

## Loi des grands nombres

Auteur : Thomas Léchenne

Niveaux scolaires :  
Seconde

Mots clés :  
Algorithmes/Programmation  
Probabilités/Statistiques

### ENONCE

Observer la loi des grands nombres à l'aide d'une simulation sur Python.

On ne rappellera pas ici l'énoncé mathématique rigoureux de la loi des grands nombres (ou plutôt d'une des lois des grands nombres puisqu'il existe la loi faible et la loi forte des grands nombres). Néanmoins, on peut rappeler brièvement le principe avec des termes compréhensibles par un élève de seconde. Plus on effectue de fois la même expérience aléatoire, plus les fréquences respectives des résultats obtenus semblent se rapprocher de leurs probabilités respectives. Ainsi, on se retrouve presque face à un paradoxe. Plus on effectue d'expérience au hasard, moins le dit hasard a un effet sensible.

Par exemple, lorsqu'on lance un dé équilibré à 6 faces, la probabilité d'obtenir 1 est de  $\frac{1}{6}$  (nombre de cas favorables / nombre de cas possibles). Si on lance le même dé un grand nombre de fois, la fréquence d'apparition du 1 va tendre vers cette probabilité.

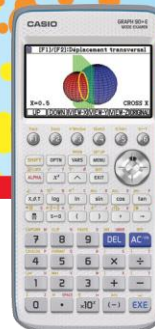
Voyons cela avec une simulation sur Python :

Commençons ce nouveau programme par charger la bibliothèque "random". Pour cela, allons chercher la commande "from random import \*" disponible dans le catalogue (). Cette bibliothèque est indispensable pour pouvoir générer un nombre entier au hasard entre 1 et 6 (avec la commande randint **SHIFT** **4**).

```
loi.py 001/002
from random import *
```

FILE RUN SYMBOL CHAR A↔a ▶

# Graph 90+E



Nous allons fabriquer une fonction Python nommée DE qui aura pour argument N, le nombre de fois où on va simuler notre lancer de dé.

Nous aurons besoin de deux variables : la première (appelons la C) qui comptabilisera le nombre de 1 obtenus et la deuxième (nommée L) qui sera une liste des fréquences tous les 20 lancers (ce qui nous permettra de suivre l'évolution).

La variable L est initialisée comme une liste vide, C à 0.

Nous allons reproduire chacun des lancers un par un à l'aide d'une boucle Pour (pour i allant de 1 à N ce qui se traduit par "for i in range(1,N+1) :").

Simulons le lancer de dé en lui-même : la commande randint(1,6) permet de générer un nombre entier aléatoire entre 1 et 6.

On teste si ce nombre est 1 : dans ce cas, on ajoute 1 à notre compteur. Sinon, il ne se passe rien.

A noter que le test d'égalité se fait avec un double signe égal (pour le différencier de l'affectation qui se fait, elle, avec le simple signe égal).

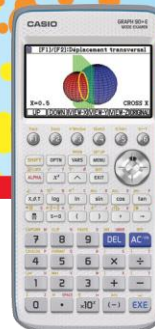
```
loi.py 003/004
from random import *
def DE(N):
```

```
loi.py 004/005
from random import *
def DE(N):
    C=0
    L=[]
```

```
loi.py 006/007
from random import *
def DE(N):
    C=0
    L=[]
    for i in range(1,N+1):
```

```
loi.py 006/007
from random import *
def DE(N):
    C=0
    L=[]
    for i in range(1,N+1):
        if randint(1,6)==1:
```

# Graph 90+E



Pour ajouter 1 à notre variable C, on peut taper "C=C+1" ou "C+=1".

Ensuite, tous les 20 lancers, on ajoute à notre liste L, la fréquence du 1.

Pour tester si le nombre de lancers est un multiple de 20, il suffit de tester si le reste de la division euclidienne de i par 20 est nul. C'est-à-dire, en langage Python, si  $i\%20$  est égal à 0.

Dans ce cas, on ajoute la fréquence des 1 déjà obtenus à notre liste. Pour que cela soit plus lisible, on peut arrondir cette fréquence à 3 chiffres après la virgule avec la commande "round" (disponible dans le catalogue **SHIFT** **4**).

Pour ajouter une valeur à notre liste il suffit de l'ajouter entre crochets à L.

La fréquence des 1 obtenus à ce moment-là est égale à  $C/i$ .

Il ne reste plus qu'à renvoyer la liste L.

Attention à sortir des indentations de la structure conditionnelle et de la boucle (mais pas de celle de la fonction DE).

On peut utiliser cette fonction pour déterminer les fréquences des 1 obtenus tous les 20 lancers pour un total de 100 ou 200 lancers.

On voit que cette fréquence se rapproche de plus en plus de  $\frac{1}{6}$  (environ 0,167 arrondi au millième) au fur et à mesure que le nombre de lancers augmente. Cela permet d'illustrer les bases de la loi des grands nombres.

```
loi.py 009/010
C=0
L=[]
for i in range(1,N+1):
    if randint(1,6)==1:
        C+=1
    if i%20==0:
        |
```

```
loi.py 009/010
=0
=[]
or i in range(1,N+1):
    if randint(1,6)==1:
        C+=1
    if i%20==0:
        L+=[round(C/i,3)]
```

```
loi.py 010/010
L=[]
for i in range(1,N+1):
    if randint(1,6)==1:
        C+=1
    if i%20==0:
        L+=[round(C/i,3)]
return L
```

```
>>>DE(200)
[0.25, 0.25, 0.25, 0.
>>>DE(100)
[0.3, 0.2, 0.233, 0.1
>>>DE(200)
◀ 0.169, 0.156, 0.161
>>>
RUN A↔a CHAR
```