TP 4. Simuler l'aléatoire

October 15, 2021

1 Rappel

Pour faire des calculs efficacement et afficher des graphiques, on importera toujours les modules numpy et matplotlib via les commandes :

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Dans ce TP, on simule des variables aléatoires, c'est à dire qu'on simule le résultat (par essence aléatoire) d'un lancer de dé, d'un tirage à pile ou face ou de tout autre tirage "au hasard"… On va explorer ici la commande random() (de la bibliothèque random) de Python et l'utiliser dans divers contextes, en particulier pour simuler d'autres variables aléatoires.

[ ]:
```
from random import random
```

Exécuter le code suivant (plusieurs fois) pour voir les résultats fournis par la fonction random.

[ ]:
```
for i in range(5):
    x = random()
    print(x)
```

2 Introduction à random

Le premier objectif est de montrer que la variable aléatoire $X = \text{random}()$ suit la loi uniforme sur $[0, 1]$, ce qui signifie que :

- la probabilité que X soit plus grand que 1 est égale à 0, on notera cela

$$\mathbb{P}(X \geq 1) = 0;$$

- la probabilité que X soit plus petit que 0 est égale à 0, on notera cela

$$\mathbb{P}(X \leq 0) = 0;$$

- la probabilité que X soit dans l'intervalle $[a, b]$, avec $0 \leq a \leq b \leq 1$ est égale à $b - a$, on notera cela

$$\mathbb{P}(a \leq X \leq b) = b - a.$$

2.1 Coder une fonction tirages, qui prend en argument deux réels a, b ($a < b$), un entier positif n, et qui :

- fait n tirages aléatoires avec random() ;
- compte le nombre de fois où le résultat est entre a et b, on note m ce résultat ;
- renvoie finalement $\frac{m}{n}$.

2.2 Comment intepréter le nombre m dans la fonction tirages ?

2.3 Fixer a et b (par exemple $a = 0.2$ et $b = 0.7$) et tracer le graphe de la fonction tirages$(a, b, \cdot)$, sur $[1, 300]$. On tracera sur le même graphe et en rouge, la droite passant par les points $(1, b - a)$ et $(300, b - a)$.

2.4 Que constatez-vous? Pouvait-on s'y attendre?

3 D'autres lois

Pour illustrer graphiquement les variables aléatoires de cette section, on pourra utiliser la commande plt.hist :

[ ]:
```
valeurs = [1, 2, 1, 1, 1, 3, 2, 4, 2, 1, 3, 1, 4, 2, 2]

plt.figure(figsize = (9, 7))
plt.hist(valeurs)
```

3.1 Simuler une variable aleatoire Y qui vaut 1 avec probabilité 0.5 et -1 avec probabilité 0.5. On pourra évidemment utiliser les résultats des questions précédentes...

3.2 Simuler une variable aleatoire Z qui vaut :

- 0 avec probabilité 0.25
- 1 avec probabilité 0.5
- 2 avec probabilité 0.25

3.3 Simuler le résultat d'un lancer de dé à six faces, puis celui d'un lancer de deux à six faces.

4 Ruine au casino

Un joueur arrive au casino avec x euros ($0 < x < 100$) et il espère gagner 100 euros. Si tel est le cas, il s'arrête mais il est aussi contraint de s'arrêter si, bien sûr, il ne lui reste plus d'argent, à chaque tirage il perd un euro ou gagne un euro..

On suppose qu'à chaque fois qu'il joue, il a une probabilité 0.5 de gagner et de perdre.

4.1 Coder une fonction partie qui prend en argument un entier x (entre 1 et 99), qui simule une partie et renvoie deux informations :

- si le joueur a gagné ou perdu ;
- le nombre de tours que ça a pris.

4.2 En déduire une fonction proba qui prend en argument un entier x (entre 1 et 99) et un entier strictement positif n, et qui :

- fait n parties avec la fonction partie ;
- compte le nombre de fois où le joueur gagne, on note m ce résultat ;
- renvoie finalement $\frac{m}{n}$.

4.3 Comment intepréter le nombre m dans la fonction proba ?

4.4 Fixer n (par exemple $n = 200$) et tracer le graphe de la fonction proba, sur $[1, 99]$. On tracera sur le même graphe et en rouge, la droite d'équation $y = \frac{x}{100}$.

4.5 Comment interpréter ce résultat ?

4.6 Modifier votre programme pour prendre en compte une probabilité de gain de 0.49 et donc une probilité 0.51 de perdre.

4.7 Que constatez-vous?

5 La planche de Galton

Le dispositif est expliqué ici.

Ici la planche est modélisée par le rectangle $[-1, 1] \times [0, 1]$. On lache une bille en $x = 0$ et à la hauteur $y = 1.0$. On se donne un entier $N \in \mathbb{N}^*$.

À chaque passage de la bille à une hauteur $1 - \frac{k}{N}$ $(1 \leq k \leq N - 1)$, la bille est déviée aléatoirement soit sur sa droite soit sur sa gauche avec probabilité $1/2$, d'une quantité $1/N$. Après ces $N-1$ déviations aléatoires, elle arrive en $y = 0$ en un point aléatoire X qui va se situer dans une des "boites" de réception

$$B_\ell = \left[\frac{\ell}{N} - \frac{1}{2N}, \frac{\ell}{N} + \frac{1}{2N}\right], \quad -(N-1) \leq \ell \leq N-1.$$

5.1 Simuler la variable aléatoire X (écrire une fonction de N qui donne la position aléatoire de la bille quand elle atteint $y = 0$).