

## Une somme de hasards

### Compétences visées

- **modéliser**, faire une simulation, valider ou invalider un modèle ;
- **raisonner**, démontrer, trouver des résultats partiels et les mettre en perspective ;
- **calculer**, appliquer des techniques et mettre en œuvre des algorithmes.

« En classe de seconde, on formalise la notion de loi (ou distribution) de probabilité dans le cas fini en s'appuyant sur le langage des ensembles et on précise les premiers éléments de calcul de probabilités. On insiste sur le fait qu'une loi de probabilité (par exemple une équiprobabilité) est une hypothèse du modèle choisi, et ne se démontre pas. Le choix du modèle peut résulter d'hypothèses implicites d'équiprobabilité (par exemple, lancers de pièces ou dés équilibrés, tirage au hasard dans une population) qu'il est recommandable d'explicitier : **il peut aussi résulter d'une application d'une version vulgarisée de la loi des grands nombres, où un modèle est construit à partir de fréquences observées pour un phénomène réel** (par exemple : lancer de punaise, sexe d'un enfant à la naissance). Dans tous les cas, on distingue nettement le modèle probabiliste abstrait et la situation réelle. »

Il est précisé dans cet extrait de programme pour la classe de 2<sup>nde</sup> que les fréquences sont observées sur un phénomène réel ; on l'élargit ici à une simulation numérique dans le cas où la probabilité de réalisation de l'événement attendu ne se calcule pas facilement. Si le fait de réaliser des essais 'physiques' est important, une simulation numérique est un complément intéressant comme illustration de « la loi des grands nombres ».

### Situation déclenchante

#### Somme de quatre dés ...

On lance quatre dés à six faces (dés bien équilibrés) et on s'intéresse à la somme des numéros sortis.

Quelle est la probabilité que cette somme soit inférieure ou égale à 18 ?



### Problématique

Comment traiter cette situation : modèle probabiliste ou simulation ?

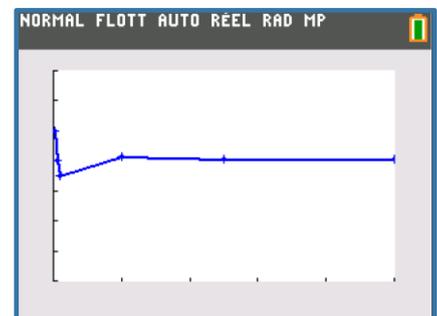
## Fiche méthode

### Proposition de résolution

On pourra effectuer des simulations grâce à un programme sous Python.

- La fonction **simul()** qui a pour paramètre  $n$ , le nombre de répétitions, simule l'expérience d'un lancer de quatre dés à six faces.
- On effectue la somme des résultats obtenus et on teste si le résultat est inférieur ou égal à 18. Si tel est le cas, un compteur est incrémenté. La fonction retourne la fréquence de cas positifs.
- On peut d'une part observer l'évolution de la fréquence en fonction du nombre d'essais. On visualise alors une stabilisation de cette fréquence vers une certaine valeur, illustrant ainsi la « loi des grands nombres ».
- D'autre part, on peut dès à présent observer la fluctuation d'échantillonnage en testant plusieurs fois la fonction **simul( $n$ )** avec la même valeur de  $n$ .

```
PYTHON SHELL
>>> simul(10)
0.8
>>> simul(1000)
0.8970000000000001
>>> simul(5000)
0.9013999999999999
>>> simul(10000)
0.9002000000000001
>>> simul(10000)
0.8978
>>> |
Fns... a A # Outils Éditer Script
```



### Remarque

Quelle que soit la méthode utilisée, il faudra un générateur de nombres (pseudo) aléatoires et donc importer la bibliothèque « random » par « **from random import \*** » |

Cette importation est en préambule du code.

On peut aussi choisir de n'importer que la fonction **randint** issue de la bibliothèque random en saisissant « **from random import randint** ». |

Lorsque l'on saisit « **from random import \*** », cela sous-entend que l'on importe toutes les fonctions de cette bibliothèque.

```
ÉDITEUR : LOIGDNBR
LIGNE DU SCRIPT 0011
from random import *
from random import randint
_
Fns... a A # Outils Exéc Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

### Etapes de résolution

La fonction **simul()** simule le lancer de quatre dés. Elle a pour paramètre  $n$  qui représente le nombre de répétitions.

- On initialise un compteur (représenté par la variable  $c$ ) à 0.
- Chaque fois que la somme des quatre nombres entiers aléatoires compris entre 1 et 6 est inférieure ou égale à 18, ce compteur est incrémenté.
- Au final, le compteur est égal au nombre de cas favorables.
- En le divisant par le nombre d'essais ( $n$ ), la fonction retourne la fréquence de réussite de l'expérience aléatoire.

```
EDITEUR : LOIGDNBR
LIGNE DU SCRIPT 0009
from random import randint

def simul(n):
    c=0
    for i in range(n):
        if randint(1,6)+randint(1,6)
           +randint(1,6)+randint(1,6)<
           =18:
            c=c+1
    return c/n

Fns... a A # Outils Exéc Script
```

### Remarque

Pour un élève de 2<sup>nde</sup>, il n'est peut-être pas évident que :

- $\text{randint}(1,6) + \text{randint}(1,6) + \text{randint}(1,6) + \text{randint}(1,6)$
- $\text{randint}(1,6) \times 4$
- $\text{randint}(4,24)$

ne représentent pas la même simulation.

Il peut être intéressant, pour convaincre les sceptiques, de tester par chacune de ces simulations la valeur de la fréquence de réalisation de l'expérience « être inférieur ou égal à 18 ».

On observe des valeurs significativement différentes qui confirment que ces simulations ne sont pas équivalentes.

```
ÉDITEUR : LOIGDNBR
LIGNE DU SCRIPT 0030
def sim2(n):
    c=0
    for i in range(n):
        if randint(1,6)*4<=18:
            c=c+1
    return c/n

def sim3(n):
    c=0
    for i in range(n):
        if randint(4,24)<=18:
            c=c+1
    return c/n

Fns... a A # Outils Exéc Script
```

```
PYTHON SHELL
>>> simul(1000)
0.9
>>> sim2(1000)
0.64
>>> sim3(1000)
0.706
>>> |

Fns... a A # Outils Éditer Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

### Un complément

On peut tenter une approche 'probabiliste' en dénombrant le nombre de cas favorables.

**Lorsque l'on lance quatre dés à six faces, dans combien de cas la somme des numéros sortis est-elle inférieure ou égale à 18 ?**

Si on se lance dans un dénombrement « à la main », cela risque d'être long et fastidieux, et le risque de se tromper n'est pas négligeable.

**Un programme peut dénombrer le nombre de cas favorables** ; la calculatrice ne rechigne pas en effet à effectuer ce type de tâche longue et fastidieuse.

On va donc tester toutes les sommes de quatre entiers compris entre 1 et 6 et comptabiliser le nombre de cas où la somme est inférieure ou égale à 18.

Ceci permet d'obtenir la probabilité de réalisation de l'événement voulu, et de comparer a posteriori les fréquences obtenues avec cette probabilité.

On obtient  $p = \frac{1170}{1296} \sim 0,90278$ .

```
ÉDITEUR : LOIGDNBR
LIGNE DU SCRIPT 0018
return c/n

def test():
    c=0
    for i in range(1,7):
        for j in range(1,7):
            for k in range(1,7):
                for l in range(1,7):
                    if i+j+k+l<=18:
                        c=c+1
    return c/6**4_

Fns... a A # Outils Exéc Script
```

```
PYTHON SHELL
>>> sim(1000)
0.9120000000000001
>>> sim(1000)
0.9130000000000001
>>> sim(1000)
0.8880000000000001
>>> test()
0.9027777777777779
>>> |

Fns... a A # Outils Éditer Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

