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 $F : \mathbb{R} \to]0,1[$.
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 $U \hookrightarrow \mathcal{U}(]0,1[)$.

► Démontrer le résultat du théorème.

- La fonction F est bijective .
- Notons G la fonction de répartition de la v.a.r. $F^{-1}(U)$.
  Pour tout $x \in \mathbb{R}$, on a : $G(x) \ =$.

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.

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

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

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

- Si $x < a$:

- Si $x \in [a,b]$:

- Si $x > b$:

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()`.

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 $G : [0, 1[\to [0, +\infty[$.

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

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

- Si $x < 0$:

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

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()`.

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. $V = -\dfrac{1}{\lambda} \ln(1 - U)$?*
- *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 : $F(x) \ = \ \exp\left(-\mathrm{e}^{-\lambda x}\right)$.

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

▸ *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.*

On suppose maintenant que $\lambda = 1$.

▸ *Expliciter la bijection réciproque G de la fonction F.*

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

```
x = np.linspace(-2,2,400)
y = np.exp(-np.exp(-x))
plt.plot(x,y)
plt.plot(y,x)
```

Le réel 0 fait-il partie des nombres contenus dans le tableau `x = np.linspace(-2,2,400)` *?*

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

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

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