

# Préparation à l'oral de Centrale maths 2 — Probabilités

## 1. DÉNOMBREMENT ET PROBABILITÉS

Bon à savoir hors memento : la commande `scipy.special.binom(n, p)` renvoie un flottant dont la valeur numérique est  $\binom{n}{p}$ . La commande `scipy.special.factorial(n)` renvoie  $n!$ .

**Exercice 0.** Ce premier exercice n'est pas une annale, mais un exercice technique introductif.

1. Écrire une fonction de paramètre une liste `L` et simulant une loi uniforme sur cette liste (on choisit au hasard et de manière équiprobable un élément de la liste).

2. Écrire une fonction de paramètre  $p \in ]0, 1[$  simulant la loi géométrique  $\mathcal{G}(p)$  en utilisant la commande *ad hoc* du module `np.random`. Écrire une deuxième fonction faisant la même chose sans utiliser cette commande.

3. Écrire une fonction prenant en argument un vecteur de probabilité, *i.e.* un vecteur  $(p_0, p_1, \dots, p_{n-1}) \in \mathbb{R}_+^n$  tel que  $\sum_{k=0}^{n-1} p_k = 1$ , et simulant la loi de  $X$ , où  $X(\Omega) = \llbracket 0, n-1 \rrbracket$  et, pour  $k \in X(\Omega)$ ,  $\mathbf{P}(X = k) = p_k$ .

**Exercice 1.** Soit  $p \in [0, 1]$ . On s'intéresse à une particule pouvant occuper quatre positions sur la droite réelle, les points d'abscisses 0, 1, 2 et 3 aux instants  $t = n$ , où  $n \in \mathbb{N}$ . On suppose que :

- 0 et 3 sont des puits : si la particule s'y trouve, elle y reste *ad vitam æternam* ;
- si la particule est en 1, alors elle saute à la position 0 à l'instant suivant avec la probabilité  $p$  ou à la position 2, ce avec la probabilité  $1 - p$  ;
- si la particule est en 2, alors elle saute en 1 avec la probabilité  $p$  ou en 3, ce avec la probabilité  $q = 1 - p$ .

À  $t = 0$ , la particule est en 1.

1. *Expérimentation.*

a. Écrire un programme `saut` prenant en argument le paramètre  $p$  et une position et retournant la position de la particule à l'instant suivant.

b. Écrire un programme `position` prenant en argument le paramètre  $p$  et un entier naturel  $n$  non nul et renvoyant la position de la particule au bout de  $n$  sauts.

c. Écrire un programme `histogramme` d'arguments  $p$ ,  $n$  et  $N$  indiquant le nombre de fois où, en moyenne sur  $N$  simulations, la particule a occupé chaque position en  $n$  sauts. Le tester avec  $n = 10$ ,  $N = 10^4$  et  $p \in \{0.3, 0.5, 0.7\}$ .

2. *Analyse mathématique.* On note  $x_n$  la position de la particule à l'instant  $n$  et  $X_n$  le vecteur-colonne de probabilité correspondant (ses coordonnées sont les  $\mathbf{P}(x_n = j)$  pour  $j \in \llbracket 0, 3 \rrbracket$  ; ainsi  $x_n$  est la variable aléatoire et  $X_n$  le vecteur colonne exprimant sa loi).

a. Montrer qu'il existe une matrice  $A$  indépendante de  $n$  que l'on explicitera telle que, pour tout entier  $n$ ,  $X_{n+1} = AX_n$ .

b. Montrer que  $A$  est diagonalisable si, et seulement si,  $0 < p < 1$ . En déduire que la suite  $(X_n)_n$  converge quel que soit  $p \in [0, 1]$ .

c. Écrire une fonction `limite` de paramètre  $p \in ]0, 1[$  renvoyant  $\lim X_n$  (on pourra utiliser que la recherche des éléments propres par python met toujours ici la valeur propre 1 en premier). L'appliquer pour  $p \in \{0.3, 0.5, 0.7\}$ . Utiliser l'une des fonctions de la question 1 pour illustrer empiriquement le résultat.

d. Comment utiliser la mise sous forme matricielle pour confirmer théoriquement les calculs de la question 1.c ?

**Exercice 2.** Soit  $n \in \mathbb{N}^*$ . On considère  $A_n = \llbracket 1, n \rrbracket$  et  $\mathcal{P}_n = \mathcal{P}(A_n)$ , l'ensemble des parties de  $A_n$ . On appelle *n-combinaison* tout  $j$ -uplet  $(p_1, \dots, p_j)$  avec  $1 \leq j \leq n$  formé de parties non vides de  $A_n$  et formant une partition de  $A_n$ . On note  $a_n$  le nombre de  $n$ -combinaisons. On a ainsi  $a_1 = 1$  et  $a_2 = 3$ , l'unique 1-combinaison de  $A_1$  étant  $\{1\}$  et les 2-combinaisons de  $A_2$  étant  $(\{1\}, \{2\})$ ,  $(\{2\}, \{1\})$  et  $(\{1, 2\})$ . On pose  $a_0 = 1$ .

1. Quel est le nombre de  $n$ -combinaisons dont tous les éléments sont de cardinal 1 ? Déterminer  $a_3$ .

2. Montrer que  $a_n = \sum_{k=1}^n \binom{n}{k} a_{n-k}$  pour tout  $n \geq 1$ .

3. Écrire un programme d'argument  $n$  renvoyant la liste des  $a_k$  pour  $k \in \llbracket 0, n \rrbracket$ . Déterminer  $a_n$  pour  $n \leq 10$ .

4. On pose  $b_n = \frac{a_n}{n!}$  et  $c_n = (\ln 2)^{-n}$ .

a. Tracer avec Python les points  $(n, c_n/2)$ ,  $(n, b_n)$  et  $(n, c_n)$ . Que peut-on conjecturer sur  $b_n$  ?

b. Montrer que

$$\forall n \in \mathbb{N} : 2 \leq \sum_{k=0}^{n-1} \frac{(\ln 2)^k}{k!} + 2 \frac{(\ln 2)^n}{n!}.$$

c. Démontrer la conjecture faite à la question 4.a.

**Exercice 3.** On considère une urne comportant initialement une boule rouge et une boule bleue. On tire des boules de l'urne l'une après l'autre. Chaque boule tirée est remise dans l'urne avec une boule supplémentaire de la même couleur. On note  $X_n$  le nombre de boules rouges et  $Y_n$  le nombre de boules bleues présentes dans l'urne après le  $n^{\text{ème}}$  tirage. Pour  $n \in \mathbb{N}$ , on définit

$$P_n(u, v) = \sum_{(i,j) \in \mathbb{N}^* \times \mathbb{N}^*} \mathbf{P}(X_n = i, Y_n = j) u^i v^j.$$

1. Écrire une fonction `Simulation(n)` simulant la variable aléatoire  $X_n$ .
2. Écrire une fonction `Moyenne(n, N)` qui simule  $N$  fois une expérience de  $n$  tirages et renvoie une liste dont le terme d'indice  $i$  est le nombre d'expériences se terminant avec  $i$  boules rouges. Que remarque-t-on ?
3. Soient  $n \in \mathbb{N}$ ,  $i$  et  $j$  dans  $\mathbb{N}^*$ . Exprimer  $\mathbf{P}(X_{n+1} = i, Y_{n+1} = j)$  en fonction de  $\mathbf{P}(X_n = i - 1, Y_n = j)$  et de  $\mathbf{P}(X_n = i, Y_n = j - 1)$ . Montrer que

$$\forall n \in \mathbb{N}: P_{n+1}(u, v) = \frac{1}{n+2} \left[ u^2 \frac{\partial P_n}{\partial u}(u, v) + v^2 \frac{\partial P_n}{\partial v}(u, v) \right].$$

4. Montrer que pour tout  $n \in \mathbb{N}$ ,  $P_n(u, v) = \frac{1}{n+1} \sum_{k=0}^n u^{k+1} v^{n+1-k}$ . Est-ce en adéquation avec la question 2 ?
5. Modifier la fonction `Simulation` afin qu'elle renvoie 1 si la dernière boule tirée est rouge et 0 sinon. Modifier la fonction `Moyenne` afin qu'elle renvoie le nombre moyen de tirages se terminant par une boule rouge. Commenter.

**Exercice 4.** Soit  $(U_n)_{n \geq 1}$  une suite de variables aléatoires mutuellement indépendantes suivant la loi uniforme sur  $[1, N]$  avec  $N \in [2, +\infty[$ . Pour  $n \in \mathbb{N}^*$ , on pose  $S_n = U_1 + \dots + U_n$ .

1. Calculer  $m = \mathbf{E}(U_1)$  et la série génératrice  $G_U$  de  $U_1$ .
2. En déduire une expression de  $G_{S_n}$ .
3. Que calcule le programme ci-dessous ?

```
P = [0] + [1]*4 ; P = Polynomial(P)**6 ; print(sum(list(P)[0:10]))
```

4. On pose  $u_n = \mathbf{P}(S_n \leq nm)$ . Écrire un programme Python qui calcule et représente graphiquement les  $u_n$  pour  $1 \leq n \leq 100$  (on prendra  $N = 11$  pour l'application numérique). Que peut-on conjecturer ?
5. On pose  $v_n = \mathbf{P}(S_n > nm)$  et  $w_n = \mathbf{P}(S_n = nm)$ . En considérant la suite de variables aléatoires  $(N+1-U_n)_{n \geq 1}$ , montrer que pour  $n \geq 1$  on a  $u_n = v_n + w_n$ .
6. Écrire un programme Python qui représente  $(w_n)_n$ . Conjecture ?
7. En admettant la conjecture de la question précédente, démontrer celle de la question 4.
8. On pose  $S_{n,k} = \#\{(i_1, \dots, i_n) \in (\mathbb{N}^*)^n; i_1 + \dots + i_n = k\}$ . Démontrer par une approche combinatoire que  $S_{n,k} = \binom{k-1}{n-1}$ .
9. Dans le cas  $N = 2$ , démontrer la conjecture de la question 6.

**Exercice 5.** Soit  $(X_n)_{n \geq 1}$  une suite de v.a.i.i.d. de loi géométrique de paramètre  $p \in ]0, 1[$  définie sur un espace probabilisé  $(\Omega, \mathcal{A}, \mathbf{P})$ . On pose  $q = 1 - p$ . Pour  $\omega \in \Omega$ , on définit

$$C(\omega) = \max \{n \in \mathbb{N}^*; X_1(\omega) \leq X_2(\omega) \leq \dots \leq X_n(\omega)\}.$$

1. Pour  $k \in \mathbb{N}^*$ , calculer  $\mathbf{P}(X_1 \geq k)$ ,  $\mathbf{P}(X_2 \geq X_1, X_1 = k)$ , puis  $\mathbf{P}(C \geq 2)$ .
2. Écrire une fonction `GeomCr(q)` renvoyant une réalisation de la variable aléatoire  $C$ .
3. Écrire une fonction `NbGeomCr(q, N)` renvoyant la moyenne de  $C$  pour  $N$  réalisations. On admet que  $C$  a une variance. Pourquoi peut-on penser que `NbGeomCr(q, 5000)` est une valeur approchée acceptable de  $\mathbf{E}(C)$  ?
4. Tracer `GeomCr(q)` en fonction de  $q \in ]0, 1[$ . Conjecturer les limites de  $\mathbf{E}(C)$  quand  $q$  tend vers 0 (resp. 1).
5. Pour  $k \in \mathbb{N}^*$ , montrer que

$$\mathbf{P}(k \leq X_1 \leq \dots \leq X_n) = \frac{(1-q)^n q^{n(k-1)}}{(1-q)(1-q^2) \dots (1-q^n)}.$$

6. Exprimer  $\mathbf{E}(C)$  en fonction de  $q$  sous la forme d'une série que l'on ne cherchera pas à calculer. Justifier que  $C$  admet une espérance et démontrer les propriétés conjecturées à la question 4.