## Agro-Veto – Planche 4

### Question de cours

Énoncer le théorème de Pythagore dans $\mathbb{R}^n$.

### Exercice avec préparation

Toutes les variables aléatoires de cet exercice sont définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.
Soit $X$ une variable aléatoire à densité, de densité $f$ telle que :

$$f(x) = \begin{cases} \dfrac{2}{x^3} & \text{si } x > 1 \\ 0 & \text{sinon} \end{cases}$$

1. Vérifier que $f$ est une densité de probabilité.
2. Déterminer la fonction de répartition de $X$.
3. Soit $U$ une variable aléatoire de loi uniforme sur $]0,1[$. Montrer que la variable aléatoire $V = \dfrac{1}{\sqrt{1-U}}$ suit la même loi que $X$.
4. Écrire un programme Python simulant une réalisation de la variable aléatoire $X$.
5. La variable aléatoire $X$ admet-elle une espérance ? une variance ? Si oui, les calculer, et vérifier vos réponses à l'aide du programme de la question 4.
6. Soit $(X_n)_{n \geqslant 1}$ une suite de variables aléatoires indépendantes de même loi que $X$. On définit, pour tout entier $n$ non nul, la variable aléatoire $T_n$ par :

$$T_n = \frac{\max(X_1, \ldots, X_n)}{\sqrt{n}}$$

   (a) Soit $n \in \mathbb{N}^*$. Déterminer la fonction de répartition de $T_n$.

   (b) Calculer alors pour tout réel $x$ :

$$G(x) = \lim_{n \to +\infty} \mathbb{P}(T_n \leqslant x)$$

   (c) On admet que $G$ est la fonction de répartition d'une variable aléatoire $T$ à densité. Montrer que $T$ admet pour densité la fonction $x$ donnée par :

$$g(x) = \begin{cases} \dfrac{2}{x^3} e^{-\frac{1}{x^2}} & \text{si } x > 0 \\ 0 & \text{sinon .} \end{cases}$$

   (d) À l'aide du changement de variable $x = \dfrac{1}{u}$, montrer que $T$ admet une espérance et la déterminer.

**Exercice sans préparation**

1. Écrire une fonction qui prend en argument un nombre entier naturel $n$ et qui renvoie la liste des diviseurs entiers naturels de $n$.
2. Un entier naturel $n$ est dit **parfait** si la somme de ses diviseurs est égale à $2n$.
   Par exemple, les diviseurs de 4 sont $1, 4, 2$ donc 4 n'est pas parfait car $1 + 4 + 2 \neq 8$.
   Les diviseurs de 6 sont $1, 2, 3, 6$ donc 6 est parfait.
   (a) Écrire une fonction qui prend en entrée un entier naturel et renvoie `True` s'il est parfait et `False` sinon.
   (b) Écrire une fonction qui prend en entrée un entier $n$ et qui renvoie les $n$ premiers nombres parfaits.

# Éléments de correction–Planche 4

## Exercice avec préparation

Soit $X$ une variable aléatoire à densité, de densité $f$ telle que :

$$f(x) = \begin{cases} \dfrac{2}{x^3} & \text{si } x > 1 \\ 0 & \text{sinon} \end{cases}$$

1. On vérifie :
   - $f$ est une fonction continue sur $\mathbb{R}$ sauf en 1 ;
   - $f$ est une fonction positive ;
   - Pour $x \geq 1$, $\displaystyle\int_1^x f(t)\ dt = \left[-t^{-2}\right]_1^x = 1 - x^{-2}$ et donc, en passant à la limite lorsque $x \to +\infty$, l'intégrable généralisée $\displaystyle\int_1^{+\infty} f(t)\ dt$ converge et vaut 1. Comme $f = 0$ sur $]-\infty, 1]$, $\displaystyle\int_{-\infty}^1 f(t)\ dt$ converge et vaut 0 et donc

$$\int_{-\infty}^{+\infty} f(t)\ dt = 1$$

   La fonction $f$ est donc une densité de probabilité.

2. Notons $F$ la fonction de répartition de $X$ : pour $x \in \mathbb{R}$, $F(x) = \mathbb{P}(X \leq x)$.
   - Si $x \leq 1$,
   $$F(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^x f(t)\ dt = 0$$
   - Si $x \geq 1$,
   $$F(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^x f(t)\ dt = \int_1^x f(t)\ dt = 1 - x^{-2}$$

3. La fonction $Q : ]0,1[ \to \mathbb{R}$ définie par

$$\forall u \in ]0,1[, \quad Q(u) = \frac{1}{\sqrt{1-u}}$$

   est continue strictement croissante, d'intervalle image $]1, +\infty[$, bijective de $]0,1[$ dans $]1, +\infty[$, de fonction réciproque $F$.
   Soit $U$ une variable aléatoire de loi uniforme sur $]0,1[$. Posons $V = Q(U)$, à valeurs dans $]1, +\infty[$ et calculons la fonction de répartition de $V$. Pour $v \in ]1, +\infty[$, on a

$$\mathbb{P}(V \leq v) = \mathbb{P}(Q(U) \leq v) = \mathbb{P}(U \leq Q^{-1}(v)) = \mathbb{P}(U \leq F(v)) = F(v)$$

   Comme la fonction de répartition de $V$ est celle de $X$, la variable aléatoire $V = \dfrac{1}{\sqrt{1-U}}$ suit la même loi que $X$.

4. 
```python
def X():
    u = np.random.rand() #tirage uniforme sur [0,1]
    return 1/np.sqrt(1-u)
```

5. La variable aléatoire $X$ admet une espérance si $\int_1^{+\infty} xf(x)\ dx$ est (A)CV. Comme

$$\int_1^{+\infty} |x|f(x)\ dx = \int_1^{+\infty} xf(x)\ dx = [-2/x]_1^{+\infty} = 2$$

alors $X$ admet une espérance $\mathbb{E}(X)$ et $\mathbb{E}(X) = 2$.
La variable aléatoire $X$ admet une variance si $\int_1^{+\infty} x^2 f(x)\ dx$ est (A)CV. Comme

$$\int_1^{+\infty} x^2 f(x)\ dx = \int_1^{+\infty} \frac{2}{x}\ dx \text{ DV}$$

alors $X$ n'admet pas de variance.
On écrit une fonction Python calculant moyenne et variance sur un grand échantillon de tirages au sort indépendants, suivant la loi de $X$. D'après la loi des grands nombres , la valeur `EX` a de grandes chances d'être proche de $\mathbb{E}(X)$.

```
def EXVX_LFGN(NS=10_000):
    S = 0.0
    Ech = []
    for _ in range(NS):
        Ech.append(X())
        S += Ech[-1]
    EX = S/NS
    return EX
```

6. Soit $(X_n)_{n\geqslant 1}$ une suite de variables aléatoires indépendantes de même loi que $X$. On définit, pour tout entier $n$ non nul, la variable aléatoire $T_n$ par :

$$T_n = \frac{\max(X_1, \ldots, X_n)}{\sqrt{n}}$$

(a) La variable $T_n$ est à valeurs dans $I_n = \left[\frac{1}{\sqrt{n}}, +\infty\right[$ (on a donc $\mathbb{P}(T_n \leq t) = 0$ si $t < \frac{1}{\sqrt{n}}$ et on a, pour $t \in I_n$,

$$\begin{array}{rcl} \mathbb{P}(T_n \leq t) & = & \mathbb{P}(\max(X_1, \ldots, X_n) \leq \sqrt{n}t) \\ & = & \mathbb{P}([X_1 \leq \sqrt{n}t] \cap \cdots \cap [X_n \leq \sqrt{n}t]) \\ & = & \mathbb{P}(X_1 \leq \sqrt{n}t) \ldots . \mathbb{P}(X_n \leq \sqrt{n}t) \\ & = & \left(1 - \frac{1}{nt^2}\right)^n \end{array}$$

(b) Pour tout réel $x$ :

— Soit $x \leq 0$, auquel cas, $\forall n \in \mathbb{N}^*$, $\mathbb{P}(T_n \leq x) = 0$ et donc

$$G(x) = \lim_{n \to +\infty} \mathbb{P}(T_n \leqslant x) = 0$$

— Soit $x > 0$ et à ce moment, pour tout $n$, à partir du rang où tel que $\frac{1}{\sqrt{n}} < x$,

$$\mathbb{P}(T_n \leq x) = \left(1 - \frac{1}{nx^2}\right)^n = \exp\left(n \ln\left(1 - \frac{1}{nx^2}\right)\right)$$

Or, lorsque $x$ tend vers 0, $\ln(1+x) \sim x$ et donc

$$n \ln\left(1 - \frac{1}{nx^2}\right) \sim n\left(-\frac{1}{nx^2}\right) = -\frac{1}{x^2}$$

et donc, après passage à l'exp,

$$G(x) = \lim_{n \to +\infty} \mathbb{P}\left(T_n \leqslant x\right) = e^{-\frac{1}{x^2}}.$$

(c) Comme $G$ est croissante, continue (y compris en 0 où il faut observer que $-\frac{1}{x^2} \to -\infty$ lorsque $x \to 0^+$), $\mathcal{C}^1$ sur chacun des intervalles $]-\infty, 0[$ et $]0, +\infty[$, $G$ est la fonction de répartition d'une variable aléatoire $T$ à densité $g$ dont une densité est donnée par dérivation de $G$ :

$$\forall x \in ]-\infty, 0], \quad g(x) = 0$$

et

$$\forall x \in ]0, +\infty[, \quad g(x) = G'(x) = \frac{2}{x^3} e^{-\frac{1}{x^2}}$$

(d) La v.a. $T$ admet une espérance si et seulement si $\int_0^{+\infty} x g(x)\, dx$ est (A)CV et on a alors, en cas d'(A)CV :

$$\mathbb{E}(X) = \int_0^{+\infty} x.\frac{2}{x^3} e^{-\frac{1}{x^2}}\, dx = \int_0^{+\infty} \frac{2}{x^2} e^{-\frac{1}{x^2}}\, dx$$

À l'aide du changement de variable $x = \frac{1}{u}$ ($u = \frac{1}{x}$, $du = -\frac{1}{x^2} dx$, $u$ variant de $+\infty$ à 0 lorsque $x$ varie de 0 à $+\infty$)–la fonction $\phi : x \mapsto u = \frac{1}{x}$ étant de classe $\mathcal{C}^1$, strictement décroissante, bijective de $]0, +\infty[$ sur lui-même–, cette intégrale généralisée est de même nature (et à même valeur, en cas de convergence) que :

$$E := 2 \int_0^{+\infty} e^{-u^2}\, du$$

Cette IG est convergente (cours sur les intégrales gaussiennes !) et vaut $\mathbb{E}(T) = \sqrt{\pi}$.

### Exercice sans préparation

1. ```python
def diviseur(n):
    L = []
    s = 1
    while s<np.sqrt(n)+1:
        if n%s ==0:
            if not (s in L):
                L.append(s)
            if not (n//s in L):
                L.append(n//s)
        s += 1
    return L
```

2. (a) ```python
def is_perfect(n):
    L = diviseur(n)
    return sum(L)==2*n
```

   (b) ```python
def perfect(n):
    L = []
    k = 1
    while len(L) < n:
        if is_perfect(k):
            L.append(k)
        k+=1
    return L
```
```