Séance Python 3 — Préparation à l'oral Agro/Véto

Ex 1 : Soit X une variable aléatoire de densité f :

$$f(x) = 1 \quad \text{si } x \in \left[-\frac{1}{2};\ \frac{1}{2}\right] \quad \text{et} \quad f(x) = 0 \ \text{ sinon}$$

Question : Ecrire une fonction qui simule la réalisation de la variable X.

Ex 2 : On considère une urne contenant initialement une boule blanche et une boule noire.
On tire successivement une boule dans l'urne puis :

- si on tire une boule blanche, on remet la boule blanche dans l'urne ainsi qu'une autre boule blanche
- si on tire une boule noire, on remet la boule noire dans l'urne et le jeu s'arrête.

On suppose que les boules sont indiscernables au toucher. On note N le nombre de boules présentes dans l'urne à la fin de l'expérience.
On admet que : $\forall k \in \mathbb{N}\backslash\{0,1\}, \quad P(N=k) = \dfrac{1}{k(k-1)}$.

Question : (*au choix*)

1) Ecrire une fonction Python qui simule l'expérience et renvoie la valeur de N.
2) Ecrire une fonction Python qui simule la réalisation de la variable aléatoire N.

Ex 3 : On considère pour tout entier $n \geqslant 2$, la fonction f_n définie sur $\mathbb{R}_+^*$ par : $f_n(x) = x^n - \ln x - n$.
D'après le tableau de variations :

x	0		u_n		$n^{-\frac{1}{n}}$		v_n		$+\infty$
$f_n'(x)$	‖		$+$		0		$-$		
f_n	$+\infty$	↘	0	↘	$f_n\left(-n^{\frac{1}{n}}\right) < 0$	↗	0	↗	$+\infty$

Il existe un unique réel $v_n \in]\, n^{-\frac{1}{n}}, +\infty\,[$ tel que $f_n(v_n) = 0$ et on admet que : $\forall n \geqslant 2, \quad v_n \leqslant (2n)^{\frac{1}{n}}$.

Question :
En utilisant l'algorithme de dichotomie, déterminer des valeurs approchées à 10^{-3} près des termes v_n pour n allant de 2 à 30. Représenter la suite (v_n) graphiquement.

Ex 4 : Soit n un entier naturel non-nul. On dispose de n jetons et de trois urnes numérotées de 1 à 3 . Pour chaque jeton, on choisit une des trois urnes au hasard et avec équiprobablité et on place le jeton dans l'urne choisie. Le placement de chaque jeton est indépendant du placement de tous les autres jetons.
On note X la variable aléatoire égale au nombre de jetons contenus dans l'urne 1 à la fin de l'expérience, et on note Y le nombre d'urnes restées vides à la fin de l'expérience.

Question : Ecrire une fonction Python qui prend en argument un entier naturel n non nul, simule l'expérience aléatoire décrite ci-dessus, et renvoie les valeurs de X et de Y obtenues.

Ex 5 : On pose $H_0(X) = 1$ et pour tout $k \in \mathbb{N}^*, H_k(X) = \prod_{i=0}^{k-1}(X - i)$.

En Python, un polynôme de $\mathbb{R}_n[X]$ est codé en listant ses $n+1$ coefficients par ordre croissant de degré. Par exemple, dans $\mathbb{R}_4[X]$, le polynôme $P = 5X^3 - 2X + 3$ est représenté par la liste $[3, -2, 0, 5, 0]$.

1) Programmer une fonction Python qui prend en argument une liste de longueur $n+1$ modélisant un polynôme P de degré inférieur ou égal à $n-1$ dans $\mathbb{R}_n[X]$ et un réel a, et qui renvoie alors la liste modélisant $(X - a)P$.
2) Programmer une fonction Python qui prend en argument un entier naturel non nul n et qui renvoie la liste modélisant le polynôme H_n.

Ex 6 : Soit X une variable aléatoire à densité, de densité f telle que :

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

On admet que si U est une variable aléatoire de loi uniforme sur $]0, 1[$ alors la variable aléatoire $V = \dfrac{1}{\sqrt{1-U}}$ suit la même loi que X.

Question : Écrire un programme Python simulant une réalisation de la variable aléatoire X.

Ex 7 : On étudie la descendance d'une fleur dont le nombre de descendants suit la loi binomiale $\mathscr{B}(2, p)$, avec $p \in]0, 1[$ fixé. Les descendantes de la première fleur ont des descendantes de façon mutuellement indépendantes et dans les mêmes conditions que la première fleur.

1) Ecrire une fonction `binomiale(n, p)` qui simule la réalisation d'une variable suivant la loi $\mathscr{B}(n, p)$.
2) Compléter la fonction Python `temps_extinction()` récursive qui simule l'expérience et renvoie la valeur de la première année où la plante a disparu.

```
def temps_extinction():
    a = binomiale(2, p)
    if a == 0:
        return  ...
    elif a == 1:
        t = temps_extinction()
        return ...
    t1 = ...
    t2 = ...
    return max(t1,t2)+1
```

3) À l'aide d'un programme Python, estimer la durée de vie moyenne de la descendance pour $p = \dfrac{1}{4}$ par la moyenne de $N = 10000$ simulations, et vérifier que cette valeur est bien inférieure à 2.

Ex 8 : Une urne contient initialement une boule bleue et une boule rouge, si on tire la boule bleue on retire ensuite dans la même urne sans avoir remis la boule bleue, si on tire une boule rouge on refait un tirage dans une urne contenant une boule bleue et une boule rouge. On enchaîne ainsi des tirages indéfiniment. *Après une boule bleue on a toujours une boule rouge et après une rouge on obtient de manière équiprobable une rouge ou une bleue.*

On note Z_n le nombre de boules bleues tirées au cours des n tirages

1) Ecrire une fonction Python `simul_Z(n)` qui permet de simuler la réalisation de Z_n.
2) Trouver une valeur approchée de l'espérance de Z_n pour n allant de 2 à 10.