Simuler un jet de dé

Préparation du TP

On place dans le répertoire de travail une bibliothèque `paquetdes` (voir annexe 1). Cette bibliothèque contient des fonctions *de*1, *de*2, *de*3, *de*4 et *de*5 qui retournent un entier strictement positif :

• la fonction *de*1 choisit un nombre de type **float**, pseudo-aléatoire, qui suit la loi uniforme sur $[0; 6[$, calcule l'arrondi à l'unité de ce nombre, et le retourne s'il est non nul. S'il est nul la fonction retourne 1,
• la fonction *de*2 retourne un nombre entier compris entre 1 et 6. Le nombre retourné suit la loi équirépartie sur les 6 issues possibles,
• la fonction *de*3 fait la même chose que la fonction *de*2 mais avant de retourner l'entier, elle change un 5 en 4 avec une probabilité de 5%,
• la fonction *de*4 affichera dans l'ordre suivant et de façon cyclique les nombres : 6, 2, 4, 3, 5, 1, 2, 5, 1, 3, 4, 6, 4, 1, 5, 2, 6, 3,
• la fonction *de*5 retourne un nombre entier compris entre 1 et 7. Le nombre retourné suit la loi équirépartie sur les 7 issues possibles.

On peut ajouter d'autres fonctions. Par exemple, une fonction *de*6 qui ne retourne jamais de 4. Etc.

Objectif du TP

Échanger avec la classe sur la notion de hasard, calculer des fréquences. Le but du TP est de déterminer la fonction *de* qui simule le mieux un lancé de dé équilibré à six faces. Plusieurs activités peuvent être proposées. Par exemple :
Le TP(Version 1) offre une question ouverte à une classe de seconde. Les élèves pourront éliminer certaines fonctions facilement (par exemple, la fonction *de*5 qui donne régulièrement un 7). Les élèves seront contraints d'écrire des programmes s'ils veulent obtenir des informations sur ces fonctions.
Le TP(Version 2) est a proposer à une classe de seconde. Le déroulement de la séance est dirigé.
Le TP(Version 3) demande de réaliser une tâche difficile. Il peut être envisagé de le donner en question supplémentaire pour des bons élèves.
Le TP(Version 4) est abordable dès que la bibliothèque `matplotlib` a été rencontrée.
Le TP(Version 5) peut être proposé à une classe de première S.

Proposition de TP (version 1)

1. Tapez et exécutez plusieurs fois le programme suivant:

```
from paquetdes import *

print("le dé n°1 donne :",de1())
print("le dé n°2 donne :",de2())
print("le dé n°3 donne :",de3())
print("le dé n°4 donne :",de4())
print("le dé n°5 donne :",de5())
```

2. Déterminez laquelle des fonctions $de1, de2, de3$, $de4$, ou $de5$ simule le mieux un lancé de dé équilibré à six faces.

Proposition de TP (version 2)

1. Tapez et exécutez plusieurs fois le programme suivant:

```
from paquetdes import *

print("le dé n°1 donne :",de1())
print("le dé n°2 donne :",de2())
print("le dé n°3 donne :",de3())
print("le dé n°4 donne :",de4())
```

2. Étude de la fonction $de1$.

- **a.** Écrivez un programme qui affiche 10 valeurs obtenues avec la fonction $de1$.
- **b.** Modifiez le programme pour que, de plus, l'utilisateur choisisse le nombre de valeurs à afficher.
- **c.** Modifiez le programme pour que, de plus, il affiche le nombre de fois où l'on obtient le nombre 1.
- **d.** Modifiez le programme pour que, de plus, il affiche le nombre de fois où l'on obtient le nombre 2, le nombre 3, ..., le nombre 6.
- **e.** Modifiez le programme pour qu'il n'affiche plus des effectifs mais des fréquences.
- **f.** Que concluez-vous au sujet de la fonction $de1$?

3. Étudiez de la même façon les fonctions $de2$, $de3$ et $de4$.

4. Quelle fonction simule le mieux le lancé d'un dé équilibré à six faces ?

Proposition TP (version 3)

1. Tapez et exécutez plusieurs fois le programme suivant:

```
from paquetdes import *

print("le dé n°1 donne :",de1())
print("le dé n°2 donne :",de2())
print("le dé n°3 donne :",de3())
print("le dé n°4 donne :",de4())
```

2. Écrivez un programme qui détecte si une liste de 10 000 entiers positifs donnée contient un cycle (par exemple, $[1, 7, 5, 1, 7, 5, 1, 7, 5, 1, 7, \ldots]$ contient un cycle de longueur 3 dont le motif est [1,7,5]). Vous pourrez vous limiter à des cycles dont la longueur est comprise entre 1 et 20.

3. Testez le programme avec la fonction *de*4. Que concluez-vous ?

Proposition TP (version 4)

1. Tapez et exécutez plusieurs fois le programme suivant:

```
from paquetdes import *

print("le dé n°1 donne :",de1())
print("le dé n°2 donne :",de2())
print("le dé n°3 donne :",de3())
print("le dé n°4 donne :",de4())
```

2. Pour la fonction *de*1, écrivez un programme qui affiche les fréquences d'obtention des nombres $1, 2, \ldots, 6$ en fonction du nombre de lancers.

3. Modifiez le programme en utilisant la bibliothèque `matplotlib` pour tracer un graphique donnant la fréquence d'obtention des nombres $1, 2, \ldots, 6$ en fonction du nombre de lancers qui varie entre 1 et 10 000.

Proposition TP (version 5)

1. Tapez et exécutez plusieurs fois le programme suivant:

```
from paquetdes import *


print("le dé n°1 donne :",de1())
print("le dé n°2 donne :",de2())
print("le dé n°3 donne :",de3())
print("le dé n°4 donne :",de4())
```

2. **a.** Donnez les bornes d'un intervalle de fluctuation au seuil de 95% de la fréquence de 1 obtenus pour un échantillon de 10 000 lancers d'un dé équilibré à six faces.

b. Écrivez un programme qui affiche si on peut considérer que la fonction $de1$ simule correctement un dé équilibré à six faces au risque d'erreur de 5%.

c. Modifiez le programme pour qu'il détermine l'ensemble des dés convenables.

Éléments pour des réponses

TP (version 1)

Rien à dire. Les programmes ci-dessous devraient convenir.

TP (version 2)

Pour la question **2.a.** :

```
from paquetdes import *
for i in range(10):
    print(de1())
```

Pour la question **2.b.** :

```
from paquetdes import *
texte=input("combien de lancers ? ")
n=int(texte)
for i in range(n):
    print(de1())
```

Pour la question **2.c.** :

```
from paquetdes import *
texte=input("combien de lancers ? ")
n=int(texte)
effectif=0
for i in range(n):
    a=de1()
    print(a)
    if a==1:
        effectif=effectif+1
print("le numéro 1 apparaît ", effectif, " fois.")
```

Pour la question **2.d.**: c'est le bon moment pour mesurer l'importance du type `list` . Comparez avec un programme qui n'utilise pas ce type.

```
from paquetdes import *
texte=input("combien de lancers ? ")
n=int(texte)
effectif=[0,0,0,0,0,0,0]          # le terme de rang 0 ne servira pas
for i in range(n):
    a=de1()
    print(a)
    effectif[a]=effectif[a]+1     # le terme de rang i contient le nombre de i obtenus
for i in range(1,7):
    print("le numéro ",i," apparaît ", effectif[i], " fois.")
```

Pour la question **2.e.**:

```
from paquetdes import *
texte=input("combien de lancers ? ")
n=int(texte)
effectif=[0,0,0,0,0,0,0]          # le terme de rang 0 ne servira pas
for i in range(n):
    a=de1()
    print(a)
    effectif[a]=effectif[a]+1     # le terme de rang i contient le nombre de i obtenus
for i in range(1,7):
    print("le numéro ",i," apparaît avec une fréquence de ", effectif[i]/n)
```

Commentaire : Il est intéressant de comparer la vitesse d'exécution du programme si on enlève la ligne qui contient «print(a)».

Pour la question **2.f** : on conclut que la fonction *de*1 ne convient pas pour simuler un dé équilibré.

Pour la question **3.** : les fonctions *de*2 et *de*4 sont intéressantes.

Pour la question **4.** : Une étude des nombres sortis permet d'éliminer la fonction *de*4.

TP (version 3)

Pour la question **3.**:

```
from paquetdes import *

def listetest(L,long):              # extraction du motif de la longueur souhaitée
    liste=L[0:long]
    return liste

def tester(L,T):                    # teste si le motif T cycle dans la liste L
    n=len(T)
    i=0
    j=0
    test=True
    while test and i<len(L):
        if L[i]!=T[j]:
            test=False
        i=i+1
        j=(j+1)%n

    return test



L=[]                                # création de la liste que l'on veut tester
for i in range(10000):
    L=L+[de4()]                     # Question : Quel est le rôle des crochets autour de de4() ?

reponse=False                       # À priori il n'a pas de cycle dans L
longueur=1                              # longueur du cycle testé
while reponse==False and longueur<21:
    T=listetest(L,longueur)
    reponse=tester(L,T)
    longueur=longueur+1

if reponse:
    print("il y a un cycle de période ", longueur-1)
else:
    print("pas de cycle détecté !")
```

TP (version 4)

Pour la question **3.**:

```
from paquetdes import *
from matplotlib.pyplot import *

N=10000
axis([0,N,0,0.3])
plot([0,N],[0.1667,0.1667],"r-")

def frequence(n):
    liste=[0,0,0,0,0,0,0]
    for i in range(n):
        a=de4()
        liste[a]=liste[a]+1
    for i in range(1,7):
        liste[i]=liste[i]/n

    return liste

for n in range(1,N,50):
    L=frequence(n)
    plot(n,L[1],"b+")        # du bleu
    plot(n,L[2],"y+")        # du jaune
    plot(n,L[4],"c+")        # du cyan
    plot(n,L[4],"m+")        # du magenta
    plot(n,L[5],"g+")        # du vert
    plot(n,L[6],"k+")        # du noir

show()
```

TP (version 5)

Rien à dire. (L'intervalle est déterminé avec la calculatrice ou bien avec des fonctions existantes ou bien des fonctions à écrire. Le TP (version 2) permet de calculer les fréquences souhaitées).

Annexe 1

Voici le contenu du fichier «paquetdes.py»

```
import random
B=362514643152153426

def de1():
    n=random.random()*6        # n est dans [0;6[
    if n<0.5:
        return 1
    a,b=int(n)+1-n,n-int(n)    # distances entre n et les entiers les plus proches
    if a<b:
        return int(n)+1
    else:
        return int(n)

def de2():
    n=random.randint(1,6)      # une valeur entière dans [1;6] (équirépartion)
    return n

def de3():
    n=random.randint(1,6)
    if n==5:
        if random.randint(1,100)<=95:
            n=5                        # les 5 sont sous représentés
        else:
            n=4                        # les 4 sont sur représentés
    return n

def de4():                    # un cycle apparaît dans les nombres obtenus
    global B
    r=B%10
    B=B//10
    if B<1:
        B=362514643152153426
    return r

def de5():
    return random.randint(1,7)
```