Histoire 1

Jacques Bernoulli (1654-1705) énonce la **loi binomiale** dans son ouvrage *Ars Conjectandi* édité par son neveu Nicolas Bernoulli. L'oncle appelle *cas féconds* les cas dans lesquels un événement peut se produire et *cas stériles*, ceux où il ne peut pas se produire. Il établit que si la probabilité des cas féconds est de r et celle des cas stériles de s, sur une succession de nt expériences : « *les degrés de probabilités [...] pour lesquels il peut arriver que toutes les expériences soient fécondes ou toutes le soient sauf une qui est stérile, ou toutes sauf deux, 3, 4 qui sont stériles s'expriment respectivement par :*

$$r^{nt}, \frac{nt}{1}r^{nt-1}s, \frac{nt(nt-1)}{1\times 2}r^{nt-2}s^2, \frac{nt(nt-1)(nt-2)}{1\times 2\times 3}r^{nt-3}s^3 \dots$$

c'est-à-dire par les termes de la puissance nt du binôme $r+s$. »

1 Succession d'épreuves indépendantes

Définition 1

On considère n expériences aléatoires **successives**. Si les résultats de chacune d'elles ne dépendent pas des résultats des autres expériences, on dit que ces expériences sont **indépendantes**.

Définition 2

Soit n un entier naturel. Soit une expérience aléatoire E qui est constituée d'une succession d'expériences aléatoires $E_1, E_2, \dots, E_n$, d'univers finis $\Omega_1, \Omega_2, \dots \Omega_n$.

- L'univers Ω de l'expérience E est le produit cartésien $\Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ des univers de la succession d'expériences aléatoires.
- Une issue de Ω est un n-uplet (ou liste) $(x_1, x_2, \dots, x_n)$ d'issues du produit cartésien $\Omega_1 \times \Omega_2 \times \dots \times \Omega_n$.

Propriété 1

Soit n un entier naturel. Soit une expérience aléatoire E qui est constituée d'une succession d'expériences aléatoires **indépendantes** $E_1, E_2, \dots E_n$, d'univers finis $\Omega_1, \Omega_2, \cdots \Omega_n$.
La probabilité d'une liste d'issues $(x_1, x_2, \cdots, x_i, \cdots, x_n)$ est égale au produit des probabilités des composantes x_i

$$\mathbb{P}(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n) = \mathbb{P}(\mathrm{x}_1) \times \mathbb{P}(\mathrm{x}_2) \times \cdots \mathbb{P}(\mathrm{x}_n)$$

Capacité 1 ***Modéliser une situation par une succession d'épreuves indépendantes et calculer des probabilités***

Une urne contient des jetons blancs et noirs tous indiscernables au toucher.
Une partie consiste à prélever au hasard successivement et avec remise deux jetons de cette urne.
On établit la règle de jeu suivante :

- un joueur perd 9 euros si les deux jetons tirés sont de couleur blanche ;
- un joueur perd 1 euro si les deux jetons tirés sont de couleur noire ;
- un joueur gagne 5 euros si les deux jetons tirés sont de couleurs différentes.

1. On considère que l'urne contient 2 jetons noirs et 3 jetons blancs.

a. Modéliser la situation à l'aide d'un arbre pondéré.

b. Calculer la probabilité de perdre 9 € sur une partie.

2. On considère maintenant que l'urne contient 3 jetons blancs et au moins deux jetons noirs mais on ne connait pas le nombre exact de jetons noirs. On appellera N le nombre de jetons noirs.

a. Soit X la variable aléatoire donnant le gain du jeu pour une partie.
Déterminer la loi de probabilité de cette variable aléatoire.

b. Résoudre l'inéquation pour x réel :

$$-x^2+30x-81>0$$

c. En utilisant le résultat de la question précédente, déterminer le nombre de jetons noirs que l'urne doit contenir afin que ce jeu soit favorable au joueur.

d. Combien de jetons noirs le joueur doit-il demander afin d'obtenir un gain moyen maximal ?

3. On observe 10 joueurs qui tentent leur chance en effectuant une partie de ce jeu, indépendamment les uns des autres. On suppose que 7 jetons noirs ont été placés dans l'urne (avec 3 jetons blancs).

Quelle est la probabilité d'avoir au moins 1 joueur gagnant 5 euros ?

2 Épreuve et schéma de Bernoulli

2.1 Épreuve de Bernoulli

Définition 3

Une **épreuve de Bernoulli** de paramètre p (avec $0 \leqslant p \leqslant 1$) est une expérience aléatoire E dont l'univers Ω ne comporte que deux issues appelées « succès » (noté S) et échec (noté $\overline{\text{S}}$) de probabilités respectives p et $1-p$.
La variable aléatoire $\text{X}:\Omega \rightarrow \{0,1\}$ qui prend la valeur 1 en cas de succès et 0 en cas d'échec, suit une **loi de Bernoulli** de paramètre p,qu'on note $\mathscr{B}(p)$.
La loi de probabilité d'une variable aléatoire de Bernoulli de paramètre p, est donnée par le tableau suivant :

k	1	0
$\mathbb{P}(\mathrm{X}=k)$	p	$1-p$

Propriété 2

Soit X une variable aléatoire qui suit une loi de Bernoulli de paramètre p avec $0 \leqslant p \leqslant 1$.

1. **L'espérance** de X est égale à $\boxed{\mathbb{E}(X) = p}$.

2. **La variance** de X est égale à $\boxed{\mathbb{V}(X) = p(1-p)}$ et **l'écart-type** de X est égal à $\boxed{\sigma(X) = \sqrt{p(1-p)}}$.

Démonstration ***Voir manuel Indice p.366***

Soit X une variable aléatoire suivant une loi de Bernoulli de paramètre p.

....

....

....

....

....

....

....

....

....

....

Méthode

La bibliothèque `random` de `Python` propose des fonctions permettant de générer des nombres pseudo-aléatoires :

- La fonction `random` permet de générer un nombre décimal choisi aléatoirement dans l'intervalle $[0;1[$ avec l'appel `random()`.

- La fonction `randint` permet de générer un nombre entier choisi aléatoirement entre deux entiers `a` et `b` (bornes incluses), vérifiant `a` $\leqslant$ `b`, avec l'appel `randint(a, b)`.

```
In [1]: from random import random, randint

In [2]: [randint(1, 6) for k in range(3)]
Out[2]: [2, 3, 4]
```

```
In [3]: [random() for k in range(2)]
Out[3]: [0.6222757409726671, 0.965200387415264]
```

Algorithmique 1 *Simuler une variable aléatoire de Bernoulli*

On lance un dé cubique équilibré dont les faces sont numérotées de 1 à 6. On appelle « succès » l'apparition de la face 6.
Soit X la variable aléatoire qui prend la valeur 1 si la face est 6 et la valeur 0 sinon.

1. X suit-elle une loi de Bernoulli ? Si oui, déterminer son paramètre.

2. On rappelle que `randint(a, b)` est un entier choisi aléatoirement entre deux entiers `a` et `b` compris. Compléter la fonction `Python` ci-dessous pour qu'elle simule une réalisation de la variable aléatoire X :

```
from random import randint

def simulX():
    if .......................:
        return ......
    else:
        return ......
```

3. De quelle valeur devraient se rapprocher `mystere(1000)`, `mystere(10000)` et `mystere(100000)` ? Justifier.

```
from random import randint

def mystere(n):
    s = 0
    for k in range(n):
        s = s + simulX()
    return s / n
```

2.2 Schéma de Bernoulli

Définition 4

On appelle **schéma de Bernoulli** de paramètres n et p la répétition de n épreuves de Bernoulli (avec $n \geqslant 1$) de paramètre p (avec $0 \leqslant p \leqslant 1$) identiques et indépendantes (c'est à dire que l'issue d'une épreuve ne dépend pas des issues des autres épreuves).

Capacité 2 *Modéliser une situation par un schéma de Bernoulli*

Pour chacune des expériences aléatoires suivantes, déterminer si elle peut être modélisée par un schéma de Bernoulli et si oui, préciser ses paramètres.

1. On lance dix fois une pièce équilibrée et on compte le nombre de « Face » obtenues.

2. On lance cinq fois un dé cubique à 6 faces numérotées de 1 à 6 et on compte le nombre de six obtenus.

3. On lance cinq fois un dé cubique à 6 faces numérotées de 1 à 6 et on fait la somme des faces obtenues.

4. On tire trois fois et avec remise une boule dans une urne contenant 10 noires et 7 boules rouges et on note le nombre de boules rouges obtenues.

5. On tire trois fois et sans remise une boule dans une urne contenant 10 noires et 7 boules rouges et on note le nombre de boules rouges obtenues.

3 Loi binomiale

3.1 Loi du nombre de succès

Définition 5

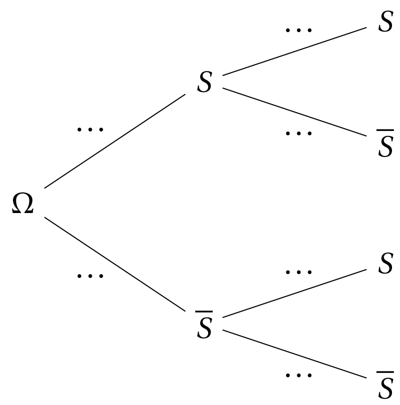
Soit un **schéma de Bernoulli** de paramètres n et p et soit X la variable aléatoire qui à une liste de n résultats $\left(\text{comme } \left(S, \overline{S}, \overline{S}, \cdots, S\right)\right)$ associe le nombre de succès dans cette liste.
On dit que X suit une **loi binomiale** de paramètres n et p et on note $X \hookrightarrow \mathscr{B}\left(n, p\right)$.

Capacité 3 ***Vers la loi de probabilité d'une loi binomiale***

On considère l'épreuve de Bernoulli E de paramètre $p = \dfrac{1}{6}$ qui consiste à lancer un dé équilibré à 6 faces numérotées de 1 à 6 et à compter comme *succès* l'obtention d'un 6.

1. On répète 2 fois l'expérience aléatoire E de façon indépendante.

   a. Compléter l'arbre pondéré ci-dessous. On a noté S un succès et $\overline{S}$ un échec.



   b. Soit la variable aléatoire X_2 qui compte le nombre de *succès*. Déterminer la loi suivie par X_2.

      i. Quelles sont les valeurs prises par la variable aléatoire X_2 ?

      ii. Déterminer sa loi de probabilité puis son espérance.

2. On répète 3 fois l'expérience aléatoire E de façon indépendante.

a. Représenter ce schéma de Bernoulli de paramètres $n=3$ et $p=\dfrac{1}{6}$ par un arbre pondéré en notant S un succès et $\overline{S}$ un échec.

b. Soit la variable aléatoire X_3 qui compte le nombre de *succès*. Déterminer la loi suivie par X_3.

i. Quelles sont les valeurs prises par la variable aléatoire X_3 ?

ii. Déterminer les probabilités $\mathbb{P}(X_3=3)$ et $\mathbb{P}(X_3=0)$.

iii. Combien de chemins dans l'arbre réalisent 2 succès ? Exprimer ce nombre à l'aide d'un coefficient binomial et en déduire une formule de calcul de $\mathbb{P}(X_3=2)$.

iv. Exprimer de même $\mathbb{P}(X_3=1)$.

v. Dresser un tableau de la loi de probabilité de X_3 et déterminer son espérance.

3.2 Loi de probabilité d'une loi binomiale

Propriété 3

Soit X une variable aléatoire qui suit une loi binomiale $\mathscr{B}(n,p)$ avec n entier non nul et p réel entre 0 et 1.

Pour tout entier $k \in \{0,\ 1,\ \cdots,\ n\}$ on peut exprimer $\mathrm{P}(X=k)$ en fonction de $\dbinom{n}{k}$:

$$\boxed{\mathrm{P}(X=k)=\ \dots\dots\dots\dots\dots\dots\dots\dots\dots}$$

Démonstration ***Au programme, voir manuel Indice p. 368***

Dans un schéma de Bernoulli de paramètres n et p, une issue réalisant l'événement $\{X=k\}$ avec k entier compris entre 0 et n, est un n-uplet comportant k succès S et $n-k$ échecs $\overline{S}$.

....

....

....

....

....

....

....

....

....

Capacité 4 ***Connaître la loi de probabilité d'une loi binomiale***

Soit $n \in \mathbb{N}^*$ et soit $p \in [0;1]$. Soit X une variable aléatoire qui suit une loi binomiale $\mathscr{B}(n\,;\,p)$.
Calculer les probabilités suivantes :

1. $\mathbb{P}(X=0)$
2. $\mathbb{P}(X=n)$
3. $\mathbb{P}(X=1)$
4. $\mathbb{P}(X=n-1)$
5. $\mathbb{P}(X=2)$
6. $\mathbb{P}(X=n-2)$

3.3 Calculer des probabilités pour une loi binomiale

Méthode ***Fonctions préprogrammées de la calculatrice ou du tableur, voir manuel Indice p.372 et p.373***

	Menu	$P(X=k)$	$P(X \leqslant k)$
Casio	Touche **MENU** puis **STAT** puis **DIST**, puis **BINM**	Choisir **Bpd** et **Var** puis saisir les paramètres	Choisir **Bcd** et **Var** puis saisir les paramètres
Texas	Menu **Distrib** (2nde var) puis **binomFdp**, ou **binomFRep**	**binomFdp(n,p,k)**	**binomFrép(n,p,k)**
Numworks	Menu **Calculs** puis **Probabilités** dans la **Boîte à outils**	**binompdf(k, n, p)**	**binomcdf(k, n, p)**

Soit X une variable aléatoire suivant la loi binomiale $\mathscr{B}(10\,;\,0{,}3)$, si on utilise le tableur `Calc` de `LibreOffice` :

- pour calculer $\mathbb{P}(X=2)$ dans une cellule, on saisit `=LOI.BINOMIALE(2;10;0,3;0)`
- pour calculer $\mathbb{P}(X\leqslant 2)$ dans une cellule, on saisit `=LOI.BINOMIALE(2;10;0,3;1)`
- pour calculer $\mathbb{P}(X\geqslant 2)$, on passe à l'événement contraire et $\mathbb{P}(X\geqslant 2)=\mathbb{P}(X>1)=1-\mathbb{P}(X\leqslant 1)$.

Capacité 5 ***Calculer des probabilités pour une loi binomiale***

On considère une variable aléatoire X qui suit la loi binomiale de paramètres $n=30$ et $p=0{,}4$.

1. Calculer les valeurs exactes des probabilités $\mathbb{P}(X=0)$, $\mathbb{P}(X=1)$ et $\mathbb{P}(X\geqslant 1)$.
2. Calculer les valeurs exactes des probabilités $\mathbb{P}(X\leqslant 2)$ et $\mathbb{P}(X\geqslant 29)$ puis comparer avec les valeurs approchées obtenues avec les fonctions préprogrammées `binomcdf` de la calculatrice.

Algorithmique 2

1. Compléter les fonctions Python ci-dessous pour qu'elles répondent aux spécifications fixées en commentaire.

```
def factorielle(n):
    # renvoie n!
    f = 1
    for k in range(1, n + 1):
        f = .......
    return f

def binom(n, k):
    # renvoie le coefficient binomial k parmi n
    return factorielle(n) // (factorielle(n - k) * factorielle(k))

def binompdf(n, p, k):
    # renvoie P(X=k) pour une va X de loi binomiale B(n,p)
    return .............................

def binomcdf(n, p, k):
    # renvoie P(X <= k) pour une va X de loi binomiale B(n,p)
    proba_cumul = 0
    for i in range(0, k + 1):
        proba_cumul = .................
    return proba_cumul
```

2. Soit X une variable aléatoire qui suit une loi binomiale de paramètres $n = 10$ et $p = 0,3$.

   Exprimer les probabilités ci-dessous à l'aide des fonctions Python définies précédemment puis calculer des valeurs approchées avec les fonctions préprogrammées de la calculatrice.

   a. $\mathbb{P}(X = 3)$ **c.** $\mathbb{P}(X \leqslant 4)$ **e.** $\mathbb{P}(4 \leqslant X \leqslant 8)$

   b. $\mathbb{P}(X \leqslant 3)$ **d.** $\mathbb{P}(X \geqslant 4)$ **f.** $\mathbb{P}((X \leqslant 4) \cup (8 \leqslant X))$

3.4 Modéliser une situation par une loi binomiale

Capacité 6 *Modéliser par une loi binomiale*

Une urne contient trois boules rouges et cinq boules bleues, indiscernables au toucher. On tire successivement, avec remise, 10 boules dans cette urne. On note X le nombre de boules rouges tirées.

1. Déterminer la loi suivie par la variable aléatoire X.
2. Donner une expression de $\mathbb{P}(\mathrm{X} = 0)$.
3. Donner une expression de $\mathbb{P}(\mathrm{X} \geqslant 1)$.
4. Donner une expression de $\mathbb{P}(\mathrm{X} \geqslant 2)$.
5. Donner une expression de la probabilité conditionnelle $\mathbb{P}_{\mathrm{X} \geqslant 1}(\mathrm{X} \geqslant 2)$.

3.5 Loi binomiale et problèmes de seuil

Capacité 7 ***Utiliser l'expression de la loi binomiale pour résoudre un problème de seuil***

Un test de dépistage d'une maladie a été mis au point. Si l'individu est malade, dans 94% des cas le test est positif. Pour un individu choisi au hasard dans cette population, la probabilité que le test soit positif vaut $0,158$. De plus la proportion de malades dans la population est de 15%.

1. On teste un individu choisi au hasard dans la population : le test est positif. Déterminer une valeur arrondie au centième de la probabilité que la personne soit malade.
2. On prélève un échantillon aléatoire dans la population, et on fait passer le test aux individus de cet échantillon. Déterminer la taille minimale de l'échantillon pour que la probabilité qu'au moins un individu soit testé positivement, soit supérieure ou égale à $0,99$.

Capacité 8 ***Déterminer un seuil s pour lequel la probabilité $\mathbb{P}(X > s)$ est inférieure à une valeur donnée α***

Une roseraie livre des fleuristes par lots de 100 roses. Afin de fidéliser ses clients, elle rembourse un lot lorsqu'il contient strictement plus de 5 roses fanées. Une étude statistique a montré que 4% des roses livrées sont fanées. On choisit un lot au hasard et on note X la variable aléatoire qui donne le nombre de roses fanées de ce lot.

1. **a.** Déterminer la loi de probabilité suivie par X.

   b. Calculer la probabilité que le lot soit remboursé au fleuriste. Arrondir au centième.

2. La roseraie souhaite réduire le nombre de lots remboursés. Pour cela elle décide d'augmenter le nombre k de roses fanées à partir duquel le lot est remboursé. On veut déterminer le seuil s pour lequel la probabilité de rembourser le lot soit inférieure ou égale à $0,01$.

   a. Quelle inégalité est vérifiée par $\mathbb{P}(X \leqslant s)$?

   b. Compléter les fonctions `Python` ci-dessous pour que `seuil()` ait la valeur s recherchée.

   `comb(n, k)` est égal au coefficient binomial $\binom{n}{k}$ avec n et k entiers naturels tel que $0 \leqslant k \leqslant n$.

```python
from math import comb

def binompdf(n, p, k):
    #renvoie P(X=k) pour X suivant la loi binomiale B(n, p)
    return ..........................

def seuil():
    k = 0
    probacum = binompdf(100, 0.04, 0)
    while probacum ..... :
        k = k + 1
        probacum = .................
    return k
```

Table des matières