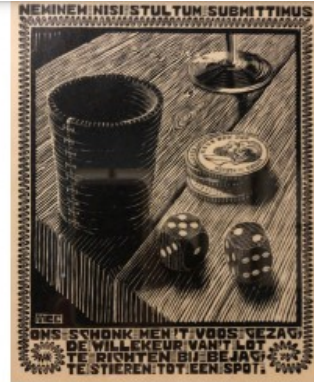


ÉCHANTILLONNAGE



▶ Vidéo <https://youtu.be/EXEcSJE31QY>

Partie 1 :

1. Définition

Exemples :

- 1) Sur l'ensemble des cartes à puce produites par une entreprise en une semaine, on en prélève 200. On dit que cet ensemble de 200 cartes à puce constitue un échantillon de la population de toutes les cartes à puce produites en une semaine.
- 2) On s'intéresse aux intentions de vote lors d'une élection. On sonde 1000 personnes en leur demandant leur intention de vote. L'ensemble de ces 1000 personnes constitue un échantillon de la population totale des électeurs.
- 3) On lance une pièce de monnaie 50 fois de suite et on note les résultats obtenus. L'ensemble de ces 50 lancers constitue un échantillon.

Définition :

Un échantillon est constitué des résultats de n répétitions indépendantes de la même expérience sur l'ensemble des personnes ou objets sur lesquels porte l'étude statistique (la population).

Un échantillon issu d'une population est donc

2. Simulation d'une expérience aléatoire

- On considère l'expérience aléatoire qui consiste à lancer un dé à 6 faces. Le programme Python suivant permet de simuler cette expérience.

La fonction `randint` renvoie un nombre aléatoire entier de 1 à 6.

→

```
from random import*  
r=randint(1,6)
```



On exécute le programme et on obtient l'affichage ci-contre. Cela signifie que le logiciel a simulé un lancer de dé et on a obtenu un « 3 ».

- La règle du jeu veut que si le résultat est « 1 » ou « 6 », on gagne. Dans le cas contraire, on perd. On répète n fois de suite cette expérience à deux issues (gagner ou perdre) consistant à lancer le dé.

Ainsi, on modifie et complète le programme Python afin de simuler n lancers de dé. Le programme affiche le nombre de fois que l'on gagne.

La variable n désigne le nombre de lancers.

→

```
from random import*
s=0
for k in range(n):
    r=randint(1,6)
    if r==1 or r==6:
        s=s+1
```

La variable s permet de compter le nombre de fois que l'on gagne (le dé s'arrête sur « 1 » ou sur « 6 »).

→

```
>>>
```

```
-
```

On exécute le programme et on obtient l'affichage ci-contre. Cela signifie que sur 10 lancers, on a gagné 3 fois.

Partie 2 : Loi des grands nombres

Calcul d'une fréquence (Rappel) :

Sur 10 lancers, on a gagné 3 fois. La fréquence de jeu gagné est égale à :

Modifions le programme afin d'afficher en sortie la fréquence de jeux gagnés sur un échantillon de n lancers de dé.

```
from random import*
s=0
for k in range(n):
    r=randint(1,6)
    if r==1 or r==6:
        s=s+1
```

Il suffit de remplacer dans la dernière ligne `return(s)` (l'effectif) par `return(s/n)` (la fréquence).

→

```
>>> dé(10)
0.2
>>> dé(100)
0.32
>>> dé(1000)
0.328
>>> dé(5000)
0.3372
>>> dé(10000)
```

On exécute le programme pour des valeurs de n de plus en plus grandes. Ci-contre les résultats obtenus à l'aide du logiciel.

Partie 3 : Estimation d'une probabilité

On se propose maintenant de répéter N fois la simulation de l'expérience aléatoire précédente. Dans chaque cas, pour n suffisamment grand, la fréquence observée f devrait être proche de la probabilité théorique $p = \frac{1}{3}$.

On veut calculer la proportion des cas pour lesquels l'écart entre f et p est inférieur ou égale à $\frac{1}{\sqrt{n}}$.

On complète alors le programme précédent avec

```
from math import*
c=0
for k in range(N):
    f=dé(n)
    if abs(f-1/3)<=1/sqrt(n):
        c=c+1
print (c/N)
```

On exécute le programme pour différentes valeurs de N en choisissant n suffisamment grand, soit $n = 10000$.

On trouve des valeurs proches de 0,95 ce qui signifie que dans 95% des cas, l'écart entre la fréquence observée f et la probabilité p est inférieur ou égale à 0,01.

En effet :

