

Chapitre 4 – Échantillonnage

1. Nombres aléatoires

1a. Génération en Python

En Python, le module `random` permet de **générer des nombres aléatoires**. Un module est une bibliothèque de fonctions qui doit d'abord être importée, ici avec la commande `from random import *`

On peut ensuite utiliser les fonctions :

- `random()` pour générer des nombres compris entre 0 et 1
- `randint(a,b)` pour générer des nombres entiers compris entre a et b.

Exemple : le code ci-contre :

```
1 from random import *
2 print(random())           # Nombre décimal entre 0 et 1
3 print(randint(2,5))      # Entier compris entre 2 et 5
```

peut afficher ces résultats :

```
0.316719082345 0.499167382717 0.62313068322
3                2                4
```

Rappel : avant de générer des nombres aléatoires, vous devez écrire la ligne `from random import *`.

- `random()` génère des nombres compris entre 0 et 1
- `randint(a,b)` génère des nombres entiers compris entre a et b.

Exemple 1 Écrire un programme très simple qui affiche le résultat d'un lancer de dé à six faces.

Exemple 2 À l'aide d'une boucle, écrire un programme qui affiche une suite aléatoire de 30 nombres compris entre 1 et 10.

Exemple 3 Écrire un programme qui affiche la somme de deux lancers de dé à six faces. Quels sont les nombres que peut afficher un tel programme ?

Exemple 4 A l'aide d'une structure `if/else`, écrire un programme qui simule un pile ou face : il doit afficher `Pile` ou `Face` avec équiprobabilité.

Exemple 5 Modifier votre programme précédent pour qu'il affiche `Succès` avec une probabilité de 80% et `Echec` sinon.

Exemple 1

```
from random import *
print(randint(1,6))
```

Exemple 2

```
from random import *
for k in range(30):
    print(randint(1,10))
```

Exemple 3

```
from random import *
print(randint(1,6) + randint(1,6))
```

Cela affichera des nombres entiers entre 2 et 12.

Exemple 4

```
from random import *
if randint(0,1) == 0 :
    print("Pile")
else :
    print("Face")
```

Exemple 5

```
from random import *
if random() < 0.8 :
    print("Succès")
else :
    print("Échec")
```

On utilise random pour avoir un nombre entre 0 et 1, qui a 80% de probabilité d'être inférieur à 0,8.

1b. Échantillons

Définition : Lorsqu'on réalise **plusieurs fois une expérience aléatoire**, l'ensemble des résultats obtenus est appelé **échantillon**.

Donner des **exemples d'échantillons de taille 10** pour les expériences aléatoires suivantes :

- pile ou face :
- lancer de dé à six faces :
- pile ou face, mais avec une pièce truquée qui a 80% de probabilité de tomber sur pile :

En Python, on utilise les **boucles for** pour générer des échantillons de taille voulue.

Exemple 1 Écrire un programme qui génère un échantillon de 50 lancers de dé à six faces.

Exemple 2

Écrire un programme qui génère un échantillon de 200 pile ou face avec la pièce truquée décrite ci-dessus.

Exemple 3 Modifier le programme ci-dessus pour qu'au lieu d'afficher « pile » ou « face », le programme compte le nombre de « pile ». Quel est le nombre moyen de « pile » que le programme devrait afficher ?

Exemples d'échantillons

- pile ou face : {pile, pile, face, pile, face, face, face, pile, face, face}
- lancer de dé : {4, 6, 6, 1, 2, 1, 4, 5, 2, 4}
- pile ou face truqué : {pile, face, pile, pile, pile, pile, face, pile, pile, face}

Exemple 1

```
from random import *
for k in range(50):
    print(randint(1, 6))
```

Exemple 2

```
from random import *
for k in range(200):
    if random() < 0.8 :
        print("pile")
    else :
        print("face")
```

Exemple 3

```
from random import *
n = 0
for k in range(200):
    if random() < 0.8 :
        n = n + 1
print(n)
```

Cela devrait afficher **160 en moyenne**, mais le résultat fluctue.

2. Fluctuation

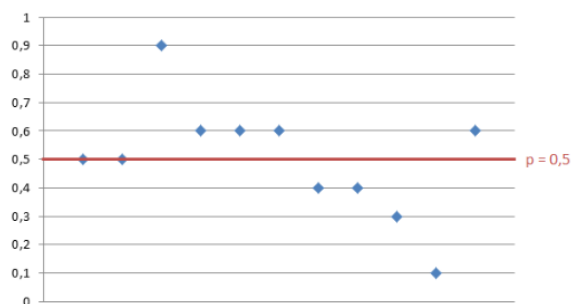
Définition : Pour une même expérience, deux échantillons ne seront pas nécessairement identiques. On appelle cela la **fluctuation d'échantillonnage**.

Si on s'intéresse à une issue de **probabilité p** , plus le nombre de répétitions n est grand, plus la **fréquence f** de cette issue dans l'échantillon **se rapproche de p** . En général, l'écart entre p et f (la marge d'erreur) est **inférieur ou égal à $\frac{1}{\sqrt{n}}$** .

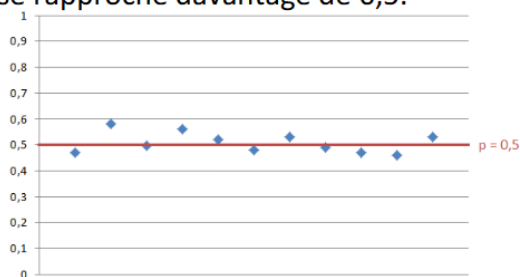
Si on effectue 10 pile ou face, la moyenne du nombre de piles obtenus est 5. Mais il est loin d'être certain qu'on obtient 5 piles à chaque fois !

Le graphique ci-contre montre la fréquence du **nombre de piles obtenus en lançant 10 pièces plusieurs fois**.

On voit que parfois, $f = 0,1$ (on n'a obtenu qu'un pile sur les 10) ou même $f = 0,9$ (on a obtenu neuf piles sur les dix).



Si on réalise **100 pile ou face**, la fréquence f de pile se rapproche davantage de 0,5.



Si on réalise **1 000 pile ou face**, la fréquence f est toujours très proche de 0,5.

