

La simulation en statistique (suite)

Les nombres éleatoires

Bernard Egger

Simuler, c'est substituer au réel un virtuel contrôlé, avons-nous dit dans l'article précédent.

Cette opération demande un certain matériel. Les nombres au hasard en sont à coup sûr un élément essentiel. Sous forme de tables, ou à partir d'outils informatiques, ils sont indissociables de l'activité de simulation. Chacun sait que la naissance des ordinateurs doit beaucoup au besoin impérieux de disposer d'une liste considérable de nombres aléatoires afin de simuler les réactions en chaîne dans la fission nucléaire. Et il y a une certaine ironie à penser que simuler le hasard a participé grandement dans la création de ces objets purement déterministes que sont les ordinateurs.

Car la question est bien là : ce dont il s'agit c'est de simuler le hasard. Face à une liste de nombres, la question que l'on se pose n'est pas : « a-t-on une liste de nombres au hasard ? », question qui n'aurait d'ailleurs pas beaucoup de sens, mais plutôt : « ces nombres peuvent-ils être considérés comme des nombres au hasard ? ». Cette formulation a alors deux conséquences essentielles : il faut d'abord que nous soyons capable de définir ce que l'on considère comme une liste de nombres au hasard ; ceci fait toute liste répondant aux critères précédents pourra être considérée comme liste de nombres au hasard, quel que soit le moyen utilisé pour la produire. Ces critères vont se traduire sous forme de tests que devra remplir une liste pour être une candidate acceptable à la fonction de liste de nombres aléatoires. Un mode de génération de listes de nombres aléatoires est jugé au travers de nombreuses listes qu'il a permis de construire et des résultats obtenus par ces listes à de nombreux tests.

Toutefois comme le fait remarquer Ivar IKELAND dans « Au hasard », on ne peut espérer que tout soit parfait car : « L'ordinateur peut tromper un test tout le temps, et tous les tests quelque temps, mais il ne peut pas tromper tous les tests tout le temps. » et plus loin, « le hasard, lui, saute spontanément tous les obstacles que l'on met sur son chemin ».

Tester la ressemblance avec le hasard

Qu'attendons-nous du hasard ? En premier lieu l'imprévisibilité, mais une imprévisibilité ordonnée. Cette double contrainte a pour traduction mathématique : l'équiprobabilité. Le premier test à effectuer est donc celui de l'uniformité de la répartition. Sur un tableur (Excel par exemple), on peut facilement réaliser l'expérience pour les nombres aléatoires fournis par le logiciel

La fonction ALEA() retourne des nombres compris entre 0 et 1. Si l'on procède à une subdivision régulière de $[0; 1[$, par exemple en 20 intervalles, et que l'on procède au tirage d'une liste de 2000 nombres, on s'attend à trouver des effectifs à peu près égaux à 100 dans chacun des intervalles $[\frac{k}{20}, \frac{k+1}{20}]$, avec k entier variant de 0 à 19.

Pour simplifier le comptage, on peut s'apercevoir simplement que $x \in [\frac{k}{20}, \frac{k+1}{20}]$ si et seulement si $E(20x) = k$ (E représentant la partie entière). Par exemple sur trois échantillons de 2000 nombres aléatoires on obtient les résultats suivants :

0	1	2	3	4	5	6	7	8	9	10	11	12	13
95	93	94	100	111	104	103	95	96	100	108	91	108	86
116	80	88	87	100	92	116	112	96	93	91	96	127	97
101	97	91	95	105	101	112	92	94	90	111	89	94	110

14	15	16	17	18	19
112	91	97	108	108	100
97	110	111	96	88	107
111	102	91	95	110	109

Les résultats obtenus sont satisfaisants, car n'oublions pas qu'il ne s'agit pas de dire : ces résultats prouvent que la répartition est uniforme, mais plutôt à la lecture de ces résultats on ne peut pas rejeter l'hypothèse que la répartition soit uniforme. (Lire à ce propos l'article de Daniel SCHWARZ dans www.unil.ch/penombre/08/03.htm).

En effet la fluctuation d'échantillonnages, ces caprices du hasard selon Daniel SCHWARZ, même dans le cas d'une répartition totalement uniforme, écarte le plus souvent des valeurs égales à 100 pour chaque cas.

Quoi qu'il en soit, il est bien clair que l'on ne saurait en rester à une sensation : il faut des outils mathématiques permettant de confirmer la valeur du modèle.

Les pièges de l'uniformité

Il s'agit bien entendu du premier test à effectuer, mais son résultat globalement positif ne peut pas suffire, comme le montre l'exemple ci-dessous.

Dans le dernier numéro de 3'33, magazine consacré aux calculatrices Casio, un article s'intitule : *Un algorithme de nombres aléatoires*.

L'auteur nous y présente une suite récurrente : $\begin{cases} 0 < u_0 < 1 \\ u_{n+1} = D(147u_n) \end{cases}$ avec $D(x)$ = partie décimale de x .

Cet algorithme est supposé produire d'une façon très simple des nombres aléatoires, et la vérification qui est proposé est bien sûr celle du test d'uniformité.

Examinons sur Excel ce que cela donne (du moins après avoir réglé quelques problèmes dont je parlerai à la fin). Pour une liste de 2000 nombres à partir de $u_0 = 0,124679$ comme le préconise l'auteur, on trouve les résultats suivants :

0	1	2	3	4	5	6	7	8	9	10	11	12	13
102	90	89	97	112	104	99	95	93	125	85	94	107	108

14	15	16	17	18	19
107	101	101	109	87	95

Résultats qui à première vue ne sont pas vraiment moins satisfaisant que les nombres au hasard fournis par Excel. Mais si l'on examine ces « nombres aléatoires » avec un peu plus d'attention, un « petit » problème apparaît, alors qu'on ne l'avait pas identifié immédiatement (c'est le cas de l'auteur) :

0,124679
0,327813
0,188511
0,711117
0,534199
0,527253
0,506191
0,410077

Voici les huit premiers nombres de la liste.

Il saute aux yeux qu'ils ont tous 6 chiffres après la virgule.

Hasard ? Certainement pas !

Que se passe-t-il en fait ? Qu'est donc la partie décimale du produit de l'un de ces nombres par 147 ? Par exemple de 0,188511 par 147 : bien évidemment le reste de 188511×147 dans la division euclidienne par 1000000.

Ce qui implique bien entendu qu'il n'y a au plus que 1000000 « nombres aléatoires » possibles.

En fait, des considérations simples sur le chiffre des unités dans la multiplication par 7 permettent de réduire dans un premier temps ce nombre à 250000.

Excel ne nous est plus d'une aide suffisante.

Le logiciel Derive permet de mieux cerner le problème

```

deci(x) := x - FLOOR(x)
ITERATE(deci(147.x), x, 0.124679, 100000) =
.124679
ITERATE(deci(147.x), x, 0.124679, 100001) =
.327813

```

La fonction `deci` correspond à la partie décimale, et `ITERATE`, de syntaxe : `ITERATE(f(x), x, a, n)`, retourne le terme de rang n de la suite récurrente $u_{n+1} = f(u_n)$ avec $u_0 = a$.

Voilà une liste de « nombres aléatoires » qui se réduit comme une peau de chagrin : au plus 100000 maintenant.

Bien entendu sur 2000 ou 3000 valeurs comme le propose l'article, on n'y voit que du feu. Pourtant nous avons ici *une suite périodique* (quel que soit le choix du terme initial, la seule différence étant alors la longueur de la période).

Et contrairement à ce que dit l'auteur, on peut dire bien d'autres choses sur cette suite que simplement qu'elle est bornée.

Si elle n'est pas un bon exemple d'un générateur de nombres aléatoires, elle en est un magnifique contre-exemple (et un bon exercice d'arithmétique).

Construire le hasard

Toutefois, et nous y reviendrons, la démarche imparfaite de l'article met le doigt sur ce qu'est « notre hasard » : un chaos déterministe, si l'on entend par chaos comme le fait Pascal DELAHAYE sur le site :

www.crdp.ac-grenoble.fr/imel/delahaye/chaos/infogene.htm

« un phénomène entièrement déterministe qui prend une allure aléatoire » (c'est-à-dire qui correspond à ce que l'on attend du « vrai hasard »).

Les générateurs des calculatrices sont des suites récurrentes chaotiques.

Celle de l'article est bien récurrente, mais assez peu chaotique. Pourtant si l'on n'y prend pas garde, elle peut paraître acceptable.

Un point essentiel aurait pu permettre facilement de la rejeter : on peut attendre d'une liste de nombres vraiment au hasard que les décimales d'un rang donné suivent elles aussi une répartition uniforme. Dans cet exemple la sixième décimale est plutôt gênante.

Cet article a donc une vertu (involontaire) : en dévoilant le mode de construction des générateurs qui sont essentiellement des suites chaotiques (si ce n'est plus tout à fait vrai pour certains logiciels sophistiqués, cela l'est encore pour les tableurs et les calculatrices, ce qui explique d'ailleurs que l'on peut théoriquement ou pratiquement réinitialiser ces générateurs-suites, même si les constructeurs ne le permettent pas toujours), il met en évidence un problème important : la précision permise par la calculatrice ou par le logiciel implique la finitude des possibles et donc la périodicité de la suite.

On peut donc peut-être imaginer qu'avec le choix d'un terme initial ayant 15 décimales par exemple après la virgule, l'algorithme proposé par l'article pourrait « fonctionner » aussi bien que le générateur de la calculatrice.

Pour trancher il va falloir affiner notre test. Ou plutôt en construire d'autres. Dans le prochain article, nous parlerons de deux d'entre eux, le premier pourrait s'appeler le test d'indépendance, et le deuxième est connu sous le nom de test du poker.

1 Les déboires des tableurs

Pour terminer il est bon de signaler les déboires d'Excel vis à vis de ... la multiplication.

	A
1	0,107
2	0,729
3	0,163
4	0,961
5	0,266999997
6	0,248999628

Le tableau ci-contre contient les premiers termes de la suite précédente avec $u_0 = 0,107$ calculé sur Excel (à gauche) et sur Star Office (à droite). La formule de la cellule A2 est bien sûr : $=147*A1-ENT(147*A1)$

0,107000000
0,729000000
0,163000000
0,961000000
0,267000003
0,249000441

Puis on recopie la formule vers le bas.

Nous avons expliqué plus haut que le nombre de décimales des termes successifs de la suite ne changeait pas. Quelle surprise donc avec A5, puis A6.

Il existe plusieurs autres situations où ce problème existe ; essayez par exemple la fonction « tente »

définie sur $[0; 1]$ par :
$$\begin{cases} f(x) = 2x & \text{si } x \leq \frac{1}{2} \\ f(x) = 2 - 2x & \text{si } x \geq \frac{1}{2} \end{cases} .$$

Construisez alors une suite récurrente de premier terme 0,26 par exemple et telle que $u_{n+1} = f(u_n)$. Examinez alors comment les erreurs dans la multiplication par 2 conduisent au bout d'une soixantaine de recopies à des résultats totalement aberrants (sur les autres tableurs également).

L'erreur commence très tôt comme le montre le tableau ci-dessous qui reprend le précédent avec 15 décimales :

0,107000000000000
0,728999999999999
0,162999999999830
0,9609999999828030

Comment y remédier ? En utilisant systématiquement l'instruction ARRONDI :

	A
1	0,10700000000000000000000000000000
2	0,72900000000000000000000000000000
3	0,16300000000000000000000000000000
4	0,96100000000000000000000000000000
5	0,26700000000000000000000000000000
6	0,24900000000000000000000000000000
7	0,60300000000000000000000000000000

La cellule A2 contient maintenant la formule :

=ARRONDI(147*A1-ENT(147*A1);14)

L'affichage à 30 chiffres reste correct tout le temps.

À suivre ...