

Code de partage avec Capytale : 376d-1541121

## Préambule

Quelques précisions pour apprivoiser les commandes `random` permettant de modéliser les lois usuelles. Il faut dans un premier temps importer la librairie `random` que l'on renomme `rd`

```
import numpy.random as rd
```

### Lois finies :

- `rd.randint(a,b)` simule une v.a.r. suivant la loi  $\mathcal{U}([a, b-1])$  (**⚠ le  $b$  est exclu avec cette librairie**)
- `rd.binomial(1,p)` simule une v.a.r. suivant la loi  $\mathcal{B}(p)$
- `rd.binomial(n,p)` simule une v.a.r. suivant la loi  $\mathcal{B}(n, p)$

### Lois infinies :

- `rd.geometric(p)` simule une v.a.r. suivant la loi  $\mathcal{G}(p)$
- `rd.poisson(lambda)` simule une v.a.r. suivant la loi  $\mathcal{P}(\lambda)$

Les 5 types commandes ci-dessus renvoient un nombre entier.

### Répétition d'expériences aléatoires

Les commandes ci-dessous renvoient des tableaux de nombres entiers.

- `rd.loi(parametre1,parametre2,1)` : permet de simuler une v.a.r. suivant la loi indiquée avec son(ses) paramètre(s). On obtient un tableau de nombres entiers de taille  $1 \times 1$  (i.e. contenant un seul nombre).
- `rd.loi(parametre1,parametre2,N)` : permet de simuler une v.a.r. suivant la loi indiquée avec son(ses) paramètre(s). On obtient un tableau de nombres entiers de taille  $1 \times N$
- `rd.loi(parametre1,parametre2,[m,N])` : permet de simuler une v.a.r. suivant toutes la loi indiquée avec son(ses) paramètre(s). On obtient un tableau de nombres entiers de taille  $m \times N$

## Exercices

### Exercice 1

1. Créer un vecteur  $x$  contenant 10 simulations de la loi  $\mathcal{B}(15; 0, 8)$ . Refaire et comparer.
2. Même question avec une loi de Poisson de paramètre 1. Combien de valeurs ont été prises dans votre simulation ?

### Exercice 2 - loi binomiale

1. Concrètement, quelle expérience aléatoire permet de simuler une loi  $\mathcal{B}(30; 0, 6)$  ?
2. Méthode algorithmique : à l'aide d'une boucle `for`, écrire un programme qui simule « à la main » cette loi.
3. Quelle est la commande prédéfinie de Python qui permet de faire la même chose ?

### Exercice 3 - loi géométrique

1. Concrètement, quelle expérience aléatoire permet de simuler une loi  $\mathcal{G}(0, 1)$  ?

2. A l'aide d'une boucle `while`, écrire un programme qui permet de simuler « à la main » une loi  $\mathcal{G}(0, 1)$
3. Quelle est la commande prédéfinie de Python qui permet de faire la même chose ?

**Exercice 4** (inspiré d'Edhec E 2015)

Un joueur réalise des lancers indépendants d'une pièce truquée donnant « pile » avec la probabilité  $p$ . On note  $N$  la variable aléatoire égale au rang d'apparition du premier « pile ». Si  $N$  prend la valeur  $n$ , le joueur place  $n$  boules numérotées de 1 à  $n$  dans une urne, puis il extrait une boule au hasard de cette urne. On dit que ce joueur a gagné si le numéro porté par la boule tirée est impair. On appelle  $X$  la variable aléatoire égale au numéro de la boule extraite.

1. Montrer que avec  $m \in \mathbb{N}$ ,  $m$  est pair si et seulement si  $2 \left\lfloor \frac{m}{2} \right\rfloor = m$
2. Compléter le programme suivant afin qu'il simule l'expérience ci-dessus.

```
import numpy.random as rd
import numpy as np

p=float(input("entrer p"))
N=.....
while ..... :
    .....
X= .....
if ..... :
    .....
else :
    .....
```

3. Raccourcir ce programme en utilisant une des fonctions de la bibliothèque `numpy.random`