

TD Math & Python (2)

Les questions informatiques sont marquées d'un (♣)

On charge d'abord les modules dont on a besoin (en prenant les alias du memento)

```
from math import *  
  
import matplotlib.pyplot as plt  
  
import numpy as np ; import numpy.linalg as alg  
  
import scipy.integrate as integr  
  
import scipy.optimize as resol  
  
import numpy.random as rd  
  
from numpy.polynomial import *
```

Exercice A

Soit $(X_k)_{k \in \mathbb{N}}$ une suite de v.a. i.i.d. et N une v.a. entière indépendante de $(X_k)_{k \in \mathbb{N}}$. On pose $Y = \sum_{k=1}^N X_k$.

1) (♣) On prend $X = X_1 \sim \mathcal{B}(50, \frac{1}{50})$ et $N \sim \mathcal{P}(\frac{1}{10})$.

a) Définir en PYTHON une fonction permettant de calculer Y .

b) Réaliser 10 fois de suite une série de 1000 expériences et évaluer la moyenne et l'écart-type de Y .

2) On revient au cas général. On suppose X_1 et N de moments d'ordre 2 finis.

a) Exprimer G_Y en fonction de G_X et G_N . En déduire $E(Y)$.

b) On prend $X_1 \sim \mathcal{B}(n, p)$ et $N \sim \mathcal{P}(\lambda)$. Calculer $E(Y)$ et $V(Y)$.

Exercice B

1) (♣) On considère la matrice réelle $M(a, b) = \begin{pmatrix} 3a - 2b & -6a + 6b + 3 \\ a - b & -2a + 3b + 1 \end{pmatrix}$.

On définit l'écart par $e(a, b) = |\lambda - \mu|$, où λ et μ sont les valeurs propres complexes de $M(a, b)$.

Définir la fonction `ecart` qui étant donnés a et b renvoie $e(a, b)$.

2) (♣) Soient A et B deux variables aléatoires indépendantes, à valeurs dans $\mathbb{N}^*$ et de loi géométrique $\mathcal{G}(p)$.

a) Ecrire une fonction `hasard` qui étant donné un réel $p \in]0, 1[$ réalise la simulation de 500 valeurs (a, b) du couple de variables aléatoires (A, B) et renvoie le nombre de fois où $e(a, b) \geq \frac{1}{100}$.

b) Pour $p = \frac{1}{500}, \frac{2}{500}, \dots, \frac{499}{500}$, construire un graphique représentant les $M_p = \left(p, \frac{1}{500} \text{hasard}(p)\right)$.

c) Comparer la courbe obtenue avec le graphe de la fonction $p \mapsto \frac{2 - 2p + p^2}{2 - p}$ définie sur $[0, 1]$.

3) a) Montrer (*ou bien admettre*) que la matrice $M(a, b)$ est semblable à $N(a, b) = \begin{pmatrix} a + 1 & 1 \\ 0 & b \end{pmatrix}$.

b) A quelle condition la matrice $N(a, b)$ est-elle diagonalisable ?

c) Calculer la probabilité pour que $M(A, B)$ soit diagonalisable.

Exercice C

Soit $A = (X_{ij})_{1 \leq i \leq n, 1 \leq j \leq n} \in \mathcal{M}_n(\mathbb{R})$ une matrice dont les coefficients X_{ij} sont des v.a. i.i.d. de loi de Rademacher.

1) (♣) On note $D_n = \det A$. Avec PYTHON, évaluer $E(D_n)$ et $V(D_n)$.

2) Justifier mathématiquement que $E(D_n) = 0$ et $V(D_n) = n!$

Indication : Raisonner par récurrence sur n : utiliser $D_n = \sum_{i=1}^n X_{i1} C_{i1}$.

En appliquant l'hypothèse de récurrence aux C_{i1} (de loi $\sim D_{n-1}$), montrer que $E(D_n^2) = nE(X^2)E(D_{n-1}^2)$.

Exercice D

1) (♣) Une urne contient trois boules distinctes. On effectue des tirages avec remise. Pour $i \in \mathbb{N}^*$, on note Y_i la variable aléatoire telle que $Y_i = k$ si l'on a, pour la première fois, i boules distinctes dans les k premiers tirages. On a $Y_1 = 1$.

Écrire une fonction `test(i)` de paramètre $i \in \{1, 2, 3\}$ renvoie Y_i . On limitera le nombre de tirages à 30.

2) (♣) Écrire une fonction `repartition(i,n)` donnant la répartition sur n expériences (on limitera à $k \leq 30$).

3) Calculer $P(Y_3 = n + k, Y_2 = n)$ pour $n \geq 1$ et $k \geq 1$. En déduire la loi de $Y_3 - Y_2$.

Exercice E (★)

Soit $(\varepsilon_n)_{n \in \mathbb{N}^*}$ une suite de v.a. i.i.d. de Rademacher : $\forall n \in \mathbb{N}^*, P(\varepsilon_k = +1) = P(\varepsilon_k = -1) = \frac{1}{2}$.

Pour $n \in \mathbb{N}^*$, on pose $X_n = \sum_{k=1}^n 2^{-k} \varepsilon_k$.

1) Vérifier mathématiquement que $P(X_n \in [-1, 1]) = 1$.

2) Soit $(X_{n,k})_{(n,k) \in \mathbb{N}^* \times \mathbb{N}^*}$ une suite de v.a. indépendantes où chaque $X_{n,k}$ suit la même loi que X_n .

Soient $\varepsilon > 0$ et $t \in \mathbb{R}$. Montrer que $\lim_{N \rightarrow +\infty} P\left(\left|\frac{1}{N} \sum_{k=1}^N \cos(tX_{n,k}) - E(\cos(tX_n))\right| \geq \varepsilon\right) = 0$.

3) (♣) Écrire en PYTHON un programme modélisant les variables aléatoires

$$Y_{n,N} = \frac{1}{N} \sum_{k=1}^N \cos(tX_{n,k})$$

Utiliser cette fonction avec $N = 1000$ pour évaluer $E(\cos(tX_n))$ où $n \in \{3, 4, \dots, 10\}$.

4) On pose $\phi_{X_n} : \mathbb{R} \rightarrow \mathbb{C} \quad t \mapsto E(\exp(itX_n))$. Montrer mathématiquement que $\phi_{X_n}(t) = \frac{1}{2^n} \frac{\sin(t)}{\sin(t/2^n)}$.

Exercice F

1) (♣) Montrer avec PYTHON que $(5+i)^4(239-i) = 114244(1+i)$.

En déduire : $4 \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right) = \frac{\pi}{4}$.

2) Montrer : $\forall x \in [0, 1], \left| \arctan(x) - \sum_{k=0}^n \frac{(-1)^k}{2k+1} x^{2k+1} \right| \leq x^{2n+3}$.

En déduire une valeur approchée de π à 10^{-15} près.

Remarque : PYTHON fonctionne en double précision, c'est-à-dire que les nombres flottants sont codés sur 64 bits.

La mantisse étant codée sur 53 bits, la manipulation des nombres flottants entraîne systématiquement une erreur de 2^{-53} , c'est-à-dire environ 10^{-16} sur la mantisse, donc sur le 17-ième chiffre significatif du nombre.