

Pile ou face

Marc - Lorenzi - 16 avril 2018

```
In [1]: import matplotlib.pyplot as plt
import random
%matplotlib inline
```

Succès, échec, quelques remarques philosophiques sur les générateurs de nombres aléatoires, attente d'un succès ...

1. Succès, échec

1.1 Variables de Bernoulli

Soit (Ω, P) un espace probabilisé. Soit $p \in [0, 1]$. Soit $A \subset \Omega$ un événement de probabilité p . Les éléments de Ω sont les issues d'une expérience. Pour tout $\omega \in \Omega$ nous dirons que l'expérience réussit (relativement à l'ensemble A) lorsque $\omega \in A$, sinon l'expérience échoue. Le réel p est donc la probabilité de réussite de l'expérience. Par exemple, $p = \frac{1}{2}$ pour un jeu de pile ou face avec une pièce parfaite, $p = \frac{1}{5}$ lorsque A est l'événement "je suis gaucher", $p \simeq 5 \cdot 10^{-8}$ pour l'événement "je gagne au loto".

Parlons en termes de variables aléatoires, ce sera plus efficace. Soit $X : \Omega \rightarrow \{0, 1\}$ une variable aléatoire ne pouvant prendre que les valeurs 0 et 1. Une telle v.a. est appelée variable aléatoire de Bernoulli. On pose $p = P(X = 1)$. Le réel p est appelé le paramètre de Bernoulli de la v.a. et on dit que X suit une loi de Bernoulli de paramètre p .

1.2 Simulation

On peut simuler un variable de Bernoulli par la fonction ci-dessous. La fonction calcule un réel aléatoire appartenant à $[0, 1]$ (fonction `random.random()`). Si ce réel est dans $[0, p]$ elle renvoie 1, sinon elle renvoie 0.

```
In [2]: def rand(p):
        if random.random() <= p: return 1
        else: return 0
```

```
In [3]: print([rand(0.3) for k in range(100)])
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0,
0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Ainsi, la fonction `rand` est une variable de Bernoulli de paramètre p . Euh mais non, ça ne va pas du tout !!! On voit bien que l'ensemble d'arrivée est $\{0, 1\}$ mais il n'y a pas d'ensemble de départ ! En fait `rand` n'est même pas une fonction au sens mathématique puisque deux appels successifs à cette fonction renvoient des valeurs qui peuvent être différentes. On peut contourner ce problème en imaginant qu'il y a un paramètre ω implicite qui appartient à un univers Ω qui est ... implicite aussi. Lors de n évaluations successives de `rand(p)`, on obtient n valeurs $X_1(\omega), \dots, X_n(\omega)$ où X_1, \dots, X_n sont des v.a. qui suivent une loi de Bernoulli de paramètre p .

Un générateur de nombres aléatoires bien écrit nous permettra d'affirmer avec confiance que ces v.a. sont indépendantes (enfin autant qu'on peut l'espérer).

Remarque : les générateurs de nombres aléatoires ne sont pas des objets magiques, ils sont pour la plupart bâtis autour de suites récurrentes du type $u_{n+1} = f(u_n)$, le paramètre u_n appartenant à un ensemble plus ou moins compliqué et caché à l'utilisateur. Pour information, à l'heure où est écrit ce notebook, Python utilise comme générateur le "Mersenne Twister".

1.3 Moyennes

Tout le monde le sait, le nombre moyen de fois que `rand(p)` renvoie 1 tend vers p lorsque le nombre de tirages tend vers l'infini. Ceci soulève quelques questions :

- Est-ce vrai ?
- Si oui, pourquoi ?
- Si non, pourquoi pas ? Et c'est non non ou un peu non ?

Voici une fonction effectuant n tirages aléatoires avec une probabilité de réussite p . La fonction renvoie la liste des tirages.

```
In [4]: def tirages(n, p):
         return [rand(p) for k in range(n)]
```

```
In [5]: print(tirages(50, 0.3))
```

```
[0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1]
```

Que contient réellement cette liste ? Une fois encore les variables aléatoires sont nos amies. Sans justifier complètement ce que je vais dire, je peux considérer que le k ème élément de cette liste est $X_k(\omega)$ où $\omega \in \Omega$ est un élément inconnu d'un univers inconnu, et X_k est une variable aléatoire suivant une loi de Bernoulli de paramètre p . Mieux, les v.a. X_1, \dots, X_n sont indépendantes.

La fonction `moyennes` prend en paramètres une liste $[x_1, \dots, x_n]$. Elle renvoie la liste $[y_1, \dots, y_n]$ où $y_k = \frac{1}{k} \sum_{i=1}^k x_i$.

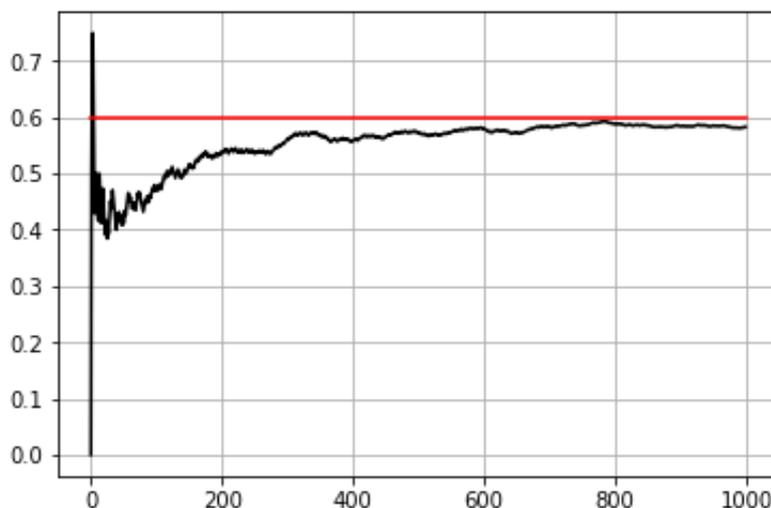
```
In [6]: def moyennes(s):
        somme = 0.
        s1 = []
        for k in range(len(s)):
            somme += s[k]
            s1.append(somme / (k + 1))
        return s1
```

On regroupe en une seule fonction les appels aux deux fonctions précédentes : la fonction `pile_face` renvoie les moyennes des tirages.

```
In [7]: def pile_face(n, p):
        s = tirages(n, p)
        return moyennes(s)
```

Avec les notations ci-dessus, `pile_face(n, p)` renvoie la liste $[Y_1(\omega), \dots, Y_n(\omega)]$ où $\omega \in \Omega$ et $Y_k = \frac{1}{k} \sum_{i=1}^k X_i$ pour $k = 1, \dots, n$. Il n'y a plus qu'à tester. Plusieurs fois, parce que à chaque fois c'est différent.

```
In [8]: n, p = 1000, 0.6
s = pile_face(n, p)
plt.plot(s, 'k-')
plt.plot(n* [p], 'r-')
#plt.axis([1, n, p - 0.3, p + 0.3])
plt.grid()
plt.show()
```



Oui, bon, ben oui. Les moyennes ont bien l'air de converger vers p . Mais qu'entend-on par là ? La réponse à laquelle on rêve est que cela signifie que pour tout $\omega \in \Omega$, $Y_n(\omega) \rightarrow p$ lorsque $n \rightarrow \infty$. Mais cela NE PEUT PAS être vrai. Imaginons que j'aie beaucoup, beaucoup de chance, je réussis à tous les coups. On aura alors $Y_n(\omega) = 1$ pour tout n , la limite sera 1 et pas p . En fait, en continuant à imaginer, on voit que n'importe quel réel entre 0 et 1 est une limite possible pour la suite (Y_n) . Pire, il se pourrait très bien que pour certains ω , $Y_n(\omega)$ diverge ... Toute la question est : qu'entend-on par beaucoup, beaucoup de chance ?

Prenons-nous à rêver que $Y_n(\omega)$ tend vers p pour **presque tous** les $\omega \in \Omega$?

Mathématiquement, cela revient à considérer l'événement $\{Y_n \rightarrow p\}$. A-t-on $P(Y_n \rightarrow p) = 1$? Affirmatif, mais c'est difficile à montrer. C'est la **loi forte des grands nombres**. Que signifie-t-elle exactement ? Soit $E = \{\omega \in \Omega, Y_n(\omega) \not\rightarrow p\}$. La loi forte des grands nombres nous dit que $P(E) = 0$. Cela ne signifie pas que E est vide, mais que si on tombe sur un élément de E on n'a vraiment, vraiment pas de chance (ou beaucoup, beaucoup :-).

Nous allons prouver le résultat plus faible suivant :

Loi faible des grands nombres : Pour tout $\varepsilon > 0$, $P(|Y_n - p| > \varepsilon) \rightarrow 0$ lorsque n tend vers l'infini.

1.4 Un peu de théorie

Soit $(X_n)_{n \geq 1}$ une suite de v.a. indépendantes suivant une même loi de Bernoulli de paramètre $p \in [0, 1]$. Pour tout $n \geq 1$, posons $S_n = \sum_{k=1}^n X_k$. Soit $Y_n = \frac{1}{n} S_n$. Soit $\varepsilon > 0$. On utilise l'inégalité de Bienaymé-Tchebychev, démontrée en cours :

$$P(|S_n - E(S_n)| > n\varepsilon) \leq \frac{V(S_n)}{n^2 \varepsilon^2}$$

Maintenant, par linéarité de l'espérance, $E(S_n) = \sum_{k=1}^n E(X_k) = np$. Et comme les X_k sont indépendantes, $V(S_n) = \sum_{k=1}^n V(X_k) = np(1-p)$. On réinjecte pour obtenir

$$P(|S_n - np| > n\varepsilon) \leq \frac{np(1-p)}{n^2 \varepsilon^2}$$

Bien évidemment, $|S_n - np| > n\varepsilon$ si et seulement si $|nY_n - np| > n\varepsilon$, ou encore $|Y_n - p| > \varepsilon$. Ainsi,

$$P(|Y_n - p| > \varepsilon) \leq \frac{C}{n}$$

où $C = \frac{p(1-p)}{\varepsilon^2}$. D'où le résultat cherché par le théorème des gendarmes.

2. Premier succès dans un tirage à pile ou face

Voici deux affirmations maintes fois entendues dans les bars, autour d'un apéro pastis-cacahuètes :

1- Si je joue vingt fois au loto j'ai plus de chances de gagner que si je joue 10 fois ...

2- Ça fait dix fois de suite que je perds au loto ! Donc les 10 prochaines fois j'aurai plus de chances de gagner ...

Ne remettons pas en doute la première affirmation (elle est vraie, mais cela ne m'intéresse pas ici). Intéressons nous plutôt à la seconde, elle est la raison d'être du joueur addict :-).

2.1 Introduction

Soit (X_1, X_2, \dots) une suite de variables aléatoires indépendantes suivant une loi de Bernoulli de paramètre $p \neq 0$, définies sur un espace probabilisé (Ω, P) . Soit $N : \Omega \rightarrow \mathbb{N}$ définie par $N(\omega) = \min\{n \in \mathbb{N}^*, X_n(\omega) = 1\}$. On voit tout de suite un petit problème : et si pour tout entier n , $X_n(\omega) = 0$? On peut dans ce cas convenir que $N(\omega) = \infty$ où ∞ est un symbole sans signification particulière.

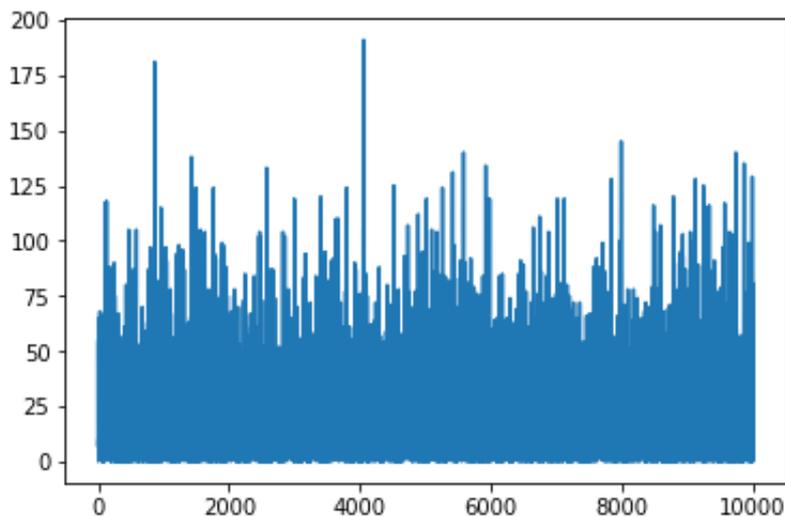
Question : quelle est la loi de N ?

Commençons par une petite simulation. La fonction `premier_succes` est en fait la fonction N .

```
In [9]: def premier_succes(p):  
        count = 0  
        while rand(p) != 1: count += 1  
        return count
```

```
In [10]: s = [premier_succes(0.05) for k in range(10000)]  
plt.plot(s)
```

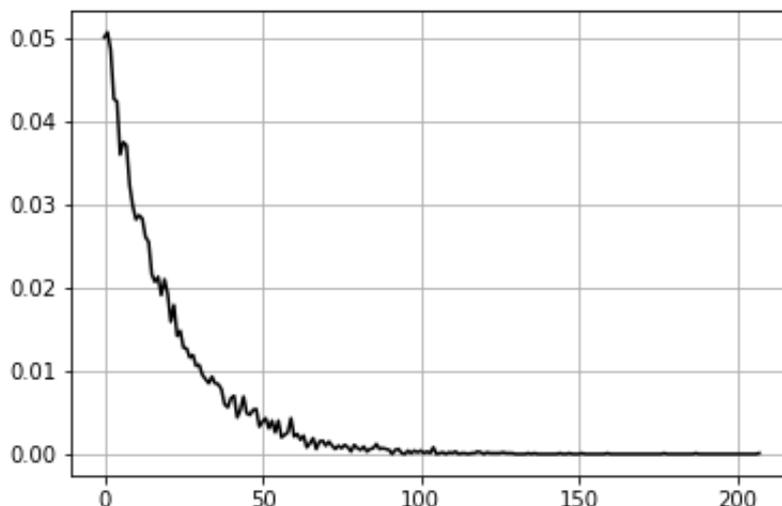
```
Out[10]: [<matplotlib.lines.Line2D at 0x108c30278>]
```



Bon, c'est le fouillis. Mais c'est la loi de N qui nous intéresse, pas les valeurs prises individuellement par N . Déterminons donc les probabilités $P(N = k)$ pour $k \in \mathbb{N}$. La fonction `histogramme` fait le travail.

```
In [11]: def histogramme(s):  
        m = max(s)  
        h = (m + 1) * [0]  
        c = 0  
        for k in s:  
            h[k] = h[k] + 1  
            c = c + 1  
        for i in range(len(h)): h[i] = h[i] / c  
        return h
```

```
In [12]: s = [premier_succes(0.05) for k in range(10000)]
plt.plot(histogramme(s), 'k-')
plt.grid()
plt.show()
```



C'est plus joli. La fonction $k \mapsto P(N = k)$ est assez décroissante et tend super vite vers 0 lorsque k tend vers l'infini. Apparemment. Vrai ou faux ?

2.2 Un peu de théorie

Avant de démarrer, voici un petit résultat sur les séries.

Proposition : soit $q \in]-1, 1[$. La série $\sum q^n$ est une série convergente, et $\sum_{n=0}^{\infty} q^n = \frac{1}{1-q}$.

Démonstration : Soit $N \in \mathbb{N}$. On a $\sum_{n=0}^N q^n = \frac{1-q^{N+1}}{1-q}$ et cette quantité tend bien vers $\frac{1}{1-q}$ lorsque N tend vers l'infini. Une telle série est appelée série géométrique de raison q .

Après cette petite digression, revenons à notre fonction de premier succès. Nous reprenons les notations de l'introduction. Soit $\omega \in \Omega$. On a $N(\omega) = k$ si et seulement si pour tout $i < k$, $X_i(\omega) = 0$, et $X_k(\omega) = 1$. Ainsi, $\{N = k\} = \left(\bigcap_{i=1}^{k-1} \{X_i = 0\}\right) \cap \{X_k = 1\}$. Mais les X_i sont indépendantes, la probabilité de l'intersection est donc le produit des probabilités :

$$P(N = k) = P(X_k = 1) \prod_{i=1}^{k-1} P(X_i = 0) = p(1-p)^{k-1}$$

On a fait un petit oubli ... et $P(N = \infty)$? Eh bien $\{N = \infty\}$ est le complémentaire de la réunion de tous les événements $\{N = k\}$, événements deux à deux incompatibles. Donc, $P(N = \infty) = 1 - \sum_{k=0}^{\infty} P(N = k)$. Poursuivons :

$$P(N = \infty) = 1 - \sum_{k=0}^{\infty} p(1-p)^k = 1 - p \sum_{k=0}^{\infty} (1-p)^k$$

On reconnaît là une série géométrique de raison $0 \leq 1-p < 1$. Donc

$$P(N = \infty) = 1 - p \frac{1}{1 - (1-p)} = \dots 0$$

L'événement $\{N = 0\}$ n'est pas impossible, mais il est de probabilité nulle.

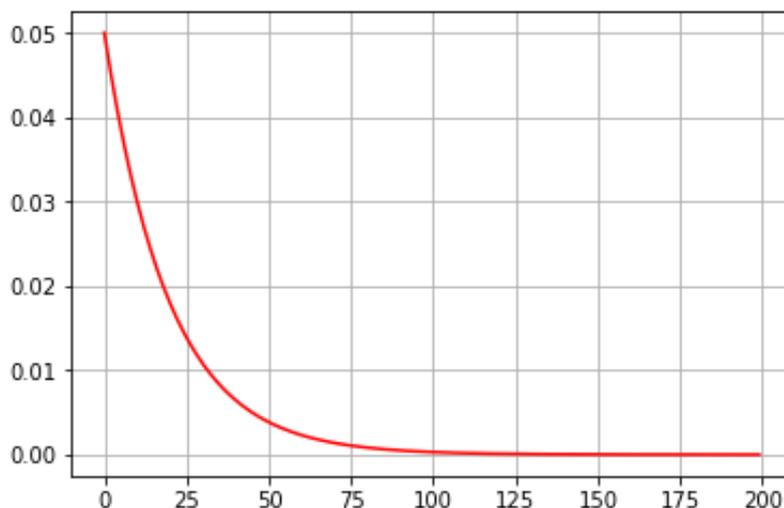
Tout ceci suggère la définition suivante :

Définition : On dit qu'une variable aléatoire X à valeurs dans \mathbb{N} suit une loi géométrique de paramètre $p \in]0, 1]$ lorsque, pour tout $k \in \mathbb{N}$, $P(X = k) = p(1-p)^{k-1}$.

Voici une illustration de la loi géométrique ...

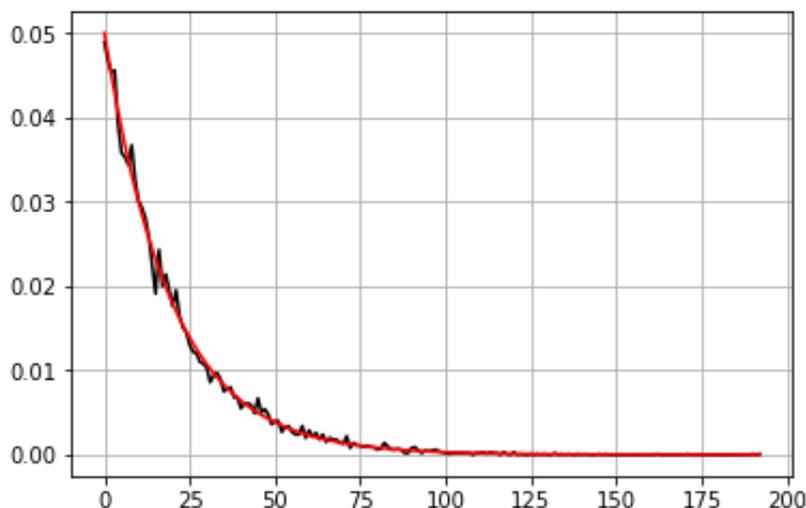
```
In [13]: def geom(p, k):
         return p * (1 - p) ** (k - 1)
```

```
In [14]: s = [geom(0.05, k) for k in range(1, 10000)]
         plt.grid()
         plt.plot(s[:200], 'r-')
         plt.show()
```



Superposons la simulation et la théorie.

```
In [15]: n, p = 10000, 0.05
s = [premier_succes(p) for k in range(n)]
s1 = [geom(p, k) for k in range(1, n)]
plt.plot(histogramme(s), 'k-')
plt.plot(s1[:max(s) + 1], 'r-')
plt.grid()
plt.show()
```



Ça colle. Et même plutôt bien. Et le loto dans tout ça ?

2.3 La propriété d'oubli de la loi géométrique

Soit X une v.a. suivant une loi géométrique de paramètre p . Soit $n \in \mathbb{N}$. On a $P(X > n) = \sum_{k=n+1}^{\infty} p(1-p)^{k-1} = p(1-p)^n \sum_{k=n+1}^{\infty} (1-p)^{k-1-n}$. Le changement d'indice $k' = k - n - 1$ dans la série nous montre que sa somme est $\frac{1}{1-(1-p)} = \frac{1}{p}$. Ainsi, $P(X > n) = (1-p)^n$.

Prenons maintenant deux entiers m et n . On a

$$P(X > m + n | X > n) = \frac{P(X > m+n, X > n)}{P(X > n)} = \frac{P(X > m+n)}{P(X > n)} = \frac{(1-p)^{m+n}}{(1-p)^n} = (1-p)^m = P(X > m).$$

On a utilisé dans notre calcul le fait que $\{X > m + n\} \subset \{X > n\}$. Donc $\{X > m + n\} \cap \{X > n\} = \{X > m + n\}$.

Revenons à notre loto gagnant. Quelle est la probabilité de devoir jouer au moins $m + n$ fois avant de gagner, sachant qu'on a perdu n fois ? Eh bien c'est la probabilité de devoir jouer au moins m fois avant de gagner, sachant rien du tout. La morale de l'histoire c'est que ce n'est pas parce qu'on sait qu'on a déjà joué avant qu'on a plus de chances de gagner après. Et donc celui qui n'a jamais joué a autant de chances de gagner que celui qui a passé sa vie à perdre :-).

