`Python` td 7 - Simulation de variables à densité

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

I. Rappel : générateur `rd.random`

L'instruction `x=rd.random()` renvoie un réel pris au hasard entre 0 et 1. Autrement dit, il s'agit de la simulation d'une variable X suivant la loi uniforme **continue** $X \hookrightarrow \mathcal{U}([0,1])$. Comme X prend presque sûrement ses valeurs dans $[0,1]$, `X=rd.random()` simule aussi la loi $\mathcal{U}([0,1])$.

L'instruction `L=rd.random(n)` renvoie une matrice ligne L à n cases (matrice unidimensionnelle), chaque coefficient de cette matrice prenant pour valeur un réel au hasard entre 0 et 1.

L'instruction `A=rd.random([n,p])` renvoie une matrice rectangulaire A à n lignes et p colonnes dont chaque coefficient prend pour valeur un réel au hasard entre 0 et 1.

II. Loi uniforme continue sur $[a,b]$

Soit $(a,b) \in \mathbb{R}^2$, où $a < b$. La commande

```
X= .......  *rd.random() +............
```

simule une variable uniforme à densité $X \hookrightarrow \mathcal{U}([a,b])$.

La commande

```
A= .......  *rd.random([n,p]) +............
```

donne une matrice A de taille $n \times p$ dont chaque case est la simulation d'une variable aléatoire suivant la loi $\mathcal{U}([a,b])$.

Exercice 1
Simuler 1000 fois la loi uniforme sur $[1,10]$ puis tracer l'histogramme des résultats obtenus (à répartir en 9 classes)

La commande `X=rd.uniform(a,b)` permet de simuler directement $X \hookrightarrow \mathcal{U}([a,b])$.

III. Loi exponentielle

Soit $\lambda > 0$. Rappelons que $X \hookrightarrow \mathcal{E}(\lambda)$ si X possède une densité de la forme

$$\forall t \in \mathbb{R}, \quad f_X(t) = \begin{cases} 0 \text{ si } \ t < 0 \\ \lambda.e^{-\lambda.t} \text{ si } t \geq 0 \end{cases}$$

Une telle variable admet une espérance et une variance, et $E(X) = \frac{1}{\lambda}$, $V(X) = \frac{1}{\lambda^2}$.

Exercice 2
Ecrire un programme Python qui trace la courbe d'une densité de $X \hookrightarrow \mathcal{E}(2)$ sur l'intervalle $[0,3]$.

La commande Python `rd.exponential(1/lambda)` simule une variable $X \hookrightarrow \mathcal{E}(\lambda)$.

Attention : le paramètre de la fonction est l'inverse du paramètre λ. Autrement dit on met en paramètre de la fonction Python l'espérance de X.

Comme d'habitude, la commande `rd.exponential(1/lambda),n)` simule n fois la loi exponentielle (matrice ligne) et `rd.exponential(1/lambda),[n,p])` simule $n \times p$ fois cette loi (matrice de taille $n \times p$).

Exercice 3
Simuler 1000 fois la loi exponentielle de paramètre 2 puis tracer un histogramme des résultats obtenus. Que remarque-t-on ?

IV. Loi normale

Rappelons que $X \hookrightarrow \mathscr{N}(m,\sigma^2)$ lorsque X est une variable aléatoire à densité et qu'une de ses densités est définie de la manière suivante:

$$\begin{array}{rccl} f: & \mathbb{R} & \to & \mathbb{R} \\ & t & \mapsto & f(t) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(t-m)^2}{2\sigma^2}} \end{array}$$

Dans ce cas, $E(X) = m$ et $V(X) = \sigma^2$.

La commande Python `rd.normal(m,sigma)` simule une variable $X \hookrightarrow \mathcal{N}(m,\sigma^2)$.
Attention : le deuxième paramètre de la fonction Python est l'écart-type σ.

Exercice 4
Soit $X \hookrightarrow \mathcal{N}(3,4)$. Simuler X.

Comme d'habitude, la commande `rd.normal(m,sigma,n)` simule n fois la loi ,normale (matrice ligne) et `rd.normal(m,sigma,[n,p])` simule $n \times p$ fois cette loi (matrice de taille $n \times p$).

Exercice 5
Simuler 10000 fois la loi normale centrée réduite. Tracer un histogramme des résultats obtenus.

V. Loi petit gamma

Rappelons que $X \hookrightarrow \gamma(\nu)$ lorsque X est une variable aléatoire à densité et qu'une de ses densités est :

$$\begin{array}{rrcll} f: & \mathbb{R} & \to & \mathbb{R} & \\ & t \leqslant 0 & \mapsto & f(t) & = 0 \\ & t > 0 & \mapsto & f(t) & = \dfrac{1}{\Gamma(\nu)}\, t^{\nu-1} e^{-t} \end{array}$$

X admet une espérance et une variance et $E(X) = \nu$, $V(X) = \nu$.

Exercice 6

`import scipy.special as sp`

La commande Python `sp.gamma` définit la fonction Γ. A l'aide de cette instruction, tracer la courbe de la densité de $X \hookrightarrow \gamma(\frac{1}{2})$ de $X \hookrightarrow \gamma(2)$, de $X \hookrightarrow \gamma(4)$ à chaque fois sur un intervalle du type $[0.01, A]$ où A est bien choisi.

La commande Python `rd.gamma(nu)` simule une variable $X \hookrightarrow \gamma(\nu)$.
Attention pour simuler n fois cette loi, la synthaxe est `rd.gamma(nu,1,n)`.
La commande `rd.gamma(nu,1,[n,p])` simule $n \times p$ fois cette loi (matrice de taille $n \times p$).

Exercice 7

Ecrire un programme qui simule 1000 fois la loi $\gamma(2)$ puis tracer l'histogramme des résultats obtenus.