DS 10

Devoir surveillé du Mardi 4 Mars

La calculatrice est interdite. Durée : 2h

Dans tout ce devoir, on suppose avoir importé sur `Python` les bibliothèques :

- `numpy` avec le raccourci `np` ;
- `numpy.random` avec le raccourci `rd` ;
- `numpy.linalg` avec le raccourci `al` ;
- `matplotlib.pyplot` avec le raccourci `plt`.

Exercice 1

1. Soit $\lambda > 0$. On admet que si $U \hookrightarrow \mathcal{U}([0,1])$, alors la variable $X = -\frac{1}{\lambda}\ln(1-U)$ suit une loi exponentielle de paramètre λ.

   Écrire une fonction `exponentielle(lambda)` en langage `Python` simulant une loi $\mathcal{E}(\lambda)$ à partir de la fonction `rd.random()` (sans utiliser `rd.exponential`).

2. Soit $\lambda > 0$. On admet que si $X \hookrightarrow \mathcal{E}(\lambda)$, alors $Y = \lfloor X \rfloor + 1 \hookrightarrow \mathcal{G}(1-e^{-\lambda})$.

   (a) Écrire une fonction `geom(p)` en langage `Python` simulant une loi $\mathcal{G}(p)$ à partir de la fonction `exponentielle` (sans utiliser `rd.geometric`).

   (b) On souhaite construire un vecteur `x` de taille 10000 contenant 10000 simulations de la loi $\mathcal{G}(0.2)$ à partir de la fonction `geom`. Compléter pour cela le programme `Python` suivant :

```
x = .....
for k in range(10000) :
    x[k] = .....
```

   (c) En exécutant l'instruction suivante :

```
m = np.mean(x)
v = np.mean((x-m)**2)
print(m, v)
```

   on obtient `5.0404` et `20.133581` . Ces résultats sont-ils conforme à vos attentes ? Expliquer.

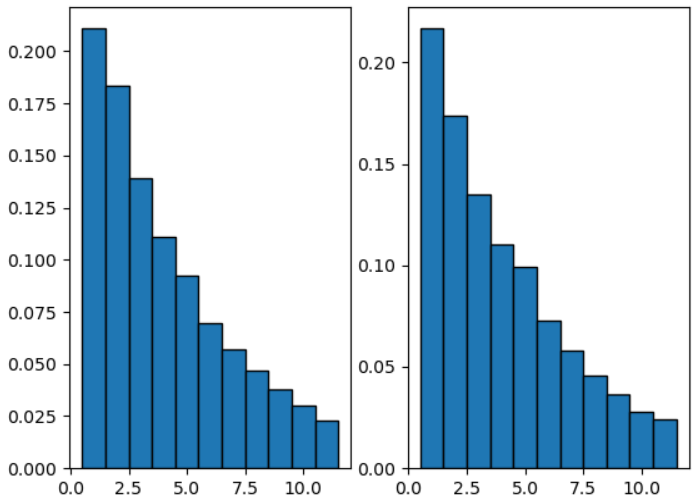
   (d) En exécutant le script :

```
c = np.arange(0.5, 12.5)
plt.subplot(1, 2, 1)
plt.hist(x, c, density='True', edgecolor='k')

plt.subplot(1, 2, 2)
plt.hist(rd.geometric(0.2, 10000), c, density='True',
edgecolor='k')

plt.show()
```

   on obtient les diagrammes en bâtons suivants :



A quoi correspondent chacun des diagrammes en bâtons ? Que peut-on conclure de ces diagrammes ?

Exercice 2

On admet que le nombre X de têtards issus des oeufs pondus en mars et avril d'une année suit une loi de Poisson de paramètre $\lambda = 20$.
Ces têtards sont soumis à des prédateurs nombreux et voraces, et on admet que chacun d'entre eux à une probabilité $p = 0,05$ de parvenir à son développement complet, et qu'ils se développent de façon indépendante.
On note Y le nombre de têtards qui parviennent à leur développement complet et se transforment donc en une grenouille.

1. Quelle est la loi de Y sachant $(X = k)$? Justifier.

2. Recopier et compléter la fonction `Python` suivante qui pour qu'elle simule une fois cette expérience et retourne la valeur y prise par la variable Y :

```
def simulY()
    x = .....
    y = .....
    return(y)
```

3. Écrire une fonction d'en-tête `def SimulY(N)` en langage `Python` donnant un échantillon de taille N de la loi de Y.

4. Recopier et compléter la fonction `Python` suivante pour qu'elle renvoie un vecteur $V = [v_0, \dots, v_{10}]$ tel que :

$$\forall k \in [\![0, 10]\!], \quad v_k = \frac{\lambda^k}{k!} e^{-\lambda}.$$

```
def loipoisson(lb):
    V = np.zeros(11)
    V[0] = .....
    for k in range(1,11):
        V[k] = .....
    return(V)
```

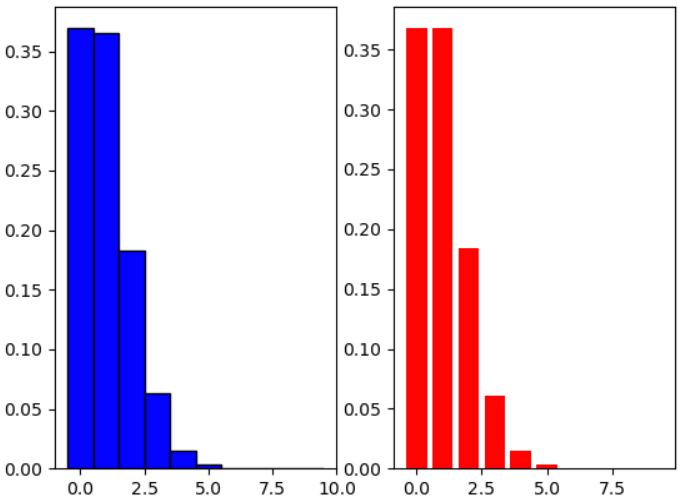
5. A la suite des instructions précédentes, on ajoute les commandes `Python` suivantes :

```
U = SimulY(100000)
c = np.arange(-0.5, 10)
plt.subplot(1, 2, 1)
plt.hist(U, c, density='True', edgecolor='k', color='blue')

n = np.arange(10)
V = loipoisson(1)
plt.subplot(1, 2, 2)
plt.bar(n, V, color='red')
plt.show()
```

Que contiennent les variables `U`, `c`, `n` et `V` ? Que fait ce programme ?

6. Après exécution, on obtient les graphiques suivants :



Quelle conjecture peut-on faire sur la variable aléatoire Y ? Justifier votre réponse.

Exercice 3

Soit $k \in \mathbb{N}^*$ et $\lambda > 0$. On considère une variable aléatoire à densité X donc la fonction de répartition est :

$$F(x) = \begin{cases} 1 - \dfrac{\lambda^k}{x^k} & \text{si } x > \lambda \\ 0 & \text{sinon.} \end{cases}$$

On dit que X suit la loi de Pareto de paramètre λ et k.

1. (a) Justifier que F réalise une bijection de $]\lambda, +\infty[$ dans $]0, 1[$ et déterminer sa bijection réciproque.
   (b) En utilisant la méthode d'inversion, en déduire une fonction `Python` qui simule une variable aléatoire suivant une loi de Pareto de paramètres λ et k.

2. Soient $X_1, \cdots, X_k$ k variables aléatoires indépendantes suivant toutes la loi uniforme sur $]0, 1]$. On pose alors $Y = \dfrac{\lambda}{\max(X_1, \ldots, X_k)}$.
   (a) Montrer que Y suit une loi de Pareto de paramètres λ et k.
   (b) En déduire une autre fonction `Python` pour simuler la loi de Pareto.

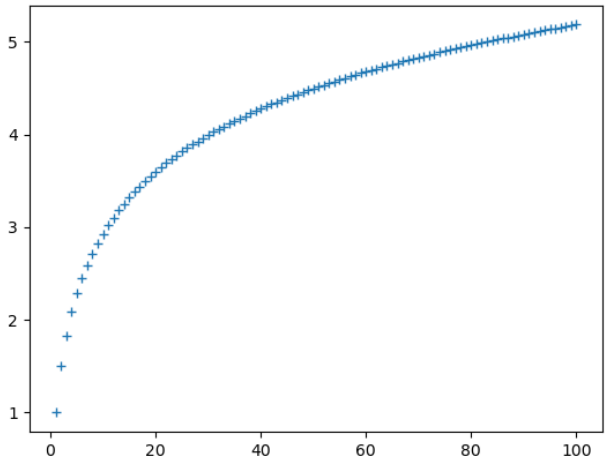
Exercice 4

On considère la série harmonique $\sum_{n\geq 1} \frac{1}{n}$. On note $H_n = \sum_{k=1}^{n} \frac{1}{k}$ la n-ième somme partielle de cette série.

1. On considère le programme `Python` suivant :

```
U = np.arange(1.,101.)
V = np.cumsum(U**(-1))
plt.plot(U,V,'+')
plt.show()
```

En exécutant ce programme, on obtient le graphe suivant :



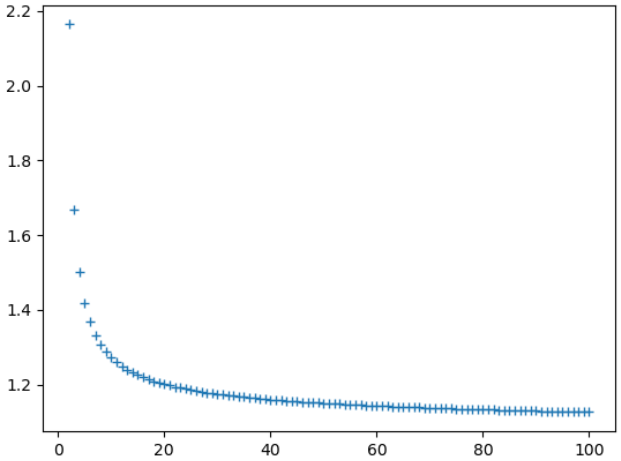
Que contient la variable `U` ? Et la variable `V` ? Que fait ce programme ? Quel résultat du cours le graphe obtenu illustre-t-il ?

2. Écrire en `Python` une fonction `seuil` qui retourne le premier entier naturel N tel que $H_N \geq 10$.

3. On considère le programme `Python` suivant :

```
U = np.arange(2.,101.)
V = 1+np.cumsum(U**(-1))
W = np.log(U)
T = V/W
plt.plot(U,T,'+')
plt.show()
```

En exécutant ce programme, on obtient le graphe suivant :



Que contiennent les variables `U`, `V`, `W` et `T` ? Que fait ce programme ? Que peut-on conjecturer à partir du graphe obtenu ?

4. On considère les suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ définies pour tout entier naturel n non nul par :

$$u_n = H_n - \ln(n) \quad \text{et} \quad v_n = H_n - \ln(n+1).$$

   (a) Montrer que, pour tout entier $n \geq 1$, $\dfrac{1}{n+1} \leq \displaystyle\int_n^{n+1} \frac{1}{t}dt \leq \frac{1}{n}$.

   (b) En déduire que les suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ sont adjacentes. On notera γ leur limite commune.

   (c) Écrire en `Python` une fonction `gamma` qui, étant donné un réel $\varepsilon > 0$, retourne une approximation de γ à ε près.
