Échantillonnage et algorithmique

Une **simulation** répétée permet d'estimer une probabilité ; un **programme** permet de la répéter des milliers de fois.

Définition — échantillon, fluctuation

Un **échantillon** de taille n est une liste de n résultats d'une même expérience aléatoire. La fréquence f d'un caractère varie d'un échantillon à l'autre : c'est la **fluctuation d'échantillonnage**. Elle diminue quand n augmente.

Principe d'estimation

Quand n est grand, la fréquence observée f est proche de la probabilité p : pour environ $95\ \%$ des échantillons, $|f-p| \leq \frac{1}{\sqrt{n}}$.

Exemple

Sur 400 tirages, $f = 0{,}31$: on estime $p \in \left[0{,}31-\frac{1}{20};0{,}31+\frac{1}{20}\right] = [0{,}26;0{,}36]$.

Python : ce qu'il faut savoir

Propriété — structures de base

```
def simulation(n):
    compteur = 0
    for i in range(n):
        if random() < 0.3:
            compteur = compteur + 1
    return compteur / n
```

Affectation =, test `if / else`, boucle bornée `for i in range(n)`, boucle non bornée `while`, fonction avec `def` et `return`.

À retenir

`range(n)` parcourt les entiers de 0 à $n-1$; == teste l'égalité, = affecte.

Questions fréquentes sur ce chapitre

Qu'est-ce que la fluctuation d'échantillonnage ?

Le fait que la fréquence d'un caractère varie d'un échantillon à l'autre. Plus l'échantillon est grand, plus les fréquences se resserrent autour de la probabilité.

Quelle est la différence entre une boucle for et une boucle while ?

La boucle for répète un nombre de fois connu à l'avance ; la boucle while répète tant qu'une condition est vraie.

Comment écrire une fonction en Python ?

Avec `def nom(paramètres):`, un bloc indenté et `return` pour renvoyer le résultat.

Que signifie l'intervalle p ± 1/√n ?

Pour un échantillon de taille n, la fréquence observée f a une forte probabilité (environ $95\ \%$) d'appartenir à $\left[p-\frac{1}{\sqrt{n}};p+\frac{1}{\sqrt{n}}\right]$.