

Travaux Pratiques 3 : Python et simulation

Éléments de correction

I Utilisation du module random

Le module random permet de générer des nombres aléatoires. Dans ce module que l'on doit importer on peut utiliser les deux fonctions Python suivantes :

- **randint(a,b)** renvoie au hasard un nombre entier compris entre a et b inclus
- **random()** renvoie un nombre décimal strictement compris entre 0 et 1

Dans votre fichier *Nom-TP3-Python* commencez par écrire l'instruction ci-dessous, ainsi vous aurez accès aux fonctions de ce module : `from random import *`

Exercice 1

1. Recopier le script suivant :

```
if randint(0,1)==0:
    print("pile")
else:
    print("face")
```

Le programme ci-dessus affiche-t-il pile ? :

- a) toujours b) une fois sur deux c) jamais d) la première fois qu'il est exécuté

b) une fois sur deux en effet la fonction randint(0,1) renvoie une fois sur deux 0 qui correspond à pile

2. Recopier le script suivant :

```
def pileouface():
    if randint(0,3)==0:
        return"P"
    else:
        return"F"
```

La fonction pileouface permet-elle de simuler le lancer d'une pièce équilibrée ?

la fonction pileouface renvoie pile une fois sur 4 la pièce n'est pas équilibrée

3. Afin de confirmer votre réponse précédente tester la fonction suivante pour n=100, puis n=1000 enfin n=20000
Quelle est le rôle de cette fonction Python ?

```
def totalpiles(n):
    t=0
    for i in range(n):
        if pileouface()=="P":
            t=t+1
    return(t/n)
```

```
totalpiles(100)
0.23
totalpiles(1000)
```

```
0.251
totalpiles(20000)
0.2472
```

la fonction totalpiles renvoie la fréquence des piles obtenus lors de n lancers qui est proche de 0.25

II Lancers de dés

Exercice 2

On lance deux dés non truqués à 6 faces numérotées de 1 à 6. On s'intéresse à la probabilité que la somme des deux dés soit égale à six

```
c=0
for i in range (100):
    d1=randint(1,6)
    d2=randint(1,6)
    if d1+d2==6:
        c=c+1
print(c/100)
```

1. Que représente la valeur affichée par la console lors de l'exécution de ce script ?

La console affiche la fréquence d'apparition de la somme 6 lorsqu'on simule en Python 100 lancers de deux dés équilibrés et que l'on ajoute les nombres obtenus

2. Modifier le programme pour que l'utilisateur puisse choisir le nombre de lancers

```
c=0
n=int(input("nombre de lancers?"))
for i in range (n):
    d1=randint(1,6)
    d2=randint(1,6)
    if d1+d2==6:
        c=c+1
print(c/n)
```

3. Déterminer la probabilité que la somme des deux dés soit égale à 6

Lors du lancers de 2 dés on obtient 36 couples dont 5 donnent une somme égale à six, dans ce cas d'équiprobabilités la probabilité cherchée est donc $\frac{5}{36}$

Exercice 3

On lance deux dés, et on observe l'écart entre les deux numéros obtenus. Par exemple si les dés ont donné 3 et 5 le résultat de l'expérience aléatoire est 2

1. Donner l'univers Ω de cette expérience aléatoire

$\Omega = \{0; 1; 2; 3; 4; 5\}$

2. Écrire, en Python, une fonction `ecart()` pour qu'elle simule l'expérience

```
def ecart():
    d1=randint(1,6)
    d2=randint(1,6)
    e=abs(d1-d2)
    return e
```

3. On veut calculer la fréquence d'apparition de chacune des issues si l'on répète 1000 fois l'expérience. Pour cela, on crée six compteurs (un par issue), qu'on initialise à 0. Compléter le script ci-dessous et l'exécuter. Quelles fréquences avez-vous obtenues pour chacune des issues ? Compléter les probabilités afin de comparer

```
e0=0
e1=0
e2=0
e3=0
e4=0
e5=0
for k in range (1000):
    e=ecart()
    if e==0:
        e0=e0+1
    elif e==1:
        e1=e1+1
    elif e==2:
        e2=e2+1
    elif e==3:
        e3=e3+1
    elif e==4:
        e4=e4+1
    elif e==5:
        e5=e5+1
print(e0/1000,e1/1000,e2/1000,e3/1000,e4/1000,e5/1000)
```

x_i	0	1	2	3	4	5
fréquences	0.169	0.282	0.226	0.016	0.0101	0.0062
$P(X = x_i)$	$\frac{6}{36} = \frac{1}{6}$	$\frac{10}{36} = \frac{5}{18}$	$\frac{8}{36} = \frac{2}{9}$	$\frac{6}{36} = \frac{1}{6}$	$\frac{4}{36} = \frac{1}{9}$	$\frac{2}{36} = \frac{1}{18}$

III Schéma de Bernoulli

Exercice 4

On considère une urne qui contient 7 boules noires et 3 boules blanches. On tire au hasard une boule dans l'urne, on note sa couleur, puis on la remet.

1. Expliquer pourquoi la fonction Python ci-dessous permet de réaliser une simulation de l'expérience précédente en Python

```
def tirage():
    a=randint(1,10)
    if a<=3:
        return "B"
    else:
        return "N"
```

La fonction `randint(1,10)` permet de générer un entier entre 1 et 10, il y a 30 % de chance qu'il soit inférieur ou égal à 3, on peut considérer dans ce cas que l'on a tiré une boule blanche et que sinon on a tiré une boule noire.

2. Écrire un programme qui effectue 1000 tirages et donne la fréquence des boules blanches observées

```
def tirage():
    a=randint(1,10)
    if a<=3:
        return "B"
    else:
        return "N"

b=0
n=0
for k in range(1000):
    if tirage()=="B":
        b=b+1
    else:
        n=n+1
print("la fréquence des B est :", b/1000, "et la fréquences des N est", n/1000)
```

3. Tester ce programme et noter les résultats obtenus

la fréquence des N est : 0.291 et la fréquences des B est 0.709

IV Problème du Grand Duc de Toscane

Un jeu en vogue à la cour de Florence au début du 17^e siècle consistait à lancer trois dés cubiques équilibrés et à faire la somme des numéros des trois faces. Le Duc observa que la somme 10 avait tendance à apparaître plus souvent que 9, alors qu'il y a autant de possibilités d'écrire 9 et 10 comme somme de trois numéros de face.

1. Écrire une fonction `duc()` qui simule le lancer de trois dés et retourne la somme obtenue.

```
def duc():
    d=randint(1,6)+randint(1,6)+randint(1,6)
    return d
```

2. Afficher dans une liste les résultats obtenus pour 1000 lancers puis calculer les fréquences d'apparitions de la somme 9 et de la somme 10

```
L=[0]
for i in range(1000): L.append(duc())
print(L)
n=0
d=0
for i in range(len(L)):
    if L[i]==9: n=n+1
    if L[i]==10: d=d+1
print("freq 9",n/1000,"freq 10",d/1000)
freq 9 0.124 freq 10 0.109
```

3. Comparer avec les probabilités

Le lancer de trois dés aboutit à un ensemble Ω contenant $6^3 = 216$ éventualités équiprobables. Pour obtenir une somme égale à 9 il y a 25 éventualités :

6 possibilités pour $1+2+6=9$			6 possibilités pour $1+3+5=9$			3 possibilités pour $1+4+4=9$			3 possibilités pour $2+2+5=9$			6 possibilités pour $2+3+4=9$			1 possibilité pour $3+3+3=9$		
1 ^{er} dé	2 ^e dé	3 ^e dé	1 ^{er} dé	2 ^e dé	3 ^e dé	1 ^{er} dé	2 ^e dé	3 ^e dé	1 ^{er} dé	2 ^e dé	3 ^e dé	1 ^{er} dé	2 ^e dé	3 ^e dé	1 ^{er} dé	2 ^e dé	3 ^e dé
														</			