

TP12 : Méthode d'inversion

Commencer par importer les bibliothèques suivantes dans chaque fichier **Python** utilisé :

```
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
```

Dans ce TP, on s'intéresse au problème suivant.

Données :

- × une v.a.r. X a priori difficile à simuler informatiquement,
- × la fonction de répartition F de la v.a.r. X .

But :

obtenir une v.a.r. V de même fonction de répartition F
(i.e. de même loi que X) plus simple à simuler informatiquement.

I. Théorème d'inversion dans le cas où F est bijective

Théorème 1. (HORS-PROGRAMME)

Soit X une v.a.r. dont la loi est donnée par sa fonction de répartition $F : \mathbb{R} \rightarrow]0, 1[$.

Soit U une v.a.r. telle que $U \hookrightarrow \mathcal{U}(]0, 1[)$.

On suppose de plus que :

- × F est continue sur \mathbb{R} ,
- × F est strictement croissante sur \mathbb{R} .

On a alors :

- F est bijective de \mathbb{R} dans $]0, 1[$.
- La v.a.r. $V = F^{-1}(U)$ a pour fonction de répartition F .

► Rappeler la fonction de répartition d'une v.a.r. U telle que $U \hookrightarrow \mathcal{U}(]0, 1[)$.

$$F_U : x \mapsto \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}$$

► Démontrer le résultat du théorème.

- La fonction F est bijective de \mathbb{R} dans $]0, 1[$ par application du théorème de la bijection.
- Notons G la fonction de répartition de la v.a.r. $F^{-1}(U)$.
Pour tout $x \in \mathbb{R}$, on a : $G(x) = \mathbb{P}([F^{-1}(U) \leq x]) = \mathbb{P}([U \leq F(x)]) = F_U(F(x)) = F(x)$.

II. Application : simulation d'une loi uniforme continue à l'aide de `rd.random()`

On considère une v.a.r. X telle que $X \hookrightarrow \mathcal{U}([a, b])$ (où a et b sont deux réels tels que $a < b$).

► Rappeler $X(\Omega)$, une densité de X ainsi que la fonction de répartition de X , notée F .

$$a) \quad X(\Omega) = [a, b]$$

$$b) \quad X \text{ admet pour densité la fonction } f : x \mapsto \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{sinon} \end{cases}.$$

$$c) \quad X \text{ admet pour fonction de répartition } F : x \mapsto \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ 1 & \text{si } x > b \end{cases}$$

► Démontrer que F réalise une bijection de $[a, b]$ dans $[0, 1]$.
Déterminer sa bijection réciproque $G : [0, 1] \rightarrow [a, b]$.

• Soit $y \in [0, 1]$ et $x \in [a, b]$. On détermine G en raisonnant par équivalence.

$$y = F(x) \Leftrightarrow y = \frac{x-a}{b-a} \Leftrightarrow (b-a)y = x-a \Leftrightarrow x = (b-a)y + a \Leftrightarrow x = G(y)$$

• Ainsi, la fonction $G : [0, 1] \rightarrow [a, b]$ est définie par : $\forall x \in [0, 1], G(x) = a + (b-a)x$.

► Déterminer la loi de la v.a.r. $V = G(U)$, où $U \hookrightarrow \mathcal{U}([0, 1])$.

Soit $x \in \mathbb{R}$.

• Si $x < a$:

$$[G(U) \leq x] = \emptyset \text{ car } G(U) \text{ est à valeurs dans } [a, b]. \text{ D'où : } \mathbb{P}([G(U) \leq x]) = 0.$$

• Si $x \in [a, b]$:

$$F_V(x) = \mathbb{P}(G(U) \leq x) = \mathbb{P}(F(G(U)) \leq F(x)) = \mathbb{P}(U \leq F(x)) = F(x).$$

• Si $x > b$:

$$[G(U) \leq x] = \Omega \text{ car } G(U) \text{ est à valeurs dans } [a, b]. \text{ D'où : } \mathbb{P}([G(U) \leq x]) = 1.$$

Ainsi, on a : $V \hookrightarrow \mathcal{U}([a, b])$.

► En déduire une fonction **Python** nommée `unifContinue(a,b)` simulant une v.a.r. X telle que $X \hookrightarrow \mathcal{U}([a, b])$. On utilisera la fonction `rd.random()`.

```

1 def unifContinue(a,b):
2     return a + (b-a) * rd.random()

```

III. Le théorème d'inversion aux concours (session 2015)

III.1. Simulation d'une v.a.r. suivant une loi exponentielle (EML 2015)

Soit $\lambda > 0$ et soit X une v.a.r. telle que $X \hookrightarrow \mathcal{E}(\lambda)$. On note F la fonction de répartition de X .

- Démontrer que F réalise une bijection de $[0, +\infty[$ dans $[0, 1[$.
Déterminer sa bijection réciproque $G : [0, 1[\rightarrow [0, +\infty[$.

- Soient $y \in [0, 1[$ et $x \in [0, +\infty[$. On détermine G en raisonnant par équivalence.

$$y = F(x) \Leftrightarrow y = 1 - e^{-\lambda x} \Leftrightarrow 1 - y = e^{-\lambda x} \Leftrightarrow -\lambda x = \ln(1 - y) \Leftrightarrow x = -\frac{\ln(1 - y)}{\lambda}$$

- Ainsi, la fonction $G : [0, 1[\rightarrow [0, +\infty[$ est définie par : $\forall x \in [0, 1[$, $G(x) = -\frac{\ln(1 - x)}{\lambda}$.

- Déterminer la loi de la v.a.r. $V = G(U)$, où $U \hookrightarrow \mathcal{U}([0, 1])$.

Soit $x \in \mathbb{R}$.

- Si $x \leq 0$:

$$[G(U) \leq x] = \emptyset \text{ car } G(U) \text{ est à valeurs positives. D'où : } \mathbb{P}([G(U) \leq x]) = 0.$$

- Si $x \in [0, +\infty[$:

$$\mathbb{P}(G(U) \leq x) = \mathbb{P}(F(G(U)) \leq F(x)) = \mathbb{P}(U \leq F(x)) = F(x).$$

Ainsi, on a : $V \hookrightarrow \mathcal{E}(\lambda)$.

- En déduire une fonction **Python** `expo(lam)` (`lam` contient la valeur de λ) simulant une v.a.r. X telle que $X \hookrightarrow \mathcal{E}(\lambda)$. On utilisera la fonction `rd.random()`.

```

1 def expo(lam)
2     return -(1/lam) * np.log(1-rd.random())

```

L'énoncé EML 2015 commençait par ce résultat d'inversion.

L'usage des fonctions de répartition n'était pas explicitement demandé.

- Soit U une v.a.r. suivant la loi uniforme sur $[0, 1[$. Quelle est la loi de la v.a.r. $V = -\frac{1}{\lambda} \ln(1 - U)$?
- Écrire une fonction en **Python** qui, étant donné un réel λ strictement positif, simule la loi exponentielle de paramètre λ .

III.2. Simulation d'une v.a.r. suivant la loi standard de Gumbel (HEC 2015)

Soit F la fonction définie sur \mathbb{R} à valeurs réelles telle que : $F(x) = \exp(-e^{-\lambda x})$.

- Justifier que F est de classe C^∞ sur \mathbb{R} et montrer que F réalise une bijection de \mathbb{R} sur $]0, 1[$.

- F est de classe C^∞ sur \mathbb{R} comme composée de fonctions C^∞ sur \mathbb{R} , et pour tout $x \in \mathbb{R}$:

$$F'(x) = -(-\lambda e^{-\lambda x}) e^{-e^{-\lambda x}} = \lambda e^{-\lambda x} e^{-e^{-\lambda x}} > 0$$

- La fonction F est continue sur \mathbb{R} et strictement croissante sur \mathbb{R} . Ainsi, F réalise une bijection de \mathbb{R} dans $F(\mathbb{R}) =]\lim_{x \rightarrow -\infty} F(x), \lim_{x \rightarrow +\infty} F(x)[$. Or :

$$\times \lim_{x \rightarrow -\infty} -e^{-\lambda x} = -\infty \quad \text{donc} \quad \lim_{x \rightarrow -\infty} e^{-e^{-\lambda x}} = 0.$$

$$\times \lim_{x \rightarrow +\infty} -e^{-\lambda x} = 0 \quad \text{et} \quad \lim_{x \rightarrow +\infty} e^{-e^{-\lambda x}} = 1.$$

F réalise bien une bijection de \mathbb{R} dans $]0, 1[$.

- En déduire que F est la fonction de répartition d'une variable aléatoire T admettant une densité f_T continue sur \mathbb{R} que l'on déterminera ; on dit que T suit la loi de Gumbel de paramètre λ .

La fonction $F : \mathbb{R} \rightarrow]0, 1[$ vérifie les propriétés suivantes.

1) F_X est croissante.

2) $\lim_{x \rightarrow -\infty} F_X(x) = 0$ et $\lim_{x \rightarrow +\infty} F_X(x) = 1$

3) F_X est continue à droite en tout point $x \in \mathbb{R}$ car continue sur \mathbb{R} .

Ainsi, F est la fonction de répartition d'une v.a.r. T . De plus, comme F est de classe C^1 sur \mathbb{R} alors T est une v.a.r. à densité. On obtient une densité en dérivant F sur l'intervalle ouvert $] -\infty, +\infty[: \forall t \in \mathbb{R}, f_T(t) = \lambda e^{-\lambda t} e^{-e^{-\lambda t}}$.

On suppose maintenant que $\lambda = 1$.

- Expliciter la bijection réciproque G de la fonction F .

- Avec $\lambda = 1$, on a $F(x) = e^{-e^{-x}}$ et on sait que F est bijective de \mathbb{R} dans $]0, 1[$.
- Soient $x \in \mathbb{R}$ et $y \in]0, 1[$. On détermine G en raisonnant par équivalence.

$$\begin{aligned} F(x) = y &\Leftrightarrow e^{-e^{-x}} = y \Leftrightarrow -e^{-x} = \ln y \Leftrightarrow e^{-x} = -\ln y \Leftrightarrow -x = \ln(-\ln y) \\ &\Leftrightarrow x = -\ln(-\ln y) \end{aligned}$$

(les opérations précédentes sont permises car $y > 0$ et $-\ln y > 0$)

- Ainsi, la fonction $G :]0, 1[\rightarrow] -\infty, +\infty[$ est définie par : $\forall x \in]0, 1[, G(x) = -\ln(-\ln x)$.

- On considère le programme **Python** suivant :

```

1 x = np.linspace(-2,2,400)
2 y = np.exp(-np.exp(-x))
3 plt.plot(x,y)
4 plt.plot(y,x)

```

Le réel 0 fait-il partie des nombres contenus dans le tableau $\mathbf{x} = \text{np.linspace}(-2,2,400)$?

- La commande `np.linspace(-2,2,400)` crée un vecteur ligne contenant 400 nombres régulièrement espacés entre -2 et 2 .
- Tout nombre créé est donc de la forme :

$$-2 + \frac{k}{399} [2 - (-2)] = -2 + \frac{4k}{399}, \text{ avec } k \in \llbracket 0, 399 \rrbracket$$

Pour obtenir 0, il faudrait : $-2 + \frac{4k}{399} = 0 \Leftrightarrow \frac{4k}{399} = 2 \Leftrightarrow k = \frac{399}{2}$

Or $\frac{399}{2}$ n'est pas un entier, donc la valeur 0 ne fait pas partie des nombres renvoyés.

Quel sera le résultat de l'exécution de ce programme ?

- \mathbf{x} est un vecteur contenant des abscisses entre -2 et 2 .
- \mathbf{y} contient les ordonnées correspondantes par F .
- `plot(x,y)` permet donc de tracer le graphe de F .
- Enfin, `plot(y,x)` effectue le tracé dans lequel le rôle des abscisses et des ordonnées est échangé. Ainsi, cette commande trace le symétrique de la courbe précédente par rapport à la droite d'équation $y = x$. On en déduit que cette commande permet le tracé du graphe de G , réciproque de la fonction F .

- Soit U une variable aléatoire suivant la loi uniforme sur l'intervalle $]0,1[$.
Quelle est la loi de la variable aléatoire $G(U)$?

- G est la réciproque de F donc, d'après la méthode d'inversion, la v.a.r. $G(U)$ a pour fonction de répartition F . Autrement dit, $G(U)$ suit la loi de Gumbel de paramètre 1.
- On peut le démontrer.
Soit $x \in \mathbb{R}$. Comme F est strictement croissante :

$$[G(U) \leq x] = [F(G(U)) \leq F(x)] = [U \leq F(x)]$$

Ainsi $F_{G(U)}(x) = F_U(F(x)) = F(x)$ car $F(x) \in]0,1[$.

- Par une méthode de votre choix, écrire en **Python** les commandes qui permettent de simuler la loi de T .

Les questions précédentes permettent d'écrire le programme :

```

1 u = rd.random()
2 t = -np.log(-np.log(u))
3 print(t)

```