

# Combinatoire du Puissance 4

| Objectif : Implémentation du jeu + Étude combinatoire du jeu

---

## Membres du binômes

### VU Hoang Thuy Duong

[LinkedIn](#) | [Github](#)

- Numéro d'étudiante : 21110221
- Groupe : 1 (*Statistique et Informatique - LU3IN005*)
- Email : [vuhoang.thduong@gmail.com](mailto:vuhoang.thduong@gmail.com)

### Halimatou DIALLO

[LinkedIn](#) | [Github](#)

- Numéro d'étudiante : 21114613
- Groupe : 1 (*Statistique et Informatique - LU3IN005*)
- Email :

## Instructeur

### M. Maxime BOUTHORS

Adresse mail : [bouthors@isir.upmc.fr](mailto:bouthors@isir.upmc.fr)

---

## Introduction

---

## Modèle basique

### Structures utilisées

#### Class `Plateau`

- 1 tableau `self.board` de dimension `self.width`x`self.length` représentant le plateau de jeu
- 2 joueurs `self.player1` et `self.player2`
- 2 variables enregistrant la dimension du plateau de jeu `self.width` et `self.length`

## Class Player

- 1 étiquette `self.etiquette` représentant la couleur possédée par ce joueur
- 1 ensemble de coordonnées `self.coordinate` de chaque tour du joueur

## Fonctions de manipulation

Nom de fonction	Utilisation
<code>show()</code>	afficher le plateau en cours
<code>reset()</code>	réinitialiser le jeu
<code>has_won()</code>	tester la victoire d'un des deux joueurs
<code>play(x,joueur)</code>	placer un jeton dans la colonne <code>x</code> pour le joueur spécifié
<code>is_finished()</code>	tester si c'est la fin du jeu (plateau plein ou victoire d'un joueur)
<code>run(joueur1,joueur2)</code>	jouer une partie entre le joueur 1 et le joueur 2

```
# Import necessary libraries
import random
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Import necessary classes
from plateau import *
from player import *
from src import *

# Définir les 2 joueurs avec leurs propres étiquettes
# Symbole :
### 1 pour le premier joueur
### -1 pour le deuxième
Joueur1 = Player(1)
Joueur2 = Player(-1)

# DÉMARRAGE DU JEU
# Définir le Plateau de Puissance 4 sur laquelle le jeu démarre

# Dimension 8x8
print("PUISSANCE 4 - DIMENSION 8x8")
P = Plateau(8,8,Joueur1,Joueur2)
P.run(Joueur1,Joueur2)
P.show()
print();print()

# Dimension 7x8
print("PUISSANCE 4 - DIMENSION 7x8")
P = Plateau(7,8,Joueur1,Joueur2)
P.run(Joueur1,Joueur2)
P.show()
print();print()
```

```
# Dimension 8x9
print("PUISSANCE 4 - DIMENSION 8x9")
P = Plateau(8,9,Joueur1,Joueur2)
P.run(Joueur1,Joueur2)
P.show()
```

PUISSANCE 4 - DIMENSION 8x8

=====

Current state of the game

=====

```
[[ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. -1.  0.  0.  0.]
 [ 0. -1.  0.  1.  1.  1.  1. -1.]]
```

PUISSANCE 4 - DIMENSION 7x8

=====

Current state of the game

=====

```
[[ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [-1.  0.  1.  0.  0.  0.  0.  0.]
 [-1.  0. -1. -1.  0.  0.  0.  0.]
 [-1.  0.  1. -1.  0.  0.  0.  0.]
 [ 1.  0. -1.  1.  1.  1.  1.  0.]
 [ 1.  1. -1. -1. -1.  1.  1.  0.]]
```

PUISSANCE 4 - DIMENSION 8x9

=====

Current state of the game

=====

```
[[ 0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [-1.  0.  0.  0.  1.  0.  1.  0.  0.]
 [-1. -1. -1.  0.  1.  1.  1.  1.  0.]
 [ 1. -1.  1.  1. -1. -1. -1.  1. -1.]]
```

# Étude de la Distribution du nombre de coups avant une victoire

```
df = pd.read_csv("freq_moves_database.csv")
```

## Processus

Afin d'étudier la distribution du nombre de coups *avant une victoire lorsque les deux joueurs jouent aléatoirement en différenciant selon que ce soit le premier ou le deuxième joueur qui gagne*, nous régénérerons 1000 fois le moteur automatique du jeu.

Après avoir obtenu les résultats et passé par des traitements afin de raccourcir les opérations et de créer des bases de données nécessaires, nous trouvons bien que le nombre de coups avant une victoire suit la **Loi Binomiale**.

## Loi binomiale

En théorie des probabilités et en statistique, la loi binomiale modélise **la fréquence du nombre de succès obtenus lors de la répétition de plusieurs expériences aléatoires identiques et indépendantes**.

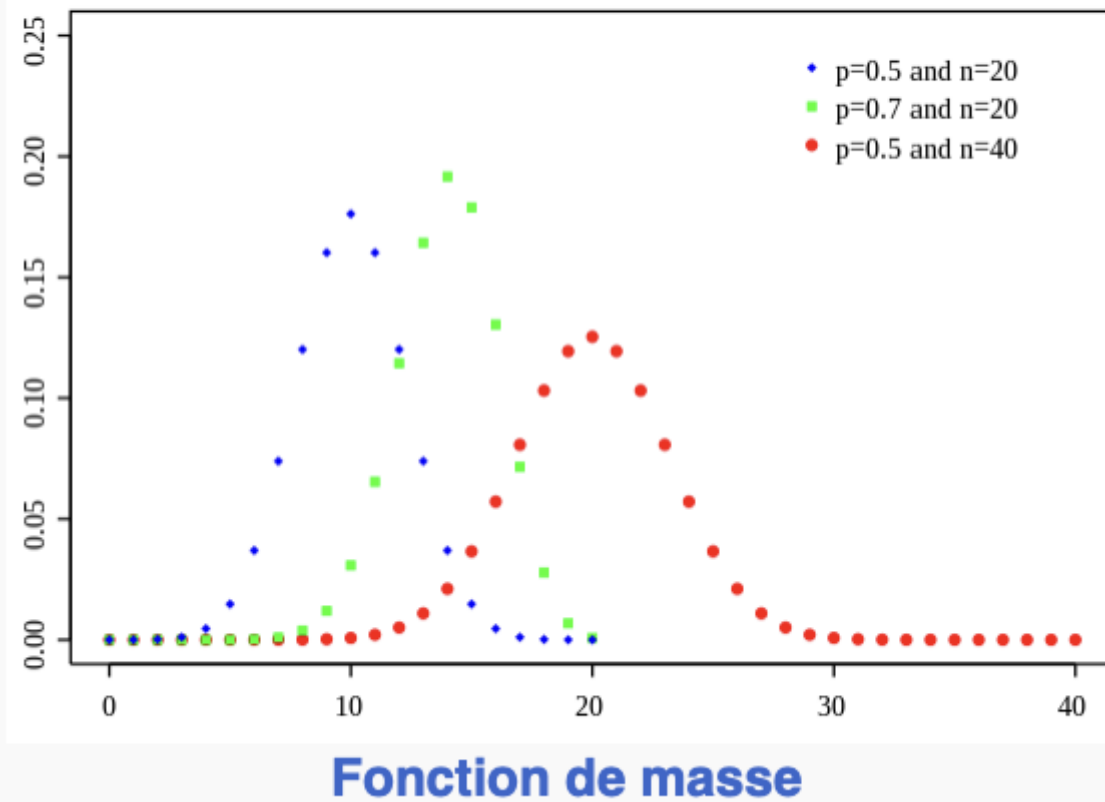
Plus mathématiquement, la loi binomiale est une loi de probabilité discrète décrite par deux paramètres :  $n$  le nombre d'expériences réalisées, et  $p$  la probabilité de succès. Pour chaque expérience appelée épreuve de Bernoulli, on utilise une variable aléatoire qui prend la valeur 1 lors d'un succès et la valeur 0 sinon. La variable aléatoire, somme de toutes ces variables aléatoires, compte le nombre de succès et suit une loi binomiale. Il est alors possible d'obtenir la probabilité de  $k$  succès dans une répétition de  $n$  expériences :

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Cette formule fait intervenir le coefficient binomial  $\binom{n}{k}$  duquel provient le nom de la loi

La loi binomiale est utilisée dans divers domaines d'étude, notamment à travers des tests statistiques qui permettent d'interpréter des données et de prendre des décisions dans des situations dépendant de l'aléa. De par la simplicité de sa définition, c'est l'une des lois de probabilité étudiées dans les cours d'introduction à la théorie des probabilités.

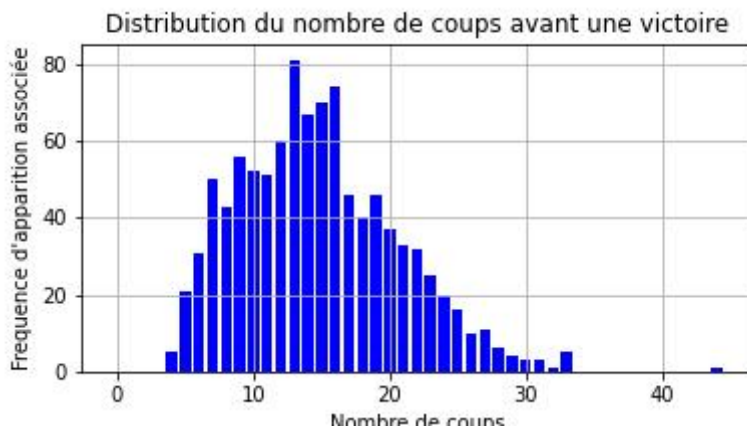
# Loi binomiale



## Explication

On cherche maintenant à modéliser la fréquence du nombre de coups avant une victoire lors de la répétition de 1000 expériences aléatoires identiques et indépendantes sur un plateau de jeu. Dans le cadre du jeu, cette fréquence suit donc une distribution binomiale avec  $n = 15 \times 15$  et  $k = 4$ .

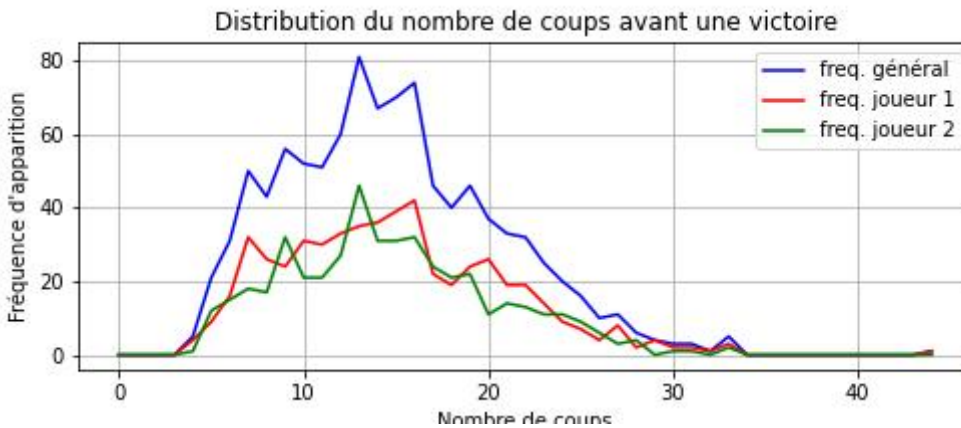
L'étude de la distribution du nombre de coups avant une victoire renvoie un graphique prenant la forme ci-dessous :



Dans ce cas-là, le nombre de coups qui possède la fréquence d'apparition la plus grande est 13.

En classifiant les fréquences selon la victoire de chaque joueur, on obtient le graphe ci-dessous, dont :

- la ligne bleue représente la fréquence des nombres de coups en général
- celle en rouge pour la fréquence des nombres de coups par le premier joueur
- et celle en verte pour l'autre



En observant le graphe, on trouve que le nombre de coups qui possède la fréquence d'apparition la plus grande est 16 pour le premier joueur, et 13 pour l'autre.

## Note

Pour accéder au code plus concrète, les fichiers sources peuvent être retrouvés dans le répertoire `side_code`.