Licence Informatique S4 - Groupe 3
Probabilités-Statistiques TP Préparatoire

April 12, 2020

1 Simuler une somme de variables aléatoires indépendantes

Somme d'une variable aléatoire uniforme et d'une variable aléatoire de loi normale

1. Ecrire une fonction $Unif(N, a, b)$ retournant N simulations d'une variable aléatoire X de loi uniforme sur $[a, b]$.

2. Ecrire une fonction $Norm(N, \mu, s)$ retournant N simulations d'une variable aléatoire Y de loi normale d'espérance μ et d'écart-type s.

3. Ecrire une fonction $SommeUnifNorm(N, a, b, \mu, s)$ en $\mathcal{R}$ retournant N simulations d'une variable aléatoire $Z = X+Y$, où X suit une loi uniforme sur $[a, b]$, Y suit une loi normale d'espérance μ et d'écart-type s et où X et Y sont indépendantes.

4. Etudier l'espérance et la variance de Z à la fois théoriquement et numériquement en fonction de a, b, μ et s. Est-il facile de déterminer une expression pour la loi de Z ?

Autres cas

1. Mêmes questions que précédemment dans le cas où X et Y sont indépendantes et de lois exponentielles de paramètre μ et λ respectivement.

2. Mêmes questions que précédemment dans le cas où X et Y sont indépendantes et de même loi uniforme sur $[a, b]$.

3. Ecrire une fonction $SommeLUnif(N, L, a, b)$ retournant N simulations d'une variable aléatoire $Z = X_1 + X_2 + \cdots + X_L$, où les X_i sont indépendantes et de même loi uniforme sur $[a, b]$. Comparer, à l'aide d'un histogramme, les résultats aux conclusions du TLC.

2 Simuler une loi uniforme sur un rectangle et sur un disque

Dans un rectangle

1. Ecrire une fonction unif(a,b,c,d) en $\mathcal{R}$ retournant une réalisation d'un point tiré uniformément dans le carré $[a, b] \times [c, d]$.

2. Ecrire une fonction unif(N,a,b,c,d) retournant N réalisations d'un point tiré uniformément sur $[a, b] \times [c, d]$.

Dans un disque

1. On souhaite désormais tirer N points aléatoirement dans un disque centré en $(0, 0)$ de rayon r. Pour ce faire, on utilisera un algorithme de rejet : choisir un point aléatoire dans un carré de coté $2r$ et ne retenir que les points qui sont dans le disque, çàd à distance inférieure ou égale à r du centre. Pour cela, on peut se servir d'un compteur qui va compter le nombre de points tirés dans le carré qui appartiennent au disque...
Par exemple : on tire un premier point dans le carré, si celui-ci appartient au disque, alors on le

conserve dans une matrice Points (la première colonne étant l'abscisse du point et la deuxième son ordonnée) qui va grandir au fil des tirages. Sinon, on recommence. Dans tous les cas, on recommence ce tirage jusqu'à ce que la matrice Points soit précisément de taille $N * 2$.

2. Dans cette dernière partie, on s'intéresse précisément à la distance d entre chacun des points tirés aléatoirement dans le disque et l'origine du disque. Tracer l'histogramme des valeurs de d obtenues. Quelle est la loi de la variable d ? Est-elle uniforme ? Expliquer le lien entre cette loi et l'histogramme obtenu.

3 Méthode de Monte-Carlo

3.1 Principe :

Utiliser la loi forte des grands nombres (LFGN) pour faire des calculs approchés, d'aires ou d'intégrales par exemple.
Supposons que l'on veuille calculer $I = \int_a^b f(x)\mathrm{d}x$. On peut interpréter I comme l'espérance d'une v.a. En effet si U est une v.a. uniforme sur $[a, b]$, on a

$$\mathbb{E}(f(U)) = \frac{1}{b-a}\int_a^b f(x)\mathrm{d}x.$$

Pour mieux comprendre cela, vous pouvez voir le cours sur la formule de transfert.

La LFGN affirme que, avec probabilité 1, on a

$$f(U_1) + \cdots + f(U_n) \xrightarrow[n\to\infty]{} \mathbb{E}(f(U))$$

où $U_1, \ldots, U_n$ sont des copies indépendantes de U.

3.2 Applications :

Exercice 1. On souhaite calculer approximativement la distance moyenne entre deux points jetés au hasard dans le carré $[0, 1] \times [0, 1]$. Élaborer et implémenter une méthode de calcul approché.

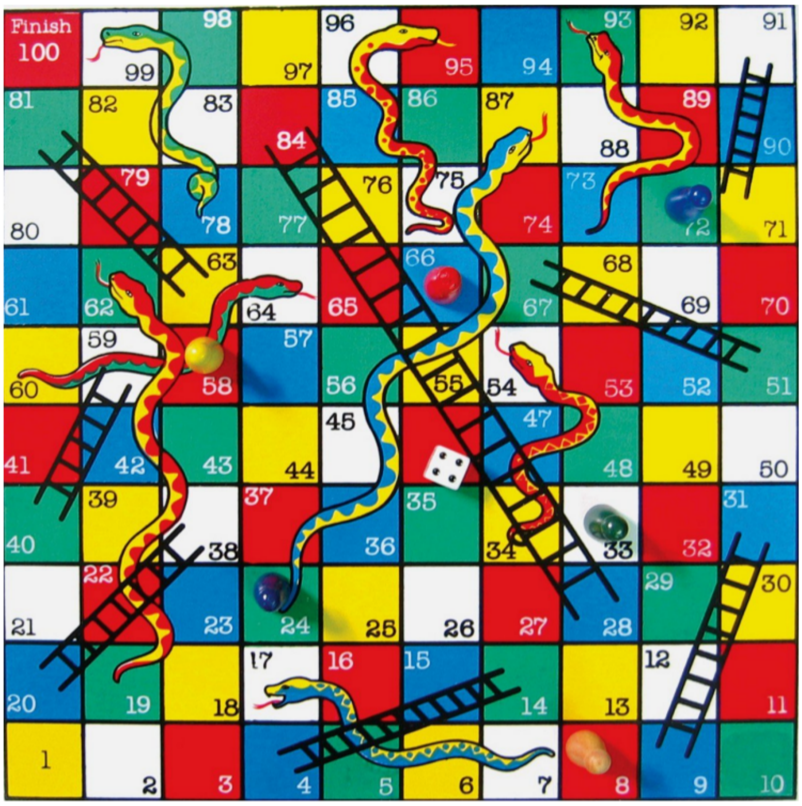
Exercice 2. En mathématiques financières, le prix C (comme "Call") d'une option (européenne) d'achat est une intégrale qui s'identifie à l'espérance suivante :

$$C = \mathbb{E}\left([e^G - 1]_+\right)$$

où $G \rightsquigarrow \mathcal{N}(0, 1)$ et $[.]_+$ désigne la partie positive, c'est-à-dire que $[x]_+ = \max(x, 0)$.
Calculer une valeur approchée de C en utilisant la LFGN.
Indication : on commencera par simuler n réalisations de la loi $\mathcal{N}(0, 1)$ avec $\mathcal{R}$.

4 Des serpents et des échelles

On s'intéresse dans cet exercice au jeu de plateau des serpents et des échelles. Les pions des joueurs commencent sur la case 1 et pour gagner il faut atteindre la case 100. À tour de rôle, chaque joueur jette le dé et avance son pion du nombre de cases correspondant. Si l'on arrive sur le bas d'une échelle, on doit la monter et si l'on arrive sur la tête d'un serpent on doit descendre le long de son corps (la disposition est donnée sur la figure 1). Deux pions peuvent se trouver sur une même case et on gagne en arrivant sur la case 100 ou en la dépassant.



Voici les cases spéciales et où elles conduisent :

Bas d'échelles	Têtes de serpents
$4 \to 14$	$17 \to 7$
$8 \to 31$	$54 \to 34$
$20 \to 38$	$62 \to 19$
$28 \to 84$	$64 \to 60$
$40 \to 59$	$93 \to 73$
$51 \to 67$	$95 \to 75$
$63 \to 81$	$99 \to 78$
$71 \to 91$	

Pour ce TP on ne prend pas en compte le grand serpent $87 \to 24$.

Question 1. Construire un vecteur *destination* de taille 100 tel que la valeur de la $n^{\text{ième}}$ coordonnée soit la case où doit aller un pion qui tombe sur la case n. Par exemple :
$destination[20] = 38$ (bas d'échelle en case 20),
$destination[10] = 10$ (rien en case 10),
$destination[17] = 7$ (tête de serpent en case 17).

Comme les pions n'interagissent pas entre eux (deux pions peuvent être sur la même case), on est ramené à simuler une partie avec un seul joueur. La variable *position* désignera la case courante où se trouve le joueur. On simulera (dans la *question 3*) un tour du joueur de la façon suivante :

1. ajouter à *position* un nombre aléatoire uniforme entre 1 et 6,

2. remplacer *position* par le minimum entre *position* et 100,
3. mettre à jour *position* grâce à destination.

Question 2. Pourquoi doit-on prendre le minimum entre *position* et 100 ?

Question 3. Écrire une fonction *Jeu(depart)* qui simule une partie d'un joueur démarrant de la case depart (un nombre entre 1 et 100) et renvoit le nombre de tours qu'il lui a fallu pour arriver à la case 100.
Indication : vous pouvez utiliser la fonction sample *pour simuler le lancé de dé.*

Question 4. Écrire une fonction *TempsMoyen(N, start)* qui effectue N parties avec un joueur démarrant de la case start et retourne le temps moyen qu'il lui a fallu pour atteindre la case 100. C'est une estimation de l'espérance du temps de jeu.

Question 5. Avec *TempsMoyen(10000,1)*, estimer l'espérance du temps de jeu d'un joueur partant de la case 1. Vous devez trouver une valeur de l'ordre de 32 tours.

Question 6. Le joueur commence à la case 1, quel est le meilleur coup de dé qu'il puisse faire (celui après lequel l'espérance de temps de jeu sera la plus courte) ? Était-ce prévisible ? Pour les simulations, prendre $N = 10000$.

Question 7. Le joueur est arrivé à la case 18, quel est le meilleur coup de dé qu'il puisse faire ensuite ? Commenter. Pour les simulations, prendre $N = 10000$.

Question 8. Qu'est ce qui n'est pas rigoureux dans les *questions 5 et 6* ? Comment pourrait-on faire pour remédier à cela ?

Question 9. Le nombre minimal de tours pour atteindre la case 100 est 7. Quelle est la probabilité pour que le joueur gagne en 7 tours ?