

Code de partage avec Cappytale : 8309-7242374

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

Exercice 1

On dispose d'une pièce donnant pile avec une probabilité p

1. Par quelle(s) fonction(s) peut-on simuler un tirage de la pièce ?

```
def Tirage1(p):  
  
    return ...  
  
def Tirage2(p):  
    if ...  
        return ...  
    else:  
        return ...
```

2. Comment simuler une réalisation de la variable X vérifiant $X(\Omega) = \{-1, 1\}$ et $P(X = 1) = p$?

```
def X(p):  
    return ...
```

3. On considère une suite de tirages de la pièce.

Compléter la fonction suivante renvoyant la première valeur $n + 1$ pour laquelle on a pile aux rangs n et $n + 1$. On dira qu'on obtient un « double pile » aux rangs n et $n + 1$

```
def DoublePile(p):  
    u=...; v=...  
    # u, v correspondent aux deux premiers tirages  
    # n=2 # nombre de tirages effectués.  
    while ... :  
        (u, v) = (v, ...)  
        n ...  
    return
```

4. Pour obtenir une valeur approchée de l'espérance d'une variable aléatoire Z , on pourra simuler la loi un « grand » nombre de fois (en pratique : 1 000 ou 10 000 fois), et en prendre la moyenne. Si les simulations sont placées dans une liste `z`, on pourra utiliser la commande `np.mean(z)`. Estimer $E(Y)$ où Y donne le rang d'apparition du premier « double pile ». On fera des tests en prenant une valeur pour p ($= 0,5$ par exemple)

Nota bene : la valeur théorique est $\frac{1+p}{p^2}$

Exercice 2

Soit $n \in \mathbb{N}^*$

Une urne U contient $2n$ boules, n rouges et n bleues. On pioche n boules et on note X_n le nombre de boules rouges obtenues.

1. Expliquer ce que renvoie le code suivant (faire plusieurs essais).

```
L=rd.randint(1, 11, 5)
M=[1<=5 for l in L]
print('L=', L)
print('M=', M)
print(np.sum(M))
```

2. En déduire une fonction simulant la loi de X_n , et la tester.
3. Estimer $E(X_n)$

Nota bene : la valeur théorique est $E(X_n) = \frac{n}{2}$ (on ne demande pas de le montrer)

4. Estimer $P(X_{10} = 5)$