

Révisions

Retour sur le programme avec des exercices de ciblés

On propose quelques exercices classés par thèmes selon les demandes des étudiant.e.s. Encore une fois, une sélection nullement exhaustive.

Suites implicites

Exercice 1. (Extrait Oral HEC 2013, SP)

Soit $n \in \mathbb{N}^*$. On définit la fonction réelle f_n par : $\forall x \in \mathbb{R}, f_n(x) = x + 1 - \frac{e^x}{n}$.

- (1) Montrer que pour tout $n \in \mathbb{N}^*$, il existe un unique nombre réel négatif x_n tel que $f_n(x_n) = 0$.
- (2) (a) Montrer que la suite $(x_n)_{n \in \mathbb{N}^*}$ est décroissante et convergente.
(b) Calculer la limite ℓ de la suite $(x_n)_{n \in \mathbb{N}^*}$.
- (3) On pose $y_n = x_n - \ell$. Déterminer un équivalent de y_n lorsque n tend vers $+\infty$.

Exercice 2. (Extrait de Oral HEC 2019, AP)

- (1) Question de cours : énoncer le théorème de la bijection.

Pour $n \in \mathbb{N}^*$, on définit la fonction $f_n : \mathbb{R}_+ \rightarrow \mathbb{R}$ par : $f_n : x \mapsto x^{2n+1} - x^{n+1} - 1$.

- (2) Soit $n \in \mathbb{N}^*$.
 - (a) Étudier les variations de f_n sur \mathbb{R}_+ .
 - (b) Montrer que f_n s'annule sur \mathbb{R}_+ en un et un seul réel, que l'on note x_n et montrer : $x_n > 1$.
- (3) (a) Montrer que la suite $(x_n)_{n \geq 1}$ décroît. *On pourra rechercher le signe de $f_{n+1}(x_n)$.*
(b) Que déduit-on des divers résultats précédents ?
- (4) (a) Montrer que la suite $(x_n)_{n \geq 1}$ converge vers 1.
(b) Montrer que la fonction $h : x \mapsto x(x-1)$ est une bijection croissante de $[1, +\infty[$ sur \mathbb{R}_+ .
(c) Pour $n \in \mathbb{N}^*$, on pose $u_n = x_n^n$. En exprimant, pour tout $n \in \mathbb{N}^*$, $h(u_n)$ en fonction de x_n , montrer que la suite $(u_n)_{n \geq 1}$ converge vers une limite ℓ à préciser.
(d) Déterminer un équivalent simple de $x_n - 1$ quand n tend vers $+\infty$.

Exercice 3. (D'après Oral HEC 2012, AP)

- (1) Question de cours : Définition d'une série convergente. Pour quels réels $x > 0$ la série de terme général $(\ln x)^n$ est-elle convergente? Calculer alors sa somme.
- (2) Pour tout entier n supérieur ou égal à 1, on note f_n la fonction définie sur l'intervalle $]0, +\infty[$, à valeurs réelles, par : $f_n(x) = (\ln x)^n - x$.
- (a) Calculer les dérivées première et seconde f'_n et f''_n de la fonction f_n .
 (b) Montrer que la fonction f_1 ne s'annule jamais.
 (c) Justifier l'existence d'un réel $a \in]0, 1[$ vérifiant l'égalité : $f_2(a) = 0$.
- (3) On suppose désormais que n est un entier supérieur ou égal à 3, et on s'intéresse aux solutions de l'équation $f_n(x) = 0$ sur l'intervalle $]1, +\infty[$. On donne : $\ln 2 \simeq 0,693$.
- (a) Dresser le tableau de variations de f_n sur $]1, +\infty[$ et montrer que l'équation $f_n(x) = 0$ admet deux racines, notées u_n et v_n , sur $]1, +\infty[$. (u_n désigne la plus petite des deux racines).
 (b) Calculer $\lim_{n \rightarrow +\infty} v_n$.
- (4) Montrer que la suite $(u_n)_{n \geq 3}$ est convergente et calculer sa limite.

Exercice 4. (D'après Oral HEC 2012, AP)

- (1) Question de cours: Convexité d'une fonction définie sur un intervalle \mathbb{R} .
- (2) (a) Justifier que $\forall x \in \mathbb{R}$, l'intégrale $\int_0^x e^{t^2} dt$ est convergente.
 On pose : $f(x) = \int_0^x e^{t^2} dt$.
 (b) Montrer que f est de classe \mathcal{C}^2 sur \mathbb{R} . Étudier la parité et la convexité de f .
 (c) Étudier les variations de f sur \mathbb{R} et tracer l'allure de la courbe représentative de f dans un repère orthogonal du plan.
- (3) (a) Établir pour tout $n \in \mathbb{N}^*$ l'existence d'un unique réel u_n vérifiant $f(u_n) = \frac{1}{n}$.
 (b) Montrer que la suite $(u_n)_{n \in \mathbb{N}^*}$ est décroissante et convergente.
 (c) Déterminer $\lim_{n \rightarrow +\infty} u_n$.
- (4) (a) Établir pour tout $u \in [0, \ln 2]$, l'encadrement : $1 + u \leq e^u \leq 1 + 2u$.
 (b) En interprétant le résultat de la question 3.c), en déduire qu'il existe un entier naturel n_0 tel que pour tout $n \geq n_0$, on a :

$$\int_0^{u_n} (1 + t^2) dt \leq \frac{1}{n} \leq \int_0^{u_n} (1 + 2t^2) dt.$$

 (c) Montrer que $\lim_{n \rightarrow +\infty} nu_n^3 = 0$ et en déduire un équivalent de u_n lorsque n tend vers $+\infty$.

Couples de v.a.d - Calculs de covariance

Exercice 1. On considère un couple de variables aléatoires (X, Y) dont la loi conjointe est donnée par le tableau ci-dessous

$X \backslash Y$	-1	0	1
0	1/4	a	1/8
1	1/5	b	1/10

- (1) Déterminer les lois marginales de X et Y en fonction de a et b .
 (2) Exprimer la covariance de X et Y en fonction de a et b .
 (3) Déterminer a et b de manière à ce que X et Y soient indépendantes.

Exercice 2. Une urne contient des jetons portant chacun une des trois lettres A , B ou C . La proportion des jetons A (resp. B , C) dans l'urne est égale à $p \in]0, 1[$ (resp. $q, r \in]0, 1[$) avec $p + q + r = 1$. On effectue n tirages (avec $n \in \mathbb{N}^*$) avec remise dans cette urne et on note X (resp. Y) la variable aléatoire correspondant au nombre de jetons A (resp. B) piochés.

- (1) Quelles sont les lois suivies par X et Y ?
- (2) Quelle est la loi conjointe de (X, Y) ?
- (3) Déterminer $\text{cov}(X, Y)$.

Exercice 3.

- (1) Rappeler le lien entre la covariance de deux variables aléatoires discrètes X et Y (pour lesquelles la covariance existe) et l'indépendance de X et Y .

Dans cet exercice on traite un cas particulier dans lequel l'implication précédente est une équivalence. On fera attention au fait que c'est vrai ici mais comme on l'a rappelé ci-avant, ce n'est pas vrai en général.

Soit a et b deux réels non nuls.

Soit X et Y deux variables aléatoires discrètes définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ telles que $X(\Omega) = \{0, a\}$ et $Y(\Omega) = \{0, b\}$.

- (2) Montrer que X , Y et XY admettent une espérance et déterminer ces espérances.

- (3) Montrer que X et Y sont indépendantes si et seulement si $\text{cov}(X, Y) = 0$.

Exercice 4. (Extrait de **Oral HEC 2012**, SP)

Soit p un réel de $]0, 1[$ et $q = 1 - p$. Soit $(X_n)_{n \in \mathbb{N}^*}$ une suite de variables aléatoires indépendantes définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$, de même loi de Bernoulli telle que :

$\forall k \in \mathbb{N}^*$, $\mathbb{P}([X_k = 1]) = p$ et $\mathbb{P}([X_k = 0]) = q$.

Pour n entier de \mathbb{N}^* , on définit pour tout $k \in \llbracket 1, n \rrbracket$ la variable aléatoire $Y_k = X_k + X_{k+1}$.

- (1) (a) Calculer pour tout $k \in \llbracket 1, n \rrbracket$, $\text{cov}(Y_k, Y_{k+1})$.
(b) Montrer que $0 < \text{cov}(Y_k, Y_{k+1}) \leq \frac{1}{4}$.
- (2) Calculer pour tout couple (k, l) tel que $1 \leq k < l \leq n$, $\text{cov}(Y_k, Y_l)$.
- (3) On note ε un réel strictement positif fixé.

Montrer que

$$\lim_{n \rightarrow +\infty} \mathbb{P} \left(\left[\left| \frac{1}{n} \sum_{k=1}^n Y_k - 2p \right| > \varepsilon \right] \right) = 0.$$

Exercice 5.

Toutes les variables aléatoires qui interviennent dans l'exercice sont supposées définies sur le même espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.

- (1) Question de cours : Définition et propriétés de la covariance de deux variables aléatoires discrètes.

Soit p, q et r des réels fixés de l'intervalle $]0, 1[$ tels que $p + q + r = 1$.

Soit $(X_n)_{n \geq 1}$ une suite de variables aléatoires à valeurs dans $\{-1, 0, 1\}$, indépendantes et de même loi donnée par :

$$\forall n \in \mathbb{N}^*, \mathbb{P}([X_n = 1]) = p, \mathbb{P}([X_n = -1]) = q, \mathbb{P}([X_n = 0]) = r.$$

On pose pour tout entier $n \geq 1$: $Y_n = \prod_{k=1}^n X_k$.

- (2) (a) Pour tout entier $n \geq 1$, préciser $Y_n(\Omega)$ et calculer $\mathbb{P}([Y_n = 0])$.
(b) Pour tout entier $n \geq 1$, calculer $E(X_n)$ et $E(Y_n)$.

(3) On pose pour tout entier $n \geq 1$, on a : $p_n = \mathbb{P}([Y_n = 1])$.

(a) Calculer p_1 et p_2 .

(b) Établir une relation de récurrence entre p_{n+1} et p_n .

(c) En déduire que pour tout entier $n \geq 1$, on a : $p_n = \frac{(p+q)^n + (p-q)^n}{2}$.

(d) Pouvait-on à l'aide de la question 2, trouver directement la loi de Y_n ?

(4) (a) Établir l'inégalité : $(p+q)^n > (p-q)^{2n}$. Calculer $V(Y_n)$.

(b) Calculer la covariance $\text{cov}(Y_n, Y_{n+1})$ des deux variables aléatoires Y_n et Y_{n+1} .

Exercice 6.

Soit (X_1, \dots, X_n) un n -uplet de v.a. finies.

(1) Rappeler la formule liant $V(X_1 + X_2)$ et $\text{Cov}(X_1, X_2)$.

(2) Montrer, par récurrence sur n , que

$$V(X_1 + X_2 + \dots + X_n) = \sum_{i=1}^n V(X_i) + 2 \sum_{i=1}^n \sum_{j=1}^{i-1} \text{Cov}(X_i, X_j).$$

(3) Qu'obtient-on sur les variables sont mutuellement indépendantes?

On met n jetons numérotés au hasard dans n boites numérotées (un seul jeton dans chaque boite).

Soit X_k la variable valant 1 si le k -ième jeton est dans la k -ième boite et 0 sinon.

Soit X le nombre de coïncidences entre les numéros des jetons et des boites.

(4) Déterminer la loi de X_k .

(5) Pour i différent de j , déterminer la loi de $X_i X_j$.

(6) Exprimer alors $\text{Cov}(X_i, X_j)$.

(7) Déterminer l'espérance et la variance de X .

Exercice 7.

Soit N un entier supérieur ou égal à 2.

On considère une *playlist* de N morceaux de musique, jouée en mode aléatoire avec les règles suivantes :

- Le premier morceau joué est le numéro 1.
- Si le morceau en train d'être joué est le numéro i , alors le numéro du prochain morceau est choisi aléatoirement et avec équiprobabilité dans $\llbracket 1, N \rrbracket \setminus \{i\}$ (autrement dit, on n'écoute pas deux fois de suite le même morceau).

On note, pour tout $n \in \mathbb{N}^*$, X_n la variable aléatoire égale au numéro du n -ième morceau écouté. En particulier, $X_1 = 1$.

(1) **Question de cours :**

(a) Définition et caractérisation d'un état stable d'une chaîne de Markov.

(b) Si la chaîne de Markov (X_n) converge en loi vers X et que π est un vecteur ligne représentant la loi de X , que peut-on dire de π ?

(2) (a) Expliquer pourquoi (X_n) est une chaîne de Markov homogène à N états.

(b) Représenter le graphe probabiliste pour $N = 2$ et pour $N = 3$ puis donner la matrice de transition, notée M , dans le cas général.

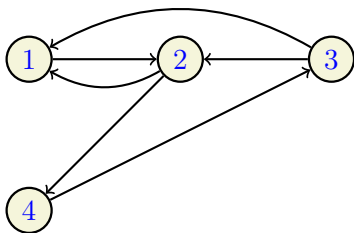
(c) Soit J la matrice de \mathcal{M}_N dont tous les coefficients sont égaux à 1. Exprimer M en fonction de J et de la matrice identité I_N .

- (d) Calculer, pour tout entier $n \geq 1$, M^n . On exprimera le résultat comme une combinaison linéaire des matrices I_N et J .
- (e) Dédire de la question précédente une fonction Python `calcul_puis_M` qui prend en entrée deux paramètres n et N et renvoie en sortie la matrice M^n .
- (3) (a) Déterminer, pour tout entier $n \geq 1$, la loi de X_n . Préciser pour $n = 2$.
La suite (X_n) converge-t-elle en loi ?
- (b) Montrer que $E_1(M) = E_N(J)$ puis déterminer les états stables de la chaîne de Markov.
- (c) Commenter ces résultats en lien avec une possible réciproque à la question **1.b**).
- (4) Soit $n \in \mathbb{N}^*$.
- (a) Les variables aléatoires X_n et X_{n+1} sont-elles indépendantes ?
- (b) Calculer $E(X_n)$ et $E(X_n^2)$ puis montrer que $E(X_n X_{n+1}) = \frac{N(N+1)}{2(N-1)} E(X_n) - \frac{1}{N-1} E(X_n^2)$.
- (c) Calculer, en cas d'existence, $\lim_{n \rightarrow +\infty} \text{cov}(X_n, X_{n+1})$ et $\lim_{N \rightarrow +\infty} \text{cov}(X_n, X_{n+1})$.

Graphes

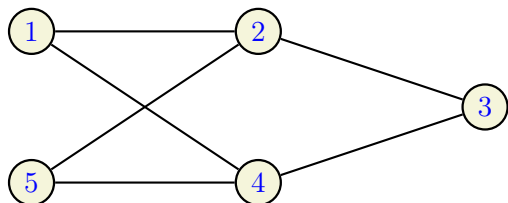
Exercice 1. Démontrer que dans un groupe de n personnes, il y en a toujours deux qui connaissent exactement le même nombre de membres du groupe

Exercice 2. On considère le graphe orienté \mathcal{G} ci-dessous.



- (1) Déterminer la matrice M d'adjacence du graphe.
- (2) Calculer M^2 et M^3 .
- (3) Montrer que \mathcal{G} est connexe.

Exercice 3. On considère le graphe \mathcal{G} ci-dessous.



- (1) Déterminer la matrice M d'adjacence du graphe.
- (2) Montrer que, pour tout $k \in \mathbb{N}$, $M^{2k+1} = 6^k M$.
- (3) En déduire le nombre de chemins de longueur 5 allant du sommet 2 au sommet 3.

Exercice 4.

On rappelle qu'une chaîne chaîne entre deux sommets d'un graphe est dite simple si elle ne passe pas deux fois par la même arête. Une chaîne (simple) est dite eulérienne si elle parcourt toutes les arêtes d'un graphe connexe une et une seule fois.

On a le résultat suivant

Théorème

Théorème d'Euler.

Soit \mathcal{G} un graphe connexe. Alors,

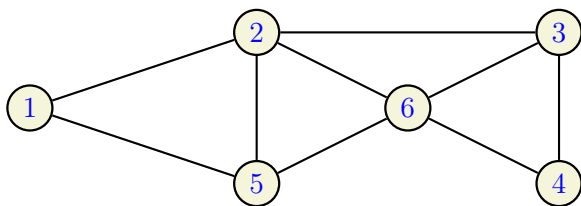
- (i) \mathcal{G} admet un cycle eulérien si et seulement si tous les sommets de \mathcal{G} sont de degré pair.
- (ii) \mathcal{G} admet une chaîne eulérienne (non fermée, c'est à dire qui n'est pas un cycle) si et seulement si le nombre de sommets de degré impair dans \mathcal{G} est 2.
Si tel est le cas, les extrémités de la chaîne eulérienne sont les deux sommets de degré impair.

- (1) (a) Écrire une fonction Python d'en-tête `def test_cycle_Euler(G)` qui prend en argument un graphe G (défini par sa liste d'adjacence) et renvoie 1 ou 0 selon qu'il existe un cycle eulérien dans le graphe ou non.
- (b) Écrire une fonction Python d'en-tête `def test_chaine_Euler(G)` qui prend en argument un graphe G (défini par sa liste d'adjacence) et renvoie 1 ou 0 selon qu'il existe une chaîne eulérienne dans le graphe ou non.

L'algorithme d'Euler consiste à déterminer dans n'importe quel graphe connexe une chaîne eulérienne. Il s'articule en quatre temps :

- Créer une chaîne simple entre deux sommets de degrés impairs.
- Tant que toutes les arêtes du graphe n'ont pas été utilisées, choisir un sommet quelconque de la chaîne précédente et trouver un cycle associé (partant de ce sommet et arrivant à ce sommet) ne contenant aucune des arêtes déjà utilisées.
- Insérer ce cycle en remplacement du sommet choisi à l'étape précédente.
- Recommencer ainsi de suite jusqu'à avoir utilisé toutes les arêtes.

On considère le graphe \mathcal{G} ci-dessous.



- (2) Le graphe est-il connexe ?
- (3) Montrer qu'il possède au moins une chaîne eulérienne. Donner les deux seules extrémités possibles de cette chaîne.
- (4) Mettre en place l'algorithme d'Euler en partant des chaînes suivantes :
 - (i) $5 - 6 - 4 - 3$
 - (ii) $5 - 2 - 3$

Exercice 5. Dans tout l'exercice n désigne un entier supérieur ou égal à 2 et p un réel de l'intervalle $]0; 1[$. Soit $S = \{s_1, \dots, s_n\}$ un ensemble fini de n sommets. On s'intéresse dans cet exercice à des graphes aléatoires construits à partir de l'ensemble S .

Plus précisément, pour tout couple $(i, j) \in \llbracket 1, n \rrbracket^2$ avec $i < j$, on introduit les variables aléatoires $T_{i,j}$ mutuellement indépendantes de même loi $\mathcal{B}(p)$. Les arêtes du graphe sont les paires de sommets $\{s_i, s_j\}$ pour lesquelles $T_{i,j} = 1$.

On dit qu'un sommet est isolé s'il n'y a aucune arête incidente à ce sommet.

On introduit N_n la variable aléatoire égale au nombre d'arêtes du graphe. Pour $k \in \llbracket 1, n \rrbracket$, on note D_k la variable aléatoire qui prend pour valeur le degré du sommet s_k (c'est à dire le nombre d'arêtes incidentes à ce sommet). Enfin, on note X_k la variable aléatoire qui vaut 1 si le sommet s_k est isolé et 0 sinon puis Z_n celle égale au nombre de sommets isolés du graphe.

- (1) Recopier et compléter la fonction Python ci-dessous qui renvoie la liste d'adjacence d'un tel graphe aléatoire en prenant pour argument la liste des sommets S et p .

On rappelle que la liste d'adjacence L d'un graphe dont l'ensemble des sommets est $S = \{s_1, \dots, s_k\}$ est la liste dont la composante L_i est la liste des sommets adjacents au sommet s_i .

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

def list_adj(S, p):
    n=.....
    l=[ [ ] for k in range(n)]
    for i in range(n-1):
        for j in range(i+1, n):
            if rd.random()<p :
                l[i].append(.....)
                l[j].append(.....)
    return l
```

(2) On exécute alors la commande suivante qui génère l'affichage suivant

```
list_adj('abcdef', 1/3)
```

Affichage Python

```
> > >
[['b', 'e', 'f'], ['a', 'c', 'f'], ['b'], [], ['a'], ['a', 'b']]
```

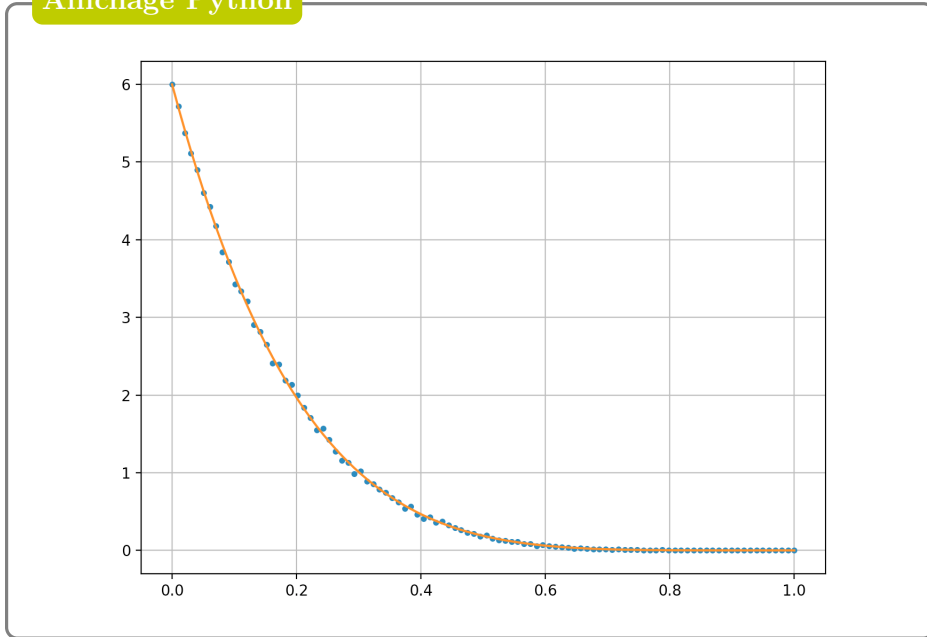
- Donner une représentation graphique du graphe aléatoire généré par cette exécution.
 - Ce graphe contient-il des sommets isolés? Si oui, combien ?
 - Expliciter la matrice d'adjacence de ce graphe. Quelle propriété a-t-elle ? Pourquoi ?
- (3) Écrire une fonction Python d'en-tête `def nb_som_is(l):` qui, prenant en argument la liste l d'adjacence d'un graphe renvoie le nombre de sommets isolés de celui-ci.
- (4) On dispose de la fonction mystère suivante

```
def mystere(S, p):
    return np.mean([nb_som_is(list_adj(S,p)) for k in range(1000)])
```

- Que renvoie-t-elle ?
- On ajoute les instructions suivantes dont l'exécution produit la figure ci-contre. Interpréter et émettre une conjecture.

```
S='abcdef'
x=np.linspace(0,1, 100)
y=[mystere(S, p) for p in x]
w=[len(S)*(1-p)**(len(S)-1) for p in x]
plt.grid()
plt.plot(x,y, '.')
plt.plot(x,w)
plt.show()
```

Affichage Python



- (5) (a) Justifier que $N_n(\Omega) \subset \llbracket 0, \binom{n}{2} \rrbracket$.
- (b) Montrer que $P(N_n = 0) = P\left(\bigcap_{i=1}^{n-1} \bigcap_{j=i+1}^n [T_{i,j} = 0]\right) = (1-p)^{n(n-1)/2}$.
- (c) Que vaut $P(N_n = \binom{n}{2})$?

- (6) (a) Justifier que, pour tout $k \in \llbracket 1, n \rrbracket$, on a

$$D_k = \sum_{i=1}^{k-1} T_{i,k} + \sum_{i=k+1}^n T_{k,i}.$$

En déduire la loi de D_k .

- (b) Montrer que, pour $1 \leq k < \ell \leq n$, on a

$$\text{cov}(D_k, D_\ell) = p(1-p).$$

Les variables D_k et D_ℓ sont-elles indépendantes ?

- (c) Déterminer $E(Z_n)$. Est-ce cohérent avec la conjecture précédente?

- (7) (a) Montrer que, pour tous $i < j$, on a

$$P([X_i = 1] \cap [X_j = 1]) = (1-p)^{2n-3}.$$

- (b) En observant que

$$Z_n^2 = \sum_{i=1}^n X_i + 2 \sum_{1 \leq i < j \leq n} X_i X_j,$$

montrer que

$$E(Z_n^2) = n(1-p)^{n-1} + n(n-1)(1-p)^{2n-3}.$$

Intervalle de confiance

Exercice 1. Soit X d'espérance m et de variance σ^2 . On considère alors un n -échantillon (X_1, \dots, X_n) de X et on pose $T_n = X_1 + X_2 + \dots + X_n$. On suppose que σ^2 est connue, mais pas m .

- (1) Déterminer $E(T_n)$ et $V(T_n)$.
- (2) Montrer que, pour tout $\varepsilon > 0$

$$P(|T_n - nm| > n\varepsilon) \leq \frac{\sigma^2}{n\varepsilon}.$$

(3) Quel est le risque de l'intervalle de confiance

$$\left[\frac{T_n}{n} - \varepsilon; \frac{T_n}{n} + \varepsilon \right]$$

pour m ?

Exercice 2. On cherche à estimer le paramètre $\theta > 0$ d'une loi uniforme $\mathcal{U}([0, \theta])$. On dispose pour cela d'un n -échantillon de cette loi et on note $Y_n = \max(X_1, \dots, X_n)$.

(1) Montrer que, pour tout réel $x > 1$, on a

$$P\left(Y_n \in \left[\frac{\theta}{x}; \theta\right]\right) = 1 - \left(\frac{1}{x}\right)^n.$$

(2) Soit $\alpha \in]0; 1[$. On note $g_n(\alpha) = n \left(\left(\frac{1}{\alpha}\right)^{1/n} - 1 \right)$.

Montrer que

$$I_{n,\alpha} = \left[Y_n, Y_n + \frac{Y_n}{n} g_n(\alpha) \right]$$

est un intervalle de confiance au niveau $1 - \alpha$ pour θ .

Exercice 3.

Soit X une variable aléatoire définie sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$, à valeurs dans $[0, \theta]$ où θ est un paramètre réel strictement positif inconnu.

Une densité f de X est donnée par $f(x) = \begin{cases} \frac{2x}{\theta^2}, & \text{si } x \in]0, \theta] \\ 0, & \text{sinon} \end{cases}$.

(1) Question de cours : Estimateur sans biais; risque quadratique d'un estimateur.

(2) Calculer l'espérance et la variance de X .

Pour tout entier $n \in \mathbb{N}^*$, soit (X_1, X_2, \dots, X_n) un n -échantillon de variables aléatoires indépendantes et de même loi que X . On pose pour tout $n \in \mathbb{N}^*$: $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$.

(3) (a) Déterminer la fonction de répartition F de X .

(b) Tracer dans un repère orthogonal l'allure de la courbe représentative de F .

(4) (a) Déterminer un estimateur T_n de θ , sans biais et de la forme $c\bar{X}_n$, où c est un réel que l'on précisera.

(b) Quels sont les risques quadratiques respectifs associés aux estimateurs \bar{X}_n et T_n de θ ?

(5) On pose pour tout $n \in \mathbb{N}^*$: $M_n = \max(X_1, X_2, \dots, X_n)$.

(a) Déterminer la fonction de répartition G_n et une densité g_n de M_n .

(b) Calculer l'espérance de M_n . En déduire un estimateur sans biais W_n de θ .

(c) Entre T_n et W_n , quel estimateur doit-on préférer pour estimer θ ?

(6) Soit α un réel donné vérifiant $0 < \alpha < 1$.

(a) Établir l'existence de deux réels a et b tels que $0 < a < 1$ et $0 < b < 1$, vérifiant $\mathbb{P}(M_n \leq a\theta) = \frac{\alpha}{2}$ et $\mathbb{P}(b\theta \leq M_n \leq \theta) = \frac{\alpha}{2}$.

(b) En déduire un intervalle de confiance pour le paramètre θ au niveau de confiance $1 - \alpha$.

Exercice 4.

On dispose d'un n -échantillon (X_1, \dots, X_n) de la loi $\mathcal{E}(\lambda)$ et on cherche à estimer le paramètre $\lambda > 0$.

On pose alors

$$S_n = \sum_{i=1}^n X_i, \quad N_n = \lambda \frac{S_n}{\sqrt{n}} - \sqrt{n}, \quad \text{et} \quad \hat{\lambda}_n = \frac{n}{S_n}.$$

(1) Justifier que $(N_n)_{n \geq 1}$ converge en loi vers une v.a Z de loi $\mathcal{N}(0, 1)$.

On note Φ la fonction de répartition de Z .

(2) Soit $\alpha \in]0, 1[$. En notant $u_\alpha = \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$, montrer que

$$I_{\alpha, n} = \left[\left(1 - \frac{u_\alpha}{\sqrt{n}}\right) \hat{\lambda}_n; \left(1 + \frac{u_\alpha}{\sqrt{n}}\right) \hat{\lambda}_n \right]$$

est un intervalle de confiance (asymptotique) au risque α pour λ .

(3) Soit $\beta \in]0, 1[\setminus \{\alpha\}$. On suppose qu'on a réussi à construire, avec le même n -échantillon (X_1, \dots, X_n) , un intervalle de confiance (asymptotique) de la même forme $I_{\beta, n}$, de risque β pour λ dont la longueur est k fois plus petite que celle de $I_{\alpha, n}$ (où k est un entier supérieur ou égal à 2).

Montrer que $\beta = 2\Phi\left(\frac{1}{k}\Phi^{-1}\left(\frac{\alpha}{2}\right)\right)$.

En déduire $\beta > \alpha$. Commenter.

Python

Dans tous les exercices qui suivent, on suppose déjà importées sous leur alias habituels les bibliothèques Python usuelles.

```
import numpy as np
import numpy.random as rd
import numpy.linalg as al
import matplotlib.pyplot as plt
import pandas as pd
```

Partie 1 : Simulation de variables aléatoires

Exercice 1.

Soit n un entier naturel non nul.

On effectue une série illimitée de tirages d'une boule avec remise dans une urne contenant n boules numérotées de 1 à n . Pour tout entier naturel k non nul, on note X_k la variable aléatoire égale au numéro de la boule obtenue au k -ième tirage.

Pour tout entier naturel k non nul, on note S_k la somme des numéros des boules obtenues lors des k premiers tirages :

$$S_k = \sum_{i=1}^k X_i.$$

On considère enfin la variable aléatoire T_n égale au nombre de tirages nécessaires pour que, pour la première fois, la somme des numéros des boules obtenues soit supérieure ou égale à n .

Écrire une fonction Python, d'en-tête `def simul_T(n)`, qui prend en argument le nombre n de boules contenues dans l'urne, qui renvoie une simulation de la variable aléatoire T_n .

Exercice 2.

On considère une urne contenant initialement une boule bleue et deux boules rouges. On effectue, dans cette urne, des tirages successifs de la façon suivante: on pioche une boule au hasard, on note sa couleur, puis on la replace dans l'urne en ajoutant une boule de la couleur de celle qui vient d'être obtenue.

On note N_n la variable aléatoire qui prend pour valeur le nombre de boules rouges piochées après n tirages.

Écrire une fonction d'en-tête `def simul_N(n)` qui renvoie une simulation de S_n où le nombre n de tirages est en argument de la fonction.

Exercice 3.

Soit N un entier supérieur ou égal à 1.

On dispose de $2N$ boules blanches et de $2N$ boules noires, ainsi que de deux urnes vides, notées U_1 et U_2 . On répartit les boules au hasard dans les deux urnes selon le protocole suivant : on place les boules blanches une

par une, en choisissant l'urne au hasard de manière équiprobable, on place ensuite les boules noires de sorte que chaque urne contienne $2N$ boules.

L'expérience consiste alors à :

- tirer une boule au hasard dans U_1 et la placer dans U_2 ,
- puis tirer une boule au hasard dans U_2 et la placer dans U_1 ,
- et recommencer les deux étapes précédentes une infinité de fois.

On note, pour tout $n \in \mathbb{N}$, X_n le nombre de boules blanches contenues dans U_1 après le n -ième tirage.

La fonction Python suivante simule la variable aléatoire X_n pour une certaine configuration initiale des urnes. Expliquer ce qu'est cette configuration initiale des deux urnes (autrement dit quelle valeur prend X_0) et compléter cette fonction.

```
def simul_X(N, n):
    U1 = [0 for k in range(N)] + [1 for k in range(N)]
    U2 = [0 for k in range(N)] + [1 for k in range(N)]
    for k in range(1, n+1):
        if np.floor(k/2) == k/2: #si k est pair
            indice = rd.randint(0, 2*N+1)
            boule = U2[indice]
            del U2[indice]
            U1.append(boule)
        else: #si k est impair
            indice = rd.randint(0, 2*N)
            boule = U1[indice]
            del U1[indice]
            U2.append(boule)
    return sum(U1)
```

Exercice 4. (Le savant fou)

Un savant fou s'ennuie dans sa tour isolée et dispose d'une pièce usée qui tombe sur *Pile* avec probabilité $p \in]0, 1[$. Il décide de se lancer dans une expérience aléatoire dont le protocole est décrit ci-dessous :

- Il lance la pièce jusqu'à ce qu'elle tombe sur *Pile*. On note N la variable aléatoire égale au rang du premier *Pile*.
- Si la pièce est tombée sur *Pile* pour la première fois au n -ième lancer, alors il remplit une urne de la manière suivante. Pour tout entier $k \in \llbracket 1, n \rrbracket$,
 - si k est pair, alors il lance un dé à 6 faces. Il place ensuite dans l'urne autant de jetons numérotés k que le résultat indiqué sur le dé.
 - si k est impair, alors il lance un dé à 8 faces. Il place ensuite dans l'urne autant de jetons numérotés k que le résultat indiqué sur le dé.
- Une fois l'urne remplie, il procède à n tirages successifs et sans remise dans cette urne, note les résultats obtenus et fait leur somme. On note S la variable aléatoire égale au résultat obtenu.

Écrire une fonction Python d'en-tête `def simul_NS()` qui renvoie une simulation du couple (N, S) . Pour cela, on rappelle que :

- Si k est un entier, la commande `k % 2 == 0` renvoie `True` si k est pair et renvoie `False` si k est impair.
- Si `L` est une liste Python possédant plus de n éléments, alors la commande `rd.choice(L, n, replace = False)` simule une succession de n tirages sans remise dans cette liste et renvoie une liste contenant les résultats successifs.

Partie 2 : Représentations graphiques

Exercice 5.

On s'intéresse à la suite récurrente (u_n) définie par

$$u_0 = 1, \quad u_{n+1} = 1 - \exp(-u_n).$$

- (1) Compléter le programme ci-dessous permettant le calcul des 100 premiers éléments de (u_n)

```
U=np.zeros(100)
U[0]=.....
for k in range(.....) :
    U[k+1] = .....
plt.plot(....., U, 'k+')
plt.show()
```

- (2) On modifie le programme précédent en remplaçant la dernière ligne par les instructions suivantes

```
X=[k for k in range(1, 101)]
S=np.cumsum(U)
Y=[np.log(x) for x in X]
plt.plot(X, S, 'k+')
plt.plot(X, Y)
plt.show()
```

- (3) Que représente le vecteur S ? Quelle conjecture pouvez-vous émettre sur la nature de la série terme général u_n ?

Exercice 6.

On considère la suite (u_k) définie, pour $k \geq 2$ par $u_k = \frac{1}{k-1} - \frac{1}{k}$.

- (1) Vérifier la suite (u_k) peut être considérée comme une distribution de probabilité. On note alors Y une variable aléatoire telle que $Y(\Omega) = \llbracket 2, +\infty \llbracket$ et $P(Y = k) = u_k$.

Soit $n \in \mathbb{N}^*$. On dispose d'une urne contenant n boules, indiscernables au toucher, numérotées de 1 à n et on effectue des tirages successifs, avec remise, dans cette urne.

On introduit la variable aléatoire X_n qui prend la valeur du rang du tirage pour lequel, pour la première fois, la boule piochée a un numéro supérieur ou égal à celle piochée lors du tout premier tirage.

Pour tout $k \in \mathbb{N}^*$, on note T_k la variable aléatoire qui correspond au numéro de la boule piochée au k -ième tirage.

- (2) (a) Recopier et compléter la fonction suivante de sorte qu'elle renvoie une simulation de X_n .

```
def simul_X(n) :
    t = .....
    k = .....
    while ..... :
        k=k+1
    return k
```

- (3) Compléter les commandes suivantes qui permettent d'obtenir une liste de fréquences des valeurs obtenues pour un 1000-échantillon de X_n .

```
def frequence(n):
    S=[simul_X(n) for k in range(1000)]
    m=max(S)
    f=np.zeros(m)
    for k in range(1000):
        f[.....] +=1
    return f/1000
```

- (4) Ajouter les commandes suivantes et les exécuter en changeant les valeurs de n comme indiqué. Interpréter.

```

def u(k):
    return 1/(k-1) - 1/k

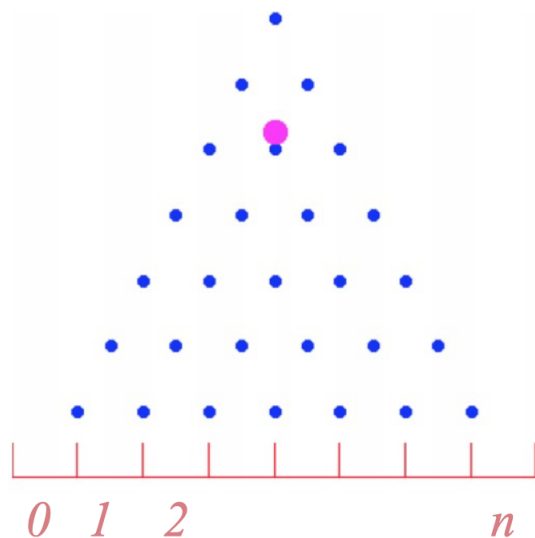
n=5 # puis n=10, puis n=25 puis n=100
obs=frequence(n)
N=min(len(obs), 14)
x=[k for k in range(2, N+2)]
U=[u(k) for k in x]
h=[obs[k] for k in range(N)]
plt.plot(x, U, 'kx')
plt.bar(x, h, width=0.1)
plt.show()

```

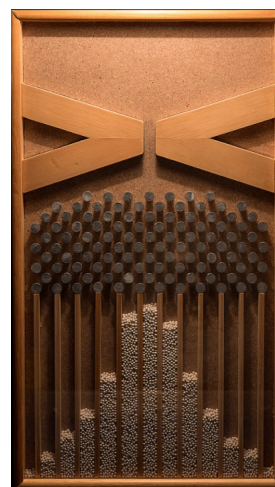
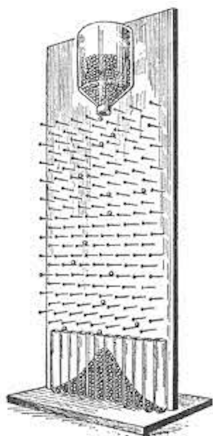
Exercice 7. Des clous sont plantés sur la partie supérieure d'une planche, en n rangées. Chaque rangée de clous contient un clou de plus que la précédente; la première en contient un, la n -ième exactement n .

On lâche une bille au "sommet" de la pyramide de clou, de telle sorte que chaque fois qu'elle rencontre un clou, celle-ci passe aléatoirement et uniformément soit à droite soit à gauche pour chaque rangée de clous.

Dans la partie inférieure de la planche, $n+1$ colonnes, numérotées de gauche à droite de 0 à n , sont disposées pour "récolter" les billes lâchées.



On lâche une bille et on note X_n la variable correspondant au numéro de la colonne dans laquelle se trouve la bille après avoir franchi les rangées de clous.



Proposer un programme permettant de simuler la chute de 1000 billes sur une planche de Galton à n rangées de clous et de représenter graphiquement la situation.

On rajoutera les commandes permettant d'observer sur la figure une certaine convergence en loi que l'on précisera.