Python, probabilités & chaînes de Markov

Les galettes des Rois

Dans une boulangerie, on vend des galettes avec des fèves en forme d'animaux (une fève par galette). Il y a 6 types de fèves : un mouton, une poule, un cheval, une vache, un poisson et un perroquet. La petite Marie collectionne les fèves.

Chaque jour, elle achète une galette qu'elle choisit au hasard dans le magasin. On souhaite étudier à quel moment Marie aura la collection complète de fèves, c'est-à-dire au moins une fève de chaque type.

Il s'agit de répondre aux deux questions suivantes :

Question Q1 :

Déterminer le nombre de jours tel que la probabilité que Marie ait la collection complète de fèves soit supérieure à 99 %

Question Q2 :

En moyenne, combien de jours faut-il pour obtenir la collection complète de fèves ?

Question Q1 :

Déterminer le nombre de jours tel que la probabilité que Marie ait la collection complète de fèves soit supérieure à 99 %

On calcule G(n), la probabilité de l'évènement : "On obtient 6 fèves différentes au bout de n jours"

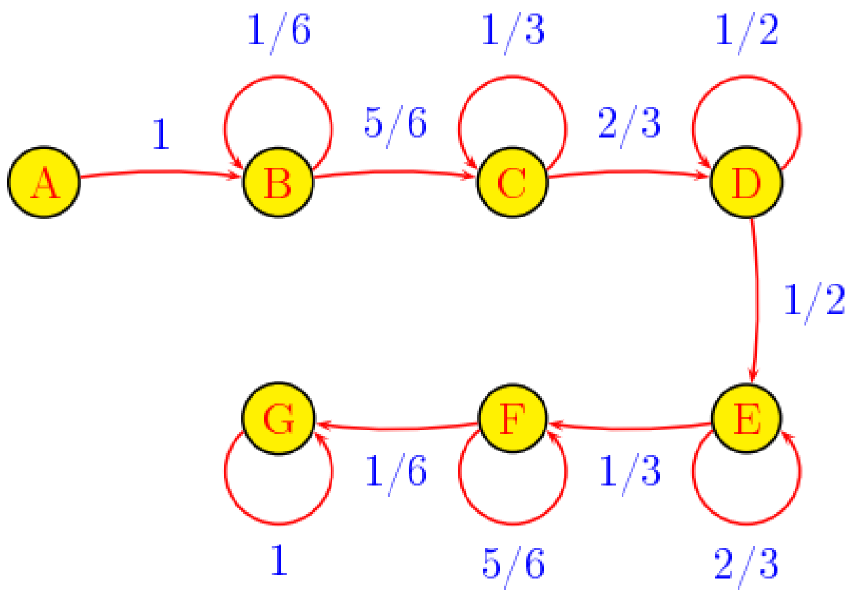
Soient les probabilités des évènements A, B, C , D, E, F et G dont les définitions sont les suivantes :
A(n) : "Probabilité d'obtenir 0 fève différente au bout de n jours"
B(n) : "Probabilité d'obtenir 1 fève différente au bout de n jours"
C(n) : "Probabilité d'obtenir 2 fèves différentes au bout de n jours"
D(n) : "Probabilité d'obtenir 3 fèves différentes au bout de n jours"
E(n) : "Probabilité d'obtenir 4 fèves différentes au bout de n jours"
F(n) : "Probabilité d'obtenir 5 fèves différentes au bout de n jours"
G(n) : "Probabilité d'obtenir 6 fèves différentes au bout de n jours"

Les valeurs initiales sont les suivantes :
A(0) = 1, B(0) = 0 , C(0) = 0, D(0) = 0, E(0) = 0, F(0) = 0 et G(0) = 0

On a les relations de récurrence suivantes :

A(n+1) = 0*A(n)
B(n+1) = 1/6*B(n) + A(n)
C(n+1) = 1/3*C(n) + 5/6*B(n)
D(n+1) = 1/2*D(n) + 2/3*C(n)
E(n+1) = 2/3*E(n) + 1/2*D(n)
F(n+1) = 5/6*F(n) + 1/3*E(n)
G(n+1) = G(n) + 1/6*F(n)

On dessine la chaîne de Markov correspondante :



On en déduit la matrice de transition :

$$X_{n+1} = K * X_n \text{ avec : } K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1/6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5/6 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2/3 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 2/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 5/6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/6 & 1 \end{pmatrix}$$

$$X_{n+1} = K^n * X_0 \text{ avec : } X_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Le programme Python

```
import numpy as np

def MIteration(n) :
    P = np.array([[1],[0],[0],[0],[0],[0],[0]])
    for k in range(n) :
        P = M.dot(P)
    return P

M = np.array([[0,0,0,0,0,0,0],[1,1/6,0,0,0,0,0],[0,5/6,1/3,0,0,0,0], \
[0,0,2/3,1/2,0,0,0],[0,0,0,1/2,2/3,0,0],[0,0,0,0,1/3,5/6,0],[0,0,0,0,0,1/6,
1]])
P = np.array([[1],[0],[0],[0],[0],[0],[0]])

T = 0

while P[6,0] < 0.99 :
    P = M.dot(P)
    T = T + 1

print("P(j) > 0.99 pour j = ", T, " jours.")
print("P(",T,") = ",P[6,0])
```

```
*** Console de processus distant Réinitialisée ***
>>>
P(j) > 0.99 pour j =  36  jours.
P( 36 ) =  0.9915418784295547
>>>
```

A partir du 36ième **jour**, la probabilité que Marie ait la collection complète est supérieure à 99 %

Quelques valeurs du vecteur colonne X :

Par exemple pour n = 6 , MIteration(6) donne **0,00154** pour la dernière ligne du vecteur X, ce qui signifie que la probabilité que Marie ait la collection complète en seulement 6 jours est égale à 0,00154, soit 0,154 % de chance d'avoir la collection complète en 6 jours.

```
>>> MIteration(6)
array([[0.00000000e+00],
       [1.28600823e-04],
       [1.99331276e-02],
       [2.31481481e-01],
       [5.01543210e-01],
       [2.31481481e-01],
       [1.54320988e-02]])
```

```
>>> MIteration(12)
array([[0.00000000e+00],
       [2.75636195e-09],
       [2.82113645e-05],
       [4.76993948e-03],
       [1.01131085e-01],
       [4.56255081e-01],
       [4.37815681e-01]])
```

```
>>> MIteration(18)
array([[0.00000000e+00],
       [5.90784025e-14],
       [3.87173265e-08],
       [7.61390754e-05],
       [9.92094274e-03],
       [2.05295763e-01],
       [7.84707116e-01]])
```

Question Q2 :

En moyenne, combien de jours faut-il pour obtenir la collection complète de fèves ?

On écrit un programme Python permettant d'approcher la solution.

On note (E) l'expérience aléatoire qui consiste à :

- tirer une galette (donc une fève) au hasard tant que la collection n'est pas complète ;
- noter le nombre p de tirages qui ont été nécessaires.

On répète N fois l'expérience (E) :

1/ Calcul de N_k : On somme l'ensemble des jours nécessaires pour chaque expérience (E) et on obtient N_k le nombre total de tirages (jours) qui ont été nécessaires (**N_k > N**). Le nombre moyen de jours, Nm, est égal à N_k/N.

2/ Calcul des N_j : On stocke dans N_j[**i**] le nombre de fois que l'expérience (E) a été réalisée en **i** jours.

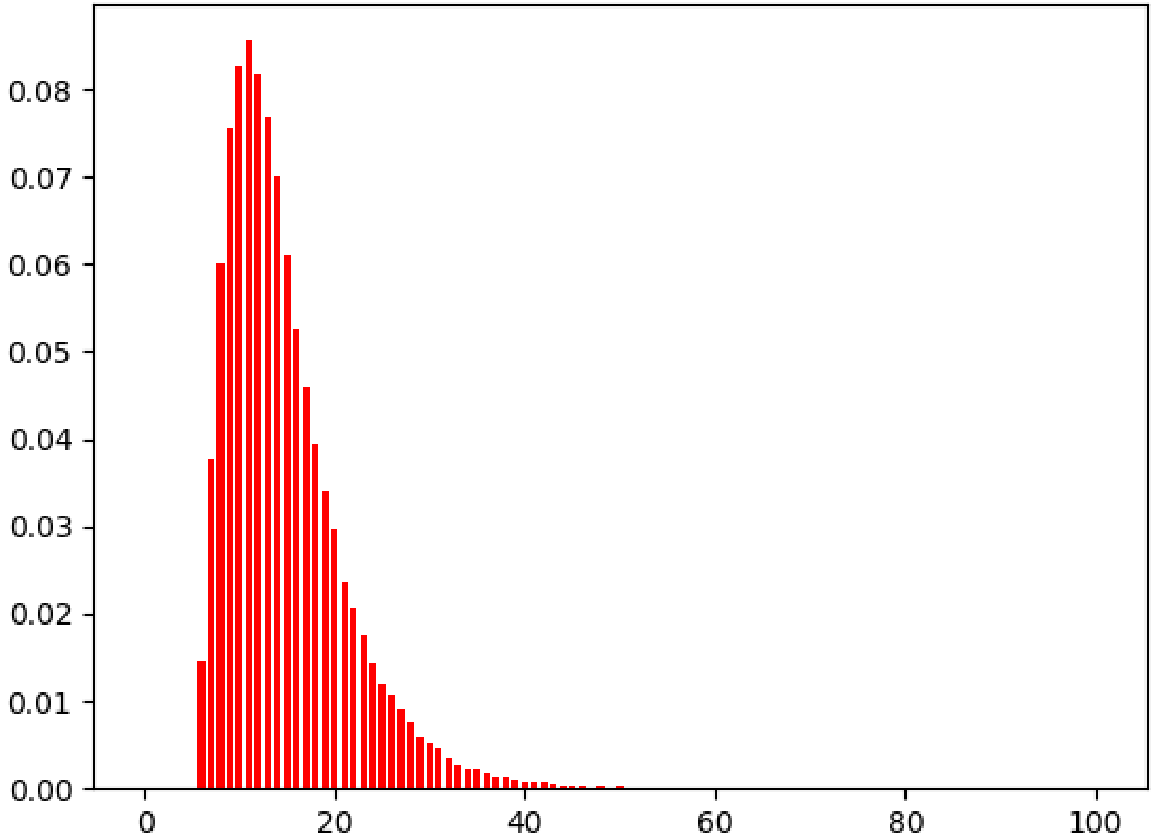
Le programme Python

```
import numpy as np
from random import *
import matplotlib.pyplot as plt

N = 70000
Nk = 0
Nj = [0]*101
Ni = np.linspace(0,100,101)

for i in range(1,N+1) :
    K = [1]*6
    p = 0
    while sum(K) != 0 :
        p = p + 1
        j = randint(1,6)
        if K[j-1] == 1 :
            K[j-1] = 0

    Nk = Nk + p
    if p < 101 :
        Nj[p] = Nj[p] + 1

print("Nombre moyen de jours Nm = ",Nk/N)
FNj = [u/N for u in Nj]
plt.bar(Ni,FNj,align ='center', width = 0.7, color = 'r')
plt.show()
```

```
*** Console de processus distant Réinitialisée ***
>>>
Nombre moyen de jours Nm =  14.713557142857143
```

Il faut en moyenne **14,7 jours** pour obtenir la collection complète de fèves.

Graphique :



En abscisse : le nombre de jours pour réaliser (E) (donc obtenir la collection complète de fèves)

En ordonnée : la fréquence $N_j[i]/N$ pour chaque nombre de jours **i** en abscisse. Il s'agit donc de la fréquence de la réalisation de (E) en **i** jours.

On remarque qu'au-delà de 60 jours, on peut considérer que toutes les fèves ont été obtenues.

Vidéo Youtube : https://youtu.be/Zh_0a_dV3DE

A propos des tutoriels

Site Internet :

Le site Internet http://www.joseouin.fr propose un ensemble de tutoriels vidéo portant sur différents domaines : Excel, Word, Libre Office, LMS Claroline, Mathématiques, Geogebra, Python, Scilab, Visual Basic, Photofiltre, CMS Joomla, Ubuntu, Windows & Autres. De nouveaux tutoriels sont régulièrement mis en ligne.

Chaîne Youtube :

Toutes les vidéos sont hébergées sur la chaîne Youtube « Mathématiques Magiques » :

https://www.youtube.com/c/MathematiquesMagiques

Pensez à vous abonner et cliquez sur le petit pouce bleu de la vidéo qui vous avez visionnée si le contenu vous a convenu. Merci à vous.

J'espère que ce tutoriel vous rendra service.

José OUIN.