DM 4

A rendre le Vendredi 18 Octobre

Exercice 1

Dans tout cet exercice, N désigne un entier naturel supérieur ou égal à 3.
On dispose de deux urnes opaques U_1 et U_2, d'apparence identique et contenant chacune N boules indiscernables au toucher. L'urne U_1 contient $(N-1)$ boules blanches et une boule noire. L'urne U_2 contient N boules blanches.

I - Une première expérience aléatoire

On effectue des tirages **sans remise** dans l'urne U_1, jusqu'à l'obtention de la boule noire.
On note X la variable aléatoire qui prend pour valeur le nombre de tirages nécessaires pour l'obtention de la boule noire.
On notera pour tout entier naturel i non nul :

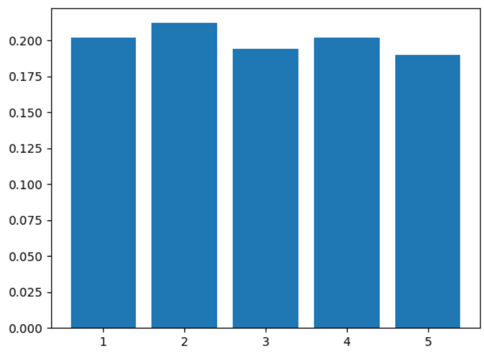
- N_i l'évènement "on tire une boule noire lors du i-ième tirage".
- B_i l'évènement "on tire une boule blanche lors du i-ième tirage".

1. On simule 1000 fois cette expérience aléatoire.

   Recopier et compléter le programme `Python` suivant pour qu'il affiche l'histogramme donnant la fréquence d'apparition du rang d'obtention de la boule noire :

```python
N = int(input('Donner un entier naturel non nul'))
x = list(range(1,N+1))
S = [0. for k in range(N)]
for k in range(1000) :
    i = 1
    M = N
    while ..... :
        i = i+1
        M = .....
    S[i-1] = S[i-1]+1/1000
plt.bar(x,S)
plt.show()
```

2. On exécute le programme complété ci-dessus. On entre 5 au clavier et on obtient l'histogramme suivant :



   Quelle conjecture pouvez-vous émettre sur la loi de la variable aléatoire X ?

Pour les questions suivantes, on revient au cas général où $N \geq 3$.

3. En écrivant soigneusement les évènements utilisés, calculer $P(X=1)$, $P(X=2)$ et $P(X=3)$.

4. Déterminer la loi de la variable aléatoire X.

5. Préciser le nombre moyen de tirages nécessaires à l'obtention de la boule noire.

II - Une deuxième expérience aléatoire

On choisit une des deux urnes au hasard (chaque urne a la même probabilité d'être choisie) et on tire dans l'urne choisie une par une les boules **sans remise** jusqu'à être en mesure de pouvoir connaître l'urne choisie.
On note Y la variable aléatoire qui prend pour valeur le nombre de tirages ainsi effectués.
On note :

- C_1 l'évènement "on choisit l'urne U_1".
- C_2 l'évènement "on choisit l'urne U_2".

6. Justifier que pour tout entier $j \in [\![1, N]\!]$:
$$P_{C_1}(Y=j) = \frac{1}{N}.$$

7. Calculer $P_{C_2}(Y=j)$ pour tout entier $j \in [\![1, N]\!]$.
On distinguera les cas $j = N$ et $1 \leq j \leq N-1$.

8. Montrer que :
$$P(Y=j) = \begin{cases} \dfrac{1}{2N} & \text{si } j \in [\![1, N-1]\!] \\ \dfrac{1}{2} + \dfrac{1}{2N} & \text{si } j = N \end{cases}$$

9. Calculer l'espérance de Y.

III - Une troisième expérience aléatoire

On effectue une succession infinie de tirages **avec remise** dans l'urne U_1. On admet qu'on obtient presque-sûrement au moins une boule blanche et au moins une boule noire lors de ces tirages.
On note T la variable aléatoire prenant pour valeur le nombre de tirages nécessaires jusqu'à l'obtention d'au moins une boule noire et d'au moins une boule blanche.
On note U la variable aléatoire prenant pour valeur le nombre de boules blanches tirées jusqu'à l'obtention d'au moins une boule noire et d'au moins une boule blanche.
Par exemple, si les tirages ont donné successivement : noire, noire, noire, blanche, blanche, noire,..., alors $T = 4$ et $U = 1$.

10. Préciser les valeurs prises par T.

11. Montrer soigneusement que pour tout entier $k \geq 2$,
$$P(T=k) = \frac{1}{N}\left(\frac{N-1}{N}\right)^{k-1} + \frac{N-1}{N}\left(\frac{1}{N}\right)^{k-1}.$$

12. Montrer que la variable aléatoire T admet une espérance que l'on calculera.

13. (a) Calculer $P([U=1] \cap [T=2])$.
    (b) Calculer $P([U=1] \cap [T=k])$ pour tout entier $k \geq 3$.

14. Soit j un entier tel que $j \geq 2$.
    (a) Calculer $P([U = j] \cap [T = j + 1])$.
    (b) Que vaut $P([U = j] \cap [T = k])$ pour tout entier $k \geq 2$ tel que $k \neq j + 1$?
15. Calculer $P(U = 1)$ puis déterminer la loi de U.

Exercice 2

On lance indéfiniment une pièce donnant *"Pile"* avec la probabilité p et *"Face"* avec la probabilité $q = 1 - p$. On suppose que $p \in]0, 1[$ et on admet que les lancers sont mutuellement indépendants.
Pour tout entier naturel k, supérieur ou égal à 2, on dit que le $k^{ième}$ lancer est un changement s'il amène un résultat différent de celui du $(k - 1)^{ième}$ lancer.
On note P_k (resp. F_k) l'événement : *"on obtient Pile (resp. Face) au $k^{ième}$ lancer"*.
Pour ne pas surcharger l'écriture on écrira, par exemple, P_1F_2 à la place de $P_1 \cap F_2$.
Pour tout entier naturel n supérieur ou égal à 2, on note X_n la variable aléatoire égale au nombre de changements survenus durant les n premiers lancers.

Partie 1 : Étude de quelques exemples.

1. Donner la loi de X_2.
2. (a) Donner la loi de X_3.
   (b) Vérifier que $E(X_3) = 4pq$ et que $V(X_3) = 2pq(3 - 8pq)$.
3. (a) Trouver la loi de X_4.
   (b) Calculer $E(X_4)$.

Partie 2 : Étude du cas $p \neq q$.

Dans cette partie, n désigne un entier naturel supérieur ou égal à 2.

4. Exprimer $P(X_n = 0)$ en fonction de p, q et n.
5. En décomposant l'événement $(X_n = 1)$ en une réunion d'événements incompatibles, montrer que :
$$P(X_n = 1) = \frac{2pq}{q - p}\left(q^{n-1} - p^{n-1}\right).$$
6. En distinguant les cas n pair et n impair, exprimer $P(X_n = n - 1)$ en fonction de p et q.
7. Pour tout entier naturel k, supérieur ou égal à 2, on note Z_k la variable aléatoire qui vaut 1 si le $k^{ième}$ lancer est un changement et 0 sinon (Z_k est donc une variable de Bernouilli).
   Écrire X_n à l'aide de certaines des variables Z_k et en déduire $E(X_n)$.

Partie 3 : Simulation

On rappelle que :

- En `Python`, on code pile par 1 et face par 0. Un lancer consiste donc à simuler une variable de Bernoulli de paramètre p (i.e. une binomiale de taille 1 et de paramètre p).
- L'instruction `L = rd.binomial(N,p,n)` crée une matrice à 1 ligne et n colonnes dont chaque coefficient est la simulation d'une variable binomiale de paramètres N et p. Ainsi, l'instruction `L = rd.binomial(1,p,n)` simule n variables de Bernoulli de paramètre p (n lancers de pièce).
- L'opérateur mathématique "$\neq$ " est codé par l'opérateur "`!=`" en `Python`.

8. Compléter les instructions suivantes pour créer une fonction qui, étant donnés n et p, effectue n lancers de pièces (avec proba p d'obtenir pile) et renvoie en sortie le nombre X de changements survenus au cours des n lancers :

```
def nbr_changements(n,p) :
    X = 0
    L = rd.binomial(1,p,n)
    for k in range(1,n) :
        if ..... :
            X = .....
    return(X)
```

9. On rajoute ensuite les instructions suivantes. Expliquer ce que font ces instructions en précisant ce qui est calculé dans la variable f.

```
n = 10
p = 1/4
q = 1-p
S = 0
for k in range(10000) :
    X = nbr_changements(n,p)
    if X==1 :
        S = S+1
f=S/10000
```

10. On rajoute à la fin du programme :

```
print(f)
print(2*p*q/(q-p)*(q**(n-1)-p**(n-1)))
```

Après exécution du programme, on obtient les résultats suivants dans la console :

```
0.0564
0.0563107
```

Pourquoi les 2 valeurs affichées sont-elles proches?