DS5

Devoir Surveillé du 03/01/2023

Durée : 1h

La qualité de la rédaction, la clarté et la précision des raisonnements interviendront pour une part importante dans l'appréciation des copies. Les résultats doivent être encadrés.
La calculatrice n'est pas autorisée.

Dans tous les exercices, on suppose avoir importé les bibliothèques suivantes :

```
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
```

Exercice 1

Soient a, b deux entiers strictement positifs. Une urne contient initialement a boules rouges et b boules blanches. On effectue une succession d'épreuves, chaque épreuve étant constituée des trois étapes suivantes :

- on pioche une boule au hasard dans l'urne,
- on replace la boule tirée dans l'urne,
- on rajoute dans l'urne une boule de la même couleur que celle qui vient d'être piochée.

Après n épreuves, l'urne contient donc $a + b + n$ boules. Pour tout $n \in \mathbb{N}^*$, on note X_n le nombre de boules rouges qui ont été **ajoutées** dans l'urne (par rapport à la composition initiale) à l'issue des n premières épreuves.

On souhaite simuler l'expérience grâce à `Python`.

1. Compléter la fonction suivante, qui simule le tirage d'une boule dans une urne contenant x boules rouges et y boules blanches et qui retourne la valeur 0 si la boule est rouge et 1 si elle est blanche.

```
def tirage(x,y):
    r = rd.random()
    if ............ :
        res = 0
    else :
        res = 1
    return res
```

2. Compléter la fonction suivante, qui simule n tirages successifs dans une urne contenant initialement a boules rouges et b boules blanches (selon le protocole décrit ci-dessus) et qui retourne la valeur de X_n :

```
def experience(a,b,n):
    x = a
    y = b
    for k in range(n):
        r = tirage(x,y)
        if r == 0 :
```

```
            x = .........
        else :
            ............
    Xn = .........
    return Xn
```

3. Écrire une fonction `simulation(a,b,n,m)` qui fait appel `m` fois à la fonction précédente pour estimer la loi de X_n. Le paramètre de sortie sera un vecteur à $n+1$ composantes contenant les approximations de $P(X_n=0)$, $P(X_n=1), \dots, P(X_n=n)$.

4. On s'intéresse au cas où $a=b=1$. On rappelle les commandes suivantes :

   - Si `x` et `y` sont des vecteurs de même taille, `plt.bar(x,y)` trace le diagramme en bâtons d'abscisses contenues dans `x` et d'ordonnées dans `y`.
   - `plt.figure(n)` ouvre une fenêtre graphique et trace la figure `n` dans celle-ci.

   On exécute le code suivant :

```
for n in range(1,6):
    x = np.arange(n+1)
    y = simulation(1,1,n,100000)
    plt.figure(n)
    plt.bar(x,y)
plt.show()
```

   On obtient les figures suivantes :

Figure 1

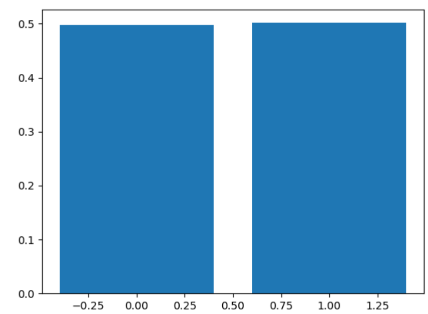


Figure 2

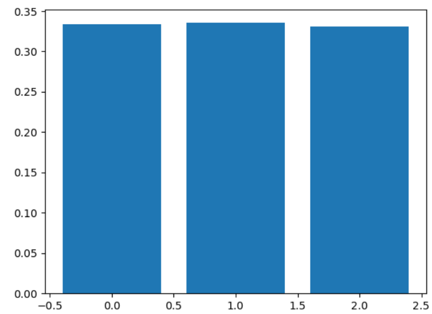


Figure 3

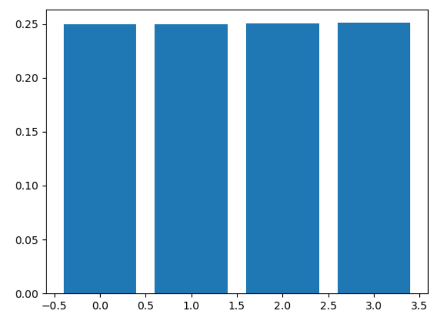


Figure 4

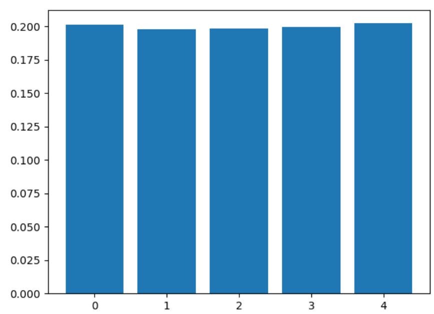
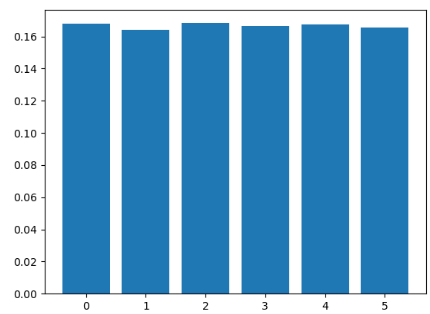


Figure 5



   À l'aide de ces résultats, conjecturer la loi de X_n.

Exercice 2

On dit qu'une variable X suit une loi de Cauchy si elle admet pour fonction de répartition la fonction :

$$F : x \in \mathbb{R} \mapsto \frac{1}{\pi}\left(\arctan(x) + \frac{\pi}{2}\right).$$

Une densité de X est alors la fonction $f : t \in \mathbb{R} \mapsto \dfrac{1}{\pi}\dfrac{1}{1+t^2}$.

1. Montrer que F réalise une bijection de $\mathbb{R}$ sur $]0,1[$, et déterminer F^{-1}.
2. (a) On rappelle que si U suit une loi $\mathscr{U}(]0,1[)$, alors $C = F^{-1}(U)$ suit une loi de Cauchy. Écrire une fonction `cauchy()` simulant une loi de Cauchy à partir de la fonction `rd.random`.

   (b) Écrire une fonction `Cauchy(N)` donnant un échantillon de taille N de la loi de Cauchy.
3. On considère la fonction `Python` suivante, qui utilise la fonction `Cauchy` :

```python
def moyenne(n):
    return np.mean(Cauchy(n))
```

Cinq appels `moyenne(1000)` donnent successivement comme résultats (arrondis à 10^{-2} près) :

`5.48,    -3.81,    0.97,    222.30,    2.92`

Que conjecturez-vous ? Démontrer votre conjecture.

Exercice 3

1. On a entré les commandes suivantes :

```python
A = np.array([[1,4,2],[3,2,8]])
print(np.min(A,0), np.min(A,1))
```

et on a obtenu : `[1 2 2] [1 2]`

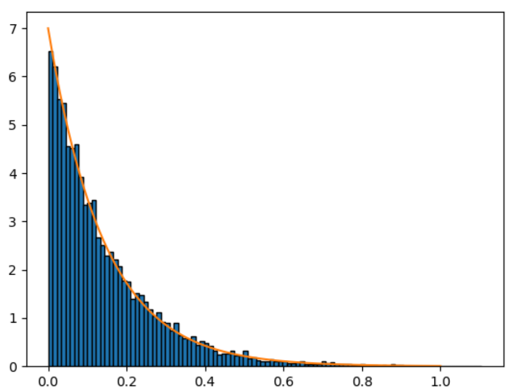
Que contient la variable `A` ? Expliquer ce que font les commandes `np.min(A,0)` et `np.min(A,1)`.

2. Soit $n \in \mathbb{N}$ avec $n \geq 2$.

   Soient $X_1, X_2, \ldots, X_n$ n variables aléatoires indépendantes et suivant la même loi exponentielle de paramètre 1. On pose $I_n = \min(X_1, \ldots, X_n)$.

   (a) On considère le programme `Python` suivant qui renvoie le graphique ci-dessous :

```python
n = 7
X = rd.exponential(1,[n,10000])
I = np.min(X,0)
plt.hist(I,100,density='True',edgecolor = 'k')
def f(x):
    y = n*np.exp(-n*x)
    return y
x = np.linspace(0,1,100)
y = f(x)
plt.plot(x,y)
plt.show()
```



   Que contient la variable `X` ? et la variable `I` ? Expliquer en détails ce que fait ce programme.

   (b) Que peut-on conjecturer quant à la loi de I_n ?
