DS8

Devoir surveillé du 10/01/2024

Durée : 0h50

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

Une urne contient 18 boules rouges et 2 boules vertes. On effectue des tirages sans remise dans cette urne, jusqu'à vider l'urne, et on note alors X le rang d'apparition de la première boule verte, et Y le rang d'apparition de la seconde boule verte.

1. Compléter la fonction suivante de sorte que la commande `un_tirage(r,v)` simule un tirage dans une urne contenant r boules rouges et v boules vertes, et retourne 0 si la boule tirée est rouge et 1 si la boule tirée est verte.

```
def un_tirage(r,v):
      if rd.random()< --- :
            y = ---
      else :
            y = ---
      return( --- )
```

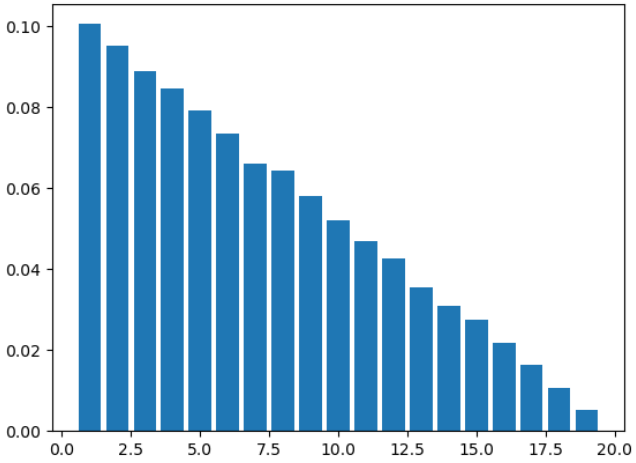
2. Compléter le programme suivant pour qu'il simule la variable aléatoire X.

```
def simulX():
      y = 1
      nb_rouges = 18
      nb_vertes = 2
      while --- :
            nb_rouges = ---
            y = ---
      return( --- )
```

3. En adaptant le programme de la question précédente, proposer une fonction `simulY()` qui simule une réalisation de la variable Y.

4. Écrire une fonction `LoiX(m)` qui fait appel `m` fois à la fonction `simulX()` pour estimer la loi de X. Le paramètre de sortie sera un vecteur contenant les approximations de $P(X = 1), P(X = 2), \ldots, P(X = 19)$.

5. Proposer un programme permettant d'obtenir le diagramme en bâtons suivant représentant une approximation de la loi de X :



Exercice 2

On désigne par p un réel de $]0,1[$. On considère une variable aléatoire Z définie sur un espace probabilisé $(\Omega, \mathscr{A}, P)$ prenant la valeur 1 avec la probabilité p et la valeur -1 avec la probabilité $1-p$.

1. (a) Écrire une fonction `simulZ(p,N)` qui prend comme paramètres d'entrée un réel $p \in]0,1[$ et un entier naturel N non nul, et qui renvoie un vecteur de taille N contenant N réalisations de la variable Z.

   (b) On considère les commandes suivantes :

```
p = 1/3 ; N = 10000
u = simulZ(p,N)
a = np.mean(u==1)
```

   De quelle valeur le contenu de la variable `a` est-il proche ?

2. On considère une suite $(Z_k)_{k\geq 1}$ de variables aléatoires toutes définies sur l'espace probabilisé $(\Omega, \mathscr{A}, P)$, mutuellement indépendantes et de même loi que Z. On assimile ces variables à des déplacements d'un point sur un axe d'origine O, les déplacements se faisant d'une unité vers la droite avec la probabilité p et d'une unité vers la gauche avec probabilité $1-p$.

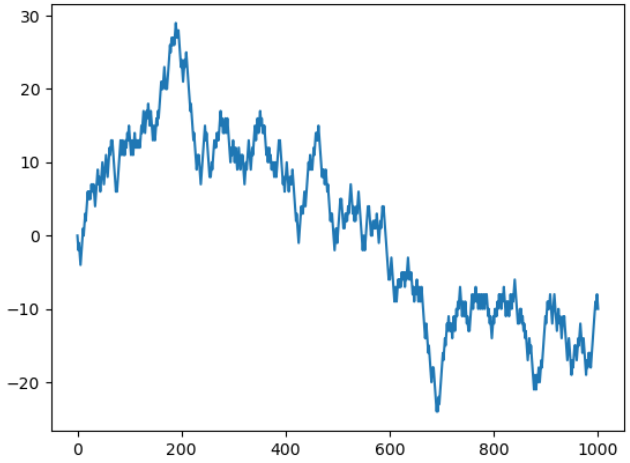
   Pour tout $n \in \mathbb{N}$, on note X_n la coordonnée du point après le n-ème déplacement, et on pose $X_0 = 0$ (le voyage du point commence à l'origine).

   (a) Pour tout $n \geq 1$, exprimer X_n en fonction des variables $Z_1, \ldots, Z_n$.

   (b) Proposer une fonction `marche(p,n)` prenant comme paramètres d'entrée un réel $p \in]0,1[$ et un entier naturel n, et qui renvoie un vecteur de longueur $n+1$ dont la k-ème composante contient la coordonnée du point après le k-ème déplacement.

   (c) Que permet d'obtenir la commande `y = np.sum(marche(p,n)==0)` ?

   (d) Proposer des commandes permettant de visualiser le trajet du point pour $n = 1000$ déplacements et $p = 1/2$. On représentera pour cela en abscisse le nombre de déplacements, et en ordonnée la position du point.



Exemple de trajet du point pour $n = 1000$ déplacements et $p = 1/2$.

Exercice 3

Pour tout $a, b \in \mathbb{N}$, on note $I_{a,b}$ le réel strictement positif défini par :

$$I_{a,b} = \int_0^1 x^a (1-x)^b \, \mathrm{d}x.$$

On considère la fonction $f_{a,b}$ définie sur $\mathbb{R}$ par $f_{a,b}(x) = \begin{cases} \dfrac{1}{I_{a,b}} x^a (1-x)^b & \text{si } x \in]0,1[, \\ 0 & \text{sinon.} \end{cases}$

1. Vérifier que $f_{a,b}$ est une densité de probabilité. On parle alors d'une loi bêta de paramètre (a, b).

2. On peut montrer à l'aide d'intégrations par parties que $I_{a,b} = \dfrac{a! \times b!}{(a+b+1)!}$ pour tout $(a, b) \in \mathbb{N}^2$ (résultat admis).

   Écrire une fonction `integrale(a,b)` qui prend en entrée deux entiers naturels a et b, et qui renvoie le réel $I_{a,b}$.

3. Écrire une fonction `f(a,b,x)` qui prend en entrée deux entiers naturels a et b et un réel $x \in [0, 1]$, et qui renvoie $f_{a,b}(x)$.

Soient $n \in \mathbb{N}^*$ et $X_1, \ldots, X_n$, n variables aléatoires indépendantes et de loi uniforme sur $[0, 1]$. On admet l'existence de variables aléatoires à densité $Y_1, \ldots, Y_n$ telles que, pour tout $\omega \in \Omega$, les réels $Y_1(\omega), Y_2(\omega), \ldots, Y_n(\omega)$ constituent un réarrangement par ordre croissant des réels $X_1(\omega), X_2(\omega), \ldots, X_n(\omega)$, de telle sorte que, pour tout $\omega \in \Omega$:

$$Y_1(\omega) \leq Y_2(\omega) \leq \cdots \leq Y_n(\omega).$$

4. On a entré les commandes suivantes :

```
A = np.array([[6,12,3],[7,1,10]])
print(np.sort(A))
```

   et on a obtenu :

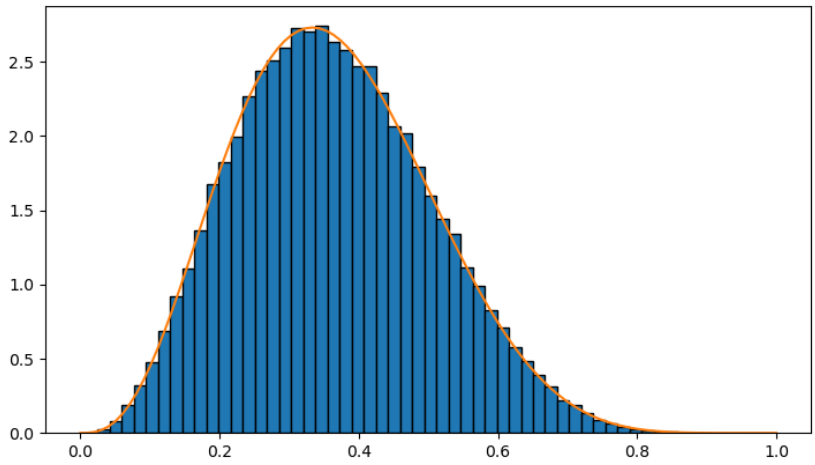
```
array([[ 3,  6, 12],
       [ 1,  7, 10]])
```

Que contient la variable `A` ? Expliquer ce que réalise la commande `np.sort(A)`.

5. On propose le programme suivant :

```
1  n = 10 ; k = 4
2
3  m = 100000
4  A = rd.random([m,n])
5  B = np.sort(A)
6  Y = B[:,k-1]
7  plt.hist(Y,50,density = 'True', edgecolor = 'k')
8
9  x = np.linspace(0,1,100)
10 y = f(k-1,n-k,x)
11 plt.plot(x,y)
12 plt.show()
```

On obtient en l'exécutant la représentation graphique suivante :



On obtient des représentations graphiques analogues pour d'autres valeurs de n et de k.

(a) Que contiennent les variables `A`, `B` et `Y` ? Expliquer en détails ce que fait ce programme.

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

(c) Quelles lignes de commandes peut-on ajouter au programme précédent pour estimer la valeur de $E(Y_k)$ et de $V(Y_k)$?

Proposer d'autres lignes de commandes utilisant la fonction `integrale` permettant d'obtenir une valeur approchée de $E(Y_k)$ et de $V(Y_k)$.