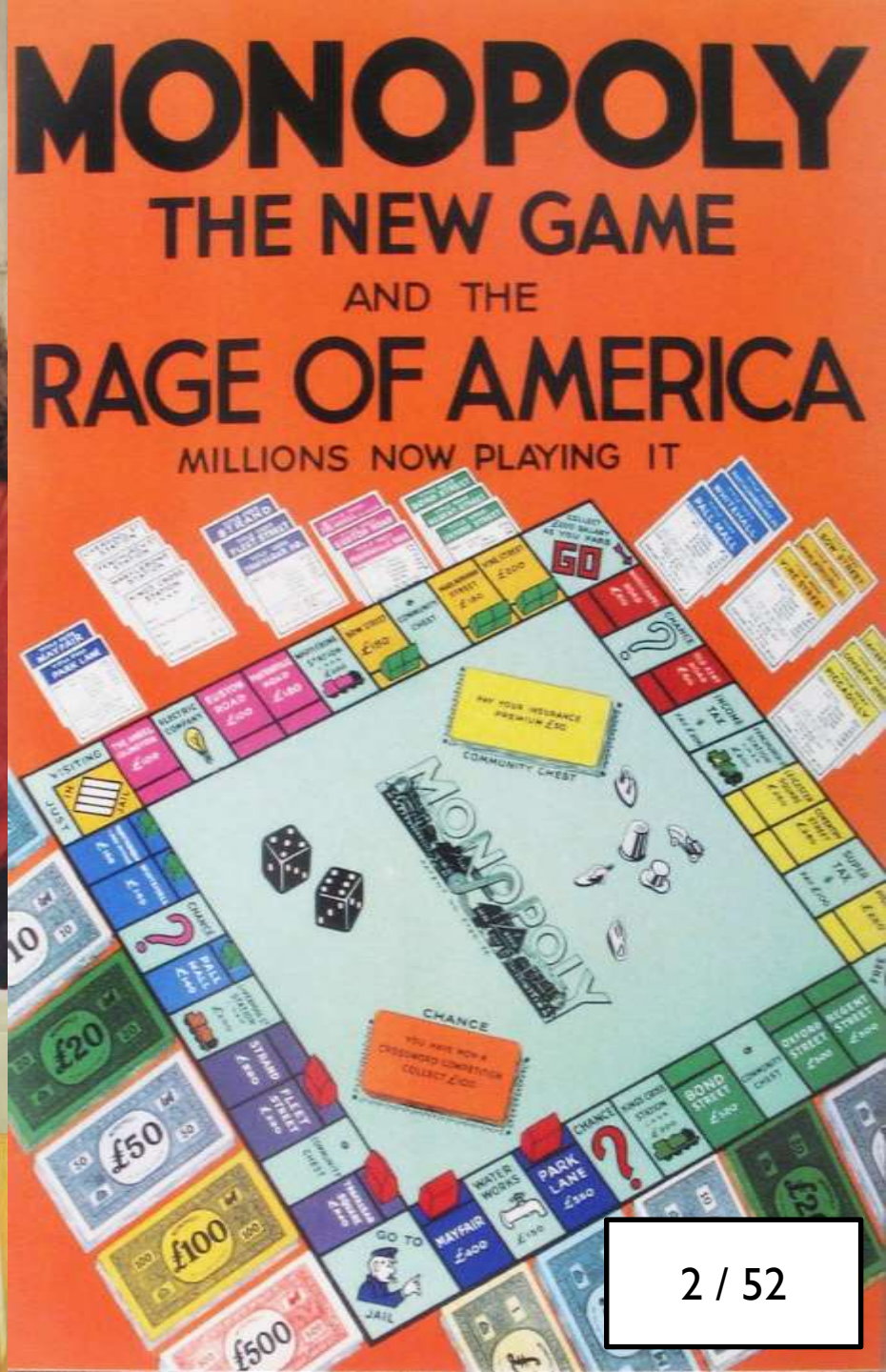
(https://www.pinterest.fr/)

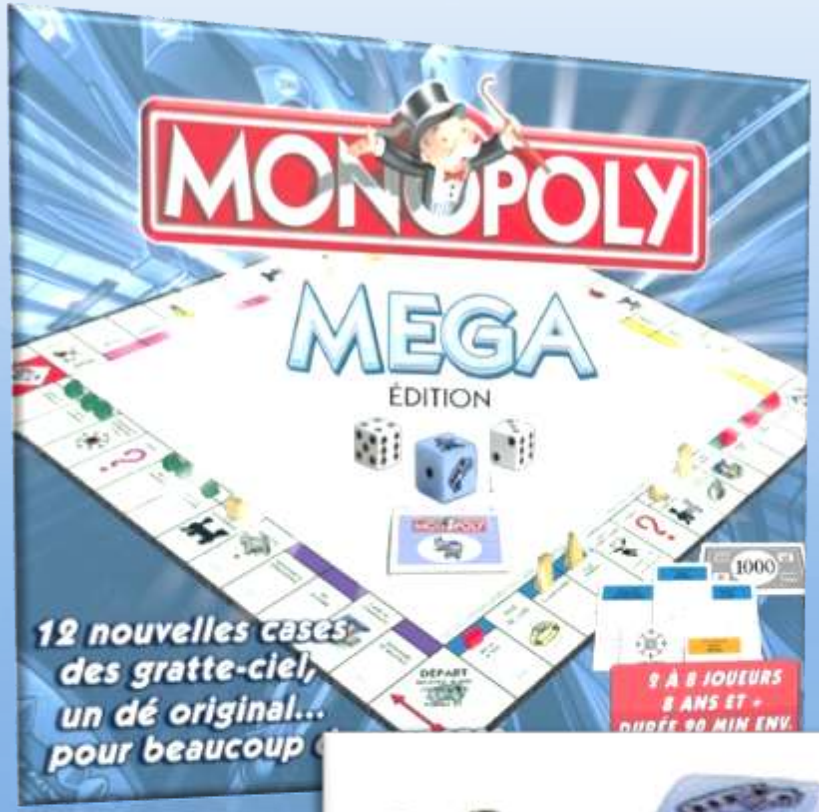
Monopoly sous l'œil de Markov:

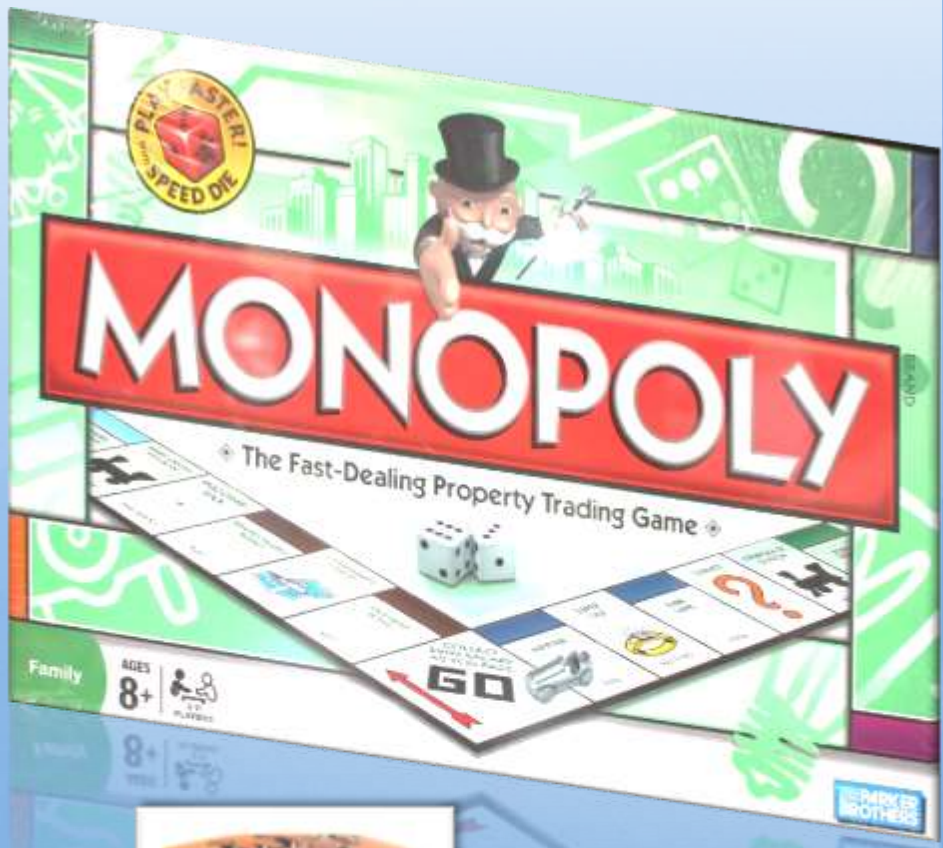
Dé Rapide et probabilité d'atterrissages

ABISOUROUR BASSMA
n° 18359

Affiche Pub 1960 de la bière SCHLITZ / 1940 (The Strong Art Musem of Play, US)







https://www.monopolypedia.fr/

Plan de la Présentation

1- La Théorie de Markov

2- Construction des Matrices

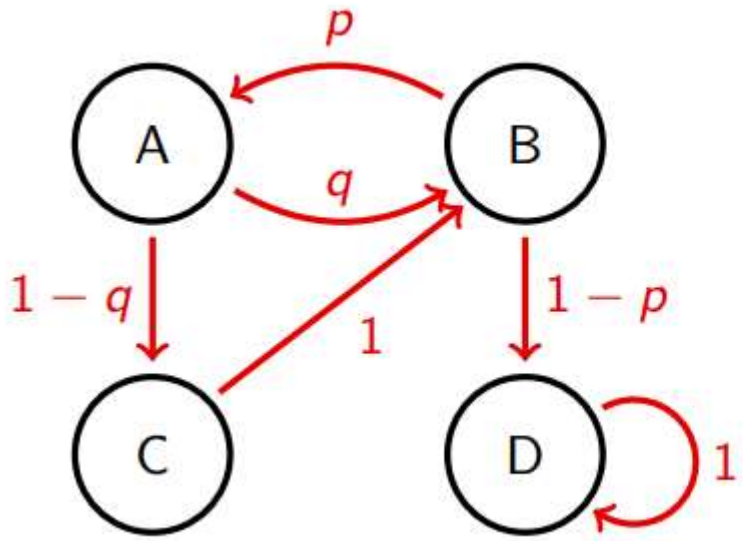
3- Comparaison des Résultats
(Classique Vs Rapide)

4- Limites d'étude & Conclusion

5- Annexe

1. LA THÉORIE DE MARKOV

Chaîne de Markov ***Finie Homogène***



$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & p & 0 & 0 \\ q & 0 & 1 & 0 \\ 1-q & 0 & 0 & 0 \\ 0 & 1-p & 0 & 1 \end{pmatrix} \end{array}$$

Adaptée de Monopoly- An Analysis using Markov Chain – Benjamin Bernard

Un exemple: une chaîne de Markov (ici homogène finie) effectue des transitions de manière aléatoire entre plusieurs états possibles A, B, C et D :

1.1. Définition. Une chaîne de Markov est une suite de variables aléatoires $(X_n)_n$ à valeurs dans l'espace d'états E telle que, pour tout $n \in \mathbb{N}$, et tous $y, y_0, \ldots, y_n \in E$,

$$\mathbb{P}(X_{n+1} = y | X_0 = y_0, \ldots, X_n = y_n) = \mathbb{P}(X_{n+1} = y | X_n = y_n).$$

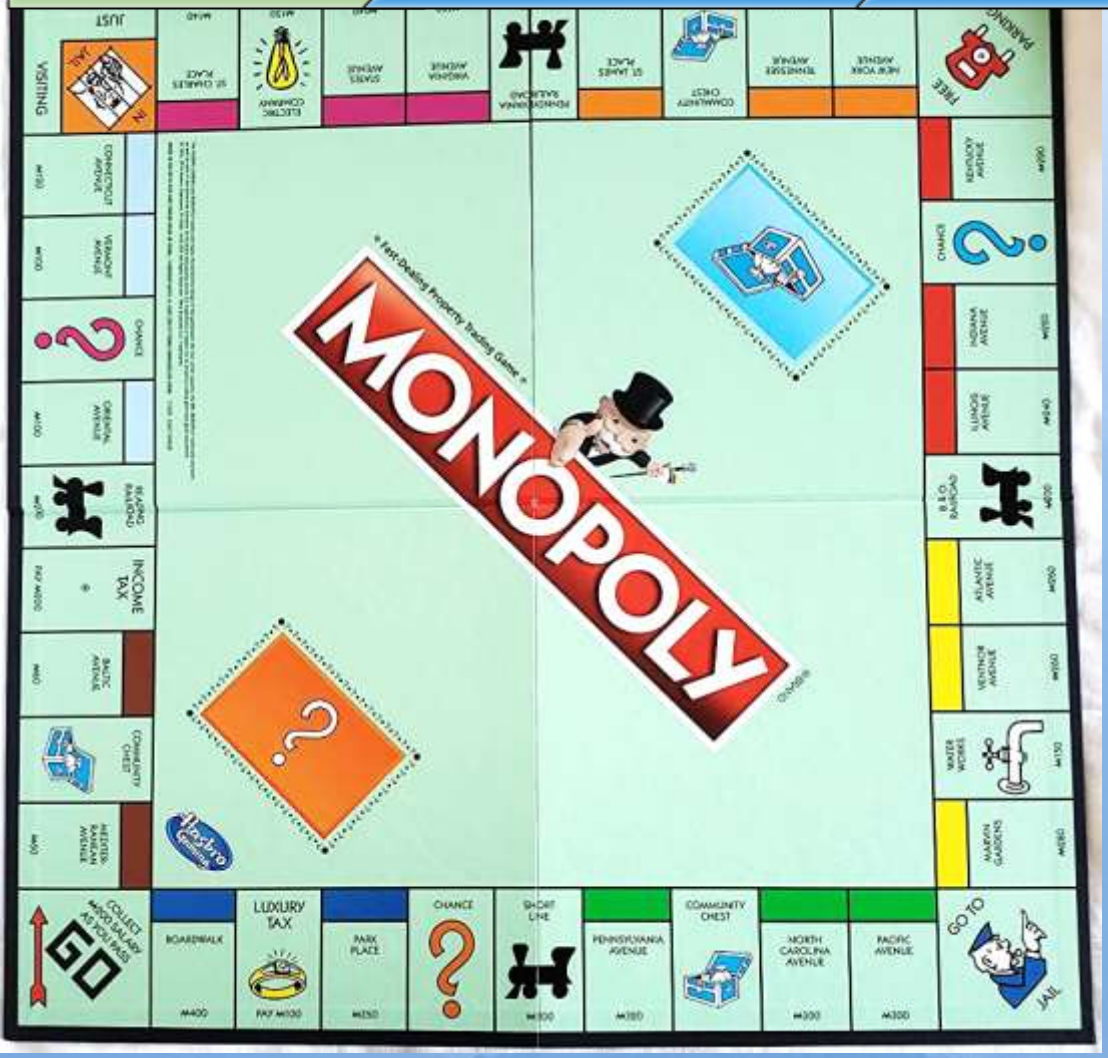
Elle est homogène si cette quantité ne dépend pas de n. Elle est finie si l'ensemble E est fini.

2.1. Définition. La matrice de transition d'une chaîne de Markov homogène finie $(X_n)_n$ d'espace d'états $E = \{e_1, \ldots, e_N\}$ est la matrice d'ordre N dont le coefficient en position (i, j) est

$$\mathbb{P}(X_1 = e_j | X_0 = e_i).$$

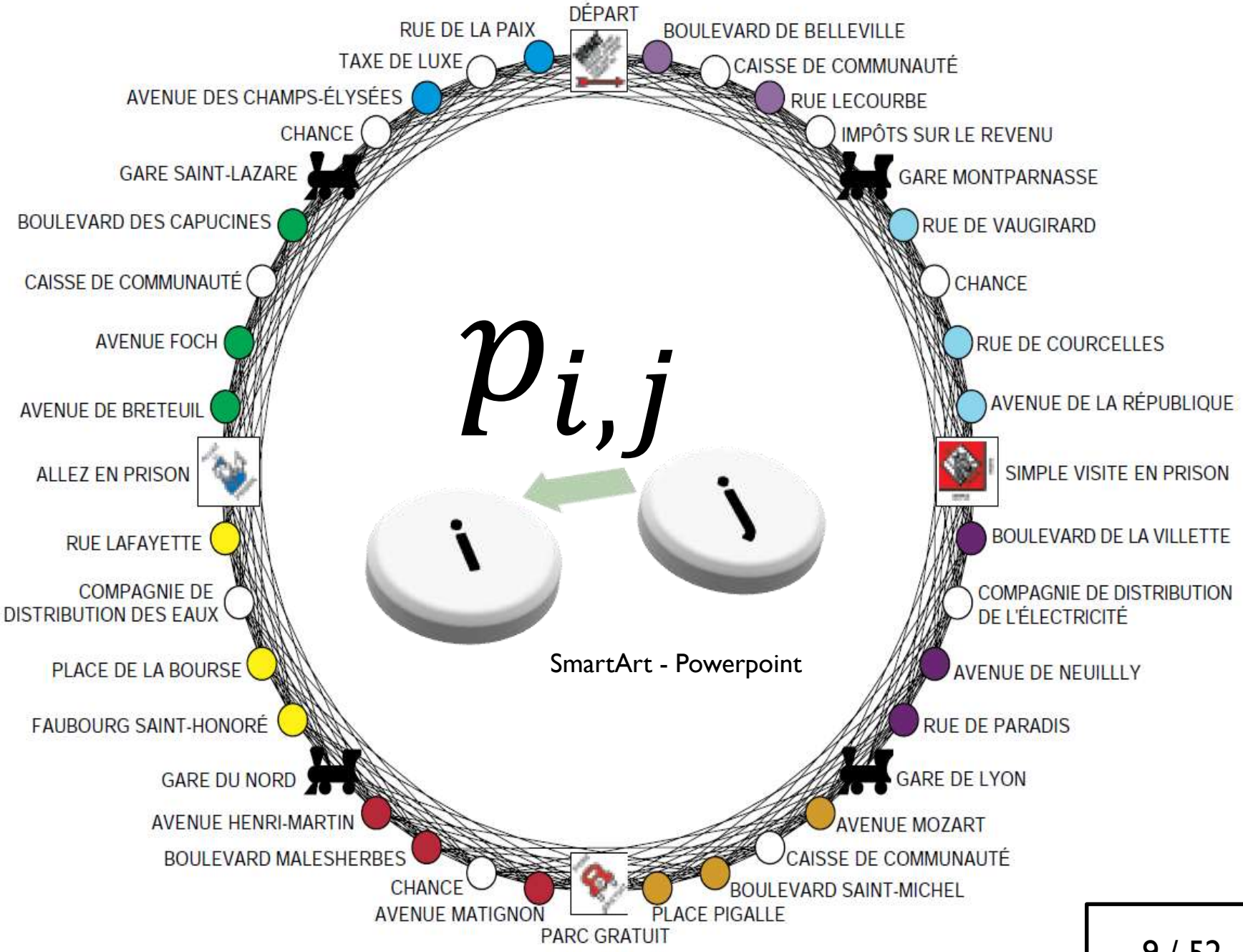
(Algèbre linéaire. Réduction des endomorphismes - Roger Mansuy, Rached Mneimné)

Présentation du jeu:

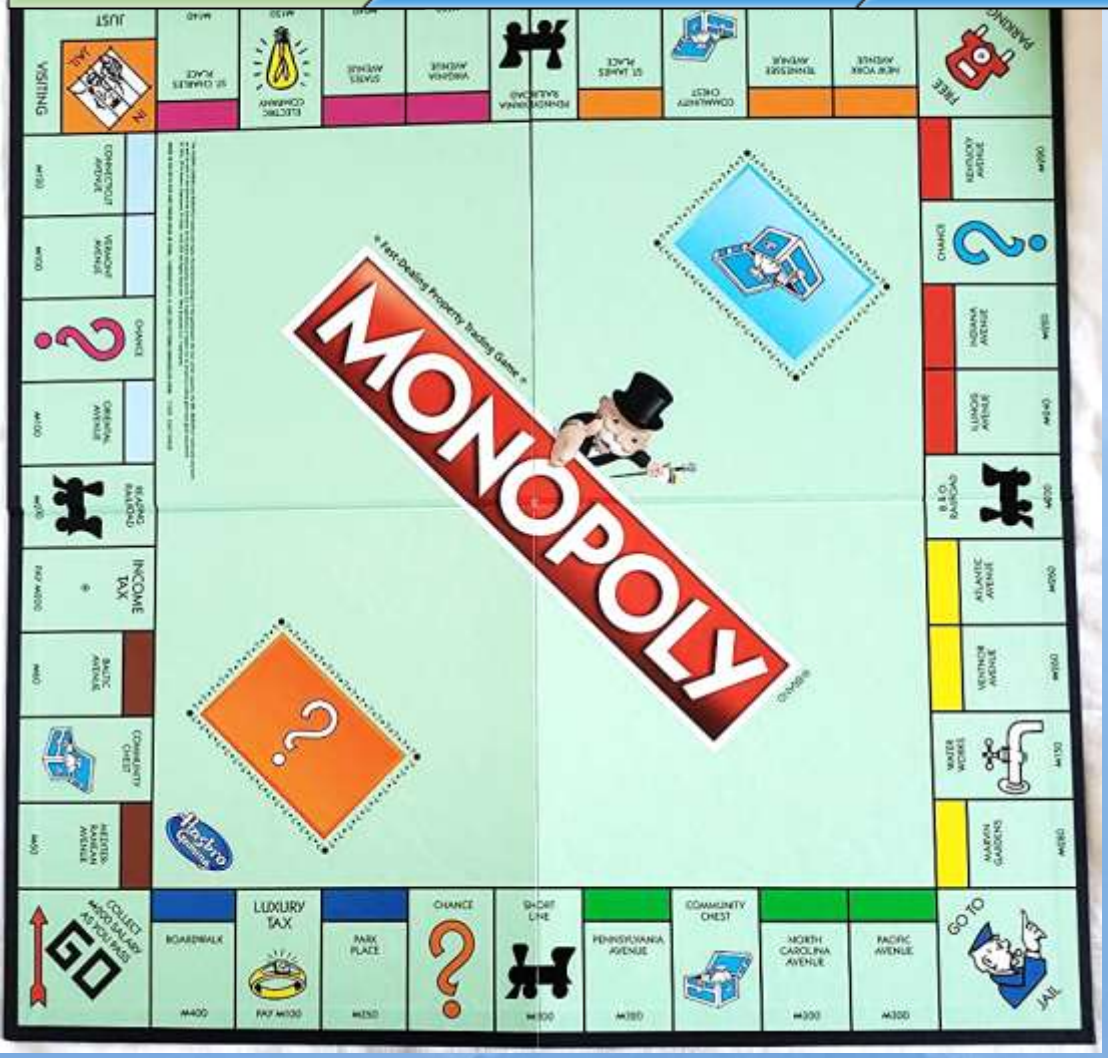


Plateau du jeu Monopoly
(https://www.ebay.com/)

Dés du jeu entourés de Pions
(https://www.ebay.com/)



Adaptée de: Représentation Mathématique d'un plateau du jeu (Trop Monopoly pour être honnête? I

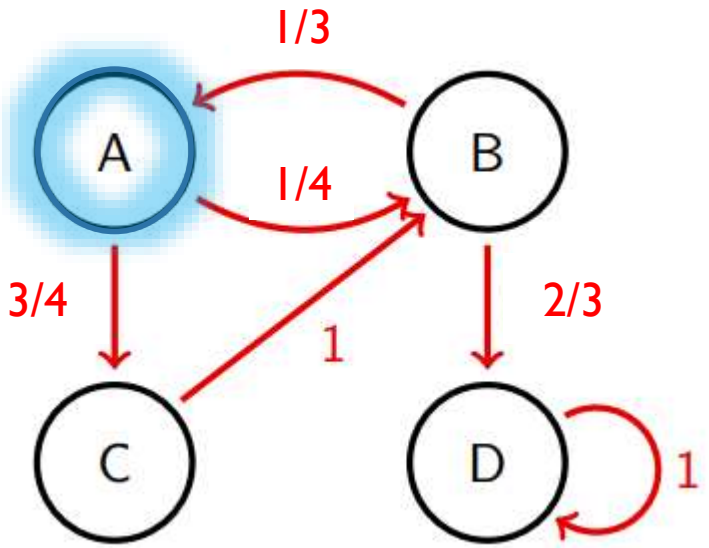


Plateau du jeu Monopoly
(https://www.ebay.com/)

40 Cases

Matrice 40x40

La Théorie de Markov



Adaptée de Monopoly- An Analysis using Markov Chain – Benjamin Bernard

$$P = \begin{bmatrix} 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{4} & 0 & 1 & 0 \\ \frac{3}{4} & 0 & 0 & 0 \\ 0 & \frac{2}{3} & 0 & 1 \end{bmatrix}$$

Desmos (https://www.desmos.com/matrix)

$$\nu = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix} \quad P \cdot \nu = \begin{bmatrix} 0 \\ \frac{1}{4} \\ \frac{3}{4} \\ 0 \end{bmatrix} \quad P^2 \cdot \nu = \begin{bmatrix} \frac{1}{12} \\ \frac{3}{4} \\ 0 \\ \frac{1}{6} \end{bmatrix} \quad P^3 \cdot \nu = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{48} \\ \frac{1}{16} \\ \frac{2}{3} \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Desmos (https://www.desmos.com/matrix)

2.8. Proposition. *Soit $(X_n)_n$ une chaîne de Markov homogène finie d'espace d'états $E = \{e_1, \ldots, e_N\}$, de matrice de transition P. Notons, pour tout $n \in \mathbb{N}$, la loi de X_n avec la matrice ligne*

$$\nu_n = \begin{pmatrix} \mathbb{P}(X_n = e_1) & \mathbb{P}(X_n = e_2) & \ldots & \mathbb{P}(X_n = e_N) \end{pmatrix}.$$

Alors, pour tout $n \in \mathbb{N}$, $\nu_{n+1} = \nu_n P$

2.9. Corollaire. *Avec les mêmes notations,* $\nu_n = \nu_0 P^n$

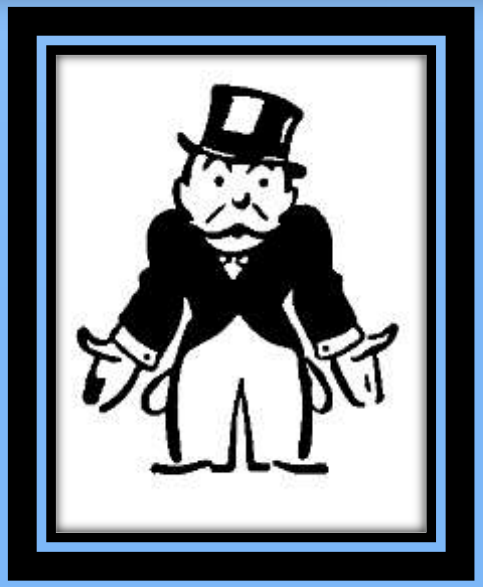
(Algèbre linéaire. Réduction des endomorphismes - Roger Mansuy, Rached Mneimné)

$$\begin{bmatrix} 1 \\ 0 \\ \ldots \\ 0 \end{bmatrix}$$

Desmos (https://www.desmos.com/matrix)

Existe-t-il une probabilité long-terme?

$$\nu_\infty = \lim_{n \to \infty} P^n \cdot \nu_0$$



2. CONSTRUCTION DE LA MATRICE DE TRANSITION

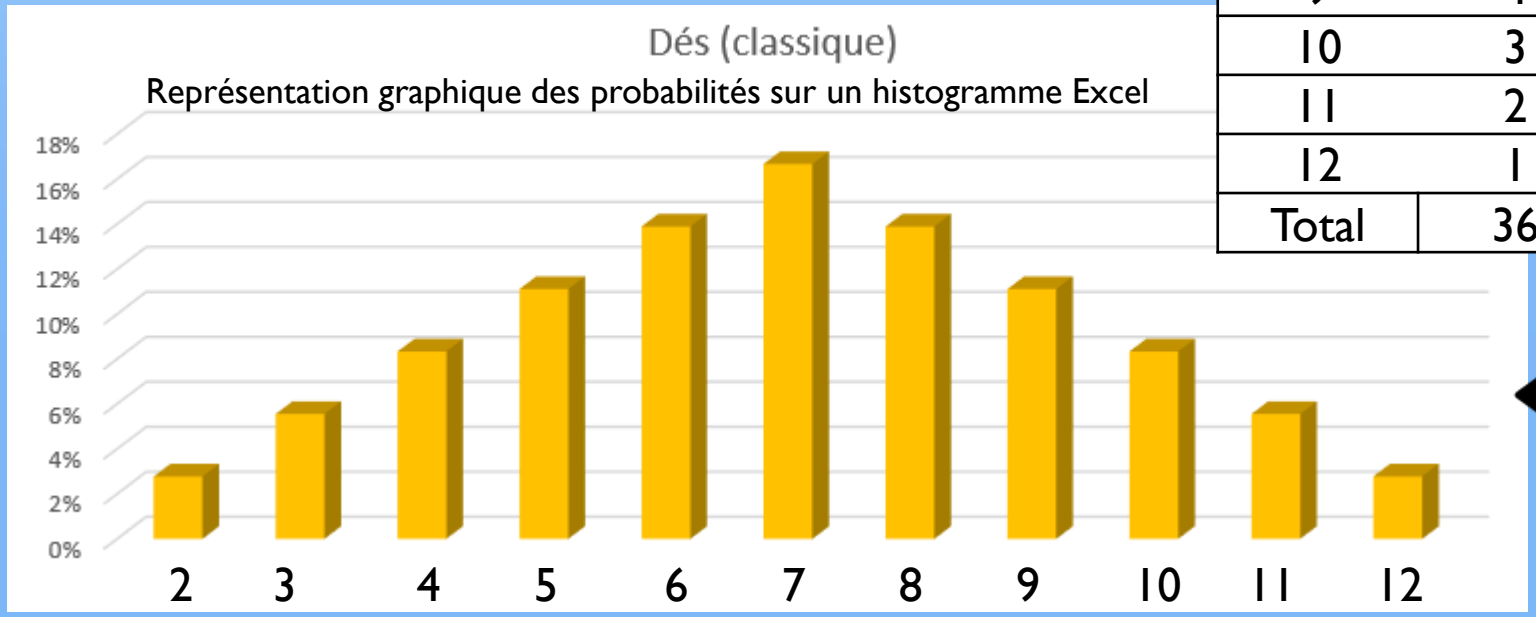
- *Matrice Lancer de Dé (Classique) / **(Rapide)***
- *Matrice Règles du jeu*

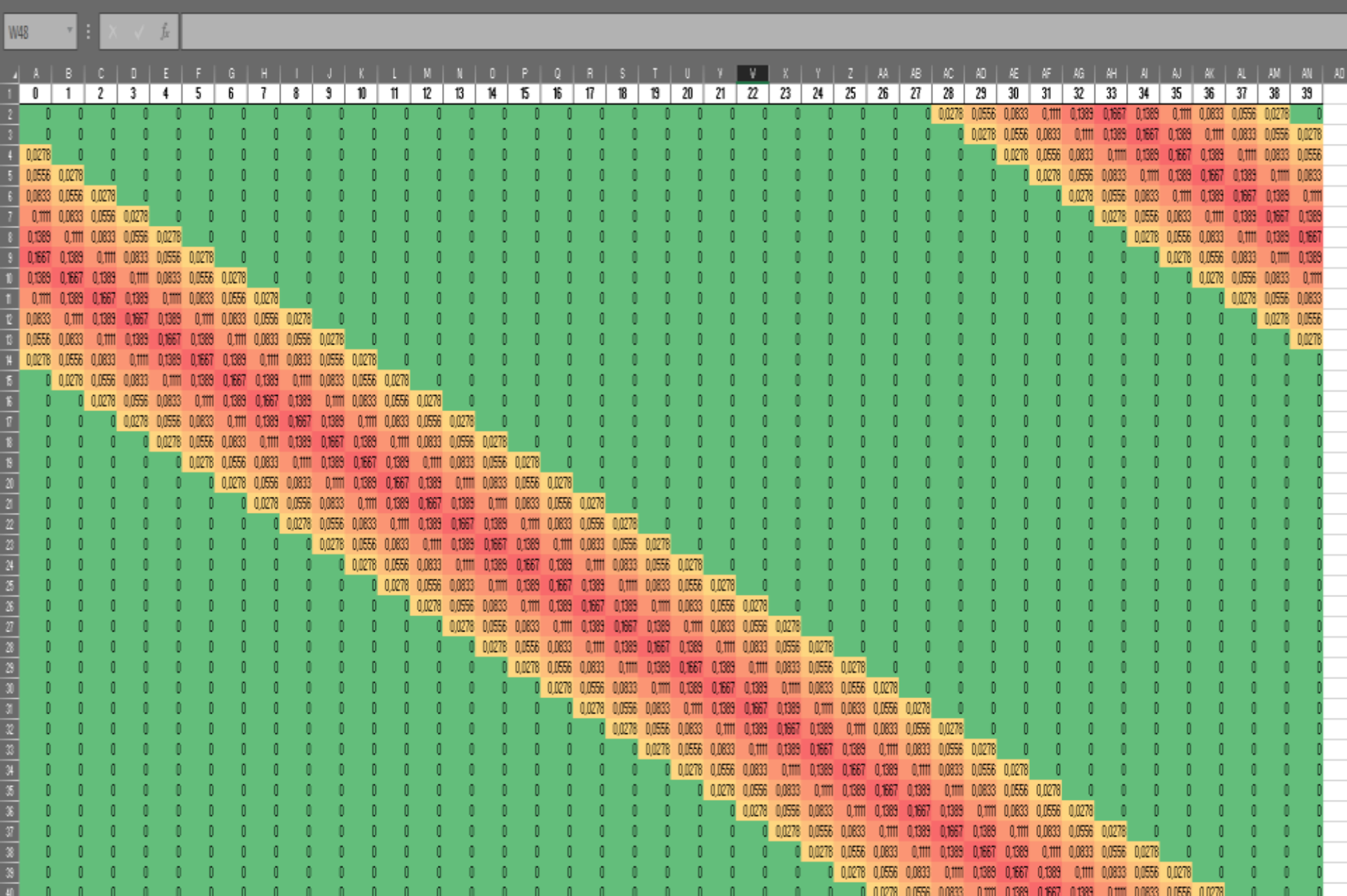
→ ***Matrice de transition finale***

2. Construction des Matrice (Matrice de lancer de dés Classique)

Dés du jeu Monopoly (https://www.laboiteauxpieces.fr/)

Résultat	Fréquence	Probabilité
2	1	1/36~ 3%
3	2	2/36~ 6%
4	3	3/36~ 8%
5	4	4/36~11%
6	5	5/36~14%
7	6	6/36~17%
8	5	5/36~14%
9	4	4/36~11%
10	3	3/36~8%
11	2	2/36~6%
12	1	1/36~ 3%
Total	36	





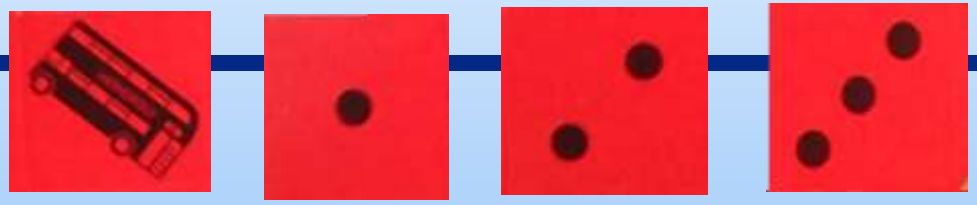
Représentation (mise en forme conditionelle) de la matrice de lancer de dé classique sur Excel

Dés du jeu (https://www.ebay.com/)

Faces du dé rapide (https://richunclepennybags.co.uk//)

- Les additionner
- Choix libre
- Prédéfini → **Les DEUX**
- Jouer puis visiter la prochaine Propriété non achetée
- Indépendance du joueur → **Prochaine propriété**

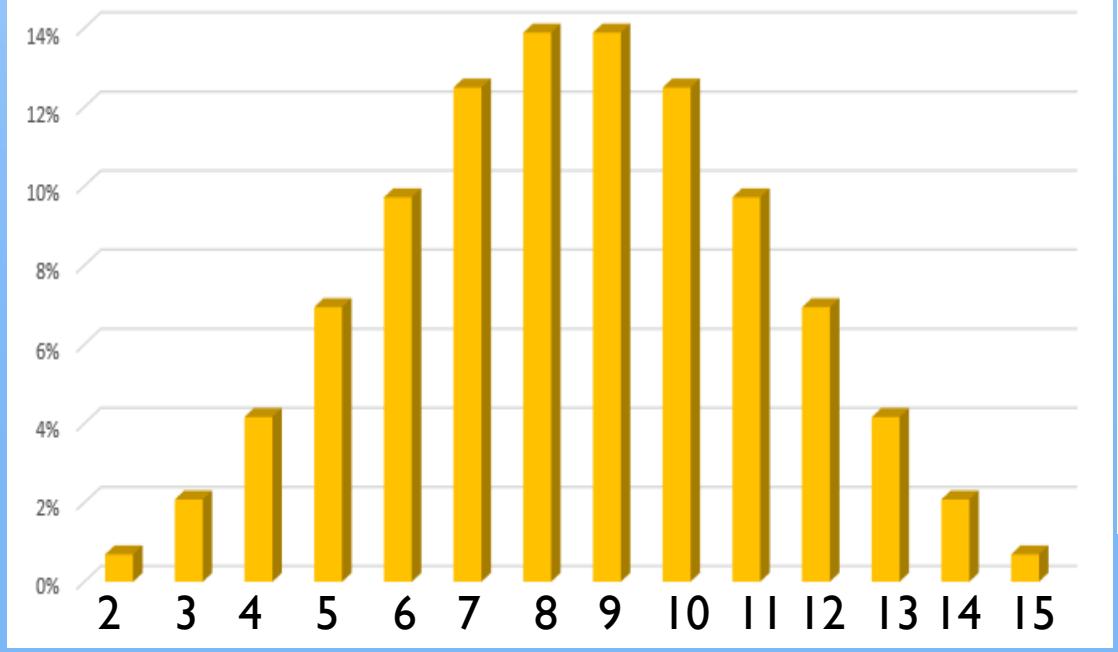
2. Construction des Matrice (Matrice de lancer de dés Rapide)



+0 +1 +2 +3

Faces du dé rapide (https://richunclepennybags.co.uk//)

Représentation graphique des probabilités sur un histogramme Excel



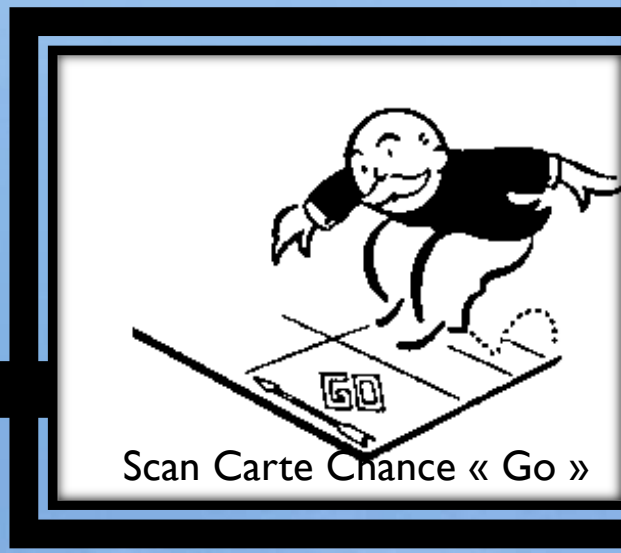
Résultat	Fréq	Proba
2	1	1%
3	3	2%
4	6	4%
5	10	7%
6	14	10%
7	18	13%
8	20	14%
9	20	14%
10	18	13%
11	14	10%
12	10	7%
13	6	4%
14	3	2%
15	1	1%
Total	144	

4/6 ≈67%

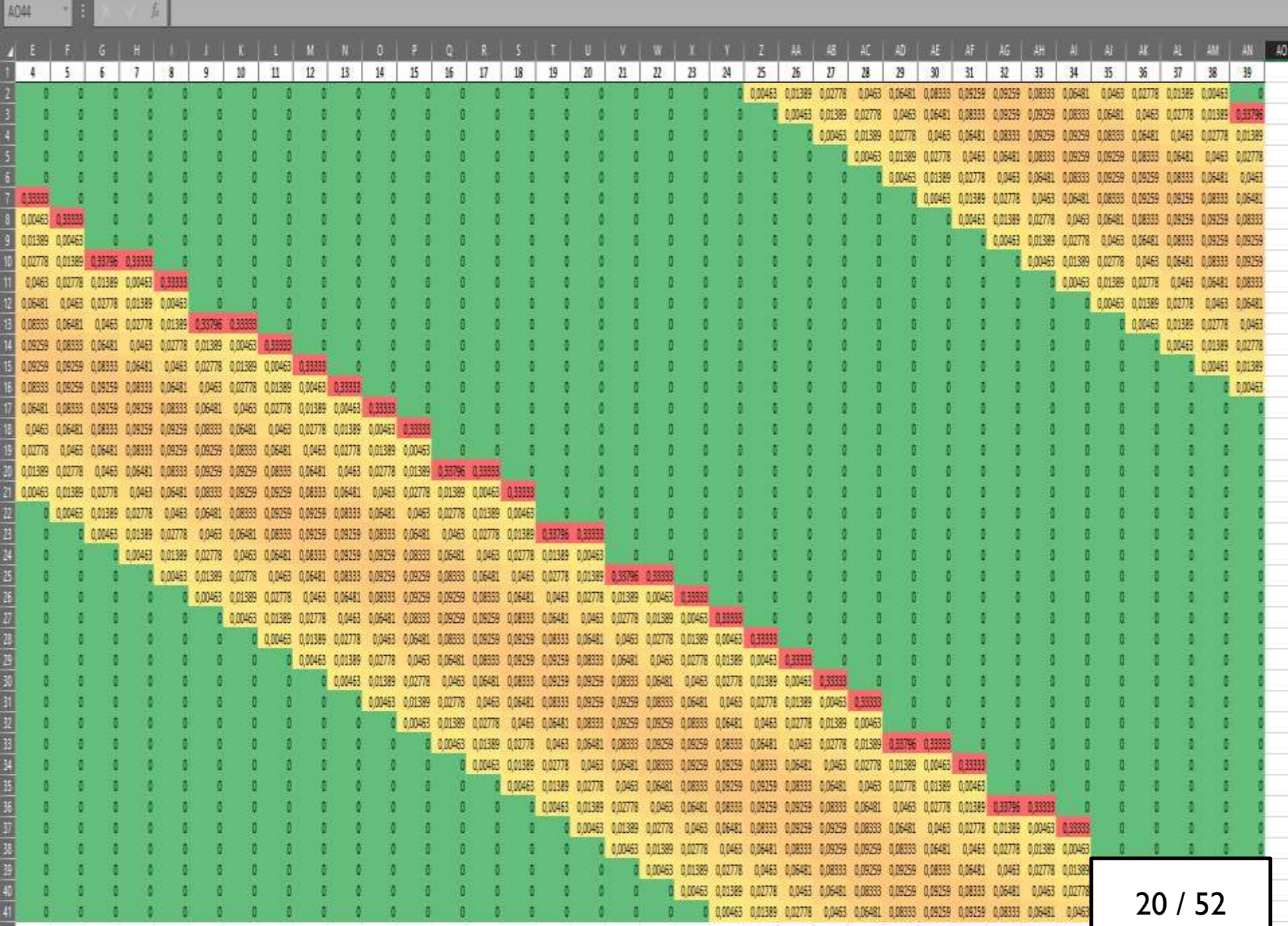
Mr Monopoly nous mène vers la prochaine propriété !

2/6

Faces du dé rapide (https://richunclepennybags.co.uk//)



Scan Carte Chance « Go »



Représentation (mise en forme conditionelle) de la matrice de lancer de dé rapide sur Excel

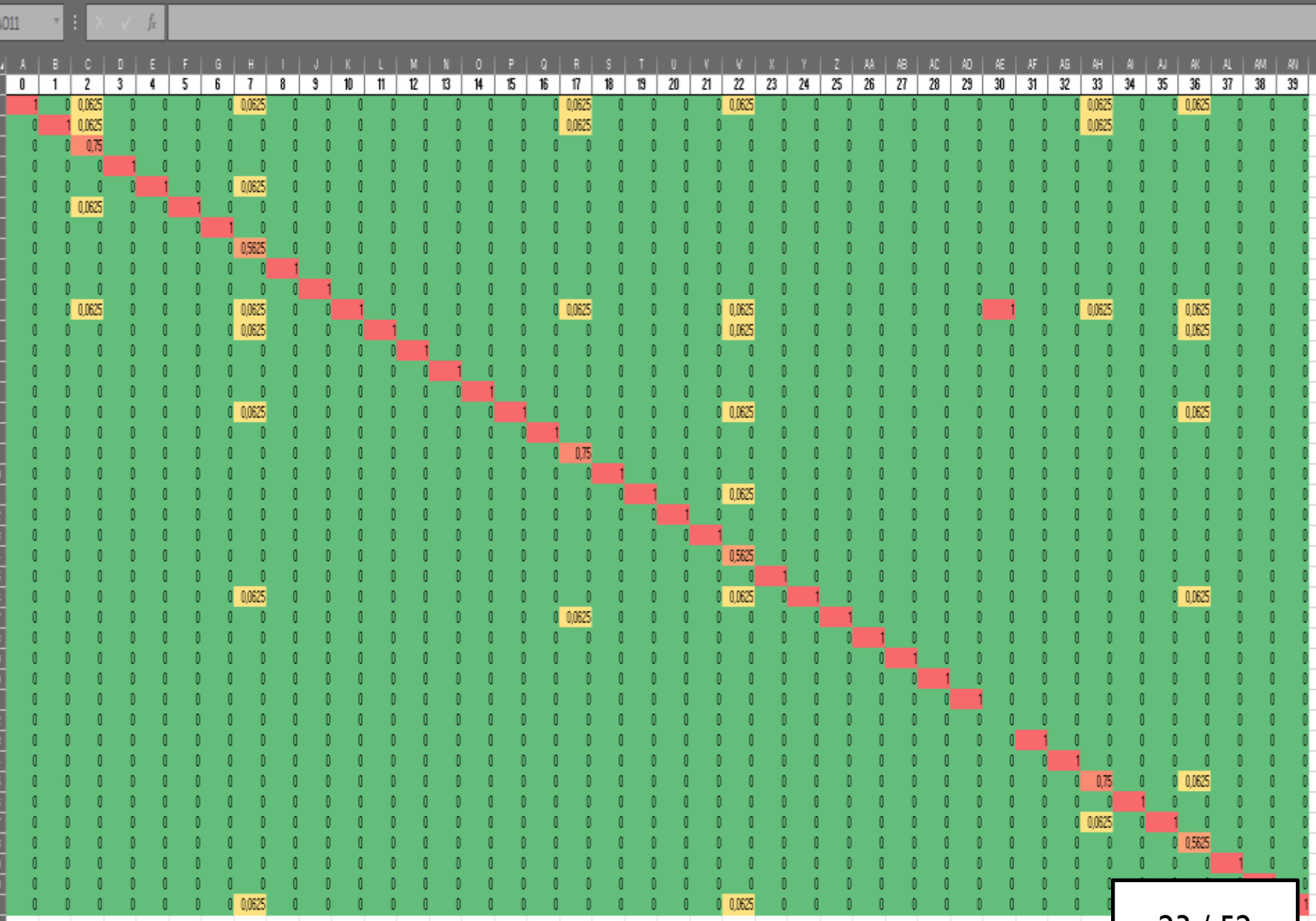
	A	B	C
1	0	1	2
2	0	0	0
3	0	0	0
4	0,02778	0	0
5	0,05556	0,02778	0
6	0,08333	0,05556	0,02778
7	0,11111	0,08333	0,05556
8	0,13889	0,11111	0,08333
9	0,16667	0,13889	0,11111
10	0,13889	0,16667	0,13889
11	0,11111	0,13889	0,16667
12	0,08333	0,11111	0,13889
13	0,05556	0,08333	0,11111
14	0,02778	0,05556	0,08333
15	0	0,02778	0,05556
16	0	0	0,02778
17	0	0	0
18	0	0	0
19	0	0	0

	A	B	C
1	0	1	2
2	0	0	0
3	0,33333	0	0
4	0,00463	0	0
5	0,01389	0,33796	0,33333
6	0,02778	0,01389	0,00463
7	0,0463	0,02778	0,01389
8	0,06481	0,0463	0,02778
9	0,08333	0,06481	0,0463
10	0,09259	0,08333	0,06481
11	0,09259	0,09259	0,08333
12	0,08333	0,09259	0,09259
13	0,06481	0,08333	0,09259
14	0,0463	0,06481	0,08333
15	0,02778	0,0463	0,06481
16	0,01389	0,02778	0,0463
17	0,00463	0,01389	0,02778
18	0	0,004	9
19	0		

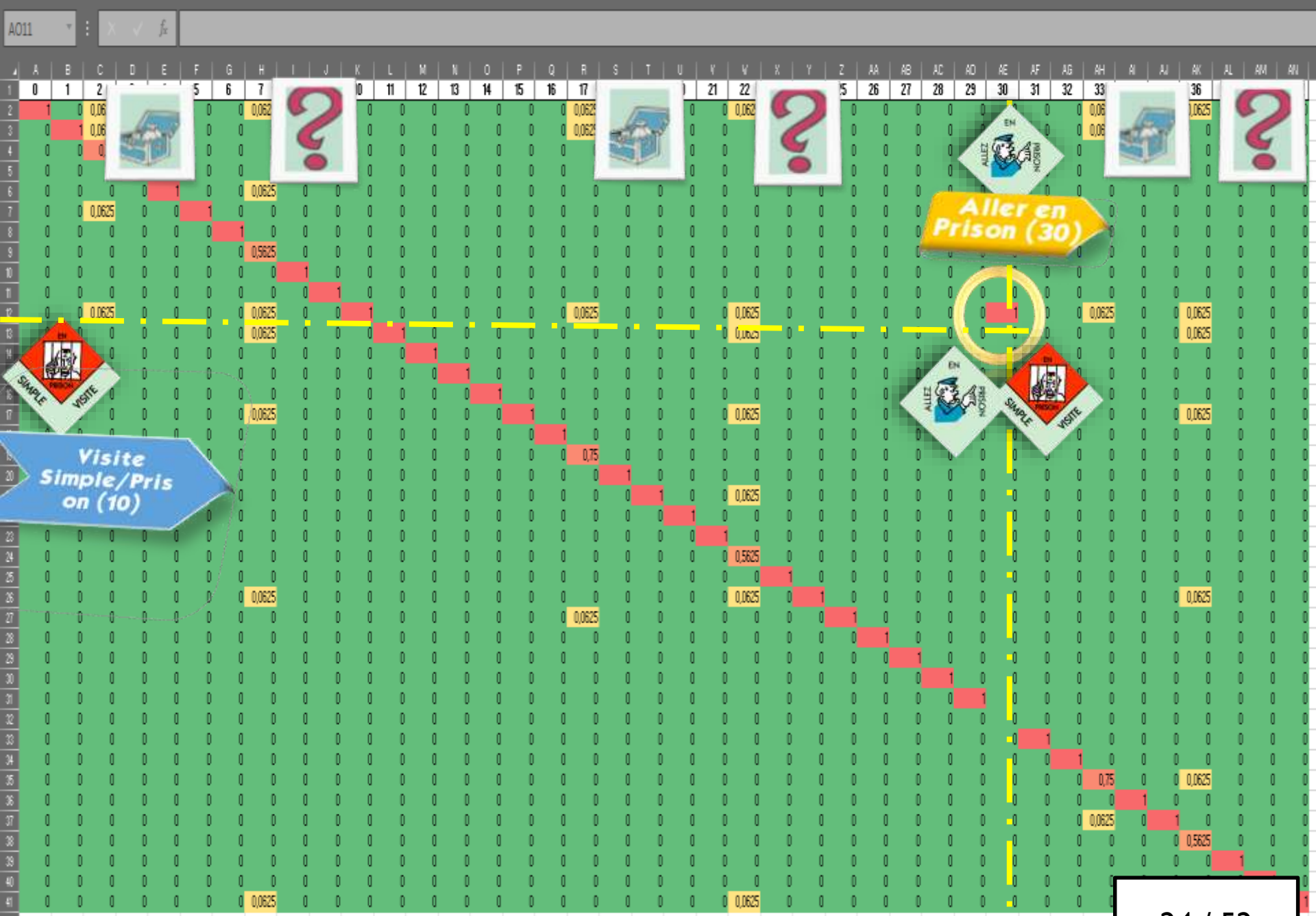
2. Construction des Matrice (Matrice des Règles du Jeu)

Plateau du jeu Monopoly (https://www.ebay.com/)

(https://:www.spriters-recource.com)



Représentation (mise en forme conditionnelle) de la matrice des règles du jeu sur Excel



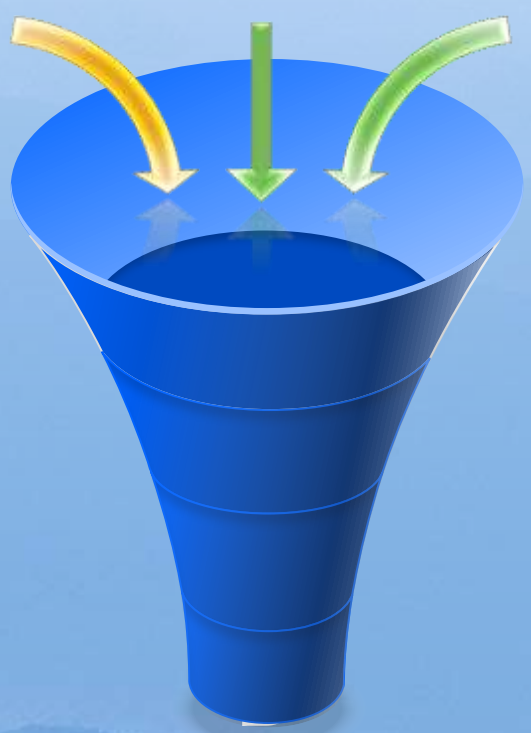
Représentation (mise en forme conditionnelle) de la matrice des règles du jeu sur Excel

Matrice Lancer de Dés (Classique)

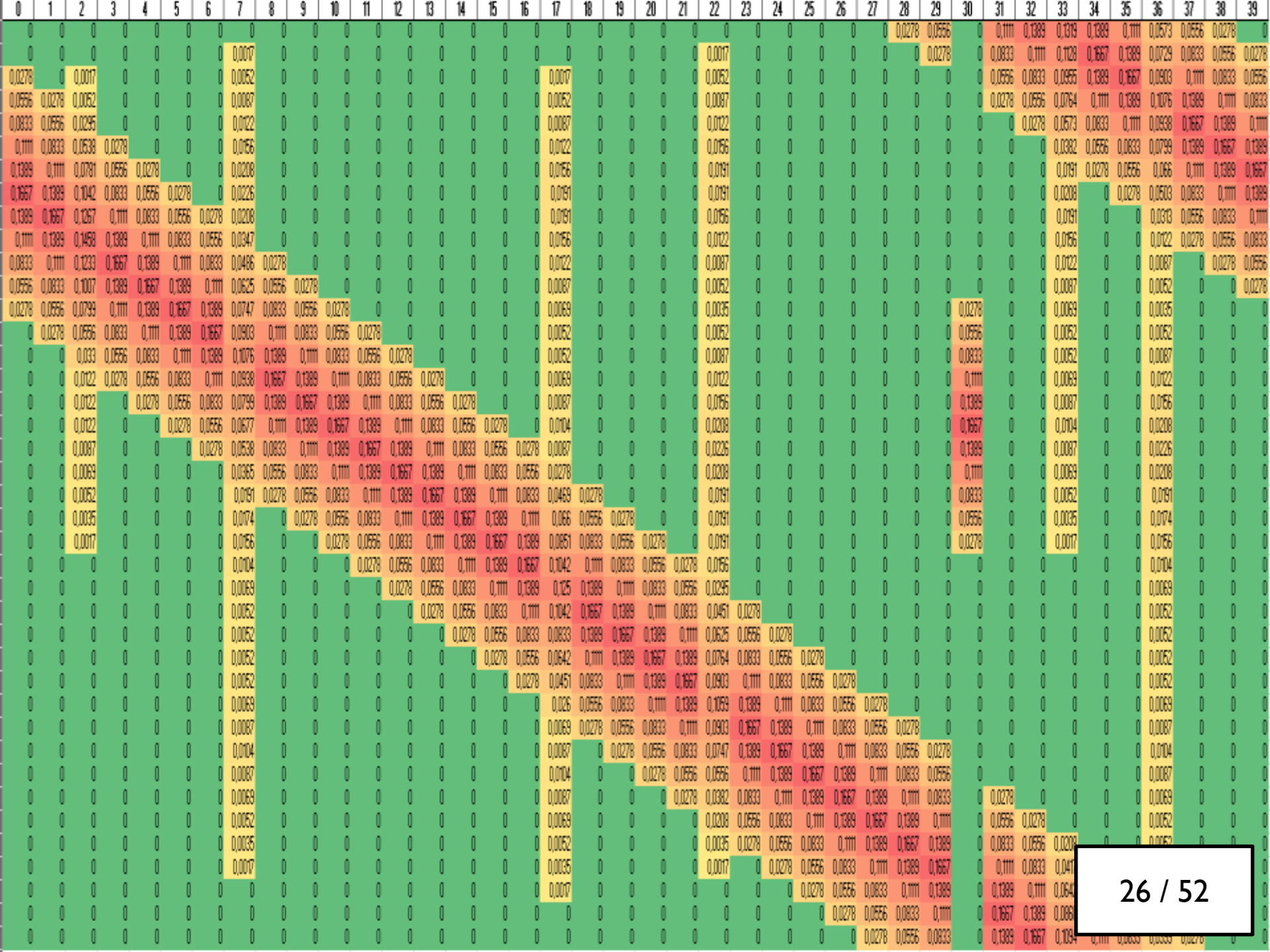
OU

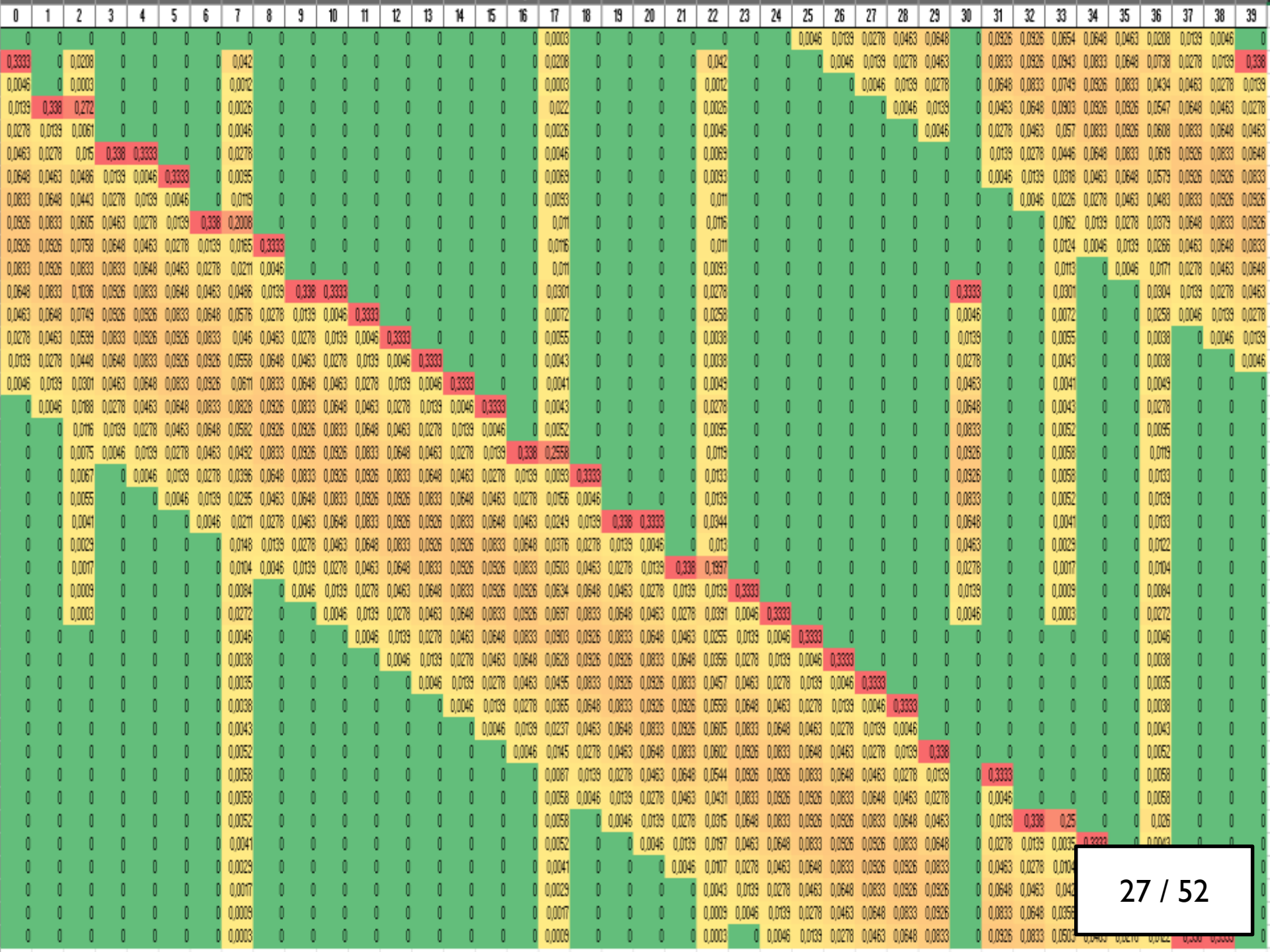
Matrice des Règles du jeu

Matrice Lancer de Dés (Rapide)



Matrice de Transition





4.2. Théorème. *Soit P une matrice ergodique de probabilité invariante ν. Alors, pour tout (i,j),*

$$\lim_n [P^n]_{i,j} = \nu_j.$$

3.5. Proposition. *Toute matrice P stochastique et irréductible admet une unique probabilité invariante ν. De plus, cette probabilité invariante est strictement positive.*

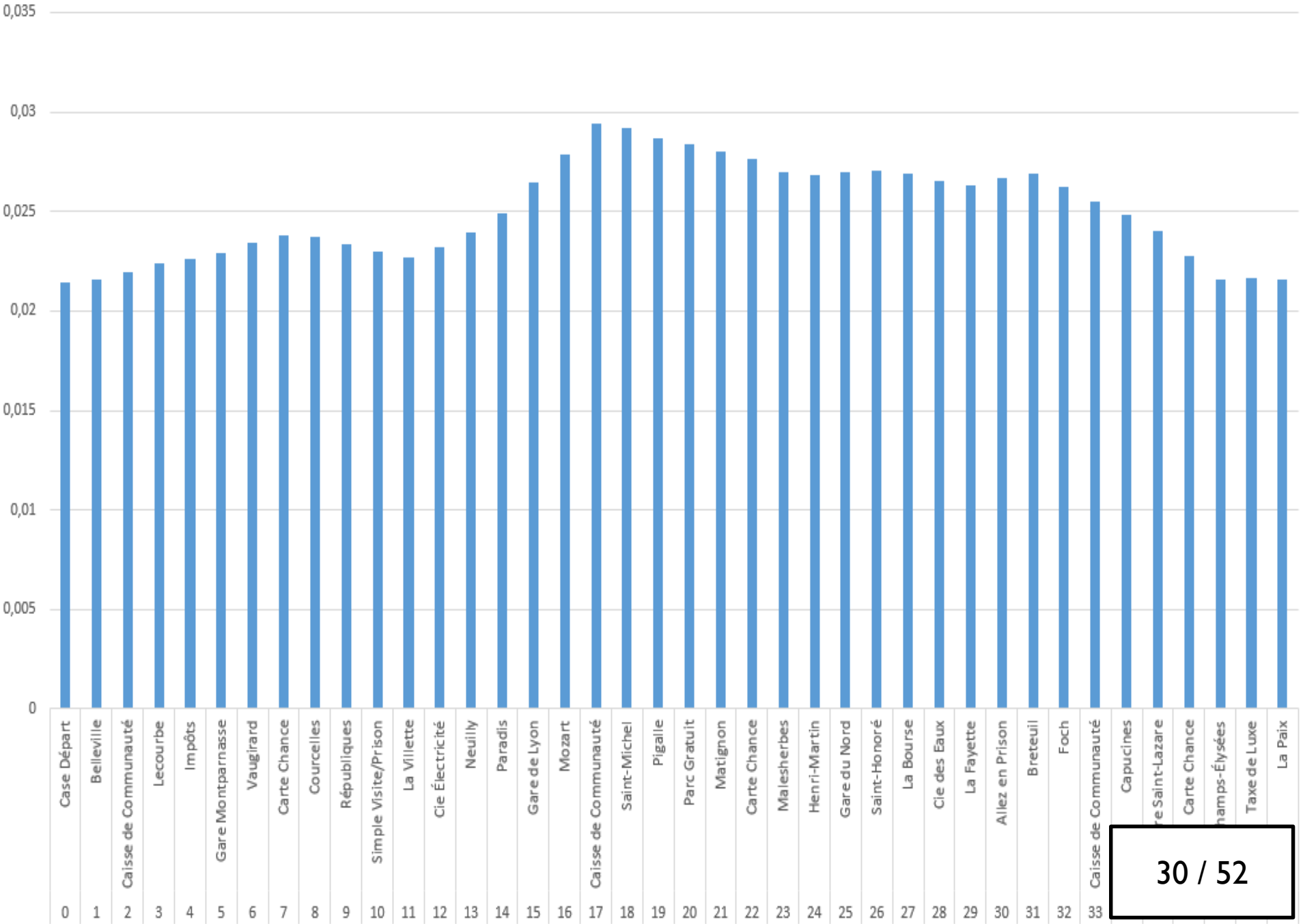
3.2. Définition. Une probabilité invariante d'une matrice stochastique P est un vecteur $\nu \in \mathcal{M}_{1,n}(\mathbb{R})$ positif et de somme 1 tel que $\nu A = \nu$.

(Adapté de Algèbre linéaire. Réduction des endomorphismes - Roger Mansuy, Rached Mneimné)

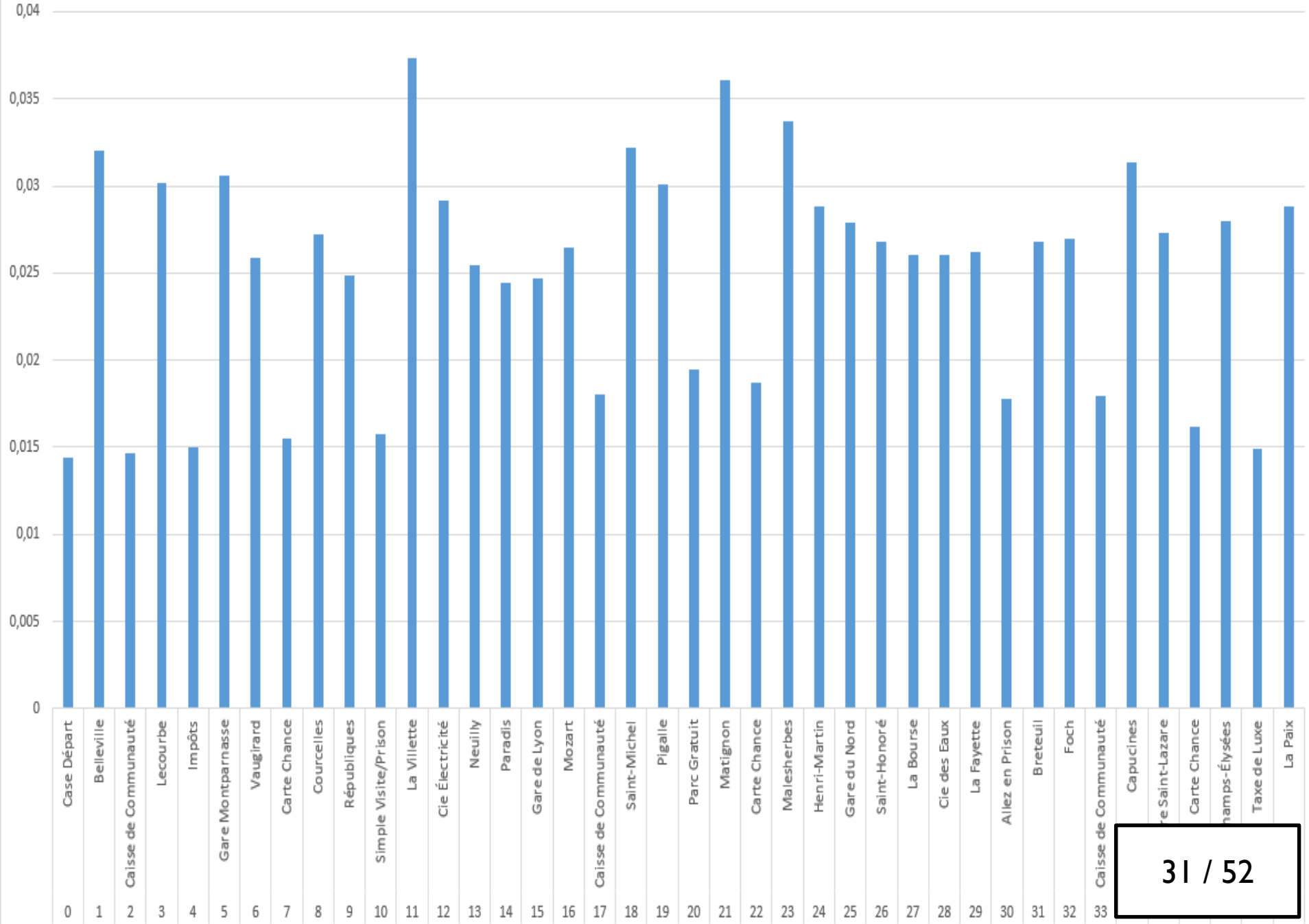
Vecteur Propre de valeur propre 1

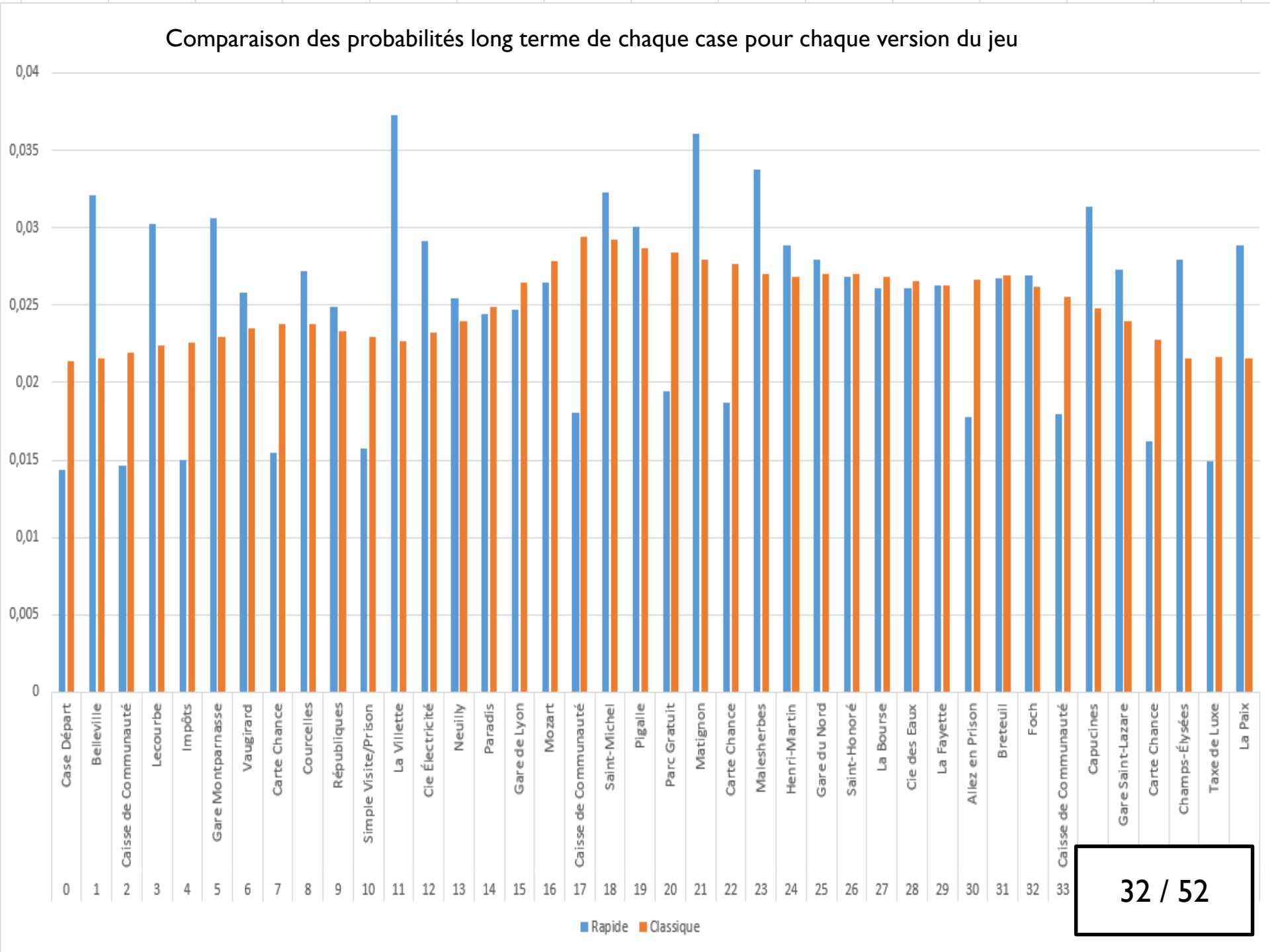
3. COMPARAISON DES RÉSULTATS

Représentation des probabilités long terme de chaque case dans une partie classique de Monopoly



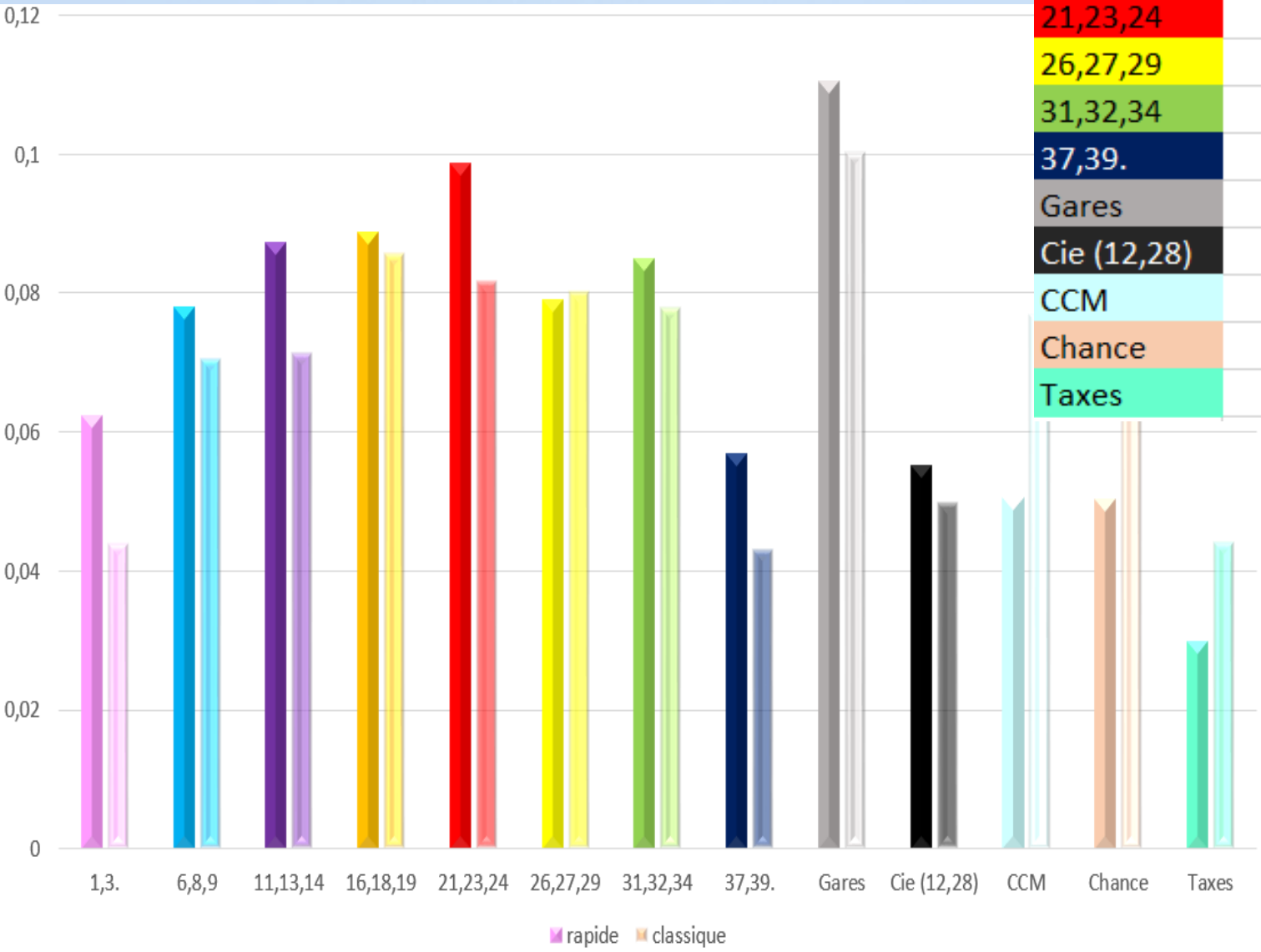
Représentation des probabilités long terme de chaque case dans une partie rapide de Monopoly



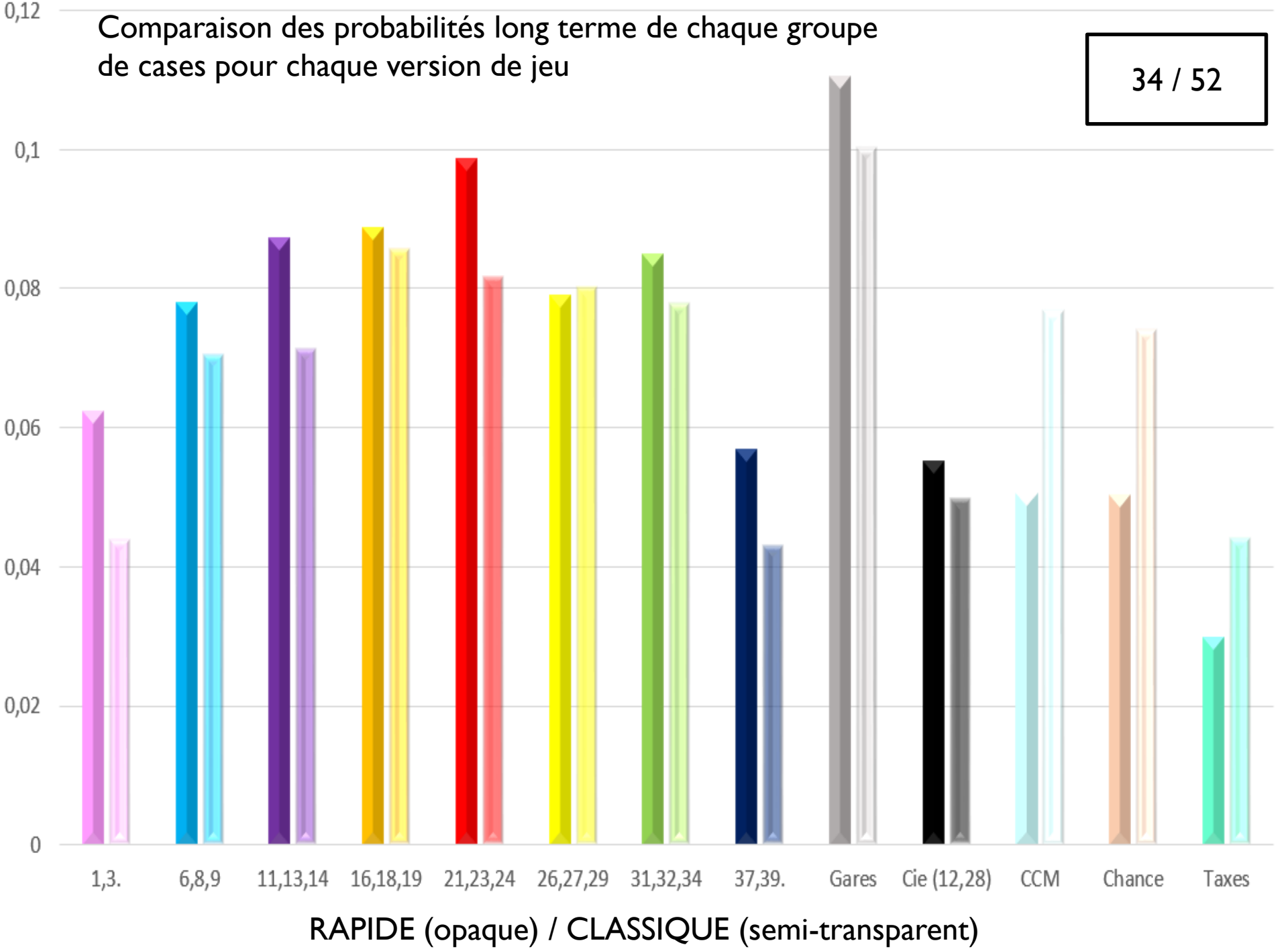

Comparaison des probabilités long terme de chaque case pour chaque version du jeu
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
Case Départ
Belleville
Caisse de Communauté
Lecourbe
Impôts
Gare Montparnasse
Vaugirard
Carte Chance
Courcelles
Républiques
Simple Visite/Prison
La Villette
Cie Électricité
Neuilly
Paradis
Gare de Lyon
Mozart
Caisse de Communauté
Saint-Michel
Pigalle
Parc Gratuit
Matignon
Carte Chance
Malesherbes
Henri-Martin
Gare du Nord
Saint-Honoré
La Bourse
Cie des Eaux
La Fayette
Allez en Prison
Breteuil
Foch
Caisse de Communauté
Capucines
Gare Saint-Lazare
Carte Chance
Champs-Élysées
Taxe de Luxe
La Paix
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Rapide
Classique
32 / 52

Comparaison des probabilités long terme de chaque groupe de cases pour chaque version de jeu

	rapide	classique
1,3.	6,23%	4,40%
6,8,9	7,79%	7,05%
11,13,14	8,72%	7,15%
16,18,19	8,88%	8,58%
21,23,24	9,87%	8,18%
26,27,29	7,91%	8,02%
31,32,34	8,50%	7,80%
37,39.	5,68%	4,32%
Gares	11,05%	10,04%
Cie (12,28)	5,52%	4,98%
CCM	5,06%	7,69%
Chance	5,04%	7,42%
Taxes	2,99%	4,42%



Comparaison des probabilités long terme de chaque groupe de cases pour chaque version de jeu



RAPIDE (opaque) / CLASSIQUE (semi-transparent)

Limites de L'Etude du Dé Rapide

- Libre Arbitre du joueur: Dé Rapide (Triple, Bus), Prison (longueur du Séjour)
- Interdépendance des joueurs (Dé Rapide/Mr Monopoly)
- Probabilités à long terme (24 lancers soit 8 heures de jeu! (5min/tour et 4 joueurs))
- Implications financières (la rentabilité et la négociation)

Conclusion:

Le Dé Rapide renforce l'importance des stratégies "classiques" en augmentant significativement les probabilités d'atterrir sur les cases propriétés. De plus, il offre au joueur une plus grande flexibilité et une certaine liberté de mouvement sur le plateau.

(https://www.pinterest.fr/)

Merci

pour votre Attention

Annexe

Num	Case	100W	MaMC	Rapide	Classique		rapide	classique
0	Case Départ	0,034	0,0368	0,0143764	0,0214219	1,3.	0,0622703	0,0439985
1	Belleville	0,0235	0,0252	0,0320569	0,021592	6,8,9	0,0779312	0,0705292
2	Caisse de Communauté	0,0237	0,0223	0,0146425	0,0219779	11,13,14	0,0872283	0,0714984
3	Lecourbe	0,0238	0,0256	0,0302134	0,0224065	16,18,19	0,0888051	0,085754
4	Impôts	0,0256	0,0277	0,015031	0,0225805	21,23,24	0,0986791	0,0817976
5	Gare Montparnasse	0,0327	0,0352	0,030573	0,0229248	26,27,29	0,0791345	0,0802096
6	Vaugirard	0,0249	0,0268	0,0258554	0,0234639	31,32,34	0,085038	0,0779843
7	Carte Chance	0,0254	0,0102	0,0154988	0,0237924	37,39.	0,0568134	0,0431879
8	Courcelles	0,0255	0,0274	0,0272181	0,0237473	Gares	0,1105182	0,1003548
9	Républiques	0,0254	0,0252	0,0248577	0,0233179	Cie (12,28)	0,0552117	0,0497986
10	Simple Visite/Prison	0,1342	0,0269	0,0157785	0,0229886	CCM	0,0506365	0,0769399
11	La Villette	0,0299	0,0321	0,0373019	0,0226624	Chance	0,0504099	0,0742179
12	Cie Électricité	0,0305	0,031	0,0291334	0,0232426	Taxes	0,0299279	0,044222
13	Neuilly	0,0254	0,0281	0,0254677	0,0239597			
14	Paradis	0,0283	0,0293	0,0244586	0,0248763			
15	Gare de Lyon	0,0275	0,0346	0,0246758	0,0264484			
16	Mozart	0,0312	0,0331	0,0264993	0,02785			
17	Caisse de Communauté	0,0306	0,0307	0,0180669	0,0294421			
18	Saint-Michel	0,0329	0,0349	0,0322381	0,0291915			
19	Pigalle	0,0328	0,0366	0,0300677	0,0287125			
20	Parc Gratuit	0,0329	0,0343	0,0194542	0,0284049			
21	Matignon	0,0305	0,0336	0,0360937	0,0279835			
22	Carte Chance	0,0325	0,0125	0,0187109	0,0276287			
23	Malesherbes	0,0299	0,0325	0,0337272	0,026972			
24	Henri-Martin	0,0349	0,0377	0,0288582	0,0268421			
25	Gare du Nord	0,0337	0,0364	0,0279336	0,0269931			
26	Saint-Honoré	0,0296	0,0321	0,0268148	0,0270424			
27	La Bourse	0,0294	0,0318	0,0260824	0,0268671			
28	Cie des Eaux	0,0309	0,0333	0,0260783	0,026556			
29	La Fayette	0,0284	0,0307	0,0262373	0,0263			
30	Allez en Prison	0,029	0,0469	0,0177867	0,026692			
31	Breteuil	0,0294	0,0318	0,0267651	0,0269294			
32	Foch	0,0289	0,0312	0,0269599	0,0262288			
33	Caisse de Communauté	0,0297	0,0282	0,0179272	0,0255198			
34	Capucines	0,0275	0,0297	0,0313129	0,0248262			
35	Gare Saint-Lazare	0,0267	0,0289	0,0273358	0,0239885			
36	Carte Chance	0,0254	0,0102	0,0162002	0,0227967			
37	Champs-Élysées	0,0241	0,026	0,0279815	0,0216038			
38	Taxe de Luxe	0,024	0,026	0,0148969	0,0216415			
39	La Paix	0,029	0,0312	0,0288319	0,0215841			

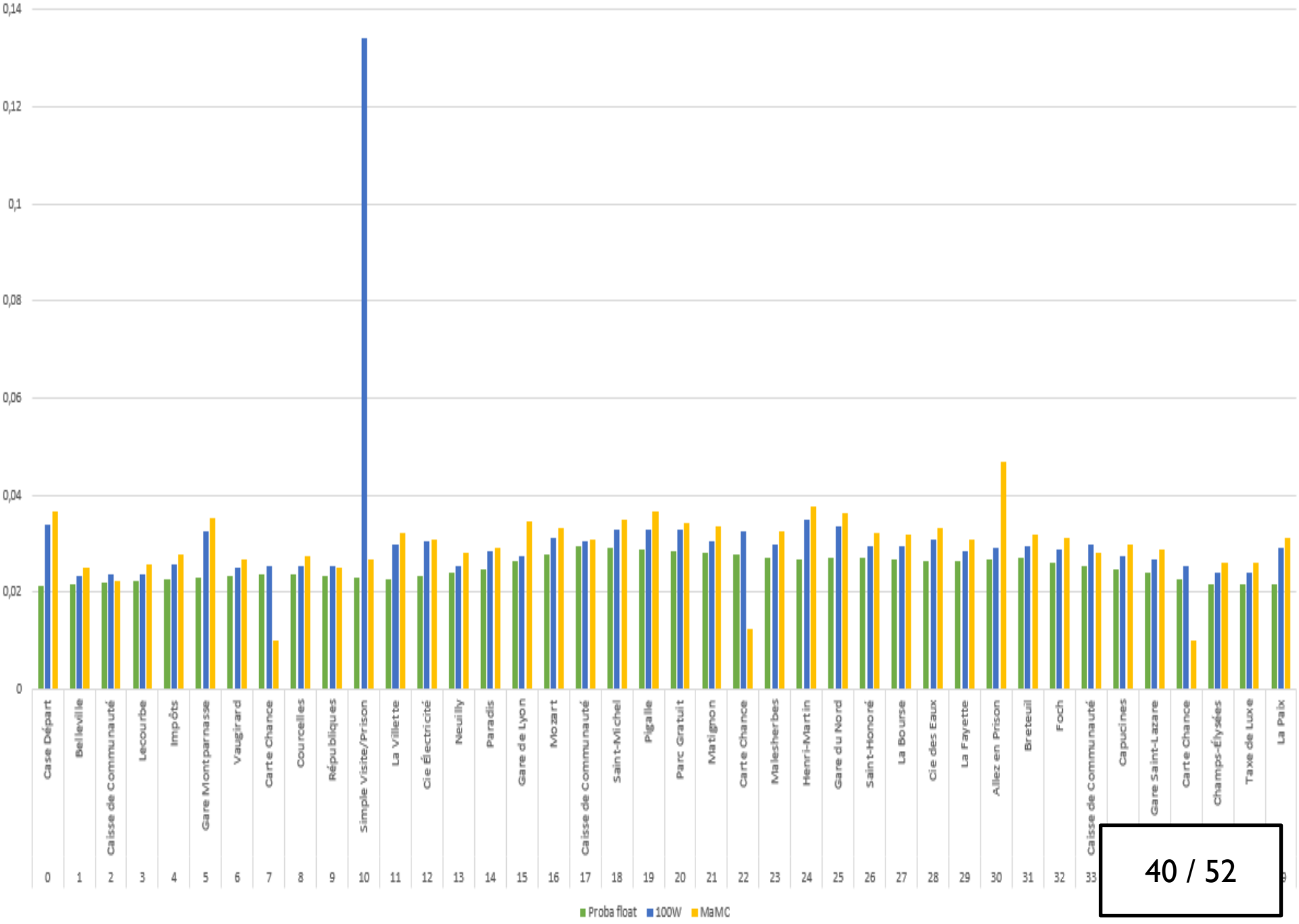
Capture d'écran du tableau Excel des différentes probabilités d'atterrissage long terme selon différents modèles

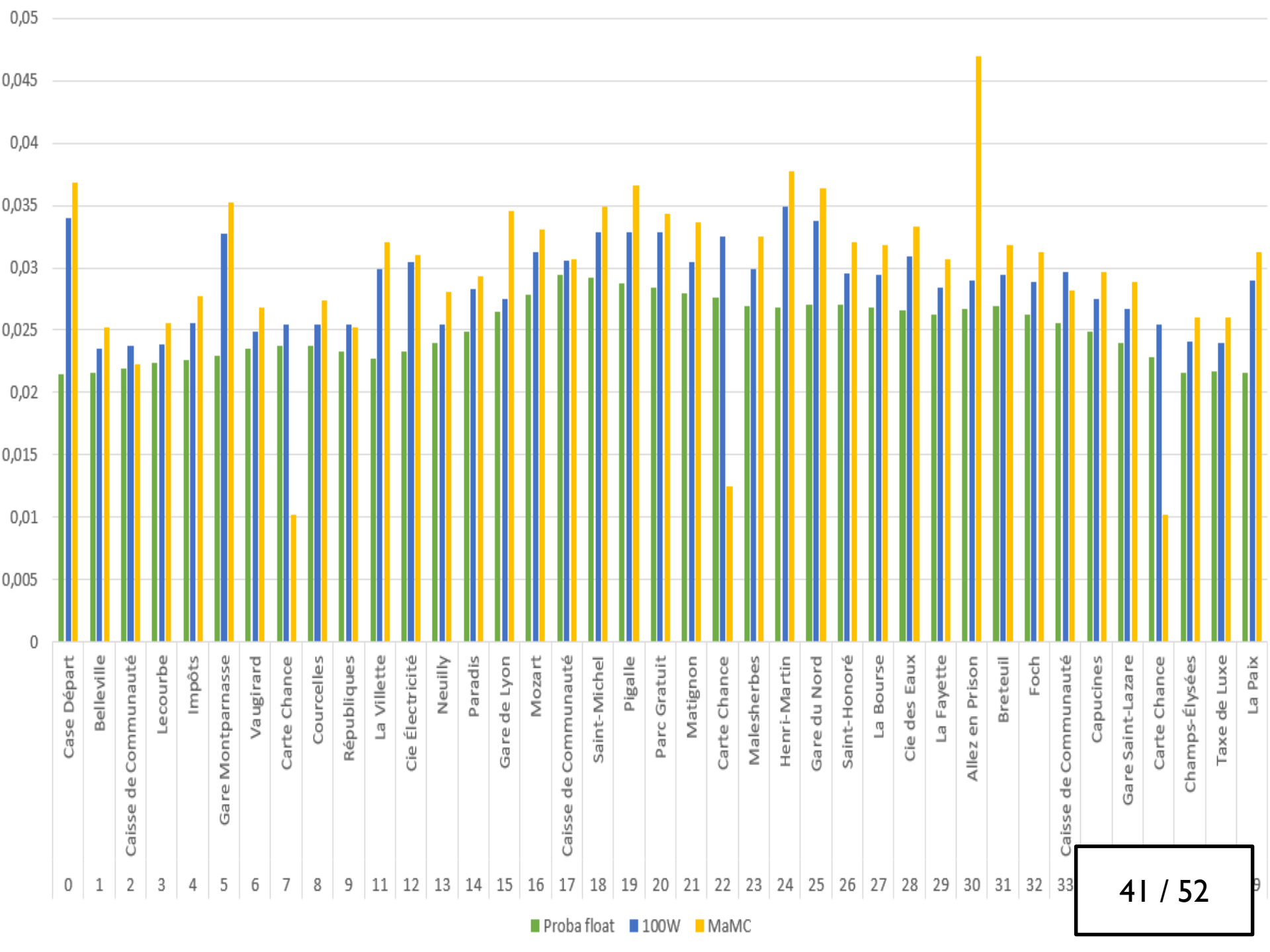
100W: 1000 Ways to Win Monopoly Games
(JAY WALKER AND JEFFREY S. LEHMAN)
1975

MaMC: Monopoly as a Markov Process
(ROBERT B.ASH AND RICHARD L. BISHOP)
1972

Rapide: Probabilités d'atterrissage long terme de chaque case, dans une partie rapide
(avec dé rapide)

Classique: Probabilités d'atterrissage long terme de chaque case, dans une partie classique
(uniqu. deux dés)


0,14
0,12
0,1
0,08
0,06
0,04
0,02
0
Case Départ
Belleville
Caisse de Communauté
Lecourbe
Impôts
Gare Montparnasse
Vaugirard
Carte Chance
Courcelles
Républiques
Simple Visite/Prison
La Villette
Cie Électricité
Neuilly
Paradis
Gare de Lyon
Mozart
Caisse de Communauté
Saint-Michel
Pigalle
Parc Gratuit
Matignon
Carte Chance
Malesherbes
Henri-Martin
Gare du Nord
Saint-Honoré
La Bourse
Cie des Eaux
La Fayette
Allez en Prison
Breteuil
Foch
Caisse de Communauté
Capucines
Gare Saint-Lazare
Carte Chance
Champs-Élysées
Taxe de Luxe
La Paix
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Proba float
100W
MaMC


0,05
0,045
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
Case Départ
Belleville
Caisse de Communauté
Lecourbe
Impôts
Gare Montparnasse
Vaugirard
Carte Chance
Courcelles
Républiques
La Villette
Cie Électricité
Neuilly
Paradis
Gare de Lyon
Mozart
Caisse de Communauté
Saint-Michel
Pigalle
Parc Gratuit
Matignon
Carte Chance
Malesherbes
Henri-Martin
Gare du Nord
Saint-Honoré
La Bourse
Cie des Eaux
La Fayette
Allez en Prison
Breteuil
Foch
Caisse de Communauté
Capucines
Gare Saint-Lazare
Carte Chance
Champs-Élysées
Taxe de Luxe
La Paix
0 1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Proba float
100W
MaMC

Hypothèses simplificatrices:

- **Dé Rapide:**
  - Stratégie fixe et indépendance du joueur
  - Ignorer la règle des triples
  - Mr Monopoly vous envoie directement vers la prochaine propriété (un lancer = un déplacement)
  - Quitter immédiatement la prison
  - Ignorer la règle des doubles
  - **CCM:**
    - Annuler l'effet des cartes « Vous êtes libéré de prison »
    - Mélanger les cartes à nouveau

```python
# Importation du module numpy pour la manipulation des matrices
import numpy as np
# Importation du module scipy.linalg pour les opérations sur les matrices (le calcul des vecteurs propres)
import scipy.linalg as la
# Importation du module pandas pour la manipulation et l'exportation des données dans un fichier Excel
import pandas as pd

# Liste des noms des cases du jeu de Monopoly
nom_cases = [
    'Case Départ', 'Belleville', 'Caisse de Communauté', 'Lecourbe', 'Impôts',
    'Gare Montparnasse', 'Vaugirard', 'Carte Chance', 'Courcelles', 'Républiques',
    'Simple Visite/Prison', 'La Villette', 'Cie Électricité', 'Neuilly',
    'Paradis', 'Gare de Lyon', 'Mozart', 'Caisse de Communauté', 'Saint-Michel',
    'Pigalle', 'Parc Gratuit', 'Matignon', 'Carte Chance', 'Malesherbes',
    'Henri-Martin', 'Gare du Nord', 'Saint-Honoré', 'La Bourse', 'Cie des Eaux',
    'La Fayette', 'Allez en Prison', 'Breteuil', 'Foch', 'Caisse de Communauté',
    'Capucines', 'Gare Saint-Lazare', 'Carte Chance', 'Champs-Élysées', 'Taxe de Luxe',
    'La Paix'
]

```

Capture d'écran de PyCharm Community Edition

```python
def proba_des():
    """
    Calcule les probabilités des résultats possibles pour deux dés standards.
    :return: Liste des probabilités.
    """
    return [x / 36 for x in [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1]]

def proba_des_rapide():
    """
    Calcule les probabilités des résultats possibles pour deux dés rapides et Mr Monopoly.
    :return: Liste des probabilités.
    """
    return [(x * (4 / 6)) / 144 for x in [1, 3, 6, 10, 14, 18, 20, 20, 18, 14, 10, 6, 3, 1]]

propriety = [1, 3, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 18, 19, 21, 23, 24, 25, 26, 27, 28, 29, 31, 32, 34, 35, 37, 39]
def monsieur_monopoly_value(pos):
    """
    Détermine la prochaine propriété de Mr Monopoly en fonction de la position actuelle.
    :param pos: Position actuelle.
    :return: Prochaine position.
    """
    if (pos == 39):
        return 1
    else:
        for x in propriety:
            if x > pos:
                return x

```

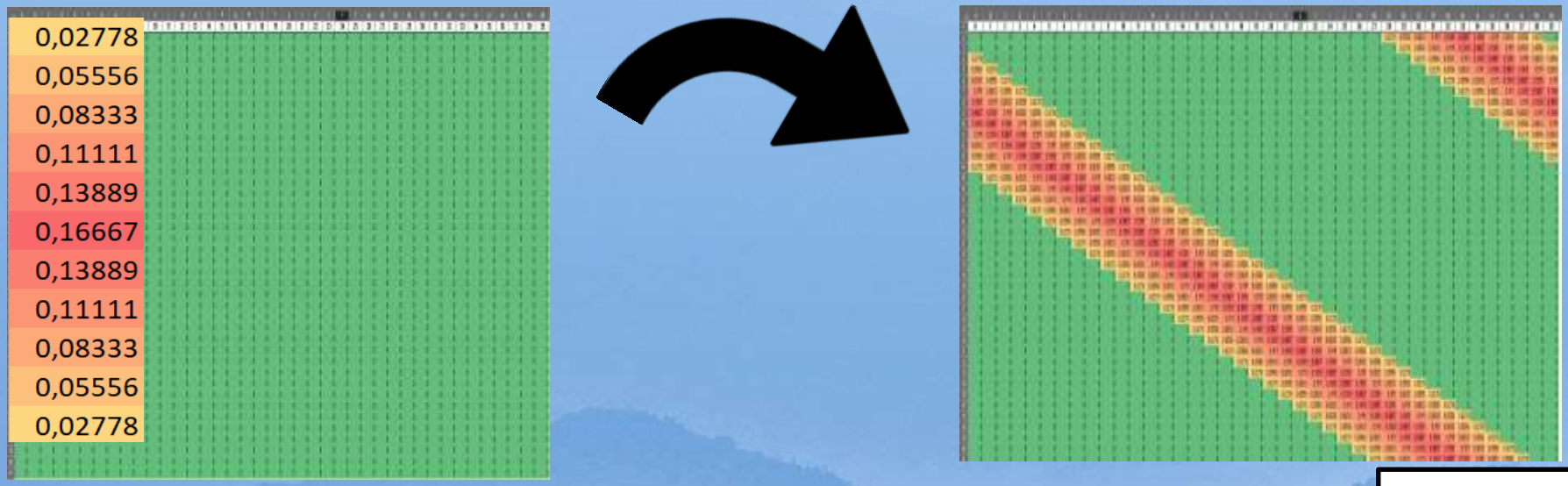
Capture d'écran de PyCharm C

```python

def creer_matrice(proba_list, matrice, compteur_colonne, dim):
    """
    Fonction récursive qui permet de remplir une matrice colonne par colonne.
    :param proba_list: Liste des probabilités.
    :param matrice: Matrice de transition à remplir.
    :param compteur_colonne: L'indicateur récursif.
    :param dim: Dimension de la matrice.
    """
    if compteur_colonne == dim:
        return matrice
    dernier_element = proba_list.pop()  # Retire le dernier élément de la liste
    proba_list = [dernier_element] + proba_list  # Ajoute cet élément au début de la liste
    matrice[:, compteur_colonne] = proba_list  # Remplit la colonne actuelle de la matrice
    creer_matrice(proba_list, matrice, compteur_colonne + 1, dim)  # Appel récursif pour la colonne suivante

```

Capture d'écran de PyCharm Community Edition

```
63
64  # Construire matrice_des qui regroupe simplement les probabilités de déplacement selon les lancers de dés
65  proba_ligne_case_depart = [0] * 40  # Initialise une ligne de probabilités avec des zéros
66  proba_ligne_case_depart[2:13] = proba_des()  # Remplit la ligne avec les probabilités des dés standards
67  matrice_des = np.zeros((40, 40))  # Crée une matrice de zéros de 40x40
68  matrice_des[:, 0] = proba_ligne_case_depart  # Remplit la première colonne avec les probabilités
69  creer_matrice(proba_ligne_case_depart, matrice_des, compteur_colonne: 1, dim: 40)  # Remplit le reste de la matrice
70
```

Capture d'écran de PyCharm Community Edition



Capture d'écran d'Excel
Représentation d'une matrice vide et la matrice de lancer de dés (Partie Classique)

```
70
71  # Construire matrice_des_rapide qui regroupe les probabilités de déplacement selon les lancers de dés rapide
72  # y compris celui de Mr Monopoly
73  proba_ligne_case_depart_rapide = [0] * 40  # Initialise une ligne de probabilités avec des zéros
74  proba_ligne_case_depart_rapide[2:16] = proba_des_rapide()  # Remplit la ligne avec les probabilités des dés rapides
75  matrice_des_rapide = np.zeros((40, 40))  # Crée une matrice de zéros de 40x40
76  matrice_des_rapide[:, 0] = proba_ligne_case_depart_rapide  # Remplit la première colonne avec les probabilités
77  creer_matrice(proba_ligne_case_depart_rapide, matrice_des_rapide, compteur_colonne: 1, dim: 40)  # Remplit le reste de la matrice
78
79  # Implémenter Mr Monopoly
80  for i in range(40):
81      matrice_des_rapide[monsieur_monopoly_value(i), i] += 2 / 6  # Ajoute les probabilités de Mr Monopoly
```

Capture d'écran de PyCharm Community Edition

```
# Construire matrice_regles qui inclut les effets des cartes Chance et Caisse de Communauté, ainsi que la prison
matrice_regles = np.eye(40)  # Crée une matrice identité de 40x40
chance_cases = [7, 22, 36]  # Les emplacements des cases chance
for x in chance_cases:
    matrice_regles[:, x] = [0] * 40  # Initialise la colonne de la case Chance avec des zéros
    matrice_regles[[0, 10, 11, 15, 24, 39, x - 3], x] = 1 / 16  # Ajoute les probabilités des cartes Chance
    matrice_regles[x, x] = 9 / 16  # Probabilité de rester sur la case

caisse_cases = [2, 17, 33]  # Les emplacements des cases caisse de communauté
gares_caisse_cases = [5, 25, 35]  # Les emplacements des gares correspondantes
for i, x in enumerate(caisse_cases):
    matrice_regles[:, x] = [0] * 40  # Initialise la colonne de la case Caisse de Communauté avec des zéros
    matrice_regles[[0, 1, 10, gares_caisse_cases[i]], x] += 1 / 16  # Ajoute les probabilités des cartes Caisse de Communauté
    matrice_regles[x, x] += 12 / 16  # Probabilité de rester sur la case

matrice_regles[10, 30] = 1  # Aller en prison
matrice_regles[30, 30] = 0  # Sortir de prison

```

Capture d'écran de PyCharm Community Edition

```python
def est_matrice_de_transition(matrice, erreur=1e-6):
    """
    Vérifie si chaque ligne d'une matrice de transition a une somme égale à 1,
    en tenant compte d'une marge d'erreur.
    :param matrice: La matrice à vérifier.
    :param erreur: La marge d'erreur acceptable.
    :return: True si chaque ligne a une somme égale à 1, False sinon.
    """
    sommes_ligne = np.sum(matrice, axis=0)  # Calcule la somme de chaque colonne
    return np.all(np.abs(sommes_ligne - 1) < erreur)  # Vérifie si les sommes sont égales à 1 dans la marge d'erreur
```

Capture d'écran de PyCharm Community Edition

2.3. Proposition. *La matrice de transition d'une chaîne de Markov homogène finie est stochastique, c'est-à-dire positive et telle que la somme des coefficients sur l'une quelconque de ses lignes vaut* 1.

(Algèbre linéaire. Réduction des endomorphismes - Roger Mansuy, Rached Mneimné)

```python
def lj_des_classique():
    """
    Calcule et enregistre les probabilités des cases pour des lancers de dés classiques.
    """
    matrice_finale = np.matmul(matrice_des, matrice_regles)  # Calcule la matrice finale en multipliant matrice_des par matrice_regles
    vecteur_propre = la.eig(matrice_finale)[1]  # Calcule les vecteurs propres de la matrice finale
    vecteur_etat_stable = vecteur_propre[:, 0]  # Sélectionne le premier vecteur propre
    vecteur_etat_stable = vecteur_etat_stable / sum(vecteur_etat_stable)  # Normalise le vecteur propre
    vecteur_etat_stable = vecteur_etat_stable.tolist()  # Convertit le vecteur en liste
    vecteur_etat_stable = [float(valeur.real) for valeur in vecteur_etat_stable]  # Convertit les valeurs en réels flottants
    afficher_valeurs(vecteur_etat_stable, nom_fichier: "lj_des_classique.xlsx")  # Affiche les valeurs dans un fichier Excel

def lj_des_rapide():
    """
    Calcule et enregistre les probabilités des cases pour des lancers de dés rapides.
    """
    matrice_finale = np.matmul(matrice_des_rapide, matrice_regles)  # Calcule la matrice finale en multipliant matrice_des_rapide par matrice_regles
    vecteur_propre = la.eig(matrice_finale)[1]  # Calcule les vecteurs propres de la matrice finale
    vecteur_etat_stable = vecteur_propre[:, 0]  # Sélectionne le premier vecteur propre
    vecteur_etat_stable = vecteur_etat_stable / sum(vecteur_etat_stable)  # Normalise le vecteur propre
    vecteur_etat_stable = vecteur_etat_stable.tolist()  # Convertit le vecteur en liste
    vecteur_etat_stable = [float(valeur.real) for valeur in vecteur_etat_stable]  # Convertit les valeurs en réels flottants
    afficher_valeurs(vecteur_etat_stable, nom_fichier: "lj_des_rapide.xlsx")  # Affiche les valeurs dans un fichier Excel

```

Capture d'écran de PyCharm Community Edition

```python
def afficher_valeurs(list_proba, nom_fichier="resultat.xlsx"):
    """
    Enregistre les probabilités de chaque case dans un fichier Excel.
    :param list_proba: Liste des probabilités pour chaque case.
    :param nom_fichier: Nom du fichier Excel à créer.
    """
    df = pd.DataFrame()  # Crée un DataFrame vide
    df['Num'] = list(range(40))  # Ajoute une colonne pour les numéros des cases
    df['Case'] = nom_cases  # Ajoute une colonne pour les noms des cases
    df['Proba'] = list_proba  # Ajoute une colonne pour les probabilités
    df.to_excel(nom_fichier, index=False)  # Sauvegarde le DataFrame dans un fichier Excel

def afficher_joliment(matrix):
    """
    Affiche une matrice de manière lisible.
    :param matrix: La matrice à afficher.
    """
    np.set_printoptions(threshold=np.inf)  # Configure l'affichage complet de la matrice
    np.set_printoptions(formatter={'int': '{: 4d}'.format})  # Formate les entiers pour l'affichage
    print(np.array2string(matrix, separator=' '))  # Affiche la matrice

def afficher_joliment_sur_excel(array):
    """
    Enregistre une matrice dans un fichier Excel.
    :param array: La matrice à enregistrer.
    """
    df = pd.DataFrame(array)  # Convertit la matrice en DataFrame
    df.to_excel( excel_writer: "test.xlsx", index=False)  # Sauvegarde le DataFrame dans un fich
```


Capture d'écran de PyCharm Community Edition

```
165  # Les tests stochastiques
166  #print(est_matrice_de_transition(matrice_regles))
167  #print(est_matrice_de_transition(matrice_des))
168  #print(est_matrice_de_transition(matrice_des_rapide))
169
170  #afficher_joliment_sur_excel(np.matmul(matrice_des, matrice_regles))
171  #lj_des_classique()
172  #lj_des_rapide()
173
```

Capture d'écran de PyCharm Community Edition