

LOI DES GRANDS NOMBRES

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: 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 :

1. Recopier et exécuter le programme suivant:

Que fait ce programme?

A quoi sert la ligne *from random import **?

Que signifie l'instruction *randint(1,6)*?

```
des.py 001/005
from random import *
for i in range(1,11):
    print(randint(1,6))
```

2. On a voulu créer une fonction *des(n)* qui renvoie la fréquence d'apparition du 1 lorsqu'on simule *n* lancers de dé. Trois erreurs ont été commises dans le script ci-contre.

Corriger les trois erreurs.

```
from random import *
def des(n):
    c=0
    for i in range(0,n+1):
        if randint(1,6)=1:
            c=c+1
    return c/n
```

3. Utiliser la fonction *des* pour faire à chaque fois 3 simulations de 50, 100, 500 et 1000 lancers de dés et noter à chaque fois la fréquence d'apparition du 1. Que peut-on remarquer?

1. Toutes les commandes sont accessibles dans le **CATALOG** (SHIFT 4).
En exécutant le programme avec F2 (RUN), on observe que ce programme permet d'afficher 10 nombres entiers aléatoires entre 1 et 6.

```
4
3
2
6
4
1
>>>|
RUN
```

La ligne **from random import *** permet de charger toute la bibliothèque **random** et donc de pouvoir utiliser la fonction **randint**. Si on enlève cette ligne on obtient **NameError: name 'randint' is not defined**.

```
* SHELL Initialized *
>>>from des import *
File "des.py", line 3
NameError: name 'rand
>>>|
RUN
```

En réalité on aurait pu écrire **from random import randint** puisque c'est la seule fonction du module **random** qu'on utilise ici.

randint(1,6) permet de générer un nombre aléatoire entier entre 1 et 6 (inclus). **Rand** vient de **random** qui signifie aléatoire en anglais et **int** vient de **integer** qui signifie nombre entier en anglais.

On n'hésitera pas à faire d'autres essais pour bien comprendre par exemple:

```
des.py 004/005
from random import *
for i in range(0,5):
    print(randint(1,2))
```

2. Voici les trois erreurs:

- Il faut que la boucle se répète **n** fois. Or lorsqu'on écrit **range(0,n+1)** on va de 0 à n donc on répète la boucle n+1 fois. Il faut écrire **range(1,n+1)** ou alors **range(0,n)**.
- Il faut indenter tout le "bloc if" pour cela on peut ajouter des espaces en appuyant sur (ALPHA □).
- Il faut ajouter un **=** en effet le test d'égalité se fait avec **==**

```
def des(n):
    c=0
    for i in range(1,n+1):
        if randint(1,6)==:
            c=c+1
    return c/n
```

3. 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.

```
>>>des(50)
0.22
>>>des(50)
0.3
>>>des(50)
0.16
```

```
>>>des(100)
0.15
>>>des(100)
0.14
>>>des(100)
0.24
```

```
>>>des(500)
0.162
>>>des(500)
0.158
>>>des(500)
0.156
```

```
>>>des(1000)
0.168
>>>des(1000)
0.171
>>>des(1000)
0.161
```