Probabilités discrètes 2018-2019

Exercice 1. Une urne contient initialement une boule blanche et une boule noire. On effectue une série de tirages aléatoires d'une boule jusqu'à obtenir une boule noire. À chaque tirage amenant une boule blanche, on replace la boule blanche puis on multiplie par 2 le nombre de boules blanches présentes dans l'urne après la remise de la boule, puis on procède au tirage suivant. Soit X la variable aléatoire égale au rang du tirage amenant une boule noire (si on obtient la boule noire), et qui vaut 0 si on n'obtient jamais de boule noire. L'objectif de l'exercice est d'évaluer la probabilité de ne jamais obtenir de boule noire, et de déterminer en particulier si cette probabilité est nulle.

1. Compléter la fonction ci-dessous affin qu'elle réalise m fois l'expérience décrite ci-dessus (en arrêtant les tirages après l'obtention de `tmax` boules blanches), et renvoie la proportion des expériences où une boule noire a été obtenue :

```
def UneExperience(tmax):
    b= ...
    succes=0
    tirages=0
    while succes== ... and tirages ... :
        tirages ...
        if random() ... :
            succes= ...
        else:
            ...
    return succes
```

```
def EstimeProbaEchec(m, tmax):
    CompteSucces=0
    for i ... :
        CompteSucces+= ...
    return ...
```

```
def UneExperience(tmax):
    b=1
    succes=0
    tirages=0
    while succes==0 and tirages<tmax:
        tirages+=1
        if random()<1/(b+1) :
            succes=1
        else:
            b=2*b
    return succes
```

```
def EstimeProbaEchec(m, tmax):
    CompteSucces=0
    for i in range(m) :
        CompteSucces+= UneExperience(tmax)
    return CompteSucces/m
```

Utiliser cette fonction avec plusieurs valeurs de `tmax` pour établir une conjecture en réponse au problème posé.

2. On appelle B_n l'événement : « Les n premiers tirages ont eu lieu et n'ont donné que des boules blanches ».
   (a) Montrer que $P(B_n) = u_n = \displaystyle\prod_{k=0}^{n-1} \frac{2^k}{1+2^k}$.
   (b) Étudier le sens de variations et la convergence de la suite (u_n).
   (c) Montrer que $\forall x \geqslant 0,\ 0 \leqslant \ln(1+x) \leqslant x$.
   (d) Étudier convergence de la suite $(-\ln(u_n))$.
3. Conclure.

Exercice 2. L'objectif est d'estimer le nombre N de poissons dans un lac. Pour cela, on capture m poissons parmi les N du lac, que l'on marque (on les colore) avant de les relâcher. Puis le lendemain, on recapture n poissons : on prend les n poissons 1 par 1 et on regarde s'ils sont colorés ou non, en les relâchant dans le lac juste après. X_n est la variable aléatoire donnant le nombre de poissons marqués parmi les n poissons pêchés. NB : un poisson ne peut être marqué qu'une seule fois, mais il peut être pêché plusieurs fois.

1. Loi de X_n.
   (a) Donner la loi de X_n.
   (b) Pour k fixé et appartenant à $X_n(\Omega)$, déterminer N tel que $P(X_n = k)$ soit maximale.
   (c) On pose $T_n = \dfrac{m(n+1)}{X_n+1}$. Calculer $E(T_n)$.
2. Modélisation

(a) Écrire une fonction `capture` prenant en entrée le nombre N de poissons du lac, et le nombre m de poissons colorés, qui simule la capture des m poissons et qui renvoie une liste L de taille N formée de 0 et de 1 : 0 pour les poissons non colorés, 1 pour ceux colorés.

(b) Proposer une fonction `recapture` prenant N, m et n en entrée, qui simule la pêche de n poissons et renvoie le nombre de poissons marqués parmi les n poissons pêchés.

Exercice 3. (concours) On dispose de N dés à 6 faces,équilibrés. On les lance une fois et on note X_1 le nombre de 6 obtenus. On retire les dés ayant obtenu 6 et on relance les autres dés (si il en reste). On note X_2 le nombre de 6 obtenus. On répète l'expérience...
On note $S_n = X_1 + ... + X_n$ qui correspond au nombre de 6 obtenus en n lancers.

1. Écrire une fonction python `X(N)` qui simule X_1.
2. En déduire une fonction python `S(N,n)` qui simule S_n.
3. On veut montrer par récurrence que S_n suit une loi binomiale de paramètres (N, p_n) et on cherche à déterminer p_n.
   (a) Question préliminaire : Soient $N, m, k \in \mathbb{N}$ avec $m \leqslant k \leqslant N$. Montrer que : $\binom{N}{m}\binom{N-m}{k-m} = \binom{N}{k}\binom{k}{m}$
   (b) Montrer que S_1 suit une binomiale et déterminer p_1.
   (c) On suppose que S_n suit une loi binomiale de paramètres (N, p_n).
      i. Soit $0 \leqslant m \leqslant N$ et $0 \leqslant k \leqslant N$, déterminer $P(X_{n+1} = k - m | S_n = m)$.
      ii. En déduire que S_{n+1} suit une loi binomiale de paramètres N et $p_{n+1} = \dfrac{1+5p_n}{6}$.
   (d) Déterminer p_n en fonction de n.
4. T est la variable aléatoire égale au plus petit entier n tel que $S_n = N$ c'est-à-dire le nombre de lancers minimal pour obtenir tous les 6 et égale à 0 si on n'obtient jamais tous les 6.
   (a) Déterminer la fonction de répartition de T.
   (b) En déduire que T admet une espérance et la calculer. On admet que T admet une espérance si la série $\sum P(T > n)$ converge et dans ce cas $E(t) = \sum_{n=0}^{+\infty} P(T > n)$.

```
def SimulX(N):
    X=0
    for k in range(N):
        h=rd.randrange(1,7)
        X+=(h==6)
    return X

def SimulS(N,n):
    S=0
    for k in range(n):
        X=SimulX(N-S)
        S+=X
    return S
```

```
from random import *
nbrExp=10000
N=10 ; n=6
s=0
for k in range(nbrExp):
    s+=S(N,n)
print(s/nbrExp,N*(1-(5/6)**n))
```

Exercice 4. Les joueurs A et B lancent une pièce. La probabilité de Pile à chaque lancer est p, celle de Face $1 - p$.
A lance la pièce jusqu'au 2e pile. Si il a obtenu k Face, il demande à B un nombre au hasard entre 0 et k.
On appelle X le nombre de Face avant 2e Pile et Y nombre choisi au hasard entre 0 et k.

1. Écrire une fonction simulant X et Y.
2. (a) Montrer que : $P(X = k) = (k+1)p^2q^k$.
   (b) Exprimer $X + 2$ comme somme de 2 variables aléatoires indépendantes judicieusement choisies.
   (c) En déduire $E(X)$ et $V(X)$.
3. (a) Trouver la loi de Y.
   (b) Calculer $E(Y)$ et $V(Y)$.
   (c) X et Y sont elles indépendantes?

(d) On pose $Z = X - Y$. Montrer que Z et Y suivent la même loi. Z et Y sont elles indépendantes ?

```
def SimulG(p):
    # nombre d'echecs avant le premier succes
    U=0
    while random()>p:
        U+=1
    return U

def X(p):
    U=SimulG(p)
    V=SimulG(p)
    return U+V

from random import *
S=0 ; p=1/7 ; N=10000
for k in range(N):
    S+=X(p)
print(S/N,2/p-2)
```

Exercice 5. Un ticket de métro vaut 2 euros. La première amende s'élève à 40euros, la deuxième à 80euros et la troisième à 160euros Un fraudeur décide de ne pas acheter de billet jusqu'au moment où il paye la deuxième amende. On appelle T la variable correspondant aux nombres de trajets effectués sans achats de billets. On note p la probabilité de contrôle d'un voyageur.

1. Écrire une fonction python donnant en retour la variable T. Utiliser cette fonction pour calculer une valeur approchée de son espérance.
2. Déterminer la loi de T.
3. Calculer l'espérance de T.
4. Calculer $P(T > n)$.
5. Application avec $n = 60$ et $p = 1/10$ puis avec $p = 1/20$. Pourquoi s'intéresse t-on à $n = 60$?
6. Conclure en donnant un avis sur la tactique du fraudeur.

Exercice 6. Une urne contient 2 lots de jetons numérotés de 1 à n.
On effectue des tirages simultanés de deux jetons. Si les jetons tirés sont identiques, alors on les enlève de l'urne. Sinon, on les y remet.
Soit T_n la variable aléatoire qui compte le nombre de tirages nécessaires à vider l'urne de son contenu.

1. Cas où l'urne ne contient qu'une paire : Quelle loi suit la variable T_1 ?
2. On suppose $n = 2$: l'urne ne contient que 2 paires. Soit k un entier naturel supérieur ou égal à 2.
   Montrer que : $\mathbb{P}(T_2 = k) = \dfrac{1}{2}\left(\dfrac{2}{3}\right)^{k-1}$. Calculer $\mathbb{E}(T_2)$
3. On revient ici au cas n quelconque
   (a) Écrire une fonction Python qui prend n comme argument et qui simule la réalisation du premier tirage : elle renverra 1 si les 2 numéros sont identiques et 0 sinon.
   (b) Écrire une fonction Python qui simule la variable T_n.
   (c) Évaluer la valeur moyenne de T_n pour plusieurs valeurs de n. Conjecturer une formule donnant l'espérance de T_n.
4. On considère l'événement C : « on tire deux jetons identiques lors du premier tirage »
   (a) Montrer que $\mathbb{P}(C) = \dfrac{1}{2n-1}$
   (b) Montrer que : $\mathbb{P}(T_n = k) = \dfrac{1}{2n-1}\mathbb{P}(T_{n-1} = k-1) + \dfrac{2n-2}{2n-1}\mathbb{P}(T_n = k-1)$
   (c) Montrer que si T_n admet une espérance, alors : $\mathbb{E}(T_n) = \mathbb{E}(T_{n-1}) + 2n - 1$. En déduire $E(T_n$. Cela semble-t-il cohérent avec les résultats du 3.(c) ?

```
def Identiques(n):
    L=[k for k in range(n)]+[k for k in range(n)]
    h1=rd.randrange(len(L))
    a=L.pop(h1)
    h2=rd.randrange(len(L))
    b=L.pop(h2)
    return a==b
```

```
def SimulTn(n):
    T=0
    for k in range(n):
        T+=1
        while not Identiques(n-k):
            T+=1
    return T

def EstimeEsperanceT(n,NExp):
    S=0
    for k in range(NExp):
        S+=SimuleT(n)
    return S/NExp
```

Exercice 7. On dispose de n urnes numérotées de 0 à $n-1$. Dans l'urne k on trouvent k boules blanches et $n-k$ boules noires. On choisit une urne au hasard et on effectue des tirages avec remise jusqu'à obtenir une boule noire. Soit X_n la variable aléatoire égale au numéro de l'urne choisie et Y_n la variable aléatoire qui correspond au nombre de tirages nécessaires pour la première noire.

1. (a) Écrire une fonction qui simule Y_n
   (b) Donner une estimation de $E(Y_n)$. Comment évolue-t-elle quand n augmente ?
   (c) Écrire une fonction d'arguments (n,k,N) qui renvoie la fréquence de réalisations de $(Y_n=k)$ au cours de N simulations. En déduire une valeur approchée de $P(Y_n=k)$ quand $k=1,2,3$. Comment évolue cette probabilité quand n tend vers l'infini.
2. Déterminer la loi de Y_n sachant $X_n=k$. En déduire la loi de Y_n (on laissera sous forme de somme).
3. (a) En utilisant des sommes de Riemann montrer que $\lim\limits_{n\to+\infty} P(Y_n=i)=\dfrac{1}{i}-\dfrac{1}{i+1}$.
   (b) On pose $\forall k\in\mathbb{N}^*,\ P(Y=k)=\lim\limits_{n\to+\infty} P(Y_n=k)$. Comparer la valeur théorique avec la valeur calculée. Montrer que $\sum\limits_{k=1}^{+\infty} P(Y=k)=1$. Y admet-elle une espérance ?

```
def SimulY(n):
    X=rd.randrange(n)
    Y=1
    p=X/n
    while rd.random()<p: Y+=1
    return Y

def EstimeEsperanceY(n,NExp):
    S=0
    for k in range(NExp):
        S+=SimuleT(n)
    return S/NExp

def Frequence(n,k,NExp):
    S=0
    for j in range(NExp):
        S+=(SimulY(n)==k)
    return S/NExp
```

Exercice 8. On lance n fois un dé équilibré (où n est un entier naturel non nul) : on appelle gain G_n remporté par le joueur la quantité définie par :
• Si le numéro 1 est obtenu au moins une fois, le gain G_n est nul,
• Si aucun numéro 1 n'est obtenu, le gain G_n est égal à la somme des nombres obtenus au cours des n lancers.
Soit X_i la variable aléatoire valant 0 si le numéro 1 est obtenu au lancer i , et valant le numéro du chiffre obtenu sinon. Soit Y_i la variable aléatoire valant 0 si le numéro 1 est obtenu au lancer i, et valant 1 sinon.

1. Créer une fonction Python dépendant d'un entier n et retournant une simulation de G_n.

```
def SimulGain(n):
    G=0
    for k in range(n):
        r=randrange(1,7)
        if r==1: return 0
        else: G+=r
    return G
```

2. Déterminer la loi de Y définie par $Y = \prod_{i=1}^{n} Y_i$.

   $(Y = 1) = (X_1 \neq 1)\bigcap\ldots\bigcap(X_n \neq 1)$. Donc

   Y suit une loi de Bernoulli de paramètre $p = P(Y = 1) = P(X_1 \neq 1)\bigcap\ldots\bigcap(X_n \neq 1) = (\frac{5}{6})^n$.

3. Ecrire une fonction permettant d'estimer $E(G_n)$ (on prendra 10 000 itérations) pour n entier naturel non nul quelconque, et une fonction permettant de tracer $E(G_n)$ en fonction de n pour n élément de $[\![2,20]\!]$.

```
def EsperanceApprocheeG(n):
    S=0
    for k in range(10000):
        S+=SimulGain(n)
    return S/10000

from random import randrange
n=15
print(EsperanceApprocheeG(n))
print(4*n*(5/6)**n)
```

4. Exprimer G_n en fonction de Y et des X_i.

   $G_n = Y\sum_{k=1}^{n} X_k$

5. Déterminer l'espérance de YX_i, en déduire l'espérance de G_n.

   $\forall k \in [\![2,n]\!]$, $P(YX_i = k) = P(X_i = k|Y \neq 0)P(Y \neq 0) = \frac{1}{5}(\frac{5}{6})^n$

   Donc $E(YX_i) = \sum_{k=2}^{6} k\dfrac{1}{5}(\dfrac{5}{6})^n = \dfrac{1}{5}(\dfrac{5}{6})^n(\dfrac{6\times 7}{2} - 1) = 4(\dfrac{5}{6})^n$

   $E(G_n) = E(\sum_{i=1}^{n} E(YX_k) = 4n(\dfrac{5}{6})^n$

6. Donner une valeur approchée de la variance de G_n (on prendra également 10 000 itérations).

7. Pour i et j entiers entre 1 et n, déterminer $E(YX_iX_j)$. En déduire la valeur de la variance de G_n.

Exercice 9. On étudie la descendance d'une fleur. Le nombre de descendants de cette fleur est une variable aléatoire qui suit la loi binomiale $\mathscr{B}(2,p)$.
À chaque génération, on suppose que le nombre de descendants de chaque fleur suit de façon indépendante la même loi que la fleur initiale.
On note E_n l'événement "il n'y a plus de descendance à la génération n" et u_n la probabilité de cet événement.

1. (a) Calculer u_1.
   (b) Montrer que $\forall n \in \mathbb{N}^*$, $u_{n+1} = ((1-p)+pu_n)^2$.
   (c) Étudier la suite (u_n). Trouver sa limite et commenter le résultat obtenu.
2. (a) Écrire une fonction qui renvoie le nombre de descendants à la première génération.
   Écrire une fonction donnant la descendance à la génération n.
   (b) Écrire un programme approchant la fréquence de l'extinction au bout de 20 générations.

```python
def DescendantsUn(p):
    X=0
    for k in range(3):
        if rd.random()<p: X+=1
    return X

def GenerationSuivante(p,G):
    G1=0
    for k in range(G):
        G1+=DescendantsUn(p)
    return G1

def Descendants(n,p):
    G=1
    for k in range(n):
        for i in range(G):
            G=GenerationSuivante(p,G)
            if G>10000: return 'trop grand'
    return G

def Extinction(n,p,NExp):
    S=0
    for k in range(NExp):
        G=Descendants(n,p)
        if G==0: S+=1
    return S/NExp
```

Exercice 10. Soit une variable aléatoire réelle X.

On suppose que X^2 suit une loi de Poisson et que pour $k \in \mathbb{N}^*$, $P(X = -\sqrt{k}) = \dfrac{1}{k} P(X = \sqrt{k})$.

1. Donner l'espérance et la variance de X^2.
2. Calculer $P(X = \sqrt{n})$ pour $n \in \mathbb{N}^*$, $P(X = 0)$ et $P(X \geqslant 0)$.
3. Démontrer que pour tous entiers naturels m et p on a : $(m^2 + p)! \geqslant (m^2)!p!$
4. Montrer que $P(|X| \geqslant m) \leqslant \dfrac{\lambda^{m^2}}{(m^2)!}$.
5. Faire un programme qui prend en entrée $\lambda > 0$ et $a > 0$ et sort la probabilité que $|X| \leqslant a$.

Exercice 11. On pose $P(X = i \bigcap Y = j) = \dfrac{\alpha}{2^{i+1} j!}$.

1. Déterminer α pour qu'on définisse ainsi la loi d'un couple de variables aléatoires.
2. Démontrer que $\forall n \in \mathbb{N}$, $P(X + Y = n) = \dfrac{1}{e 2^{n+1}} \displaystyle\sum_{j=0}^{n} \frac{2^j}{j!}$.

   Écrire un programme python qui calcule $P(X + Y = n)$.
3. Quelle est la loi de X ? Reconnaître la loi de $Z = X + 1$. En déduire $E(X)$ et $V(X)$.
4. Quelle est la loi de Y ? Donner $E(Y)$ et $V(Y)$.
5. X et Y sont-elles indépendantes ?
6. Calculer $P(X = Y)$.

Exercice 12. (INA 2009) On considère un concessionnaire de voitures. Soit X la variable aléatoire égale au nombre de voitures vendues par jour. X suit une loi de Poisson de paramètre λ ($\lambda > 0$). On considère qu'un acheteur de voiture fait un prêt pour se procurer sa voiture avec une probabilité p ($0 < p < 1$). Soit Y la variable aléatoire égale au nombre d'acheteurs par jour de voitures qui font un prêt.

1. Déterminer $P(Y = k / X = n)$. En déduire la loi conjointe de (X, Y).
2. Déterminer la loi de Y. Donner $E(Y)$, $V(Y)$.

3. On pose $Z = X - Y$. Quelle est la loi de Z ? Y et Z sont-elles indépendantes ? Calculer $\operatorname{cov}(X, Y)$.

Exercice 13. On choisit au hasard un nombre X entre 1 et n puis un nombre Y entre 1 et le premier numéro choisi. Quelle est la loi de X ? de Y ? Calculer $E(Y)$ et $\operatorname{cov}(X, Y)$.

Exercice 14. Un éditeur de livres d'histoires met sur le marché N images pour illustrer. Pour $k \in \mathbb{N}^*$ on appelle T_k la variable aléatoire égale au nombre de pochettes achetées pour avoir pour la première fois k images distinctes. On pose $S_k = T_k - T_{k-1}$ pour $k \geqslant 2$.

1. Trouver les lois de T_1 et T_2.
2. Trouver la loi de S_k. Donner son espérance.
3. Calculer $E(T_k)$.

Exercice 15. n personnes choisissent au hasard un hôtel parmi 3 hôtels H_1, H_2, H_3. On appelle X_i le nombre de personnes qui choisissent l'hôtelH_i.

1. Trouver les lois de X_1, de X_2, de $X_1 + X_2$.
2. Calculer le coefficient de corrélation linéaire de X_1 et X_2.

Exercice 16. Une personne envoie chaque jour un courrier électronique par l'intermédiaire de deux serveurs A et B. Le serveur A est choisi dans 70% des jours et le serveur B est choisi dans 30% des jours. Les choix des serveurs sont supposés indépendants les uns des autres. On note les différents serveurs utilisés par une suite de lettres.
Par exemple la suite $AABBBA$... signifie que les deux premiers jours le serveur A a été choisi, les jours 3,4,5 le serveur B, le jour 6 le serveur A... Dans cet exemple on dit que l'on a une première série de longueur 2 et une deuxième série de longueur 3. Ce qui est également le cas de la série $BBAAAB$....
On note L_1 la variable aléatoire représentant la longueur de la première série et L_2 variable aléatoire représentant la longueur de la seconde série.

1. Déterminer la loi de L_1. Vérifier que $\sum_{k=1}^{+\infty} P(L_1 = k) = 1$. Déterminer l'espérance de L_1. Montrer que $E(L_1) \geqslant 2$.
2. Déterminer la loi du couple (L_1, L_2). En déduire la loi de L_2. Calculer $E(L_2)$.
3. Calculer la covariance de L_1 et L_2. Comment interpréter son signe ? L_1 et L_2 sont-elles indépendantes ?