TP 1 : Lois discrètes

Ce TP introduit les premiers outils utilisés en Python pour la simulation de variables aléatoires. Vos programmes seront à écrire dans un fichier `TP1.py`.

La bibliothèque standard de Python inclus le module `random` qui permet de simuler des variables aléatoires. On va donc commencer notre programme par la ligne
`import random`
ou encore
`from random import *`
Dans le premier cas, les fonctions du module `random` devront être appelées avec le préfixe "`random.`" (par exemple "`random.gauss(0,1)`" pour simuler une variable de loi normale centrée réduite) dans le deuxième cas, le préfixe sera sous-entendu ("`gauss(0,1)`").

1. La fonction `random.random` renvoie un entier tiré selon la loi uniforme sur $[0, 1]$. Tester cette fonction, puis l'utiliser pour écrire une fonction prenant en argument deux réels `a` et `b` (on supposera `a<b`) et renvoyant une variable de loi uniforme sur $[\mathtt{a}, \mathtt{b}]$.

À partir d'une loi uniforme sur $[0, 1]$, il est possible de simuler une variable aléatoire à valeur dans $\{0, \ldots, N - 1\}$ de loi arbitraire.

1 Loi uniforme

2. Montrer que si U suit la loi uniforme sur $[0, 1]$, alors la partie entière de NU suit une loi uniforme sur $\{0, \ldots, N - 1\}$.
   La partie entière d'une variable `x` positive peut être obtenue par `int(x)`. Simuler une variable de loi uniforme sur $\{0, \ldots, N - 1\}$, puis écrire une fonction qui prend un argument entier `N` et renvoie une réalisation d'une variable de loi uniforme sur $\{0, \ldots, \mathtt{N-1}\}$.

La bibliothèque `matplotlib` fournit des outils graphique permettant d'illustrer nos résultats. On appellera donc
`import matplotlib.pyplot`
ou encore
`from matplotlib.pyplot import *`
La fonction `matplotlib.pyplot.hist` trace un histogramme d'une liste donnée en argument. L'affichage se fait avec la fonction `matplotlib.pyplot.show`.

3. Placer k réalisations d'une variable aléatoire uniforme sur $\{0, \ldots, N - 1\}$ dans une liste `x` et puis tracer un histogramme de `x`, en utilisant la fonction `matplotlib.pyplot.hist`. On pourra par exemple prendre $N = 30$ avec $k = 100$, puis $k = 1000$ et $k = 10000$.

2 Loi de Bernoulli, loi binomiale

4. Vérifier que si U suit la loi uniforme sur $[0, 1]$, alors l'évènement $\{U < p\}$ a pour probabilité p et $\{U \geq p\}$ a probabilité $1 - p$ (pour $p \in]0, 1[$).
   En déduire une manière de simuler une variable aléatoire de loi de Bernoulli de paramètre p, puis écrire une fonction prenant `p` comme argument et renvoyant un nombre tiré selon une loi de Bernoulli de paramètre `p`.
5. Soit $(X_k)_{k\in\mathbb{N}}$ une suite de variables aléatoires indépendantes de loi de Bernoulli de paramètre p. La variable $Y_n = \sum_{k=1}^{n} X_k$ suit la loi binomiale de paramètres n et p. Écrire une fonction qui prend en argument un entier `n` et un réel `p` et qui renvoie une variable aléatoire de loi binomiale de paramètres `n` et `p`.

6. Afficher un histogramme de N simulations de variables aléatoires de loi binomiale de paramètres n et p. À l'aide de la fonction `matplotlib.pyplot.plot`, superposer à cet histogramme le graphe des valeurs théoriques
$$\mathbb{P}(Y_n = k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}.$$
7. Pour une valeur de p fixée, simuler la suite de variables aléatoires $(Y_n/n)_{n\in\mathbb{N}}$, où les Y_n sont définis comme dans la question 5. On prendra par exemple n entre 1 et 1000. *Attention, les Y_n ne sont pas indépendants.*
Afficher la suite (Y_n/n). Qu'observe-t-on ?

3 Loi arbitraire

8. Vérifier que si $(p_0, \dots, p_N)$ est une probabilité et U est une variable aléatoire uniforme sur $[0,1]$, alors les évènements $A_i = \{p_0 + \ldots + p_{i-1} < U \leq p_0 + \ldots + p_i\}$ forment une partition de l'univers Ω et vérifient $\mathbb{P}(A_i) = p_i$ (avec $0 \leq i \leq N$).
9. À partir de cette remarque, écrire une fonction qui simule une variable aléatoire X à valeurs dans $\{0,1,2,3\}$ telle que
$$\mathbb{P}(X=0) = \frac{1}{2}, \quad \mathbb{P}(X=1) = \frac{1}{4}, \quad \mathbb{P}(X=2) = \frac{1}{6}, \quad \mathbb{P}(X=3) = \frac{1}{12}.$$
10. Afficher un histogramme de 1000 simulations de la variable de la question précédente.
11. Écrire une fonction prenant en argument un tableau de réels $(p_i)_{i=0,\dots,N-1}$ compris entre 0 et 1, et dont la somme est inférieure à 1. Cette fonction devra renvoyer une simulation d'une variable aléatoire à valeurs dans $\{0, \dots, N\}$ telle que $\mathbb{P}(X=k) = p_k$ (on posera $p_N = 1 - \sum_{k=0}^{N-1} p_k$).