Travaux Pratiques 6 : Simulations d'expériences aléatoires

La bibliothèque à importer est : `import numpy.random as rd`

Deux syntaxes sont à connaître : `rd.random()` (renvoie un réel au hasard entre 0 et 1) et `rd.randint(a,b+1)` (renvoie un entier au hasard entre a et b)

Exercice 1: Important

1. En codant pile par 1 et face par 0, écrire la syntaxe d'un lancer de pièce équilibrée
(a) à l'aide de `rd.randint(0,2)`
(b) à l'aide de `rd.random()` et d'un `if`.
2. Ecrire alors la syntaxe d'un lancer de pièce truquée où p est la probabilité d'obtenir face. *On pourra écrire une fonction python d'en-tête `def lancer(p):` .*

Exercice 2: Important

Une urne contient une boule blanche (codée 0) et deux boules noires (codées 1).

1. Ecrire la syntaxe d'un tirage avec les commandes `if` et `rd.random()`.
2. Trouver une variante avec `if` et `rd.randint(1,4)`

Exercice 3:

On lance 10 fois une pièce truquée dont la probabilité d'obtenir pile est 0.4.

Ecrire un programme qui affiche le nombre de piles obtenus.

Exercice 4: Marche aléatoire

Une puce se déplace sur un axe gradué. A l'instant $t=0$, la puce se trouve à l'origine (point d'abscisse 0).

Puis entre un instant $t=n$, et l'instant $t=n+1$, la puce se déplace de 1 vers la droite ou de 1 vers la gauche avec équiprobabilité. On arrête l'observation à l'instant $t=200$.

1. Ecrire un script qui renvoie 1 ou -1 avec équiprobabilité.
2. Compléter ce script, pour qu'il crée un vecteur x de taille 201 tel que $\forall n \in \llbracket 0, 200 \rrbracket, x[n]$ donne l'abscisse de la puce à l'instant $t=n$.
3. Exécuter plusieurs fois le script et observer les différentes trajectoires : regarder en particulier l'abscisse maximale, minimale etc.

Exercice 5:

On effectue des lancers successifs d'un dé honnête. On note X_1 , respectivement X_2 , la variable aléatoire égale au nombre de lancers effectués jusqu'à l'obtention du premier 6, respectivement du second 6.

1. Ecrire un programme qui simule X_1 .
2. Compléter le programme précédent afin qu'il simule X_2 .

Exercice 6: Edhec E 2007 adapté

On lance une pièce équilibrée et on note Z la variable aléatoire égale au rang du lancer où l'on obtient le premier "pile". Puis si Z a pris la valeur k ($k \in \mathbb{N}^*$), on remplit une urne de k boules numérotées 1, 2, $\dots$, k , et on extrait au hasard une boule de cette urne. Soit X la variable aléatoire égale au numéro de la boule tirée.

Ecrire un programme python qui simule l'expérience ci-dessus, et affiche les valeurs de Z et X .

Exercice 7: Edhec E 2005 adapté

Un mobile se déplace sur les points à coordonnées entières d'un axe d'origine 0. A l'instant 0, le mobile est à l'origine.

Le mobile se déplace selon la règle suivante : s'il est sur le point d'abscisse k à l'instant n , alors, à l'instant $(n+1)$ il sera sur le point d'abscisse $(k+1)$ avec la probabilité $\frac{2}{3}$ ou sur le point d'abscisse 0 avec la probabilité $\frac{1}{3}$.

Pour tout $n \in \mathbb{N}^*$, on note X_n l'abscisse de ce point à l'instant n et l'on a donc $X_0 = 0$.

1. Compléter la fonction suivante pour qu'elle simule l'expérience aléatoire étudiée et renvoie la valeur prise par X_n où l'entier n est le paramètre d'entrée.

```
def simuedhec(n):
    x=0
    for k in range(1,n+1):
        if ----- :
            x=-----
        else:
            x=-----
    return x
```

2. Modifier la fonction précédente afin qu'elle renvoie le vecteur contenant toutes les abscisses du mobile.
3. Dans la console, exécuter la fonction avec $n = 100$. Quelle est l'abscisse minimale et maximale dans votre simulation? N'hésitez pas à l'exécuter plusieurs fois!

Exercice 8: inspiré d'ESC 1999

Soit $n \in \mathbb{N}^*$. Une urne U_n contient n boules numérotées de 1 à n . On y effectue une succession de tirages d'une boule, en appliquant la règle suivante : si la boule tirée porte le numéro k , avant de procéder au tirage suivant, on enlève toutes les boules dont le numéro est supérieur ou égal à k .

On note X_n la variable aléatoire égale au nombre de tirages nécessaires pour vider l'urne U_n de toutes ses boules.

1. Compléter le programme ci-contre afin qu'il simule l'expérience.
2. Exécuter plusieurs fois le programme avec $n = 100$, puis changer la valeur de n . Que remarquez-vous ?

```
n=int(input('n=?'))
x=-----
while ----- :
    k=-----
    n=-----
    x=-----
print(x)
```

Exercice 9: inspiré d'Ecricome E 2015

Soit $N \in \mathbb{N}$, $N \geq 3$. On considère une urne U , qui contient initialement $N - 1$ boules blanches et 1 boule noire et on y effectue des tirages sans remise, jusqu'à l'obtention de la boule noire.

Soit X_N la variable aléatoire égale au nombre de tirages effectués.

1. Ecrire une fonction d'en-tête `def tirages(N):` qui renvoie une simulation de X_N . Quelles valeurs, cette fonction peut-elle renvoyer ?
2. On suppose dans cette question $N = 5$. Que fait le programme suivant ? A quoi sert la variable `S` ?

```
S = np.zeros(5)
for k in range(10000)
    i=tirages(5)
    S[i-1] = S[i-1] + 1
print(S/10000)
```

Si besoin, exécuter le programme en ne faisant que 10 simulations au lieu de 10000 et en rajoutant `print(i)`, `print(S)` dans la boucle `for`.

3. Que peut-on conjecturer sur la loi de X_5 ? de X_N ? *bonus* : Vérifier mathématiquement la conjecture ...

Exercice 10: Expérience de l'urne de Polya

Soit $N \in \mathbb{N}^*$. On considère une urne contenant au départ 1 boule rouge et 1 boule verte. On effectue N tirages successifs de la façon suivante : on tire une boule de l'urne, puis on la remet avec une nouvelle boule de la même couleur. On note X_N le nombre de boules rouges après ces N tirages.

Ecrire une fonction python de paramètre d'entrée N qui renvoie la valeur de X_N .

Exercice 11: Edhec E 2011

Soit $n \geq 2$ un entier. On dispose de n urnes numérotées de 1 à n , contenant chacune n boules. On répète n épreuves, chacune consistant à choisir une urne au hasard et à en extraire une boule. On note X_1 la variable aléatoire qui prend la valeur 1 si l'urne 1 contient toujours n boules au bout de ces n épreuves, et qui prend la valeur 0 sinon. On introduit de plus la variable N_1 égale au nombre de boules manquantes dans l'urne 1 à la fin des n épreuves.

1. Ecrire un programme python qui simule l'expérience et affiche les valeurs de X_1 et N_1 pour une valeur de n entrée par l'utilisateur.
2. Créer un tableau qui contient à la fin le nombre de boules dans chaque urne.

Exercice 12: Edhec S 2018

On effectue des lancers d'une pièce donnant «Pile» avec la probabilité p , élément de $]0, 1[$, et donnant «face» avec la probabilité $q = 1 - p$, les différents lancers étant supposés indépendants.

On note S_k le rang du $k^{\text{ème}}$ pile et T_k le rang d'apparition du dernier «pile» de la première série de k «piles» consécutifs. Par exemple, si les lancers donnent $F_1, P_2, F_3, P_4, F_5, P_6, P_7, P_8$, alors S_1 et T_1 prennent la valeur 2, S_2 prend la valeur 4, T_2 prend la valeur 7, S_3 prend la valeur 6, T_3 prend la valeur 8, S_4 prend la valeur 7 et S_5 prend la valeur 8.

1. Compléter les lignes 7 et 9 du script suivant pour qu'il affiche la valeur prise par S_k :


```
1. k = int(input('donnez une valeur pour k :'))
2. p = float(input('donnez une valeur pour p :'))
3. n = 0
4. c = 0
5. while c<k:
6.     n=n+1
7.     if --- :
8.         c = c+1
9. print(---)
```
2. ** On souhaite que le script précédent affiche la valeur prise par T_k . Remplacer les lignes 7 et 8 par les suivantes, dûment complétées :


```
if ---:
    c = c+1
else:
    ---
```