

Retour sur les simulations de variables aléatoires

Le but de ce TP est de simuler des variables aléatoires partant seulement de `rd.random()` qui simule une loi uniforme sur $[0;1]$.

Exercice 1. Dans cet exercice, on tirera profit du travail fait lors de l'exercice 1 du TD10 de maths.

1. À l'aide d'une simulation d'une loi uniforme sur $[0;1]$, écrire une fonction `Uniforme(a,b)` qui simule $X \sim \mathcal{U}([a;b])$.
2. À l'aide d'une simulation d'une loi uniforme sur $[0;1]$, écrire une fonction `UniformeEntier(n)` qui simule $X \sim \mathcal{U}(\llbracket 1;n \rrbracket)$.
3. Pour chacune des fonctions écrites, créer un échantillon de taille N et afficher l'histogramme des fréquences pour $N \in \{10, 100, 1000, 10^4, 10^5, 10^6\}$, prendre $a = 3$ et $b = 10$ pour simuler $X \sim \mathcal{U}([a;b])$ et prendre $n = 4$ pour simuler $X \sim \mathcal{U}(\llbracket 1;n \rrbracket)$.

Pour l'histogramme d'une liste `L`, on peut utiliser la fonction `plt.hist` :

```
plt.hist(L,edgecolor="k",density=True)
```

Seulement parfois, Python fait n'importe quoi en terme de largeur/longueur de bâtons. Pour la variable aléatoire uniforme sur $\llbracket 1;4 \rrbracket$, utiliser un `array` qui liste les abscisses de chacun des rectangles :

```
c = np.linspace(0.5,4.5,5) # Array contenant 0.5, 1.5,...,4.5
plt.hist(L,c,edgecolor="k",density=True)
# Ainsi, Python affiche pour chaque k=1,2,3,4
# un baton dont les absisses k-0.5 et k+0.5 et dont la hauteur
# est proportionnel aux nombres de simulations ayant une valeur
# entre k-0.5 et k+0.5 (donc valant k car
# on simule une variable à valeur entière)
```

Et pour $\mathcal{U}([3;10])$, utiliser une subdivision du segment $[3;10]$ avec 15 points (et donc 14 intervalles).

4. Écrire une fonction `SimulerSommeUniforme()` qui simule $X + Y$ où X et Y suivent une loi uniforme sur $[0;1]$ et afficher l'histogramme d'un échantillon, est-ce cohérent avec la densité trouvée en cours ?

Exercice 2. 1. Soit X une variable aléatoire à densité dont F_X , la fonction de répartition, est strictement croissante sur $\mathbb{R}$. Démontrer que F_X est une bijection de $\mathbb{R}$ vers $]0;1[$.

2. Soit $U \sim \mathcal{U}([0;1])$, on pose $V = F_X^{-1}(U)$, démontrer que V et X ont la même loi.
3. Si $X \sim \mathcal{E}(\lambda)$ où $\lambda > 0$, adapter ce qui précède pour écrire une fonction `SimulerExpo(lambd)` qui simule X .
4. Sur un même graphique, dessiner la loi de X ainsi que l'histogramme d'un échantillon de grande taille.

Remarque 1. Cette méthode s'appelle la méthode de simulation par inversion et est à connaître.

Exercice 3. Soit Φ la fonction de répartition de la loi normale centrée réduite.

1. Que vaut $\Phi(0)$?
2. Soit $x \in \mathbb{R}$ écrire $\Phi(x)$ à l'aide de $\Phi(0)$ et d'une intégrale sur un segment.
3. Écrire une fonction `ApproxPhi(x,N)` qui approxime $\Phi(x)$ par la méthode des rectangles (ou des trapèzes), on distinguera $x > 0$ et $x \leq 0$.
4. Justifier que $\Phi: \mathbb{R} \rightarrow]0;1[$ est bijective.
5. Soit $y \in]0;1[$, établir une fonction `Encadr(y)` qui renvoie un couple $(a,b) \in \mathbb{R}^2$ tel que $\Phi(a) \leq y \leq \Phi(b)$.
6. En déduire une fonction `PhiMoins1(y,eps)` qui applique la méthode de dichotomie pour déterminer une approximation de $\Phi^{-1}(y)$.
7. En déduire une fonction `SimulerLN()` qui simule la loi centrée réduite (modulo quelques approximations).
8. Sur un même graphique, tracer la densité de la loi normale centrée réduite et un histogramme d'un échantillon de variables simulés suivant cette loi.
9. Écrire une fonction `SimulerLN2()` qui simule la loi normale centrée réduite mais cette fois en utilisant la commande suivante qui en Python calcule $\Phi^{-1}(x)$:

```
import scipy.stats as ss
y = ss.norm.ppf(x ,0 ,1) # y = Phi^{-1}(x)
```
10. À l'aide de la fonction `SimulerLN2()`, écrire une fonction `SimulerLN(mu,sigma2)` qui simule la loi normale de paramètres μ et σ^2 .

Exercice 4. 1. Soit $(p_0, p_1, \dots, p_{n-1})$ une famille de nombres positifs tels que $\sum_{i=0}^{n-1} p_i = 1$. On tire x un nombre au hasard entre 0 et 1.

Pour $k \in \llbracket 0; n-1 \rrbracket$, on note $s_k = \sum_{i=0}^k p_i$ et $s_{-1} = 1$. Quelle est la probabilité que x appartienne à $[s_{k-1}; s_k]$?

2. Soit X une variable aléatoire discrète telle que $X(\Omega) = \{x_0, \dots, x_{n-1}\}$ et pour tout $i \in \llbracket 0; n-1 \rrbracket$, $\mathbb{P}(X = x_i) = p_i$. On stocke $x_0, \dots, x_{n-1}$ dans une liste V et $p_0, \dots, p_{n-1}$ dans une liste P . Écrire une fonction `Simuler(V,P)` qui simule X .
3. Démontrer qu'il existe une variable aléatoire X tel que pour tout $n \in \mathbb{N}$, $\mathbb{P}(X = n) = \frac{1}{(n+1)(n+2)}$. Simuler X .
4. Simuler aussi une loi de Poisson de paramètre $\lambda > 0$.

Exercice 5. Soit deux urnes U_1 et U_2 contenant à elles deux N boules avec $N \in \mathbb{N}^*$. À chaque étape, on choisit de façon équiprobable un entier entre 1 et N . Si ce nombre est inférieur ou égale au nombre de boules dans l'urne U_1 , alors on met une boule de l'urne U_1 dans U_2 , sinon on met une boule de l'urne U_2 dans l'urne U_1 . Pour tout $n \in \mathbb{N}$, on note X_n le nombre de boule dans l'urne U_1 à l'étape n .

1. Écrire une fonction `Transition(Xn,N)` qui renvoie une simulation de X_{n+1} connaissant X_n et N .
2. Écrire une `SimulerXn(X0,N,n)` qui renvoie une simulation de X_n connaissant X_0 et N .
3. Écrire une fonction `ApproxEsperanceXn(X0,N,n)` qui renvoie une approximation de $\mathbb{E}(X_n)$ en simulant assez de fois X_n .
4. Écrire une fonction `ApproxProbaXn(X0,N,n,k)` qui renvoie une approximation de $\mathbb{P}(X_n = k)$.

Exercice 6. Dans cet exercice, on pourra utiliser `rd.randint`.

1. Créer une fonction `Échantillon(n)` qui renvoie une liste de n uplet de nombres dans $\llbracket 1; 6 \rrbracket$ tirés de façon uniforme.
2. Pour un tel 100-uplet, afficher le graphe le nuage de points (k, M_k) où M_k désigne la moyenne empirique des k -premiers nombres. Quel résultat avez-vous illustré ?
3. Écrire une fonction `SimulerMnstar(n,taille)` qui renvoie un échantillon de taille `taille` de M_n^* .

4. Afficher l'histogramme de cet échantillon. Quel résultat avez-vous illustré ?

Exercice 7. Soit X une variable aléatoire à densité et f une densité. On suppose qu'il existe $(a, b, M) \in \mathbb{R}$ tel que pour tout $x \in \mathbb{R}$, $0 \leq f(x) \leq M$ et $f(x) = 0$ si $x > b$ ou si $x < a$. Pour simuler X on prend un point (x, y) appartenant au rectangle $[a; b] \times [0; M]$, si ce point est sous la courbe de f , alors on simule X en renvoyant x , sinon on recommence jusqu'à obtenir un point sous la courbe.

1. Que vaut l'aire de f et que vaut celle du rectangle $[a; b] \times [0; M]$?
2. Quelle est la loi du temps d'attente avant d'obtenir un point sous la courbe ?
3. Écrire une fonction `SimulerX(a,b,M,f)` qui simule X suivant ce principe (appelé méthode du rejet)
4. On pose $f: t \mapsto c(t-2)(4-t)\mathbb{1}_{[2;4]}(t)$, déterminer $c \in \mathbb{R}$ tel que f soit une densité. Dans ce cas, que prendre pour a , b et M ?
5. Simuler X une VA dont f est une densité à l'aide de la méthode du rejet.
6. Tracer l'histogramme des fréquences d'un échantillon et le comparer à f .