

Méthodes d'échantillonnage directes

Nombres aléatoires & pseudo-aléatoires

Il existe plusieurs manières de générer des nombres dits « aléatoires » selon des lois connues

NB : Les programmes informatiques ne génèrent pas des nombres totalement aléatoire

On parle plutôt de nombres **pseudo-aléatoires**, qui semblent aléatoires mais sont en réalité générés selon un processus déterministe (qui dépend notamment d'un paramètre appelé « **graine** » – *seed*).

Génération d'un échantillon uniforme

Algorithme congruentiel linéaire : génération d'un échantillon pseudo-aléatoire selon la loi uniforme sur $[0;1]$ (Lehmer, 1948)


- 1 Générer une suite d'entiers y_n tel que :

$$y_{n+1} = (ay_n + b) \text{ mod. } m$$

- 2 $x_{n+1} = \frac{y_{n+1}}{m-1}$

Choisir a , b et m de manière à ce que y_n ait une période très longue et que (x_1, \dots, x_n) puisse être considéré comme *iid*

avec y_0 la graine

Remarque : $0 \leq y_n \leq m-1 \Rightarrow$ en pratique m très grand (ex. 2^{19937} , le défaut dans  qui utilise l'algorithme Mersenne-Twister)

Dans la suite la génération de nombre pseudo-aléatoires selon la loi uniforme sur $[0;1]$ sera considérée comme fiable et utilisé par les différents algorithmes d'échantillonnage

Autres distributions usuelles

On s'appuie sur les **relations entre les différentes lois usuelles** en partant de $U_i \sim \mathcal{U}_{[0;1]}$

Autres distributions usuelles

On s'appuie sur les **relations entre les différentes lois usuelles** en partant de $U_i \sim \mathcal{U}_{[0;1]}$

Loi binomiale $Bin(n, p)$:

$$Y_i = \mathbb{1}_{U_i \leq p} \sim \text{Bernoulli}(p)$$

$$X = \sum_{i=1}^n Y_i \sim \text{Bin}(n, p)$$

Autres distributions usuelles

On s'appuie sur les **relations entre les différentes lois usuelles** en partant de $U_i \sim \mathcal{U}_{[0;1]}$

Loi binomiale $Bin(n, p)$:

$$Y_i = \mathbb{1}_{U_i \leq p} \sim \text{Bernoulli}(p)$$

$$X = \sum_{i=1}^n Y_i \sim \text{Bin}(n, p)$$

Loi normale $\mathcal{N}(0, 1)$ (algorithme de Box-Müller) :

U_1 et U_2 sont 2 variables uniformes $[0; 1]$ indépendantes

$$Y_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2)$$

$$Y_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2)$$

⇒ Y_1 & Y_2 sont indépendantes et suivent chacune la loi $\mathcal{N}(0, 1)$

Méthode par inversion

Définition : Pour une fonction F définie sur \mathbb{R} , on définit son **inverse généralisée** par : $F^{-1}(u) = \inf\{x \text{ tq } F(x) > u\}$

Méthode par inversion

Définition : Pour une fonction F définie sur \mathbb{R} , on définit son **inverse généralisée** par : $F^{-1}(u) = \inf\{x \text{ tq } F(x) > u\}$

Propriété : Soit

- F la fonction de répartition d'une distribution de probabilité
- U une variable aléatoire suivant une loi uniforme sur $[0;1]$

Alors $F^{-1}(U)$ définit une variable aléatoire ayant pour fonction de répartition F .

- Si
- ① on connaît la fonction de répartition F de la loi selon laquelle simuler
 - ② on est capable d'inverser F
- ⇒ on peut alors générer un échantillon suivant cette loi à partir d'un échantillon uniforme sur $[0;1]$

Méthode par inversion : illustration

Exemple : On veut générer un échantillon suivant la loi exponentielle de paramètre λ

Méthode par inversion : illustration

Exemple : On veut générer un échantillon suivant la loi exponentielle de paramètre λ

- la densité de la loi exponentielle est $f(x) = \lambda \exp(-\lambda x)$
- la fonction de répartition (son intégrale) est donc $F(x) = 1 - \exp(-\lambda x)$

Posons $F(x) = u$

On obtient alors $x = \dots$