

Modèles probabilistes

I

Tirages dans une urne (banque CCP 95)

1. Une urne contient deux boules blanches et huit boules noires. Au cours d'une partie, on tire successivement cinq boules :

- pour chaque boule blanche tirée, on gagne deux points ;
- pour chaque boule noire tirée, on perd trois points.

On note X , le nombre de boules blanches tirées et Y , le nombre de points obtenus dans une partie.

I.1 Modélisations

2. Conventions

On modélise une partie par une famille $(U_k)_{1 \leq k \leq 5}$ de cinq variables aléatoires de Bernoulli définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbf{P})$.

2.1 Pour chaque $1 \leq k \leq 5$, les événements

$$[U_k = 0] \quad \text{et} \quad [U_k = 1]$$

forment un système complet et on convient d'interpréter ces événements de la manière suivante :

k -ième tirage	
$[U_k = 1]$	Boule blanche
$[U_k = 0]$	Boule noire

2.2 On convient que, quelle que soit la composition de l'urne, chaque boule a la même probabilité d'être tirée.

2.3 La signification mathématique de [2.2] n'est pas claire : on ne sait pas ce que signifie au juste la *probabilité* dont il est question. (Est-ce une *mesure de probabilité*? Est-ce la *probabilité d'un événement*? Mais alors lequel?)

Il ne s'agit en fait pas d'un modèle mathématique mais d'un *principe heuristique*, qui va nous guider pour définir un modèle de la loi jointe des variables U_1, \dots, U_5 .

3. Loi initiale

Par hypothèse d'équiprobabilité [2.2], la probabilité de tirer une boule blanche (resp. noire) est égale à la proportion de boules blanches (resp. de boules noires) :

$$\mathbf{P}(U_1 = 1) = \frac{2}{2+8} = \frac{1}{5} \quad \text{et donc} \quad \mathbf{P}(U_1 = 0) = \frac{4}{5}.$$

Loi jointe**4. Tirage avec remise**

Si le tirage des boules est effectué avec remise, chaque boule est tirée exactement dans les mêmes conditions. Il est donc raisonnable de *convenir* dans ce cas que les variables aléatoires U_1, U_2, \dots, U_5 sont indépendantes et toutes de même loi :

$$\forall 1 \leq k \leq 5, \quad \mathbf{P}(U_k = 1) = \frac{1}{5} \quad \text{et} \quad \mathbf{P}(U_k = 0) = \frac{4}{5}.$$

5. Tirage sans remise

Si le tirage est effectué sans remise, les valeurs des variables U_1, \dots, U_{k-1} permettent de déterminer la composition de l'urne au moment du k -ième tirage.

5.1 Initialement, l'urne contenait dix boules, dont deux blanches. On a retiré $(k-1)$ boules (une à chaque tirage), dont $(U_1 + \dots + U_{k-1})$ étaient blanches. L'hypothèse d'équiprobabilité [2.2] suggère alors la *convention* :

$$\mathbf{P}(U_k = 1 \mid U_1 = \varepsilon_1, \dots, U_{k-1} = \varepsilon_{k-1}) = \frac{2 - (\varepsilon_1 + \dots + \varepsilon_{k-1})}{10 - (k-1)}$$

quelles que soient les valeurs de $\varepsilon_1, \dots, \varepsilon_{k-1}$ telles que $0 \leq \varepsilon_1 + \dots + \varepsilon_{k-1} \leq 2$ (de manière à conditionner par un événement non négligeable).

5.2 On déduit de cette convention que

$$\mathbf{P}(U_k = 0 \mid U_1 = \varepsilon_1, \dots, U_{k-1} = \varepsilon_{k-1}) = 1 - \frac{2 - (\varepsilon_1 + \dots + \varepsilon_{k-1})}{11 - k}$$

quelles que soient les valeurs de $\varepsilon_1, \dots, \varepsilon_{k-1}$.

On devine ainsi que, contrairement au cas d'un tirage avec remise [4], les variables U_k ne sont pas indépendantes : la loi de U_k dépend des valeurs prises par U_1, \dots, U_{k-1} .

5.3 On peut ainsi calculer la loi jointe de (U_1, \dots, U_5) par la formule des probabilités composées.

$$\begin{aligned} \mathbf{P}(U_1 = \varepsilon_1, \dots, U_5 = \varepsilon_5) &= \mathbf{P}(U_1 = \varepsilon_1) \times \mathbf{P}(U_2 = \varepsilon_2 \mid U_1 = \varepsilon_1) \times \mathbf{P}(U_3 = \varepsilon_3 \mid U_1 = \varepsilon_1, U_2 = \varepsilon_2) \\ &\quad \times \mathbf{P}(U_4 = \varepsilon_4 \mid U_1 = \varepsilon_1, U_2 = \varepsilon_2, U_3 = \varepsilon_3) \\ &\quad \times \mathbf{P}(U_5 = \varepsilon_5 \mid U_1 = \varepsilon_1, U_2 = \varepsilon_2, U_3 = \varepsilon_3, U_4 = \varepsilon_4). \end{aligned}$$

5.4 En appliquant la formule des probabilités totales, on vérifie que $\mathbf{P}(U_k = 1) = 1/5$ pour $2 \leq k \leq 5$ et donc que les variables U_k suivent toutes la même loi :

$$\forall 1 \leq k \leq 5, \quad \mathbf{P}(U_k = 1) = \frac{1}{5} \quad \text{et} \quad \mathbf{P}(U_k = 0) = \frac{4}{5}.$$

Ce calcul démontre qu'on obtient les mêmes lois marginales avec les deux modèles (tirage avec remise ou tirage sans remise).

I.2 Loi de X

6. Pour tout $1 \leq k \leq 5$, on obtient U_k boule blanche lors du k -ième tirage. Comme X désigne le nombre de boules blanches obtenues lors des cinq tirages,

$$X = U_1 + U_2 + U_3 + U_4 + U_5.$$

En particulier, par linéarité de l'espérance,

$$\mathbf{E}(X) = \sum_{k=1}^5 \mathbf{E}(U_k) = 1,$$

que les tirages soient effectués avec remise [4] ou sans remise [5.4].

7. Tirage avec remise

Dans le cas d'un tirage avec remise, la variable X est la somme de cinq variables aléatoires de Bernoulli indépendantes et de même paramètre $p = 1/5$ [4] et suit donc la loi binomiale $\mathcal{B}(5, 1/5)$.

$$\forall 0 \leq k \leq 5, \quad \mathbf{P}(X = k) = \frac{1}{5^5} \binom{5}{k} 4^{5-k}$$

8. Tirage sans remise

Dans le cas d'un tirage sans remise, comme l'urne contient initialement deux boules blanches, le nombre X de boules tirées reste inférieur à 2.

8.1 Cas $X = 0$

La variable X prend la valeur 0 lorsqu'on tire cinq boules noires, donc

$$[X = 0] = [U_1 = 0] \cap [U_2 = 0] \cap [U_3 = 0] \cap [U_4 = 0] \cap [U_5 = 0].$$

D'après [5],

$$\mathbf{P}(X = 0) = \frac{8}{10} \cdot \frac{7}{9} \cdot \frac{6}{8} \cdot \frac{5}{7} \cdot \frac{4}{6} = \frac{2}{9}.$$

8.2 Cas $X = 1$

La variable X prend la valeur 1 lorsqu'on tire quatre boules noires et une seule boule blanche : il existe $1 \leq k \leq 5$ tel que U_k soit égale à 1 et pour tous les autres indices $1 \leq \ell \leq 5$, la variable U_ℓ est égale à 0, donc

$$[X = 1] = \bigcup_{1 \leq k \leq 5} \left([U_k = 1] \cap \bigcap_{\substack{1 \leq \ell \leq 5 \\ \ell \neq k}} [U_\ell = 0] \right).$$

Comme une variable aléatoire ne peut prendre deux valeurs distinctes simultanément, il s'agit d'une union d'événements deux à deux disjoints :

$$[X = 1] = \bigsqcup_{1 \leq k \leq 5} \left([U_k = 1] \cap \bigcap_{\substack{1 \leq \ell \leq 5 \\ \ell \neq k}} [U_\ell = 0] \right).$$

On peut déduire de [5] que $\mathbf{P}(X = 1) = 5/9$.

8.3 Cas $X = 2$

Comme l'événement $[X = 2]$ est le dernier cas possible, le plus simple est de remarquer que

$$\Omega = [X = 0] \sqcup [X = 1] \sqcup [X = 2]$$

et d'en déduire que

$$\mathbf{P}(X = 2) = 1 - \mathbf{P}(X = 0) - \mathbf{P}(X = 1) = \frac{2}{9}.$$

I.3 Loi de Y

9. D'après [1], le *gain par lancer* est décrit par une fonction g telle que

$$g(1) = 2 \quad \text{et} \quad g(0) = -3.$$

9.1 D'après la théorie de Lagrange, on peut choisir g affine.

On peut vérifier que le gain du joueur lors du k -ième lancer est égal à la variable aléatoire

$$g(U_k) = 5U_k - 3.$$

9.2 Le *gain total* est donné par

$$Y = \sum_{k=1}^5 (5U_k - 3) = 5X - 15$$

et la loi de Y se déduit alors de la loi de X . En particulier, $\mathbf{E}(Y) = 5\mathbf{E}(X) - 15 = -10$.

9.3 La relation de liaison qui vient d'être mise en évidence entre les variables X et Y est vraie que le tirage soit effectué avec remise ou sans remise.

I.4 Simulations

Loi des grands nombres

10. Utiliser un (bon) simulateur aléatoire pour simuler N réalisations $(x_i)_{0 \leq i < N}$ d'une variable aléatoire X revient en fait à simuler *une* réalisation d'un échantillon de N variables aléatoires $(X_i)_{0 \leq i < N}$ indépendantes de même loi que X .

11. D'après la Loi des grands nombres, si X est une variable d'espérance finie et si $(X_i)_{i \in \mathbb{N}}$ est une suite de variables aléatoires indépendantes de même loi que X , alors la variable aléatoire

$$M_n = \frac{1}{n} \sum_{0 \leq i < n} X_i$$

converge presque sûrement vers la moyenne théorique $\mathbf{E}(X)$ lorsque n tend vers $+\infty$.

11.1 Si l'entier N est assez grand, la moyenne des N réalisations simulées est donc assez proche de $\mathbf{E}(X)$:

$$\frac{1}{N} \sum_{0 \leq i < N} x_i \approx \mathbf{E}(X).$$

11.2 Si X est une variable aléatoire à valeurs dans \mathbb{N} , alors la Loi des grands nombres peut s'appliquer aux variables aléatoires de Bernoulli $\mathbb{1}_{[X_i=k]}$ pour tout $k \in \mathbb{N}$, car il s'agit de variables aléatoires indépendantes, de même loi que $\mathbb{1}_{[X=k]}$.

On en déduit que, si l'entier N est assez grand, la fréquence empirique est assez proche de la probabilité théorique :

$$\frac{\#\{0 \leq i < N : x_i = k\}}{N} = \frac{1}{N} \sum_{0 \leq i < N} \mathbb{1}_{[x_i=k]} \approx \mathbf{E}(\mathbb{1}_{[X=k]}) = \mathbf{P}(X = k).$$

12. De cette manière, la simulation d'un grand nombre de réalisations d'une variable aléatoire discrète X (au moyen d'un bon simulateur aléatoire) permet d'estimer la loi et l'espérance de cette variable.

12.1 Si la loi de X est connue, on peut ainsi tester la qualité du simulateur aléatoire.

12.2 Si la loi de X est difficile à calculer, on peut s'en faire une idée, soit en calculant des valeurs approchées des probabilités $\mathbf{P}(X = k)$, soit en résumant les valeurs simulées sous la forme d'un histogramme.

Tirages avec remise

13. Pour réaliser une étude statistique du jeu, on va simuler la réalisation d'un grand nombre N de parties, chaque partie étant décrite par une famille de cinq variables aléatoires indépendantes qui suivent la loi de Bernoulli de paramètre $p = 1/5$, qui n'est autre que la loi binomiale $\mathcal{B}(1, 1/5)$.

```
import numpy.random as rd

N = 100000
T_remise = rd.binomial(1, 0.2, size=(N, 5))
```

- Q 1. On considère un tableau $T \in \mathfrak{M}_{N,5}(\mathbb{N})$ qui simule N réalisations indépendantes du vecteur aléatoire $(U_1, U_2, U_3, U_4, U_5)$ dans le cas d'un tirage avec remise.
- Q 1.a En déduire un tableau A qui simule N réalisations indépendantes de la variable aléatoire X .
- Q 1.b Déduire de ce tableau A une valeur approchée de la loi de la variable X . Comparer avec la distribution théorique [7].
- Q 1.c Déduire de la loi empirique de X une valeur approchée de $\mathbf{E}(X)$.

Tirages sans remise

14. Cette fois, chaque partie étant décrite par une famille de cinq variables aléatoires dont la loi jointe est donnée au [5].

```
def partie():
    nb_blanches, nb_total = 2, 10
    P = []
    for k in range(5):
        prop_blanches = nb_blanches/nb_total # proportion de boules blanches
        if (prop_blanches>0): # il reste au moins une boule blanche
            lancer = rd.binomial(1, prop_blanches)
            nb_blanches -= lancer # on a peut-être retiré une boule blanche
        else: # il n'y a plus que des boules noires
            lancer = 0
        P.append(lancer) # on enregistre le résultat du k-ième tirage
        nb_total -= 1 # on a retiré une boule
    return P
```

15. On en déduit comment simuler la réalisation de N parties.

```
T_sans_remise = np.array([partie() for k in range(N)])
```

- Q 2. On considère un tableau $T \in \mathfrak{M}_{N,5}(\mathbb{N})$ qui simule N réalisations indépendantes du vecteur aléatoire $(U_1, U_2, U_3, U_4, U_5)$ dans le cas d'un tirage sans remise.
- Q 2.a En déduire un tableau A qui simule N réalisations indépendantes de la variable aléatoire X .
- Q 2.b Déduire de ce tableau A une valeur approchée de la loi de la variable X .
- Q 2.c Déduire de la loi empirique de X une valeur approchée de $\mathbf{E}(X)$.
- Q 3. Les variables U_1, \dots, U_5 suivent toutes des lois de Bernoulli. Déduire des simulations précédentes les paramètres de ces lois de Bernoulli (dans le cas d'un tirage avec remise et dans le cas d'un tirage sans remise). Comparer avec [5.4].

II

Loi de Pascal (banque CCP 96)

16. Soient $0 < p < 1$ et $r \in \mathbb{N}^*$.

16.1 Une enceinte fermée contenant une bactérie reçoit un tir de rayon laser chaque seconde. Les différents tirs sont indépendants et, à chaque tir, la bactérie est touchée par le rayon laser avec la probabilité p .

16.2 L'expérience commence à l'instant $t = 0$ et le premier rayon laser est envoyé à l'instant $t = 1$.

16.3 La bactérie ne meurt qu'après avoir été touchée r fois par le rayon laser.

On note X , la durée de vie de la bactérie depuis le début de l'expérience.

II.1 Modélisation

17. On modélise les tirs de laser par une suite $(U_k)_{k \geq 1}$ de variables aléatoires indépendantes définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathbf{P})$ qui suivent toutes la loi de Bernoulli $\mathcal{B}(p)$.

Le tableau ci-dessous indique comment interpréter les événements liés à ces variables.

k -ième tir	
$[U_k = 1]$	La bactérie est touchée
$[U_k = 0]$	La bactérie n'est pas touchée

18. Pour tout $n \in \mathbb{N}^*$, on pose

$$S_n = U_1 + \cdots + U_n.$$

La variable aléatoire S_n est égale au nombre de fois que la bactérie a été atteinte par le rayon laser au cours des n premiers tirs.

19. Pour tout $\omega \in \Omega$, la série $\sum U_k(\omega)$ est une série dont le terme général, positif, est égal à 0 ou à 1.

L'entier $r \in \mathbb{N}^*$ est fixé.

19.1 Par [17], les événements $[U_n = 1]$ sont indépendants et tous de même probabilité $p > 0$, donc la série numérique $\sum \mathbf{P}(U_n = 1)$ est (grossièrement) divergente et d'après le lemme de Borel-Cantelli, la probabilité de l'événement

$$\Omega' = \bigcap_{n \in \mathbb{N}} \bigcup_{m \geq n} [U_m = 1]$$

est égale à 1.

19.2 Si $\omega \in \Omega'$, alors il existe une infinité d'entiers m tels que $U_m(\omega) = 1$, donc la série $\sum U_k(\omega)$ diverge (grossièrement).

Comme il s'agit d'une série de terme général positif, ses sommes partielles tendent vers $+\infty$ donc il existe un entier n (qui varie avec ω) tel que $S_n(\omega) \geq r$. L'entier

$$X(\omega) = \min\{n \in \mathbb{N}^* : S_n(\omega) \geq r\}$$

est donc bien défini pour tout $\omega \in \Omega'$.

19.3 En notant $n_0 = X(\omega)$, on a donc $S_{n_0}(\omega) \geq r$ et $S_{n_0-1}(\omega) < r$. Comme il s'agit d'inégalités entre entiers, on en déduit que $S_{n_0-1}(\omega) \leq r - 1$ et donc que

$$1 = r - (r - 1) \leq S_{n_0}(\omega) - S_{n_0-1}(\omega) = U_{n_0}(\omega) \leq 1.$$

Finalement, $S_{n_0-1}(\omega) = r - 1$ et $S_{n_0}(\omega) = r$, donc

$$\forall \omega \in \Omega', \quad X(\omega) = \min\{n \in \mathbb{N}^* : S_n(\omega) = r\}.$$

19.4 En pratique, dans le cadre d'une simulation informatique, toutes les réalisations des $S_n(\omega)$ atteindront la valeur r à partir d'un certain rang (éventuellement assez élevé).

II.2 Loi de X

20. On peut démontrer que X suit la **loi de Pascal** de paramètres p et r : presque sûrement, X prend des valeurs entières supérieures à r et

$$\forall n \geq r, \quad \mathbf{P}(X = n) = \binom{n-1}{r-1} p^r q^{n-r}.$$

L'espérance de X est égale à r/p .

21. Instant du premier succès

Dans le cas particulier $r = 1$, la variable X suit la **loi géométrique** de paramètre p : presque sûrement, X prend des valeurs entières supérieures à 1 et

$$\forall n \geq 1, \quad \mathbf{P}(X = n) = pq^{n-1}.$$

Le paramètre p , qui est la probabilité de succès à chacune des tentatives, est aussi l'inverse de $\mathbf{E}(X)$, c'est-à-dire l'inverse du **temps d'attente moyen** avant le premier succès.

II.3 Simulations

Loi de X

22. La fonction `instant_succes(p, r)` simule une succession de tentatives indépendantes : la probabilité de succès pour chaque tentative est égale à p et la simulation s'arrête lorsqu'on connaît le succès pour la r -ième fois.

```
def instant_succes(p, r=1):
    n, nb_succes = 0, 0
    while (nb_succes < r):
        n += 1                # nouvelle tentative
        nb_succes += rd.binomial(1, p) # peut-être un nouveau succès
    return n
```

Les variables `n` et `nb_succes` comptent respectivement le nombre total de tentatives et le nombre de succès obtenus depuis le début de l'expérience.

L'argument `r=1` signifie que, par défaut, l'expérience s'arrête dès le premier succès.

Q 4.a Utiliser la fonction `instant_succes(p, r)` pour simuler un échantillon de N réalisations de la variable aléatoire X . On pourra commencer avec $r = 1$, $p = 0,5$ et $N = 10^5$ et on fera ensuite varier ces paramètres.

Q 4.b Comparer l'espérance de X calculée théoriquement [21] [20] à la moyenne empirique déduite de cet échantillon.

Loi des deux premiers succès

23. On s'intéresse au couple (X_1, X_2) des instants des deux premiers succès : pour tout $\omega \in \Omega'$,

$$X_1(\omega) = \min\{n \in \mathbb{N}^* : U_n(\omega) = 1\} \quad X_2(\omega) = \min\{n > X_1(\omega) : U_n(\omega) = 1\}.$$

(L'événement Ω' a été défini au [19.1].)

23.1 On sait que X_1 (instant du premier succès) suit la loi géométrique de paramètre p et que X_2 (instant du second succès) suit la loi de Pascal de paramètres $r = 2$ et p .

23.2 On peut démontrer que les variables aléatoires X_1 et $X_2 - X_1$ sont indépendantes et suivent la loi géométrique de paramètre p : cela permet d'en déduire la loi du couple (X_1, X_2) .

24. En théorie, une variable aléatoire X qui suit une loi géométrique peut prendre toutes les valeurs entières supérieures à 1.

24.1 En pratique,

$$\mathbf{P}(X \geq 10) = q^9$$

et pour $p = 0,5$, la probabilité pour qu'une variable de loi géométrique prenne une valeur supérieure à 10 est inférieure à 0,2%. Il est donc raisonnable de représenter la loi géométrique de paramètre $p = 0,5$ par la famille finie $L = (g_k)_{0 \leq k \leq 10}$ où

$$g_0 = \mathbf{P}(X = 0) = 0, \quad \forall 1 \leq k < 10, \quad g_k = \mathbf{P}(X = k) = pq^{k-1} \quad \text{et} \quad g_{10} = \mathbf{P}(X \geq 10) = q^9.$$

24.2 De manière analogue, si X et Y sont deux variables aléatoires indépendantes qui suivent toutes les deux la loi géométrique de paramètre p , la loi du couple est constituée par la famille dénombrable

$$(\mathbf{P}(X = i, Y = j))_{(i,j) \in \mathbb{N}^* \times \mathbb{N}^*}$$

En pratique, nous représenterons la loi du couple par un tableau fini $T \in \mathfrak{M}_{11}(\mathbb{R})$:

$$\begin{aligned} \forall 0 \leq j \leq 10, \quad T_{0,j} &= \mathbf{P}(X = 0, Y = j) = 0 \\ \forall 0 \leq i \leq 10, \quad T_{i,0} &= \mathbf{P}(X = i, Y = 0) = 0 \\ \forall 1 \leq i, j < 10, \quad T_{i,j} &= \mathbf{P}(X = i, Y = j) = p^2(1-p)^{i+j-2} \\ \forall 1 \leq j \leq 10, \quad T_{10,j} &= \mathbf{P}(X \geq 10, Y = j) = \mathbf{P}(X \geq 10) \mathbf{P}(Y = j) \\ \forall 1 \leq i \leq 10, \quad T_{i,10} &= \mathbf{P}(X = i, Y \geq 10) = \mathbf{P}(X = i) \mathbf{P}(Y \geq 10). \end{aligned}$$

Q 5.a Créer un tableau `loi_geom` qui contient les éléments de la famille L (définie au [24.1]).

Q 5.b En déduire un tableau `loi_geom_ind` qui contient les éléments du tableau T (défini au [24.2]).

Q 6. Proposer une fonction `instants_succes(p)` qui permette de simuler les valeurs prises par le couple (X_1, X_2) . On s'inspirera de la fonction `instant_succes(p, r)` du [22].

Q 7. On utilise la fonction `instants_succes(p)` pour simuler un échantillon de N valeurs prises par le couple (X_1, X_2) .

```
ech_X1X2 = [instants_succes(0.5) for k in range(N)]
```

On pourra prendre $N = 10^5$ pour commencer. (Il faut que cette valeur soit assez grande pour produire un résultat significatif, mais pas trop grande pour éviter que la réalisation des calculs ne dure trop longtemps.)

Q 7.a Proposer une fonction `accroissements(ech)` dont l'argument est un échantillon de valeurs du couple (X_1, X_2) et qui retourne un échantillon de valeurs du couple $(X_1, X_2 - X_1)$.

Q 7.b Déduire de cet échantillon une estimation empirique de la loi du couple $(X_1, X_2 - X_1)$ sous la forme d'un tableau de fréquences $F \in \mathfrak{M}_{11}(\mathbb{R})$, puis comparer ce tableau F au tableau T défini au [24.2].

Réponses aux questions

I Tirages dans une urne (banque CCP 95)

R 1.a Il s'agit de calculer ligne par ligne la somme des coefficients de la matrice T .

```
A = np.array([np.sum(L) for L in T_remise])
```

Variante : on calcule les sommes en prenant l'indice de colonne, c'est-à-dire l'indice de rang 1, pour indice de sommation.

```
A = np.sum(T_remise, axis=1)
```

R 1.b Le tableau $A = (x_i)_{0 \leq i < N}$ contient N réalisations de la variable aléatoire X . Les valeurs prises par X sont donc les entiers compris (au sens large) entre 0 et le plus grand élément de A .

On peut également déduire de A les fréquences avec lesquelles les différentes valeurs apparaissent : pour $0 \leq k \leq \max(A)$, on compte le nombre d'indices $0 \leq i < N$ tels que $x_i = k$ et on divise ce nombre par le nombre N de valeurs simulées (c'est-à-dire la taille du tableau A).

```
def loiEmpirique(A):
    n = np.max(A)+1
    # Effectif de chaque classe
    loi_emp = np.zeros(n)
    for valeur in A:
        loi_emp[valeur] += 1
    loi_emp = loi_emp/len(A)    # Fréquence empirique de chaque classe
    return loi_emp
```

```
loi_emp_remise = loiEmpirique(A)
```

La loi théorique de X est connue [7] et peut être facilement calculée.

```
coeff_bin_5 = [1, 5, 10, 10, 5, 1] # Pourquoi faire plus compliqué ?
loiTheorique = 5**(-5)*np.array([coeff_bin_5[k]*4**(5-k) for k in range(6)])
```

Pour $N = 10^5$, on doit trouver des résultats analogues à ceux qui figurent dans le tableau suivant.

k	0	1	2	3	4	5
$\mathbf{P}(X = k)$	0,3277	0,4096	0,2048	0,0512	0,0064	0,0003
f_k	0,3265	0,4123	0,2041	0,0502	0,0066	0,0003

On notera le bon accord entre les probabilités théoriques et les fréquences calculées sur la simulation.

R 1.c Par définition,

$$\mathbf{E}(X) = \sum_{k=0}^5 k \mathbf{P}(X = k).$$

On suppose connu un tableau $L = (f_k)_{0 \leq k < n}$ où f_k est la fréquence d'apparition de la valeur k lors de la simulation de X . On en déduit que

$$\mathbf{E}(X) \approx \sum_{k=0}^5 k f_k.$$

```
def espEmpirique(L):
    n = len(L)
    valeurs = np.arange(n)      # Valeurs prises : 0 ≤ X ≤ max(A)
    return np.sum(valeurs*L)
```

Variante avec `enumerate`.

```
def espEmpirique(L):
    s = 0.
    for k, f in enumerate(L):
        s += k*f
    return s
```

Avec `espEmpirique(loi_emp_remise)`, on trouve une valeur très proche de 1 (la valeur théorique de $\mathbf{E}(X)$).

R 2.a On procède comme au [R1.a].

```
A = np.sum(T_sans_remise, axis=1)
```

R 2.b La fonction écrite au [R1.b] convient parfaitement!

```
loi_emp_sans_remise = loiEmpirique(A)
```

Avec $N = 10^5$, on trouve quelque chose comme

$$f_0 = 0,2216 \quad f_1 = 0,5559 \quad f_2 = 0,2225$$

alors que $2/9 \approx 0,222\ 222$ et $5/9 \approx 0,555\ 556$. La simulation réalisée est donc conforme aux propriétés énoncées au [8].

R 2.c Ici encore, on peut utiliser la fonction du [R1.c] pour approcher l'espérance de X et avec `espEmpirique(loi_emp_sans_remise)`, on obtient encore une valeur extrêmement proche de 1.

R 3. La loi d'une variable aléatoire de Bernoulli U est caractérisée par le paramètre $p = \mathbf{E}(U)$. On va calculer une estimation de ce paramètre au moyen des tableaux `T_remise` et `T_sans_remise`. D'après la Loi des grands nombres [11.2], il suffit de calculer la proportion de 1 dans chaque *colonne* de ces tableaux. Comme la colonne d'indice k du tableau T est donnée par `T[:,k]`, le code suivant donne le résultat voulu.

```
marginales_remise = [np.mean(T_remise[:,k]) for k in range(5)]
```

Comme il s'agit de calculer les moyennes colonne par colonne, l'indice de sommation est l'indice de rang 0 (c'est-à-dire l'indice de ligne). On peut donc aussi utiliser le code suivant.

```
marginales_sans_remise = np.mean(T_sans_remise, axis=0)
```

Avec $N = 10^5$, on obtient des valeurs approchées proches de 0,2, conformément aux modèles considérés d'après [4] et [5.4].

	$P(U_1 = 1)$	$P(U_2 = 1)$	$P(U_3 = 1)$	$P(U_4 = 1)$	$P(U_5 = 1)$
Avec remise	0,1988	0,2012	0,2009	0,1973	0,2019
Sans remise	0,2023	0,2000	0,2006	0,1999	0,1981

II Loi de Pascal (banque CCP 96)

R 4.a Puisqu'il s'agit de simuler les réalisations d'une variable aléatoire de loi géométrique, on note `ech_G` le tableau des valeurs obtenues.

```
N = 10000
p = 0.5
ech_G = [instant_succes(p) for k in range(N)]
```

On obtient un échantillon de réalisations de la loi de Pascal de paramètres 2 et p avec le code suivant.

```
ech_P = [instant_succes(p, r=2) for k in range(N)]
```

R 4.b

```
moy_emp_geom = np.mean(ech_G) # ≈ 1/p
erreur_relative_geom = np.abs(p*moy_emp_geom-1)*100

moy_emp_pascal = np.mean(ech_P) # ≈ r/p
erreur_relative_pascal = np.abs(0.5*p*moy_emp_pascal-1)*100
```

Dans les deux cas, on constate que l'erreur relative sur l'espérance reste inférieure à 2% pour $N = 10^5$ (et la plupart du temps, cette erreur relative est de l'ordre de 0,5%).

R 5.a On crée un tableau de 0 et on affecte ensuite les valeurs convenables aux différents éléments de ce tableau.

```
n, p = 10, 0.5
loi_geom = np.zeros(n+1)
for k in range(1, n):
    loi_geom[k] = p*(1-p)**(k-1)
loi_geom[n] = (1-p)**(n-1)
```

Il est inutile de chercher à optimiser ce code : il y a tellement peu d'opérations à effectuer qu'on perdrait son temps à vouloir diminuer le nombre d'opérations !

R 5.b Comme les variables X et Y sont indépendantes,

$$T = \begin{pmatrix} g_0 \\ \vdots \\ g_{10} \end{pmatrix} (g_0 \cdots g_{10}).$$

On peut donc former le tableau `loi_geom_ind` par un simple produit matriciel, en jouant sur le format de `loi_geom`.

```

ligne = loi_geom.reshape((1,11))
colonne = loi_geom.reshape((11,1))
loi_geom_ind = colonne.dot(ligne)

```

Si on inverse l'ordre de la ligne et de la colonne, on obtient un nombre (sans intérêt probabiliste) au lieu d'une matrice!

R 6. La variable `n` compte le nombre total de tentatives. À chaque succès, on incrémente la variable `nb_succes` (qui porte bien son nom) et on enregistre le numéro de la tentative dans la liste `succes`.

```

def instants_succes(p, nb_max_succes=2):
    succes = []
    n, nb_succes = 0, 0
    while nb_succes < nb_max_succes:
        n += 1
        if (rd.binomial(1, p) == 1):
            nb_succes += 1
            succes.append(n)
    return succes

```

R 7.a On fait au plus simple!

```

def accroissements(ech):
    ech2 = [[x1, x2-x1] for x1, x2 in ech]
    return ech2

```

R 7.b On commence par calculer un échantillon de N réalisations du couple $(X_1, X_2 - X_1)$.

```

ech_GG = accroissements(ech_X1X2)

```

On parcourt ensuite cet échantillon de valeurs pour calculer les fréquences empiriques des différents couples (x_i, y_i) en suivant le principe défini au [24.2] pour le tableau T .

```

def loi_couple(ech, val_max = 10):
    T = np.zeros((val_max+1, val_max+1))
    # Calcul des effectifs
    for x, y in ech:
        if x > val_max:
            i = val_max # dernière ligne
        else:
            i = x
        if y > val_max:
            j = val_max # dernière colonne
        else:
            j = y
        T[i, j] += 1
    # On retourne les fréquences empiriques en divisant tous les effectifs
    # par la taille de l'échantillon.
    return T/len(ech)

```

```

F = loi_couple(ech_GG)

```

Pour comparer le tableau F des fréquences empiriques au tableau T des probabilités théoriques, on peut calculer

$$\|T - F\|_1 = \sum_{i=0}^{10} \sum_{j=0}^{10} |T_{i,j} - F_{i,j}|.$$

(Ce n'est qu'une manière parmi d'autres : en dimension finie, toutes les normes sont équivalentes.)

```
ecart = np.max(np.abs(loi_geom_ind - F))
```

Le tableau ci-dessous présente quelques valeurs de la variable `ecart` pour $p = 0,5$ et différentes valeurs de N .

Valeurs de <code>ecart</code>		
$N = 10^4$	$N = 10^5$	$N = 10^6$
0,0081	0,00309	0,000316
0,0043	0,00173	0,000579
0,0052	0,00134	0,000469
0,0038	0,0025	
0,0034	0,00103	
0,00545	0,00212	
0,0086	0,00102	

On constate ce que suggère la Loi des grands nombres : plus l'échantillon est grand, plus les fréquences empiriques sont proches des probabilités théoriques.

D'une certaine manière, on a mis en évidence expérimentalement que les variables X_1 et $X_2 - X_1$ sont bien indépendantes et de même loi. (En fait, on a surtout mis en évidence que le générateur aléatoire utilisé est un *bon* générateur aléatoire, puisque les données simulées sont conformes à la théorie.)