Feuille de TP Python 05

Simulations probabilistes/V.a. à densité

L'objet de ce TP est essentiellement de mettre en place une méthodologie pour effectuer et traiter des simulations probabilistes faisant intervenir des variables aléatoires à densité.

Pour modéliser d'un point de vue probabiliste un programme informatique faisant intervenir le hasard, on se donne un espace probabilisé $(\Omega, \mathscr{T}, \mathbb{P})$ et une famille infinie, $(U_n)_{n\in\mathbb{N}^*}$ de variables aléatoires, uniformes sur le segment $[0,1[$, mutuellement indépendantes.

Un tirage au sort est effectué juste avant le démarrage de la session ou du programme : les « dés » sont donc jetés [1] au début, une fois pour toute et on dispose de la liste des valeurs tirées au sort $(\texttt{u[n]})_{n\in\mathbb{N}^*}$, liste dans laquelle on peut piocher [2] l'élément repéré par N, le « curseur » courant de cette liste.

Les appels par le programme à la fonction fondamentale de tirage au sort (par exemple en Python/Numpy, la fonction `numpy.random.rand`) peuvent être numérotés par les entiers dans l'ordre des appels faits.

Au démarrage du programme $N = 0$ et chaque appel à la fonction fondamentale de tirage au sort augmente le curseur de 1 et retourne `u[N]`. La fonction fondamentale de tirage au sort, à l'appel n, renvoie donc la valeur `u[n]`.

Imaginons que pour calculer le résultat d'une fonction Python `f` donnée, nous ayons besoin de 10000 valeurs aléatoires tirées indépendamment suivant une loi uniforme sur $]0,1[$:

— Mathématiquement, lors de chaque appel à la fonction `f`, on utilise les valeurs `u[N+1]`,…`u[N+10000]` des 10000 variables aléatoires *indépendantes* $U_{N+1},\dots U_{N+10000}$ où N est la position du « curseur » juste avant cet appel à `f`.

— A l'issue de cet appel à la fonction `f`, la position du « curseur » est placée à $N+10000$.

Le résultat `x` calculé à l'issue de cet appel à la fonction `f` est donc fonction des 10000 valeurs et peut-être vu comme une réalisation de la variable aléatoire [3] X, $X = f(U_{N+1},\dots,U_{N+10000})$.

Différentes exécutions au sein d'une même session Python sont considérées indépendantes au sens probabiliste (Lemme des coalitions) car faisant intervenir des variables U_n dont les indices appartiennent à des plages d'indices disjointes.

Donc, une suite (dans une même session de l'interpréteur ou une même exécution de programme) d'exécutions de `f` retourne une suite de valeurs d'un échantillon de X, c'est à dire une suite $(X_1,\dots,X_k,\dots)$ de v.a. indépendantes, de même loi que X.

Les résultats d'un programme faisant intervenir le hasard sont donc eux-aussi les réalisations d'une famille de variables aléatoires.

1 Simulation et représentation graphique de la loi d'une v.a. à densité

1.1 La problématique

Partant d'une ou plusieurs variables aléatoires uniformes, on peut créer, par simple application d'une fonction d'une ou plusieurs variables réelles d'autres variables aléatoires de tous types possibles [4].

En « encapsulant » une telle construction dans une fonction Python `X()`, on obtient une fonction Python retournant une réalisation d'une variable aléatoire réelle X. Cette fonction peut être appelée un grand nombre de fois et si l'on stocke les résultats d'un grand nombre d'appels dans une liste (ou un tableau Numpy) alors :

— le tracé d'un histogramme (avec « bins » bien choisis) de cette liste de nombres donne une représentation graphique approximative de la loi de X ;

— le tracé de la fonction de répartition de cette liste de nombres donne une estimation de la fonction de répartition de X ;

— la moyenne des nombres de cette liste est une estimation de l'espérance [5] de X ;

— l'écart-type au carré des nombres de cette liste est une estimation de la variance [6] de X ;

— ...

Voici quelques exemples.

1. Des exemples « fonctions simples d'une variable aléatoire U uniforme sur $[0,1[$ » :

1. Plus précisément, les roues de la fortune sont tournées
2. On peut aussi penser à l'image de casino suivante : on dispose d'un sabot infini de cartes portant des valeurs qui ont été mélangées à l'avance, chaque appel à la fonction fondamentale de tirage au sort consiste à piocher la carte en haut du tas, utiliser la valeur lue et mettre la carte au rebut
3. définie sur Ω
4. A support fini, comme en première année, ou alors à densité, discrètes ou encore de types mixtes
5. pourvu que celle-ci existe et en un sens à préciser, *cf.* Cours sur la loi des grands nombres
6. pourvu que celle-ci existe...

(a) $X = U^2$; sa fonction de répartition est [7] $F_X(x) = \begin{cases} 0 & \text{si } x \in]-\infty, 0[\\ \sqrt{x} & \text{si } x \in [0,1[\\ 1 & \text{si } x \in [1,+\infty[\end{cases}$; une densité est $f_X(x) = \begin{cases} 0 & \text{si } x \in]-\infty, 0] \\ \frac{1}{2\sqrt{x}} & \text{si } x \in]0,1[\\ 0 & \text{si } x \in [1,+\infty[\end{cases}$; son espérance vaut $\mathbb{E}(X) = \frac{1}{3}$, sa variance est $\mathbb{V}(X) = \frac{1}{5} - \frac{1}{9} = \frac{4}{45}$.

(b) $X = -\ln(U)$; $X \sim \mathscr{E}(1)$; sa fonction de répartition est $F_X(x) = \begin{cases} 0 & \text{si } x \in]-\infty, 0[\\ 1 - e^{-x} & \text{si } x \in [0,+\infty[\end{cases}$; une densité est $f_X(x) = \begin{cases} 0 & \text{si } x \in]-\infty, 0] \\ e^{-x} & \text{si } x \in]0,+\infty[\end{cases}$; son espérance vaut $\mathbb{E}(X) = 1$, sa variance est $\mathbb{V}(X) = 1$.

(c) $X = \tan(\pi(U - \frac{1}{2}))$; (variable de CAUCHY [8]) ; sa fonction de répartition est $F_X(x) = \dfrac{1}{\pi}\arctan x + \dfrac{1}{2}$; une densité est $f_X(x) = \dfrac{1}{\pi}\dfrac{1}{1+x^2}$; son espérance (et encore moins sa variance) n'existe pas.

(d) $X = -\ln(-\ln(U))$; (variable de GUMBEL [9]) ; sa fonction de répartition est $F_X(x) = e^{-e^{-x}}$; une densité est $f_X(x) = e^{-x}.e^{-e^{-x}}$; Son espérance [10] est la constante d'EULER-MASCHERONI $\gamma \simeq 0,5772156649$, sa variance vaut $\frac{\pi^2}{6}$.

2. Des exemples « fonctions simples de plusieurs variables aléatoires U_1, U_2,.., indépendantes, uniformes sur $[0,1[$ » :

(a) $X = \sqrt{-2\ln(U_1)}.\cos(2\pi.U_2)$; $X \sim \mathscr{N}(0,1)$; sa fonction de répartition [11] est $F_X(x) = \dfrac{1}{\sqrt{2\pi}}\displaystyle\int_{-\infty}^{x} e^{-\frac{1}{2}t^2}\,dt$; une densité est la densité Gaussienne $f_X(x) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$; son espérance vaut $\mathbb{E}(X) = 1$, sa variance est $\mathbb{V}(X) = 1$.

(b) X_1 comme dans l'exemple (à choisir dans la gamme d'exemples 1.(a),..., 1.(d), 2.(a)), X_2 (à choisir de même), indépendantes ; $X = X_1 + X_2$. (Les densités ou fonctions de répartition ne sont pas toujours facilement calculables.)

(c) X_1 à choisir dans la gamme d'exemples, X_2 à choisir de même, indépendantes ; $X = X_1 - X_2$. (Les densités ou fonctions de répartition ne sont pas toujours facilement calculables.)

(d) X_1 à choisir, X_2 à choisir, indépendantes ; $X = \max(X_1, X_2)$. La fonction de répartition est $F_X = F_{X_1}.F_{X_2}$.

(e) X_1 à choisir, X_2 à choisir, indépendantes ; $X = \min(X_1, X_2)$. La fonction de répartition est $F_X = 1 - (1 - F_{X_1}).(1 - F_{X_2})$.

(f) Variantes de 2.(b)–2.(e) avec n variables suivant toutes la même loi à choisir.

1.2 Travail à effectuer

1. Pour le premier exemple 1.(a), $X = U^2$, charger, de l'archive consacrée au TP, le script nommé `u2.py`, comprendre son fonctionnement et compléter les commentaires.
2. En choisissant deux exemples parmi 1.(b)–1.(d), 2.(a), construire un script similaire à `u2.py` présentant les caractéristiques fondamentales de la v.a. en question.
3. En choisissant deux exemples parmi ceux présentés en 2.(b)–2.(f), construire votre propre script présentant des caractéristiques fondamentales de la v.a. en question. (Vous n'avez pas besoin d'être aussi exhaustif que ce qui est fait dans `u2.py`, notamment au niveau des éléments théoriques)

2 Tirer un nombre réel au hasard, en respectant sa loi

La loi d'une variable aléatoire réelle X est caractérisée par sa fonction de prépartition F_X définie par :

$$\forall x \in \mathbb{R},\ F_X(x) = \mathbb{P}(X \leq x).$$

Une fonction des quantiles de X est une sorte de réciproque [12] de F_X : une fonction $Q_X :]0,1[\to \mathbb{R}$ est fonction des quantiles de X si et seulement si :

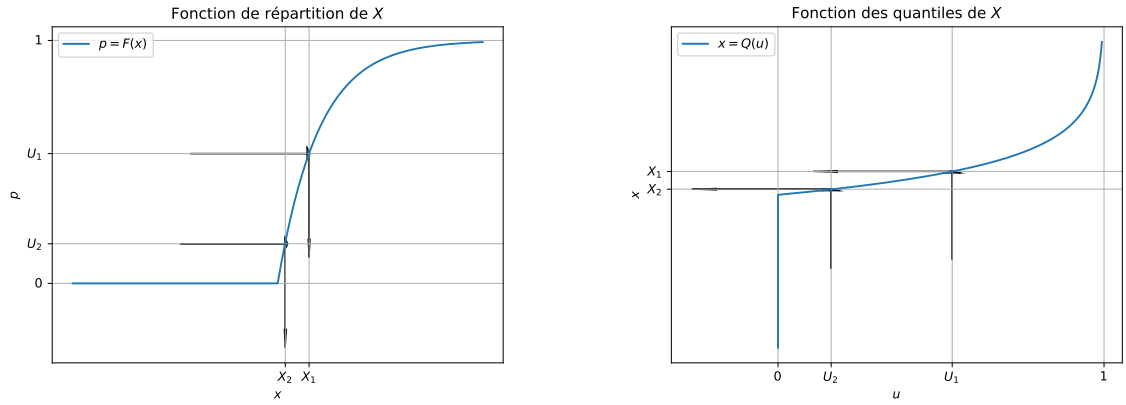
$$\forall u \in]0,1[, \forall x \in \mathbb{R},\ Q_X(u) \leq x \Leftrightarrow u \leq F_X(x)$$

7. donnée par $\forall x \in \mathbb{R}, ...$
8. terminologie hors programme
9. terminologie hors programme
10. Ces quantités sont liées à la théorie des séries que nous verrons en fin d'année
11. disponible en Python/Numpy en tant que `norm.cdf` du module `scipy.stats`
12. Une telle fonction existe toujours en la définissant par :

$$\forall u \in]0,1[, Q_X(u) = \inf\{x \in \mathbb{R}, F_X(x) \geq u\}$$

c'est à dire, vu que $\{x \in \mathbb{R}, F_X(x) \geq u\}$ est un intervalle du type $[a, +\infty[$:

$$\forall u \in]0,1[, \{x \in \mathbb{R}, F_X(x) \geq u\} = [Q_X(u), +\infty[.$$



(a) Tirer au sort sur l'axe des ordonnées, retourner l'abscisse

(b) Après inversion

FIGURE 1 – Simuler une v.a. étant donnée sa fonction de répartition

Par exemple si $X \sim \mathscr{E}(\lambda)$, les fonctions F_X et Q_X définies ci-après vérifient cette relation :

$$\forall x \in \mathbb{R}, F_X(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ 1 - e^{\lambda . x} & \text{sinon} \end{cases}, \forall u \in]0,1[, Q_X(u) = -\frac{1}{\lambda}\ln(1-u)$$

Cette équivalence donne un moyen algorithmique de construire une v.a. Y ayant même loi que X et, donc, une fonction Python simulant la v.a. X.

On démontre en effet que si $U \sim \mathscr{U}_{]0,1[}$, $Y = Q_X(U)$ a même loi que X et donc que la fonction Python `X()` construite suivant le schéma suggéré retourne une valeur aléatoire tirée suivant la loi de X.

```
import numpy as np # on importe la bibliothèque numpy avec son surnom standard

def U(): #renommage uniforme sur 0,1
    return np.random.rand()
def Q(u) :
    x = ....## Mettre ici une formule pour la réciproque "généralisée" de F
    return x
def X():
    return Q(U())
```

2.1 Travail à effectuer

1. (Variable Logistique) La fonction F définie par :

$$\forall x \in \mathbb{R}, F(x) = \frac{1}{1+e^{-x}}$$

est la fonction de répartition d'une variable aléatoire X.

Donner une fonction Python simulant X.

2. (Variable de FRECHET) Soit $\alpha > 0$. La fonction F définie par :

$$\forall x \in \mathbb{R}, F(x) = \begin{cases} e^{-x^{-\alpha}} & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$

est la fonction de répartition d'une variable aléatoire X.

Donner une fonction Python simulant X.

3. (Variable de WEIBULL) Soit $\alpha > 0$. La fonction F définie par :

$$\forall x \in \mathbb{R}, F(x) = \begin{cases} 1 - e^{-x^{+\alpha}} & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$

est la fonction de répartition d'une variable aléatoire X.

Donner une fonction Python simulant X.

3 Décroissance exponentielle d'une population d'atomes

3.1 Description sommaire

L'exercice demandé est beaucoup plus ouvert que les exercices précédents. Il doit mener à la production d'un graphe du type de celui présenté en Fig. 2.

On considère une population d'atomes d'un certain type A. On note $\mathcal{N}$ son effectif à l'instant initial. Ces atomes se désintègrent (transformation du type A en le type B) indépendamment les uns des autres, l'instant de désintégration de l'atome k, T_k, étant une variable aléatoire de loi [13] $\mathscr{E}(\lambda)$.

3.2 Travail à effectuer

Tracer, sur un même graphe, pour $NP = 10$ populations initiales de même effectif $\mathcal{N}$, la proportion p (par rapport à l'effectif initial) d'atomes de type A , en fonction du temps t.

Tracer, sur le même graphe la fonction $t \mapsto e^{-\lambda.t}$.

Constatations ? Explications ?

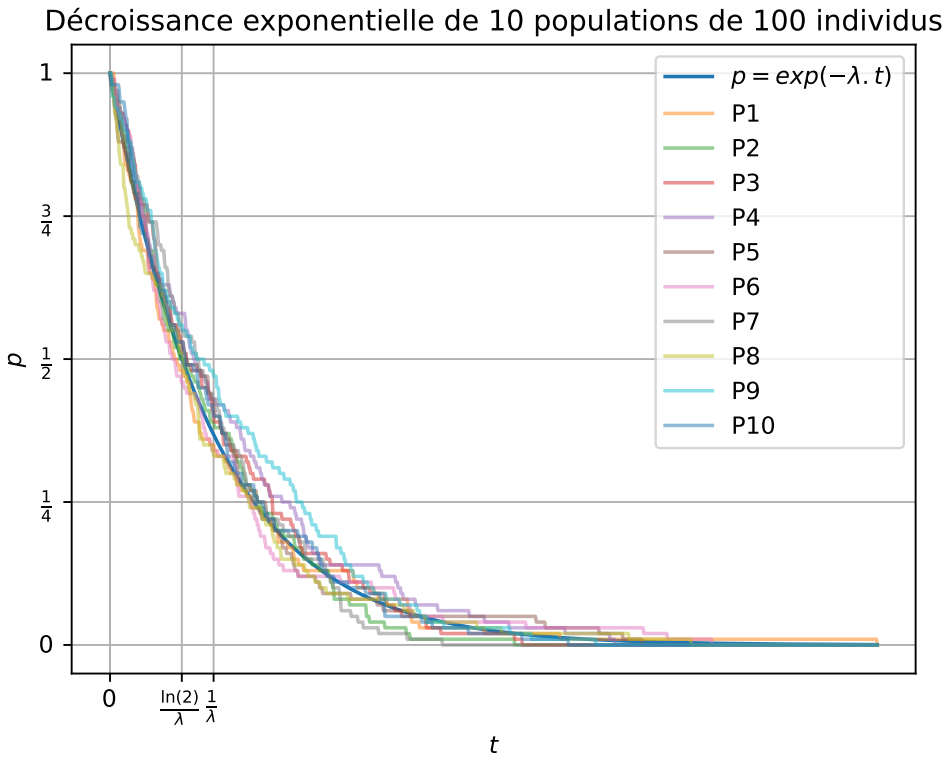


FIGURE 2 – Diverses simulation d'évolution vs. l'évolution moyenne.

13. Astuce : vous pouvez tout faire avec $\lambda = 1$, un autre λ est juste un changement linéaire d'unité temporelle