

# Simulation d'expériences aléatoires

Pour faire des probabilités avec Python, on utilise la bibliothèque `numpy.random` que l'on importe à l'aide de la commande

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

## 1 La commande `rd.randint`

- ▷ Choisissez un entier  $n$  et tester la commande `rd.randint(n)` plusieurs fois dans la console pour différentes valeurs de  $n$ . Qu'observez-vous ?
- ▷ Tester la commande `rd.randint(n,m)` pour différentes valeurs de  $n$  et de  $m$ . Qu'observez-vous ?
- ▷ Choisissez un entier  $p$  et ajoutez un troisième argument dans la commande `rd.randint(n,m,p)`. Qu'observez-vous ?
- ▷ En résumé, que font les commandes `rd.randint(n)`, `rd.randint(n,m)` et `rd.randint(n,m,p)` ?

- Exercice 1.1**
1. Ecrire une fonction `dé` qui simule le lancer d'un dé cubique non truqué.
  2. On lance 10 fois un dé équilibré à 6 faces. Ecrire une fonction appelée `comptagesix` permettant de déterminer le nombre de fois où le 6 apparaît au cours des 10 lancers. On pourra utiliser la fonction `dé`.

## 2 Simulation d'une épreuve de Bernoulli : commande `rd.binomial`

- ▷ Qu'est-ce qu'une épreuve de Bernoulli ?

- ▷ Donner des exemples d'expérience qui sont des épreuves de Bernoulli.

On peut simuler une épreuve de Bernoulli de paramètre  $p$  ( $p \in ]0; 1[$ ) à l'aide de la commande `rd.binomial(1,p)`.

- Exercice 2.1** On effectue des lancers d'une pièce telle que la probabilité de faire pile soit égale à  $\frac{1}{3}$ .
1. Construire une fonction `piece` qui permet de simuler un lancer de cette pièce et retourne 1 si on obtient pile et 0 si on obtient face.
  2. Construire une fonction `lancers` qui, étant donné un entier naturel  $n$  non nul, permet d'obtenir le nombre de piles obtenus au cours de  $n$  lancers. Cette fonction utilisera la fonction `piece`.

3. Construire une fonction `frequence` permettant d'obtenir la fréquence d'apparition du pile au cours de  $n$  lancers. Cette fonction utilisera la fonction `lancers`. Tester pour  $n = 10$ ,  $n = 100$ ,  $n = 1000$ ,  $n = 10000$  et commenter les résultats obtenus.
4. Construire une fonction `rang` qui retourne le numéro du lancer où on obtient pour la première fois pile. Cette fonction utilisera la fonction `piece`.

**Exercice 2.2** On considère une urne contenant 4 boules bleues et 5 boules rouges indiscernables au toucher. Ecrire une fonction qui simule le tirage d'une boule dans cette urne.

### 3 La commande `rd.random`

- ▷ Tester la commande `rd.random()` plusieurs fois dans la console. Qu'observez-vous ?
- ▷ Tester l'instruction `rd.random(n)` pour différentes valeurs de  $n \in \mathbb{N}$ . Qu'observez-vous ?
- ▷ En résumé, que fait la commande `rd.random` ?

La commande `rd.random` permet de simuler de nombreux phénomènes aléatoires pourvu qu'on la modifie correctement.

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle  $]0; 2[$ .

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle  $]0; 7[$ .

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle  $]2; 7[$ .

On peut simuler une épreuve de Bernoulli de paramètre  $p$  ( $p \in ]0; 1[$ ) à l'aide de la commande `rd.random`. L'idée est la suivante :

- on tire un réel  $t$  dans  $[0; 1]$  à l'aide de la commande `rd.random`,
- si  $t < p$ , on considère que l'épreuve est couronnée de succès,
- sinon, on considère que l'épreuve est un échec.

**Exercice 3.1** Reprendre l'énoncé de l'Exercice 2.2 et utiliser la commande `rd.random`.

L'avantage d'utiliser `rd.random` pour modéliser le tirage dans l'exercice précédent est que cette méthode s'étend au cas où il y a plus de 2 sortes de boules différentes dans l'urne.

**Exercice 3.2** On considère une urne contenant 4 boules bleues, 5 boules vertes et 2 boules jaunes indiscernables au toucher. Ecrire une fonction qui simule le tirage d'une boule dans cette urne.