



---

## Programme de colle n°V

Période du 05/06 au 09/06

---

### Semaine du 05/06 au 09/06

#### Programme de colle

- Variables aléatoires à gogo. On pourra aussi travailler avec des lois discrètes.

#### Questions de cours

- Définition de l'espérance, de la variance et formule de König-Huyguens pour une variable aléatoire finie.

- On considère la variable aléatoire  $X$  telle que  $X(\Omega) = \{-1, 0, 1\}$  et

$$P(X = -1) = P(X = 1) = 1/4, \quad P(X = 0) = 1/2.$$

Écrire une fonction Python permettant de simuler  $X$ .

- Une urne contient  $N$  boules dont 1 boule rouge et  $N - 1$  boules blanches. On pioche successivement et sans remise les boules de cette urne et on note  $X$  la variable aléatoire qui prend la valeur du numéro de la pioche où apparaît la boule rouge. Déterminer la loi de  $X$ .
- Lois usuelles finies. Formules. Espérance. Variance. Simulation en Python (de plusieurs façons possibles). Préciser dans quelles situations on peut les reconnaître.

#### Suggestion d'exercices

**Exercice 1.** Soit  $n \in \mathbb{N}^*$  fixé. Une urne contient une boule numérotée 1, deux boules numérotées 2, trois boules numérotées 3 et ainsi de suite jusqu'à  $n$  boules numérotées  $n$ . On pioche une boule au hasard dans cette urne et on note  $X$  le numéro de la boule piochée.

- (1) Combien y a-t-il de boules dans l'urne?
- (2) Déterminer la loi de  $X$ .
- (3) Calculer  $E(X)$ .

- (4) Montrer que  $V(X) = \frac{(n-1)(n+2)}{18}$ .

**Exercice 2.** On lance une pièce équilibrée jusqu'à obtention du premier *PILE*. Si ce premier arrive au  $k$ -ème lancer, on remplit une urne de  $k$  boules numérotées de 1 à  $k$  puis on pioche au hasard une boule dans cette urne. On note  $Z$  la variable aléatoire qui renvoie le numéro de la boule piochée. Écrire une fonction en Python qui renvoie une simulation de  $Z$ .

**Exercice 3.** Une urne contient 1 boule bleue, 2 boules blanches et 3 boules rouges. On pioche successivement et avec remise 3 boules dans cette urne. Si on obtient les 3 couleurs, on marque 2 points, si on

obtient une seule couleur 1 point et 0 point sinon. On note  $X$  la variable aléatoire qui prend la valeur du nombre de points marqués sur une partie.

- (1) Déterminer la loi de  $X$ .
- (2) Calculer  $E(X)$  puis  $V(X)$ .
- (3) Écrire une fonction Python qui simule  $X$ .

**Exercice 4.** Un immeuble de  $p$  étages est équipé d'un ascenseur.  $N$  personnes montent dans l'ascenseur au rez de chaussée et descendent chacune à un étage au hasard et de façon indépendante.

Soit  $X$  le nombre d'arrêts de l'ascenseur.

On note  $X_i$  la variable valant 1 si l'ascenseur s'arrête à l'étage  $i$  et 0 sinon et on note  $E_k$  la variable aléatoire qui prend la valeur de l'étage où descend la  $k$ -ième personne.

- (1) Pour tout  $k \in \llbracket 1; N \rrbracket$ , déterminer la loi de  $E_k$ .
- (2) En utilisant les variables aléatoires  $E_k$ , déterminer  $P(X_i = 0)$ . En déduire la loi de  $X_i$ .
- (3) Calculer  $E(X)$ .
- (4) (\*) Déterminer pour  $i$  différent de  $j$  :  $P(X_i = 0 \cap X_j = 0)$ .  
En déduire  $P(X_i X_j = 0)$ . Quelle est la loi de la variable aléatoire  $X_i X_j$ ?

**Exercice 5.** On effectue une succession infinie de lancers indépendants d'une pièce équilibrée, donnant pile avec la probabilité  $p = 1/2$  et face avec la probabilité  $q = 1 - p = 1/2$ .

On s'intéresse aux successions de lancers amenant un même côté.

On dit que la première série est de longueur  $k$  (avec  $k \in \mathbb{N}^*$ ) si les  $k$  premiers lancers ont amené le même côté de la pièce et le  $(k + 1)$ -ième l'autre côté.

De même, la deuxième série commence au lancer suivant la fin de la première série et se termine (si elle se termine) au lancer précédant un changement de côté. On définit de même les séries suivantes.

Pour tout  $i$  de  $\mathbb{N}^*$ , on note  $P_i$  (resp.  $F_i$ ) l'événement : " le  $i$ -ième lancer amène pile (resp. face) ".

Pour tout  $n$  de  $\mathbb{N}^*$ , on note  $N_n$  la variable aléatoire égale au nombre de séries obtenus lors des  $n$  premiers lancers.

- (1) Soit  $n \in \mathbb{N}^*$ .
  - (a) Justifier que  $N_n(\Omega) = \llbracket 1; n \rrbracket$ .
  - (b) Calculer les probabilités  $P(N_n = 1)$  et  $P(N_n = n)$ .
- (2) (a) Déterminer les lois des variables aléatoires  $N_1$  et  $N_2$  et calculer leurs espérances.  
(b) Déterminer la loi de  $N_3$  puis vérifier que  $E(N_3) = 2$ .
- (3) Simulation informatique sous Python.
  - (a) Écrire une fonction `def lancer(p)` : qui simule un lancer d'une pièce en renvoyant 1 si on obtient pile (avec probabilité  $p \in ]0; 1[$ ) et 0 si on obtient face (avec probabilité  $q = 1 - p$ ).
  - (b) Compléter la fonction suivante qui, étant donné un entier  $n$  de  $\mathbb{N}^*$ , simule  $n$  lancers de la pièce et renvoie la valeur de  $N_n$  obtenue. *On rappelle que dans cet exercice  $p$  vaut  $1/2$ .*

```
def simul_N(n) :
    # Simulation des n lancers stockés dans la liste L
    L = [ ]
    for k in range(.....)
        L.append(.....)
    end
    # Calcul de la valeur de Nn stockée dans la variable N
    N = .....
    for i in range(....., ..... )
        if ..... :
```

```

N = .....
return N
    
```

(4) (\*) On pose, pour tout  $n$  de  $\mathbb{N}^*$  et pour tout  $s$  de  $[0; 1]$

$$G_n(s) = \sum_{k=1}^n P(N_n = k)s^k.$$

La fonction  $G_n$  s'appelle **fonction génératrice** de la variable  $N_n$ .

- (a) Calculer, pour tout  $n$  de  $\mathbb{N}^*$ ,  $G_n(0)$  et  $G_n(1)$ .
- (b) Montrer que  $G'_n(1) = E(N_n)$ .
- (c) Soit  $n \in \mathbb{N}^*$ . En utilisant un système complet d'évènements associé à la variable  $N_n$ , montrer que :

$$\forall k \in \llbracket 1; n + 1 \rrbracket, \quad P(N_{n+1} = k) = \frac{1}{2}P(N_n = k) + \frac{1}{2}P(N_n = k - 1).$$

(d) En déduire :  $\forall n \in \mathbb{N}^*, \forall s \in [0; 1],$

$$G_{n+1}(s) = \left(\frac{1+s}{2}\right) G_n(s).$$

(e) En déduire, pour tout  $n$  de  $\mathbb{N}^*$  et pour tout  $s$  de  $[0; 1],$

$$G_n(s) = s \cdot \left(\frac{1+s}{2}\right)^{n-1}.$$

(f) En déduire, à l'aide de la question 4b, que le nombre moyen de séries lors des  $n$  premiers lancers est  $(n + 1)/2$ .

**Exercice 6.** Soit  $X$  une variable aléatoire finie telle que  $X(\Omega) = \llbracket 1, n \rrbracket$ . Montrer que

$$E(X) = \sum_{k=0}^{n-1} P(X > k).$$

(On commencera par observer que  $P(X = k) = P(X > k - 1) - P(X > k)$ .)

**Exercice 7.** Soit  $N$  un entier supérieur ou égal à 2. Un joueur lance une pièce équilibrée indéfiniment. On note  $X_N$  la variable aléatoire réelle égale au nombre de fois où, au cours des  $N$  premiers lancers, deux résultats successifs ont été différents. (On peut appeler  $X_N$  le "nombre de changements" au cours des  $N$  premiers lancers).

Par exemple, si les 9 premiers lancers ont donné successivement *Pile*, *Pile*, *Face*, *Pile*, *Face*, *Face*, *Face*, *Pile*, *Pile*, alors la variable  $X_9$  aura pris la valeur 4 (quatre changements, aux 3<sup>ième</sup>, 4<sup>ième</sup>, 5<sup>ième</sup> et 8<sup>ième</sup> lancers).

- (1) Justifier que  $X_N(\Omega) = \llbracket 0, \dots, N - 1 \rrbracket$ .
- (2) Déterminer la loi de  $X_2$  ainsi que son espérance. Déterminer la loi de  $X_3$ .
- (3) Montrer que

$$P(X_N = 0) = \left(\frac{1}{2}\right)^{N-1}$$

et que

$$P(X_N = 1) = 2(N - 1) \left(\frac{1}{2}\right)^N.$$

(4) (a) Justifier que pour tout entier  $k$  de  $\llbracket 0, \dots, N - 1 \rrbracket$

$$P_{X_N=k}(X_{N+1} = k) = \frac{1}{2}.$$

(b) En déduire que pour tout entier  $k$  de  $\llbracket 0, \dots, N-1 \rrbracket$

$$P(X_{N+1} - X_N = 0 \cap X_N = k) = \frac{1}{2}P(X_N = k).$$

(c) En sommant cette relation de  $k = 0$  à  $N-1$ , montrer que

$$P(X_{N+1} - X_N = 0) = \frac{1}{2}.$$

(d) Montrer que la variable  $X_{N+1} - X_N$  suit une loi de Bernoulli de paramètre  $\frac{1}{2}$ .

(e) En déduire la relation  $E(X_{N+1}) = \frac{1}{2} + E(X_N)$ , puis donner  $E(X_N)$  en fonction de  $N$ .

**Exercice 8.** Soient  $a, b$  deux entiers strictement positifs. Une urne contient initialement  $a$  boules rouges et  $b$  boules blanches. On effectue une succession d'épreuves, chaque épreuve étant constituée des trois étapes suivantes :

- on pioche une boule au hasard dans l'urne,
- on remplace la boule tirée dans l'urne,
- on rajoute dans l'urne une boule de la même couleur que celle qui vient d'être piochée

Après  $n$  épreuves, l'urne contient donc  $a + b + n$  boules.

Pour tout  $n \in \mathbb{N}^*$ , on note  $X_n$  le nombre de boules rouges qui ont été **ajoutées** dans l'urne (par rapport à la composition initiale) à l'issue des  $n$  premières épreuves.

Pour tout  $n \in \mathbb{N}^*$ , on notera  $R_n$  l'événement "on pioche une boule rouge au  $n$ -ième tirage".

(1) Donner l'ensemble  $X_n(\Omega)$  des valeurs prises par la variable aléatoire  $X_n$  en fonction de  $n$ .

(2) On souhaite simuler l'expérience grâce à Python.

(a) Recopier et compléter la fonction suivante, qui simule le tirage d'une boule dans une urne contenant  $x$  boules rouges et  $y$  boules blanches et qui retourne la valeur 0 si la boule est rouge et 1 si elle est blanche.

```
import numpy.random as rd
```

```
def tirage(x,y):
```

```
    if .....
```

```
        return 1
```

```
    .....
```

(b) Recopier et compléter la fonction suivante, qui simule  $n$  tirages successifs dans une urne contenant initialement  $a$  boules rouges et  $b$  boules blanches (selon le protocole décrit ci-dessus) et qui retourne la valeur de  $X_n$  :

```
def simul_X(n, a, b):
```

```
    x=a
```

```
    y=b
```

```
    for k in range( .....
```

```
        r=tirage(x,y)
```

```
        if r==0 :
```

```
            .....
```

```
        else :
```

```
            .....
```

```
    return .....
```

(3) On complète avec les commandes ci-dessous dont l'exécution permet l'affichage ci-après.

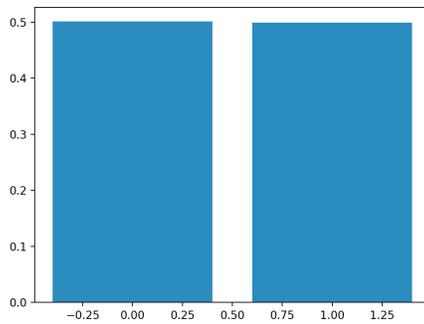
```
import matplotlib.pyplot as plt
```

```

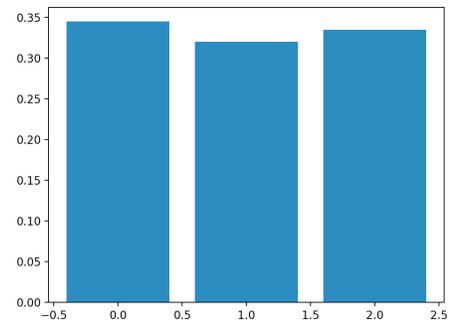
a=1
b=1
for n in range(1, 5):
    sample=[simul_X(n,a,b) for k in range(1000)]
    freq=[0]*(n+1)
    for k in range(1000):
        freq[sample[k]]+=1
    freq=[freq[k]/1000 for k in range(len(freq))]
    plt.bar([k for k in range(n+1)], freq)
    plt.show()
    plt.close()

```

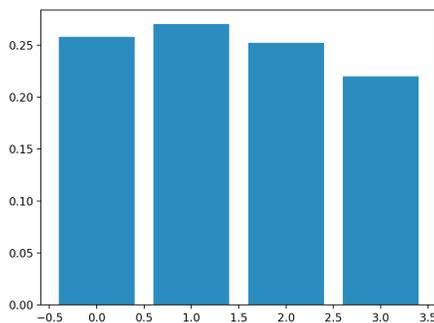
Affichage Python



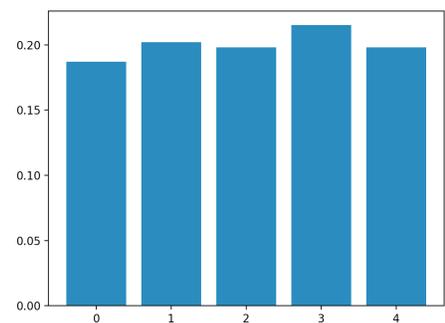
Affichage Python



Affichage Python



Affichage Python



- (a) À l'aide de ces résultats, conjecturer la loi de  $X_n$ .  
 (b) Déterminer la loi de  $X_1$ .  
 (c) Soient  $k$  et  $n$  deux entiers tels que  $0 \leq k \leq n$ . Déterminer les probabilités conditionnelles suivantes :

$$P_{[X_n=k]}(X_{n+1} = k), \quad P_{[X_n=k]}(X_{n+1} = k + 1), \quad P_{[X_n=k]}(X_{n+1} = \ell) \quad \text{avec } \ell \notin \{k, k + 1\}.$$

- (d) En raisonnant par récurrence sur  $n$ , prouver la conjecture émise au (3a).  
 (On pourra utiliser la formule des probabilités totales avec le système complet d'évènements  $\{(X_n = k)\}$  et la question précédente.)

**Exercice 9.** On considère la matrice  $M$  définie par :

$$M = \frac{1}{6} \begin{pmatrix} 6 & 3 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 2 \end{pmatrix}$$

et les trois vecteurs de  $\mathcal{M}_{3,1}(\mathbb{R})$  suivants

$$V_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, V_2 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, V_3 = \begin{pmatrix} 3 \\ -4 \\ 1 \end{pmatrix}$$

(1) La matrice  $M$  est-elle inversible ?

(2) Calculer  $MV_1$ ,  $MV_2$  et  $MV_3$  en fonction de  $V_1$ ,  $V_2$  et  $V_3$ .

Une urne contient une boule rouge et deux boules blanches. On effectue dans cette urne une succession de tirages d'une boule selon le protocole suivant :

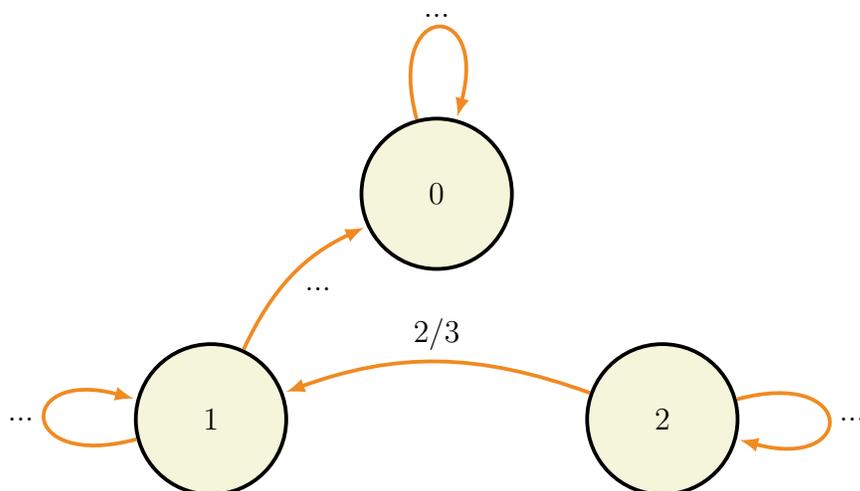
- si la boule tirée est rouge, elle est remise dans l'urne.
- si la boule tirée est blanche, elle n'est pas remise dans l'urne.

Pour tout entier  $i$  supérieur ou égal à 1, on note  $B_i$  (respectivement  $R_i$ ) l'événement "on obtient une boule blanche (respectivement une boule rouge) lors du  $i^{\text{ème}}$  tirage".

Pour tout entier  $n$  supérieur ou égal à 1, on note  $X_n$  le nombre de boules blanches contenues dans l'urne à l'issue du  $n^{\text{ème}}$  tirage et on pose  $X_0 = 2$ .

On introduit la matrice colonne  $U_n = \begin{pmatrix} P[X_n = 0] \\ P[X_n = 1] \\ P[X_n = 2] \end{pmatrix}$ .

- (4) (a) Déterminer pour tout entier naturel  $n$ , l'ensemble des valeurs prises par la variable  $X_n$  ( on distinguera les trois cas :  $n = 0, n = 1$  et  $n \geq 2$ ).
- (b) Recopier et compléter **en justifiant** le *diagramme de transition* ci-contre de la suite  $(X_n)$  (pour  $n \geq 2$ ). On précisera notamment à quelles probabilités conditionnelles correspondent les valeurs sur les flèches du diagramme.



- (c) En utilisant la formule des probabilités totales avec un système complet d'événements construit avec la variable  $X_n$ , montrer que pour tout entier  $n$  supérieur ou égal à 2, on a l'égalité suivante :

$$P(X_{n+1} = 1) = \frac{1}{2}P(X_n = 1) + \frac{2}{3}P(X_n = 2).$$

Montrer de même qu'on a

$$U_{n+1} = MU_n.$$

Vérifier que l'égalité précédente reste valable pour  $n = 0$  et  $n = 1$ .

(d) En déduire par récurrence, pour tout entier naturel  $n$ , la relation suivante :

$$U_n = V_1 + 4 \left(\frac{1}{2}\right)^n V_2 + \left(\frac{1}{3}\right)^n V_3.$$

(e) Donner la loi de la variable  $X_n$ .

(5) Calculer  $E(X_n)$ , espérance de  $X_n$ , ainsi que sa limite lorsque  $n$  tend vers  $+\infty$ .

(6) Recopier et compléter la fonction suivante permettant de représenter graphiquement la *trajectoire* de la v.a.  $X_n$ .

```
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt

def traj_X(n):
    X=np.zeros(n+1)
    X[0]=.....
    for i in range (1, n+1):
        if X[i-1]==0 :
            X[i]=.....
        elif ..... :
            X[i]= ...
        else:
            .....
    N=[k for k in range(n+1)]
    plt.grid()
    plt.plot(N, X, 'ko')
    plt.show()
```