Dans les questions faisant intervenir des instructions en langage Python, on prendra soin d'importer les bibliothèques nécessaires lors de leur première utilisation.

Exercice 1 (d'après Ecricome)

Soit n un entier naturel non nul.
Une urne contient n boules indiscernables au toucher numérotées de 1 à n.On tire une boule au hasard dans l'urne. Si cette boule tirée porte le numéro k, on place alors dans une seconde urne toutes les boules suivantes : une boule numérotée 1, deux boules numérotées 2 et plus généralement, pour tout $j \in [\![1,k]\!]$, j boules numérotées j, jusqu'à k boules numérotées k. Les boules de cette deuxième urne sont indiscernables au toucher. On effectue alors un tirage au hasard dans cette seconde urne.
On note X la variable aléatoire égale au numéro de la première boule tirée et Y la variable aléatoire égale au numéro de la deuxième boule tirée.

1. Le premier tirage se fait dans une urne qui contient n boules indiscernables au toucher et chaque numéro est alors équiprobable ; on a reconnu la loi uniforme

$$X \hookrightarrow \mathscr{U}([\![1,n]\!]).$$

En particulier, les formules du cours donnent directement $E(X) = \dfrac{n+1}{2}, \quad V(X) = \dfrac{n^2-1}{12}$.

2. Dans le *pire* des cas, on pioche lors du premier tirage la boule numérotée 1 et il n'y a dans la deuxième urne qu'une boule numérotée 1. Si on pioche au premier coup la boule numérotée n, il y a ensuite des boules numérotées de 1 à n dans la deuxième urne. Ainsi, la valeur de la deuxième boule piochée est toujours une valeur entre 1 et n et toutes les valeurs sont possibles. On a donc $Y(\Omega) = [\![1,n]\!]$.

3. Soit $k \in [\![1,n]\!]$.

a. On suppose que l'évènement $[X=k]$ est réalisé.
D'après la description de l'expérience, on a disposé dans la deuxième urne un nombre de boules égal à

$$1+2+\ldots+k = \sum_{j=1}^{k} j = \frac{k(k+1)}{2}.$$

b. Si $j \in [\![1,k]\!]$, il y a dans l'urne des boules numérotées j (et il y en a exactement j). Sinon, les boules j ne font pas partie de l'urne (et la probabilité de les piocher alors nulle). On a donc par équiprobabilité et avec la formule ci-dessus pour le total de boules :

$$P_{[X=k]}(Y=j) = \begin{cases} \dfrac{2j}{k(k+1)} & \text{si } 1 \leq j \leq k \\ 0 & \text{si } j > k \end{cases}$$

4. **a.** Commençons par mettre au même dénominateur.

$$\frac{a}{k} + \frac{b}{k+1} = \frac{a(k+1)+bk}{k(k+1)} = \frac{(a+b)k+a}{k(k+1)}.$$

On cherche donc a et b tels que $\frac{(a+b)k+a}{k(k+1)} = \frac{1}{k(k+1)}$ ou encore $(a+b)k+a=1$ en multipliant $k(k+1) \neq 0$.
Ainsi, par identification des polynômes en k :

$$(a+b)k+a=1 \iff \begin{cases} a+b &= 0 \\ a &= 1 \end{cases} \iff \begin{cases} b &= -1 \\ a &= 1 \end{cases}$$

On peut alors écrire, pour tout $k \in \mathbb{N}^*$,

$$\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}.$$

b. Soit $j \in [\![1,n]\!]$. D'après la formule des probabilités totales appliquée au s.c.e $\{[X=k],\ k\in[\![1,n]\!]\}$, on a

$$\begin{aligned}
P(Y=j) &= \sum_{k=1}^{n} P_{[X=k]}(Y=j)P(X=k) \\
&= \sum_{k=j}^{n} P_{[X=k]}(Y=j)P(X=k) && \text{(car les autres termes sont nuls)} \\
&= \sum_{k=j}^{n} \frac{2j}{k(k+1)} \times \frac{1}{n} && \text{(d'après les questions 1 et 3.b)} \\
&= \frac{2j}{n} \sum_{k=j}^{n} \frac{1}{k(k+1)} && \text{(on factorise la somme par les facteurs indépendants de } k\text{)} \\
&= \frac{2j}{n} \sum_{k=j}^{n} \left(\frac{1}{k} - \frac{1}{k+1}\right) && \text{(d'après la question précédente)} \\
&= \frac{2j}{n} \left(\frac{1}{j} - \frac{1}{n+1}\right) && \text{(par télescopage)} \\
&= \frac{2(n+1-j)}{n(n+1)},
\end{aligned}$$

5. Y a un univers image fini, elle admet donc une espérance. De plus,

$$\begin{aligned}
E(Y) &= \sum_{j=1}^{n} jP(Y=j) = \sum_{j=1}^{n} j\frac{2(n+1-j)}{n(n+1)} \\
&= \frac{2}{n(n+1)} \sum_{j=1}^{n} j(n+1-j) = \frac{2}{n(n+1)} \left((n+1)\sum_{j=1}^{n} j - \sum_{j=1}^{n} j^2\right) \\
&= \frac{2}{n(n+1)} \left((n+1)\frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6}\right) \\
&= (n+1) - \frac{2n+1}{3} \\
&= \frac{n+2}{3},
\end{aligned}$$

6. Soit $n \geq 2$. Si X et Y étaient indépendantes alors, pour tout couple $(k,j)\in[\![1,n]\!]^2$, on aurait $P_{[X=k]}(Y=j) = P(Y=j) \neq 0$.
Or, la probabilité de gauche est nulle dès que $j > k$. Les variables X et Y ne sont donc pas indépendantes.
Pour $n=1$, les deux variables aléatoires X et Y sont certaines et égales à 1... elles sont donc indépendantes.

7. **a.** Par le théorème de transfert, on a

$$\begin{aligned}
E(XY) &= \sum_{k=1}^{n}\sum_{j=1}^{n} kjP([X=k]\cap[Y=j]) = \sum_{k=1}^{n}\sum_{j=1}^{n} kjP([X=k])P_{[X=k]}([Y=j]) \\
&= \sum_{k=1}^{n}\sum_{j=1}^{k} kj \times \frac{1}{n} \times \frac{2j}{k(k+1)} \\
&= \frac{1}{n}\sum_{k=1}^{n} \frac{2}{k+1} \sum_{j=1}^{k} j^2 = \frac{1}{n}\sum_{k=1}^{n} \frac{2}{k+1} \times \frac{k(k+1)(2k+1)}{6} \\
&= \frac{1}{3n}\sum_{k=1}^{n} k(2k+1) = \frac{1}{3n}\left(2\sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} k\right) \\
&= \frac{1}{3n}\left(\frac{n(n+1)(2n+1)}{3} + \frac{n(n+1)}{2}\right) \\
&= \frac{n+1}{18}(2(2n+1)+3) \\
&= \frac{(n+1)(4n+5)}{18},
\end{aligned}$$

b. Par la formule de Koenig-Huygens :

$$\begin{aligned}\mathrm{Cov}(X,Y) &= E(XY)-E(X)E(Y)=\frac{(n+1)(4n+5)}{18}-\frac{n+1}{2}\times\frac{n+2}{3}\\ &= \frac{n+1}{18}(4n+5-3(n+2))\\ &= \frac{(n+1)(n-1)}{18}=\frac{n^2-1}{18}\end{aligned}$$

(pour $n=1$, la covariance est nulle, ce qui est cohérent avec ce qu'on a mentionné quant à l'indépendance de X et Y.)

8. **a.** Il suffit d'utiliser l'instruction append pour compléter la liste avec deux boucles `for`.

```
def seconde_urne(k):
    L=[ ]
    for j in range(1,k+1) :
        for i in range(j):
            L.append(j)
    return L
```

b. La variable X se simule grâce à la commande `rd.randint(1,n+1)`. On crée l'urne 2 à l'aide de la valeur de X. On prend ensuite le terme de la liste `urne2` à la position i (où i est choisi aléatoirement uniformément parmi le nombre de boules disponibles) pour Y. Ceci donne

```
def simul_XY(n) :
    X = rd.randint(1,n+1)
    urne2=seconde_urne(X)
    nb=len(urne2) # nombre de boules dans l'urne 2
    i=rd.randint(0, nb)
    Y=urne2[i]
    return X,Y
```

c. La variable `liste` contient la liste des fréquences de chaque valeur prise par Y lors de 10000 simulations de celle-ci. C'est à dire qu'on *estime* la loi de Y (les fréquences observées donnent des valeurs approchées des valeurs théoriques $P(Y=j)$). On commence, avec la commande `liste=[0]*n` par créer une liste de n zéros qui va être actualisé. Le $j-$ième terme de la liste (indexé en Python par $j-1$) contient la fréquence de passage par la valeur j.
En effet, la commande `j = simul_XY[1]` simule Y (c'est la deuxième composante du couple (X,Y) car on indexe en commençant par 0...) et les composantes de `liste` sont les fréquences qui augmente de 1 divisé par l'effectif total dès lors que Y a pris la valeur de la composante en question.

9. **a.** Le point moyen a pour coordonnées $(\overline{x},\overline{y})$ où $\overline{x}$ et $\overline{y}$ désignent respectivement les moyennes des 20 valeurs obtenues par les simulations de X et de Y.
Lorsque n devient grand, la *loi faible des grands nombres* permet d'affirmer (enfin il faudrait dire que X et Y admettent une variance mais c'est le cas car les univers images sont finis) que la moyenne empirique d'un $n-$échantillon fournit une bonne estimation de l'espérance. Donc le point moyen devrait avoir des coordonnées

$$\overline{x}\simeq E(X)=\frac{21}{2}=10,5, \qquad \text{et } \overline{y}\simeq E(Y)=\frac{22}{3}\simeq 7,33.$$

b. Les valeurs prises par X et Y sont entre 1 et 20 : on peut éliminer la figure 1 (le nuage de points ne correspond pas). La droite de régression doit être croissante vu l'allure des trois nuages de points. On peut éliminer la 4. La droite de régression de la figure 2 est trop haute pour être une bonne approximation.
C'est donc la figure 3 qui correspond au nuage de points étudié.

Pour le plaisir, on se permet d'ajouter quelques lignes de code qui permettent de générer un tel nuage de points. On observera que certains points sont d'ailleurs confondus.

```
import matplotlib.pyplot as plt
nuageX = []
nuageY = []
for i in range(50):
    X, Y = simul_XY(20)
    nuageX.append(X)
    nuageY.append(Y)
EX = np.mean(nuageX)
VX = np.mean(np.multiply(nuageX,nuageX))-EX**2
```

```
EY = np.mean(nuageY)
EXY = np.mean(np.multiply(nuageX,nuageY))
covXY = EXY-EX*EY
a = covXY/VX
b = EY-a*EX
plt.plot(nuageX,nuageY,'x')
plt.plot([b,20],[0,20*a+b],'--')
plt.show()
```