n \in \mathbb{N}^*,\ u_{n+1}^{\beta} - u_n^{\beta} = \left(\left(1 + \frac{1}{u_n^3}\right)^{\beta} - 1\right) \times u_n^{\beta}$.

(b) Rappeler un équivalent de $(1+x)^{\beta} - 1$ au voisinage de 0.
(c) En déduire que : $u_{n+1}^{\beta} - u_n^{\beta} \sim \beta u_n^{\beta-3}$, $n \to +\infty$.
(d) En déduire que la suite $\left(u_{n+1}^{\beta} - u_n^{\beta}\right)_{n\in\mathbb{N}^*}$ admet une limite finie non nulle **si et seulement si** $\beta = 3$.

(3) (a) Pour tout $n \in \mathbb{N}^*$, on pose : $a_n = u_{n+1}^3 - u_n^3$.
En utilisant le résultat $(\star)$ admis, montrer que : $u_n \sim (3n)^{1/3}$, $n \to +\infty$.
(b) En déduire la nature de la série $\sum\limits_{n\geq 1} \frac{1}{u_n}$.

On souhaite à présent démontrer le résultat $(\star)$ dans le cas particulier où la suite $(a_n)_{n\in\mathbb{N}^*}$ est croissante.
On suppose donc dans la suite de l'exercice que $(a_n)_{n\in\mathbb{N}^*}$ est une suite croissante qui converge vers un réel ℓ.

Pour tout entier $n \in \mathbb{N}^*$, on pose : $b_n = \frac{1}{n}\sum\limits_{k=1}^{n} a_k$.

On veut montrer que la suite $(b_n)_{n\in\mathbb{N}^*}$ converge également vers ℓ.

(4) (a) Établir la relation suivante : $\forall n \in \mathbb{N}^*,\ b_{n+1} - b_n = \frac{1}{n(n+1)}\left(n a_{n+1} - \sum\limits_{k=1}^{n} a_k\right)$.

(b) En déduire que la suite (b_n) est croissante.
(c) Justifier que, pour tout $n \in \mathbb{N}^*$, $a_n \leq \ell$.
En déduire que (b_n) converge vers un réel ℓ' qui vérifie $\ell' \leq \ell$.
(d) Montrer que, pour tout $n \in \mathbb{N}^*$, $2b_{2n} - b_n = \frac{1}{n}\sum\limits_{k=n+1}^{2n} a_k$.

En déduire l'inégalité : $2b_{2n} - b_n \geq a_n$.
(e) Déduire des questions précédentes que $\ell' = \ell$ et qu'ainsi, la suite (b_n) converge vers ℓ.

(5) Python.
On rappelle que la commande `np.cumsum(T)` (du package `numpy`) renvoie une liste contenant les sommes cumulées de la liste `T`. Par exemple : `np.cumsum([1,4,6,3])` renvoie `[1,5,11,14]`).

(a) Dans la fonction python suivante, le paramètre `a` est une liste contenant les premiers termes de la suite (a_n). Compléter cette fonction pour qu'elle renvoie les premiers termes de la suite (b_n).

```
def suite_b(a) :
    S = np.cumsum(.........)
    b = [ ................... for i in range(len(S))]
    return b
```

(b) En utilisant le résultat $(\star)$, déterminer quelle valeur (approximative) va s'afficher à l'exécution des commandes suivantes :

```
a = np.array([ 2-1/n**2 for n in range(1,1001) ])
b = suite_b(a)
print(b[999])
```

Exercice 3

Pour tout entier naturel n non nul et pour tout réel x, on pose :

$$A_n(x) = \begin{pmatrix} 1 & x/n \\ x/n & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{et} \quad R(x) = \frac{1}{2}\begin{pmatrix} \mathrm{e}^x + \mathrm{e}^{-x} & \mathrm{e}^x - \mathrm{e}^{-x} \\ \mathrm{e}^x - \mathrm{e}^{-x} & \mathrm{e}^x + \mathrm{e}^{-x} \end{pmatrix}$$

(1) Calculer P^2 puis en déduire que P est inversible et donner son inverse.

(2) Soit x un réel fixé.
  (a) Justifier que $R(x)$ est diagonalisable.
  (b) Vérifier que les vecteurs $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ sont des vecteurs propres de $R(x)$ et déterminer les valeurs propres associées (en fonction de x).
  (c) En déduire que $\mathrm{Sp}(R(x)) = \{\mathrm{e}^x, \mathrm{e}^{-x}\}$. $R(x)$ est-elle inversible ?
  (d) Déterminer une matrice diagonale $D(x)$ vérifiant : $R(x) = PD(x)P^{-1}$.
  (e) En déduire que, pour tout couple (x, y) de réels : $R(x)R(y) = R(x + y)$.
  (f) En déduire que, pour tout entier naturel n, on a : $(R(x))^n = R(nx)$.
  (g) Calculer $R(0)$ puis déduire également de la Question (2e) la matrice inverse de $R(x)$.

On dit qu'une suite de matrices (M_n) tend vers la matrice M, lorsque n tend vers $+\infty$, si chaque coefficient de M_n tend vers le coefficient situé à la même place dans M. On note alors : $\lim_{n\to\infty} M_n = M$.
On admet alors le résultat suivant noté $(\star\star)$.

Résultat admis

Si $\lim\limits_{n\to\infty} M_n = M$ et Q est une matrice inversible, alors : $\lim\limits_{n\to\infty} QM_nQ^{-1} = QMQ^{-1}$. $\quad (\star\star)$.

L'objectif de la suite de l'exercice est de déterminer la limite lorsque n tend vers $+\infty$ de la suite $((A_n(x))^n)_{n\in\mathbb{N}^*}$.

(3) Soit n un entier naturel non nul fixé et x un réel non nul fixé.
  (a) Montrer que : $\mathrm{Sp}(A_n(x)) = \left\{1 + \frac{x}{n}; 1 - \frac{x}{n}\right\}$ et déterminer les sous-espaces propres associés.
  (b) En déduire qu'il existe une matrice diagonale $D_n(x)$ vérifiant : $A_n(x) = PD_n(x)P^{-1}$.

(4) Soit x un réel non nul fixé.
  (a) Soit n un entier naturel non nul fixé. Montrer que pour tout entier k : $A_n(x)^k = PD_n(x)^kP^{-1}$.
  (b) Calculer :
$$\lim_{n\to+\infty}\left(1 + \frac{x}{n}\right)^n \quad \text{et} \quad \lim_{n\to+\infty}\left(1 - \frac{x}{n}\right)^n.$$
  (c) En déduire que $\lim\limits_{n\to+\infty} D_n(x)^n = D(x)$, où la matrice $D(x)$ est celle obtenue à la Question (2d)
  (d) En déduire, à l'aide du résultat $(\star\star)$ la limite : $\lim\limits_{n\to+\infty} A_n(x)^n$.

Problème

Partie 1 : Loi de Gumbel

Soient m un réel et a un réel strictement positif. On introduit la fonction $f_{m,a}$ définie sur $\mathbb{R}$ par

$$\forall x \in \mathbb{R}, \qquad f_{m,a}(x) = \frac{1}{a}\exp\left(\frac{m-x}{a} - e^{\frac{m-x}{a}}\right).$$

(1) Montrer que $f_{m,a}$ peut être considérée comme une densité de probabilité.

On dira qu'une variable aléatoire Z ayant pour densité $f_{m,a}$ suit la loi de Gumbel de paramètre m et a et on écrira $Z \hookrightarrow \mathcal{G}(m,a)$.

(2) Déterminer l'expression de la fonction de répartition $F_{m,a}$ d'une v.a $Z \hookrightarrow \mathcal{G}(m,a)$.

(3) Montrer que

$$Z \hookrightarrow \mathcal{G}(0,1) \Longleftrightarrow aZ + m \hookrightarrow \mathcal{G}(m,a).$$

(4) Soit $X \hookrightarrow \mathcal{E}(1)$. Montrer que $-\ln(X) \hookrightarrow \mathcal{G}(0,1)$.

(5) Python. Déduire des deux questions précédentes l'écriture d'une fonction d'en-tête `def Gumbel(m, a)` qui renvoie une simulation d'une loi de Gumbel de paramètres m et a.

(6) On note Ψ la fonction de répartition de la loi $\mathcal{G}(0,1)$. Soit $\alpha \in]0;1[$. Montrer qu'il existe deux réels c_α et d_α (que l'on déterminera) tels que

$$\Psi(c_\alpha) = 1-\alpha, \quad \text{et} \quad \Psi(d_\alpha) = \alpha.$$

Partie 2 : $n-$échantillon de lois exponentielles et loi de Gumbel

Soit $n \in \mathbb{N}^*$. On considère un $n-$échantillon $(X_1, X_2, ..., X_n)$ de la loi exponentielle $\mathcal{E}(\lambda)$ de paramètre $\lambda > 0$. On pose $L_n = \min(X_1, ..., X_n)$ et $M_n = \max(X_1, ..., X_n)$.

(7) Déterminer les fonctions de répartition des variables L_n puis M_n.

(8) Quelle est la loi de la variable $Y_n = n\lambda L_n$?

(9) On maintenant alors $Z_n = \lambda M_n - \ln(n)$.

(a) Déterminer la fonction de répartition F_n de Z_n.
(b) Pour $t \in \mathbb{R}$, déterminer la limite $F(t)$ de $F_n(t)$ lorsque $n \to +\infty$.
(c) En déduire soigneusement que Z_n converge en loi vers une variable aléatoire Z suivant une loi de Gumbel dont on précisera les paramètres.

Partie 3 : Construction de deux intervalles de confiance

La durée de vie d'une ampoule est modélisée par une variable aléatoire $X \hookrightarrow \mathcal{E}(\lambda)$ où $\lambda > 0$ est inconnu. On cherche à estimer la durée de vie moyenne $\mu = E(X) = 1/\lambda$ et on dispose d'un échantillon de n ampoules (dont les durées de vie sont supposées indépendantes).

(10) Dans cette question, on suppose que la seule information dont on dispose est la durée de vie de l'ampoule qui a *grillé* le plus tôt.

(a) À l'aide de la Question (8), proposer un estimateur $\tilde{L}_n$ de μ, construit à partir de L_n, qui soit sans biais.
(b) Quel est son risque quadratique?
(c) Montrer que, pour tout $\varepsilon > 0$

$$P\left(\left|\tilde{L}_n - \mu\right| > \varepsilon\right) = 1 - e^{-1+\lambda\varepsilon} + e^{-1-\lambda\varepsilon}.$$

(d) L'estimateur $\tilde{L}_n$ est-il convergent?
(e) Soit $\alpha \in]0;1[$. Montrer que si $Y \hookrightarrow \mathcal{E}(1)$, alors

$$P\left(Y < -\ln\left(1-\frac{\alpha}{2}\right)\right) = P\left(Y > -\ln\left(\frac{\alpha}{2}\right)\right) = \frac{\alpha}{2}.$$

(f) Montrer alors que l'intervalle $I_{\alpha,n}$ ci-dessous est un intervalle de confiance au seuil $1-\alpha$ pour μ :

$$I_{\alpha,n} = \left[\frac{\tilde{L}_n}{-\ln(\alpha/2)}, \frac{\tilde{L}_n}{-\ln(1-\alpha/2)} \right].$$

(11) Dans cette question, on suppose que l'on connait la durée de vie de la dernière ampoule. Soit $\alpha \in]0;1[$.

(a) Expliciter les valeurs de $c_{\alpha/2}$ et $d_{\alpha/2}$.

(b) À l'aide de la Question (9c), montrer que $J_{\alpha,n}$ est un intervalle de confiance asymptotique au seuil $1-\alpha$ pour μ:

$$J_{\alpha,n} = \left[\frac{M_n}{\ln(n) - \ln(-\ln(1-\alpha/2))}; \frac{M_n}{\ln(n) - \ln(-\ln(\alpha/2))} \right]$$

(12) Python. On ajoute les commandes suivantes dont l'exécution produit les affichages ci-contre. Commenter et interpréter. Vaut-il mieux connaître L_n ou M_n ?

```
def IDC1(alpha, X):
    L, n = min(X), len(X)
    return [n*L/(-np.log(alpha/2)), n*L/(-np.log(1-alpha/2))]

def h(t) :
    return -np.log(-np.log(t))
def IDC2(alpha, X):
    M, n = max(X), len(X)
    return [M/(np.log(n)+h(1-alpha/2)), M/(np.log(n)+h(alpha/2))]

alpha=0.05
for i in np.arange(10):
    sample=[rd.exponential(2) for k in range(1000)]
    [A, B]=IDC1(alpha, sample)
    plt.plot([A,B], [i, i], '--')
    plt.plot([2], [i], 'kx')
plt.show() # Figure 1
for i in np.arange(10):
    sample=[rd.exponential(2) for k in range(1000)]
    [A, B]=IDC2(alpha, sample)
    plt.plot([A,B], [i, i])
    plt.plot([2], [i], 'kx')
plt.show() # Figure 2
```

Figure 1

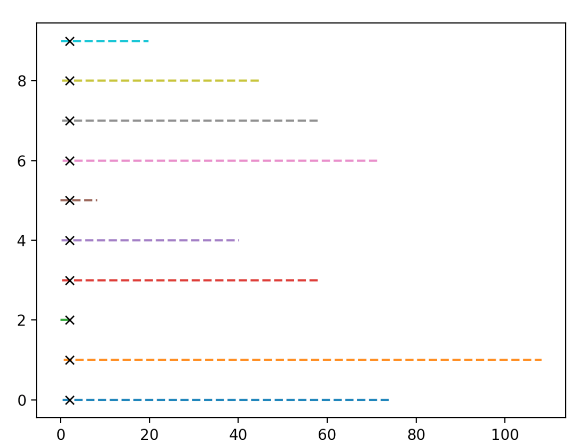

Figure 2