

Code de partage avec Cappytale : ef48-7365099

On utilisera (entre autres) la bibliothèque `numpy.random` sous l'alias `rd`

Exercice 1

Le score d'un joueur lors d'un lancer de fléchette est modélisé par une variable aléatoire X telle que $X(\Omega) = \{0, 2, 5, 10\}$ et

$$P(X = 0) = \frac{1}{5}, \quad P(X = 2) = \frac{1}{2}, \quad P(X = 5) = \frac{1}{5} \quad \text{et} \quad P(X = 10) = \frac{1}{10}$$

1. Calculer l'espérance et la variance de X
2. Découper l'intervalle $[0, 1]$ en quatre intervalles où pourrait tomber un nombre aléatoire généré avec `rd.random()` dans le but de simuler X . Écrire alors une fonction `simulX()` qui renvoie une simulation de la variable X
3. Que fait le programme suivant ?

```
def sampleX(n):
    S = []
    for k in range(n):
        S.append(simulX())
    return S
print(np.mean(sampleX(1000)))
```

Exercice 2

On désigne par n un entier naturel supérieur ou égal à 2. On lance n fois une pièce équilibrée (c'est-à-dire donnant « Pile » avec la probabilité $\frac{1}{2}$), les lancers étant supposés indépendants.

On note Z la variable aléatoire qui vaut 0 si l'on n'obtient aucun Pile pendant ces n lancers et qui, dans le cas contraire, prend pour valeur le rang du premier pile.

1. Compléter la fonction suivante permettant de simuler la variable aléatoire Z

```
def simulZ(n) :
    for i in range(1, n+1):
        if ..... :
            return ...
    return ...
```

2. Réécrire cette fonction à l'aide d'une boucle `while`.

Exercice 3

On dispose de $n \geq 1$ urnes, numérotées de 1 à n . Pour chaque $k \in \llbracket 1, n \rrbracket$ l'urne k est composée de k boules numérotées de 1 à k . On choisit une urne au hasard puis on tire une boule dans cette urne et on note X_n la variable aléatoire qui prend la valeur du numéro de la boule piochée.

Écrire une fonction `simulX(n)` qui simule X_n

Exercice 4

Chaque jour, le nombre de personnes subissant un test de dépistage dans une certaine pharmacie est modélisé par une variable N suivant une loi de Poisson de paramètre $\lambda = 5$. Chaque test a une certaine probabilité $p = \frac{1}{10}$ d'être positif et les tests sont indépendants les uns des autres. On note X le nombre de tests positifs un jour donné.

1. Ecrire une fonction permettant de simuler la variable X
2. Créer un échantillon de taille 1 000 de cette variable aléatoire et le comparer, avec un diagramme en bâtons, à un échantillon de même taille d'une loi de Poisson de paramètre $\lambda \times p = \frac{1}{2}$.
Émettre une conjecture sur la loi de X

Rappel : on pourra utiliser la fonction `plt.bar` pour le diagramme en bâtons.

3. En commençant par expliciter $P_{[N=n]}(X = k)$, démontrer le résultat précédemment conjecturé.

Exercice 5

Une urne contient des boules numérotées de 1 à n , indiscernables au toucher. On effectue alors des tirages sans remise dans l'urne et on note, pour tout $k \in \llbracket 1, n \rrbracket$, X_k la variable aléatoire qui prend la valeur de la boule obtenue au $k^{\text{ième}}$ tirage.

1. Ecrire une fonction Python `simulX` qui prend en argument deux entiers naturels non nuls k et n , et qui renvoie une simulation de X_k
2. On exécute le programme suivant, expliquer ce qu'il fait (la ligne 7 en particulier).

```
n=6
k=4
echantillon = [simulX(k,n) for j in range(1000) ]
freq = np.zeros(6)

for j in range(1000):
    freq[echantillon[j]-1] = freq[echantillon[j]-1]+1

freq = freq/1000

plt.bar([j for j in range(1,n+1)],freq)
plt.show()
```

Puis émettre une conjecture sur la loi de X_n