le cnam

ALGORITHMES DE TIRAGE

Philippe Périé, Sylvie Rousseau & Gilbert Saporta

STA108, 9 novembre 2012

NOMBRES ALEATOIRES et PSEUDO-ALEATOIRES

- Utiles pour réaliser des tirages et simuler des phénomènes aléatoires
- Nombres aléatoires: suite de réalisations indépendantes d'une variable uniforme sur [0;1]
  - Peuvent être obtenus par des procédés physiques:
    - roues de loterie,
    - éclairage à intervalles irréguliers d'un disque divisé en 10 secteurs isométriques et numérotés de 0 à 9 : table de Kendall et Babington Smith

Nombres pseudo aléatoires

- Procédés déterministes mais fournissant une suite de nombres en apparence iid sur [0; 1]
  - Suites mathématiques
    - décimales de π, des tables de logarithmes
  - Procédés arithmétiques
    - Milieu du carré de Von Neumann (1946)

- On part d'un nombre entier
- On l'élève au carré
- On extrait les chiffres du centre comme nombres aléatoires.

- Exemple : $x_0 = 7534$

$(7534)^2$	=	56 7611 56
$(7611)^2$	=	57 9273 21
$(9273)^2$	=	85 9885 29
$(9885)^2$	=	97 7132 25

....

- d'où la suite 7611 9273 9885 7132
- Inconvénients majeurs : dépendance au nombre de départ et régularités nombreuses (permanence de 0 ou de séries particulières).

Méthodes de congruence

Elles reposent sur des suites récurrentes :

- choix arbitraire d'un entier x_0 appelé germe (ou *seed ou graine*)
- génération d'une séquence (x_1 ,..., x_n) d'entiers :

$$X_{i+1} = a\, x_i + b \quad \text{(modulo } m\text{) pour } i = 1, \ldots, n ,$$

où *a, b* et *m* sont des entiers appelés respectivement multiplicateur, incrément et modulo.

On vérifie : $0 < x_i < m$ pour i 1, ..., n .

- Intérêt : les nombres $u_1 \ldots, u_n$ où $u = \dfrac{x_i}{m}$

forment un échantillon pseudo-aléatoire de la loi uniforme sur *[0,1]* si les entiers *a, b* et *m* sont « bien » choisis.

<u>Intuition de l'horloge</u> : les heures 9h et 21 sont congrues modulo 12

- Le procédé étant déterministe, ces nombres sont dits pseudo-aléatoires.
- *Exemple :* $x_0 = 1$; $a = 6$; $b = 0$; $m = 25$

  $x_0 = 1$ $x_1 = 6\,[25] = 6$ $x_2 = 36[25] = 11$

  $x_3 = 66[25] = 16$ $x_4 = 21$ $x_5 = 1 = x_0$

  Ce cycle a pour longueur 5.
- Remarque :
- La séquence x_i i =1,...,n contient au plus m termes distincts.
- Cette suite est donc périodique de période p avec $p \leq m$ Si $p = m$, la période est dite pleine.

- Choix des entiers *a, b* et *m* :

Ils sont déterminés de telle sorte que la séquence ait les meilleures propriétés possibles.

En particulier, *m* est pris aussi grand que possible pour assurer une grande variété de valeurs dans la suite *xi*

- **Hull et Dobell** (1962) ont montré que les séquences de période pleine sont obtenues si et seulement si :
  - *b* et *m* sont premiers entre eux,
  - *(a-1)* est un multiple de chaque nombre premier qui divise *m*
  - si *m* est un multiple de 4 alors *(a-1)* aussi

Un algorithme très usité est la méthode congruentielle de **Lehmer** (1948) qui pose *b = 0*.

- **Méthode de Lehmer :**

$$x_{i+1}=ax_i \ (m)$$

(Sur machines 32 bits m aussi grand que possible ➔ $m=2^{31}-1$)

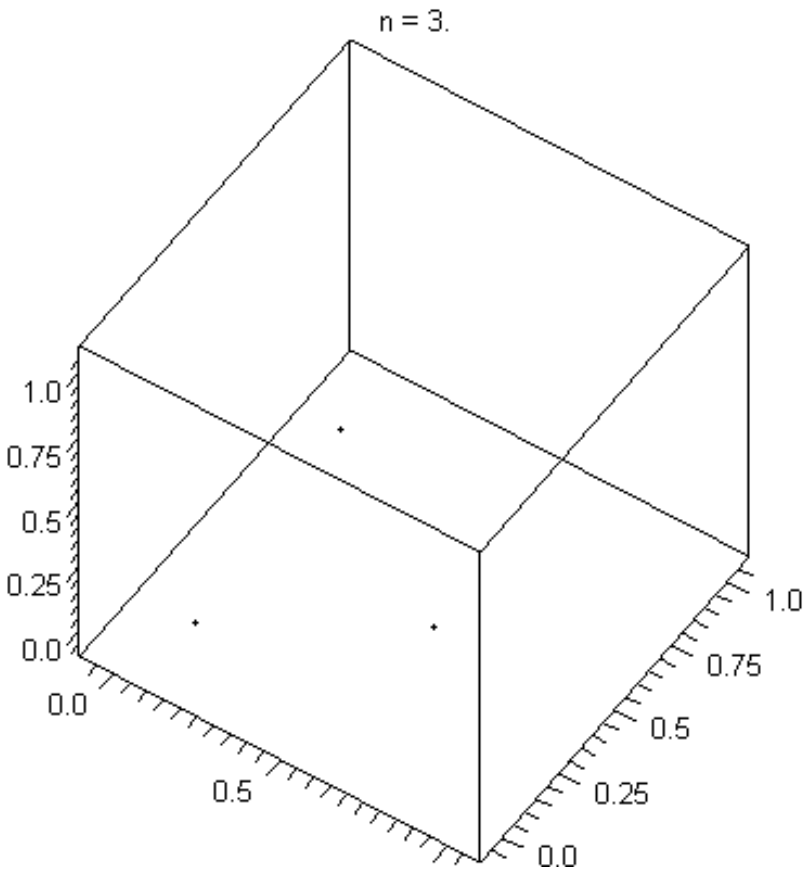
- choix classiques:
  - $a=7^5=16807$ $m=2^{31}-1$
  - $a=2^{16}+3=65539$ $m=2^{31}-1$
  - $a=279470273$ $m=4294967291$

Remarque : $a=2^{16}+3=65539$ $m=2^{31}-1$: **RANDU**

(introduit dans les années 1960, sur des machines IBM. Il est très impopulaire car il possède de nombreux biais auxquels ont dû faire face les personnes qui l'ont utilisé).

- **RANDU**

- a= $2^{16}+3=65539$ $m=2^{31}-1$
- $m = 2^{16} + 3$ ➔ $m^2=6m-9 \bmod 2^{31}$

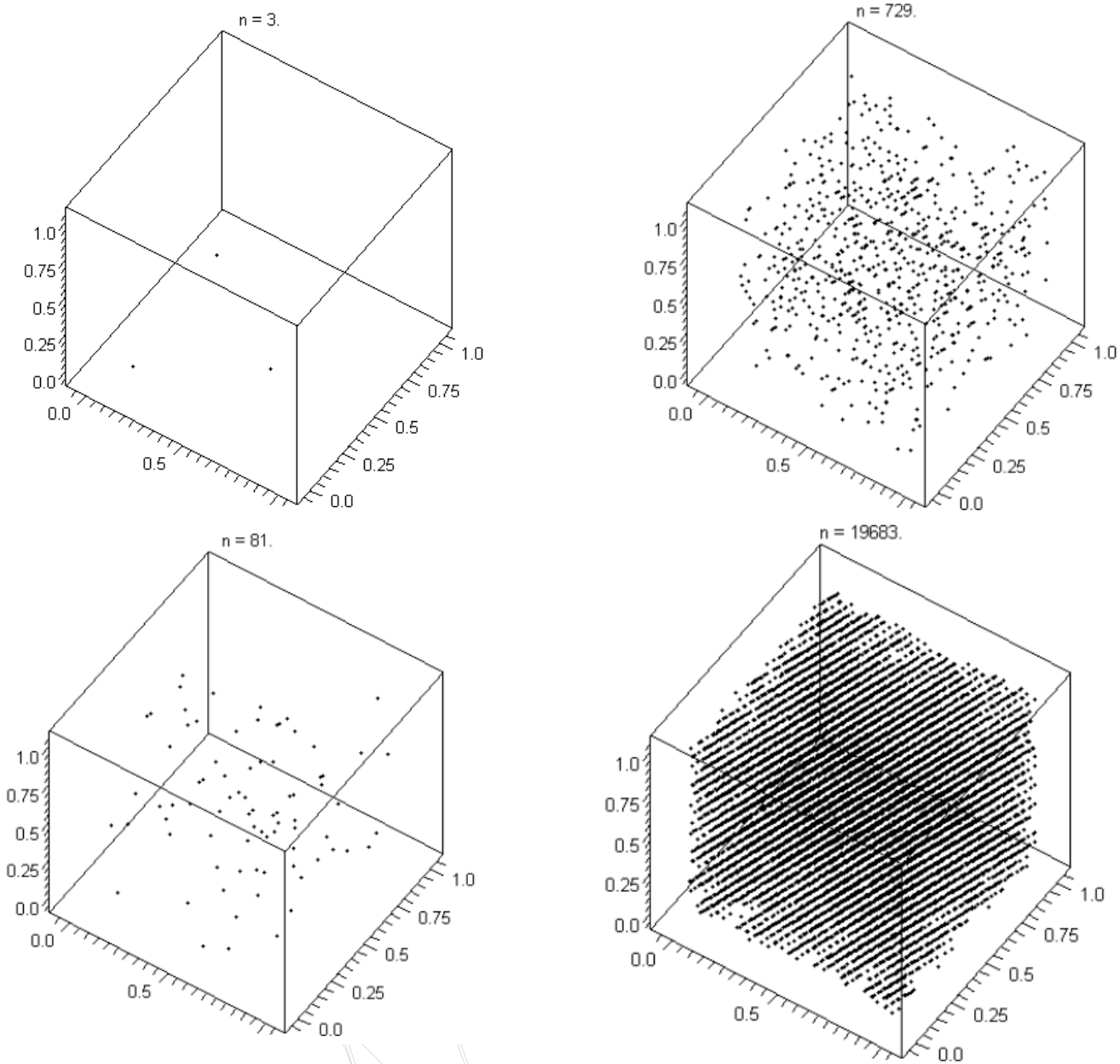


- Pb : trois nombres successifs X_n X_{n+1} et X_{n+2} vérifient toujours la relation $X_{n+2} = 6X_{n+1} - 9X_n$

- Cette relation donne un caractère ‘prédictif’ à la série pseudo aléatoire: par exemple, une modification des valeurs de X_n et X_{n+1} de l'ordre de 0,01, change la valeur de X_{n+2} d'au plus 0,15.

- Pour avoir un "bon" générateur, on souhaite une relation avec des coefficients beaucoup plus grands, de telle manière qu'une petite modification de X_n et X_{n+1} change complètement X_{n+2}

http://en.wikipedia.org/wiki/Image:Lcg_3d.gif#file

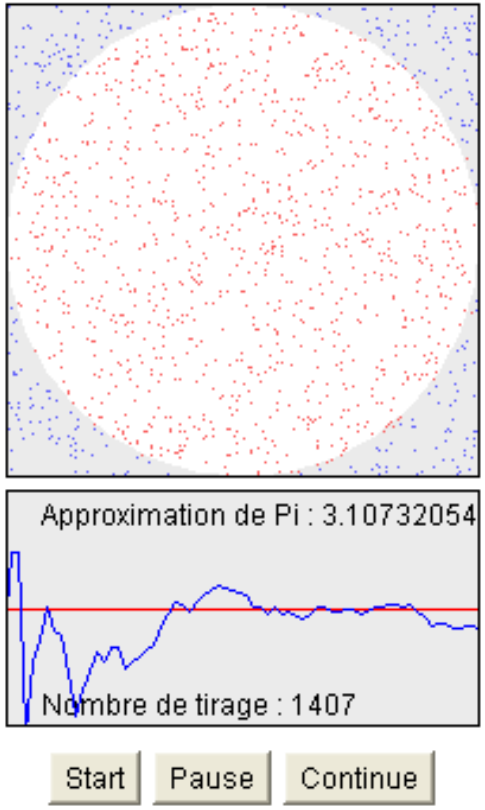


- Solutions variées: congruences avec retard $x_i = a\, x_{i-r} + b\ [m]$
  - *Exemple:* $r_{i+1} = (1664525 r_i + 1013904223)\ \ m = 2^{32}$ *(Numerical Recipes in C)*

- Nombreux tests pour valider le caractère uniforme et l'indépendance des réalisations
  - Chi-deux, Kolmogorov, tests de séquences, de non corrélation

estimation de π

- http://www-sop.inria.fr/mefisto/java/tutorial1/node15.html#SECTION00033120000000000000



Calcul d'intégrales: méthode de Monte Carlo

- **Première méthode :**

on simule *n* valeurs de *U*

$$I = \int_0^1 g(t)dt = E(g(U))$$

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} g(u_i)$$

- **Deuxième méthode:** fonction d'importance

T variable sur [0 ;1] de densité p(t)

$$I = \int_0^1 \frac{g(t)}{p(t)} p(t)dt = E\left(\frac{g(T)}{p(T)}\right)$$

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} \frac{g(t_i)}{p(t_i)}$$

Générateurs pseudo-aléatoires cryptographiques

- Doivent être capable de produire des séries dont le caractère pseudo aléatoire est moins discernable pour mériter ce titre
- ... Mais plus lents

- Un générateur congruenciel rapide et possédant de bonnes propriétés : Mersenne Twister (1997)
- Mais n'est pas considéré comme générateur cryptographique
- Utilisé dans SPSS à partir de la version 12

ALGORITHMES DE TIRAGE

- Qualités souhaitées:
  - Sans remise
  - Séquentiel
  - Rapide
  - Respecte les probabilités d'inclusion
  - De taille fixe
  - Utilisable si N est inconnu
  - Etc.

Une méthode inefficace : énumération puis sélection

(Yves Tillé, ‘Sampling Algorithms’ p 31)

- Si le plan de sondage est connu, et que la population n’est pas trop grande, une méthode pour sélectionner un échantillon est l’approche énumérative : énumérer tous les échantillons possibles, puis en sélectionner 1 au hasard.
- ... méthode pure et simple conceptuellement mais impossible dès que la population dépasse quelques dizaines

➔ L’objectif des algorithmes de tirage est de tirer un échantillon en respectant le plan de sondage et en évitant une énumération complète au préalable

Classes de méthodes (Yves Tillé pp 32 – 39)

- Martingales
- Algorithmes séquentiels
- Sélection pas à pas
- Par élimination
- Sondages réjectifs

Notion d'entropie

On appelle entropie du plan $p(.)$ *la quantité* $-\sum_s p(s) ln(p(s))$.

On montre aisément que I(p) est toujours positif.

- *Plus l'entropie est élevée, plus le plan de sondage est en un certain sens aléatoire*

- A défaut d'information auxiliaire, on peut chercher le plan le plus aléatoire (au sens de l'entropie) qui vérifie les probabilités d'inclusion fixées

Plans à probabilités égales sans remise

1. Tirage successif des unités

- On tire une unité au hasard parmi N : $p = 1/N$ *d'obtenir l'individu* $i_{(1)}$
- On ote cet individu de la base de sondage
- Et on fait un nouveau choix : $p = 1/(N-1)$ *d'obtenir* $i_{(2)}$
- etc ...

La probabilité d'obtenir les n individus $(i_{(1)}, i_{(2)}, \ldots i_{(n)})$ dans cet ordre est :

$$1/N \times 1/(N-1) \times \ldots \times 1/(N-n+1) = (N-n)!/N!$$

Comme il y a $n!$ permutations du n-uplet $(i_{(1)}, i_{(2)}, \ldots i_{(n)})$, la probabilité de sélectionner un échantillon de n unités vaut : $1/C_N^n$.

<u>*Avantage*</u> : on a bien défini un plan simple

<u>*Inconvénient*</u> : éxécution très longue (n lectures de fichiers et opérations de tri)

Plans à probabilités égales sans remise

- Tirage de Bernoulli:

  on tire N nombres aléatoires. L'unité i est retenue si $U_i < \pi$.

Exemple

k	π_k	u_k	I_k
1	0.4	0,323	1
2		0,500	0
3		0,361	1
4		0,780	0
5		0,012	1
6		0,252	1
7		0,578	0
8		0,997	0
9		0,112	1
10		0,645	0

$\mathbf{n_s = 5}$

• Tirage de Bernoulli

Avantages :

- facilité de programmation
- tirage i.i.d. des unités
- N n'a pas besoin d'être connu a priori

Inconvénients:

- La taille de l'échantillon obtenu est aléatoire
- Ce n'est pas vraiment un plan de sondage car $s=\emptyset$ est possible

↳ Eviter ce plan de sondage surtout si n petit sauf si on ne peut faire autrement

Remarques

- $\pi_k = \pi =$ constante
- $\pi_{kl} = \pi_k^2$ par indépendance des tirages
- Si $\pi = n / N$ alors $E(n_s) = n$

• Tri aléatoire

- On génère N réalisations d'une v.a de loi $U[0,1]$ sur toute la population

- On trie la base de sondage par valeurs croissantes (ou décroissantes) des réalisations $(u_1, ..., u_N)$ précédemment obtenues

- On retient les n premiers (ou derniers) individus du fichier

Avantages:

- plan simple
- inutile de connaître N a priori
- aisé à réaliser

Inconvénient : long si N est grand

- **Sélection-rejet**

si U_1<n/N on prend l'unité 1. Puis n=n-1 et N=N-1. On sélectionne l'unité 2 si U_2<n-1/N-1

Si U_1>n/N, on passe à l'unité 2 avec N=N-1. On sélectionne l'unité 2 si U_2<n/N-1 etc.

Exemple

k	π_k	u_k	j	$(n - j) / (N - k + 1)$	I_k
1	0.4	0,375	0	0,400	1
2		0,624	1	0,333	0
3		0,045	1	0,375	1
4		0,517	2	0,286	0
5		0,632	2	0,333	0
6		0,246	2	0,400	1
7		0,927	3	0,250	0
8		0,325	3	0,333	1
9		0,645	4	0,000	0
10		0,178	4	0,000	0
Total	**4**				**4**

j= nb d'unités déjà sélectionnées

Avantages: plan simple
une seule lecture de fichier suffit

Inconvénient : il faut connaître N a priori

• Méthode de mise à jour de l'échantillon

- On sélectionne les n premiers individus de la base de sondage
- On répète pour $k = n+1$ à N :
  - simulation de u selon $U[0,1[$
  - si $u < n / k$ alors :
    - sélectionner l'unité k
    - oter l'une des unités déjà tirées à probabilités égales ($p=1/n$)
  - $k = k + 1$

Exemple :

k	π_k	$I_k(1)$	u_k	n / k
1		1		
2		1		
3		1		
4		1		
5	0,4		0,354	0,8
6			0,025	0,667
7			0,987	0,571
8			0,153	0,5
9			0,485	0,444
10			0,800	0,4

Avantages: plan simple
inutile de connaître N a priori
intéressant si N est grand

Inconvénient : il faut réserver un vecteur de taille n

- Pas aléatoires

  Tirer U et trouver s tel que $U \leq 1 - \frac{C_{N-s-1}^{n}}{C_{N}^{n}}$

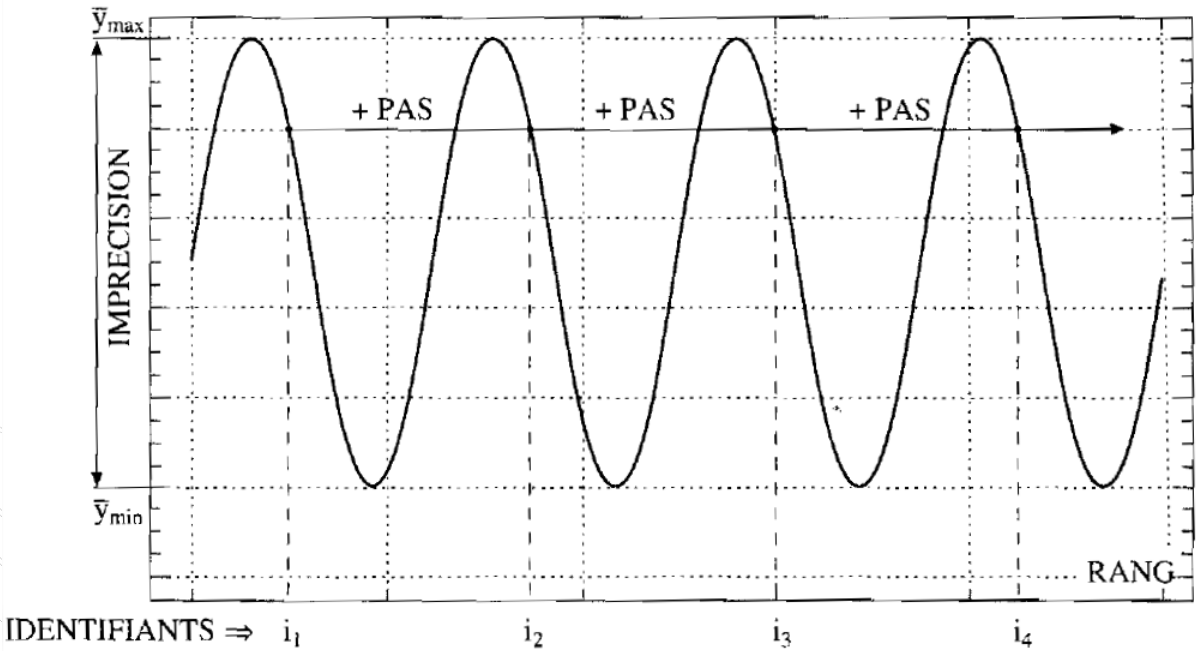
  sélectionner l'unité s+1, faire N=N-s-1 et n=n-1 etc.

- et aussi le tirage systématique...

- Tirage systématique
  - Définir un **pas** de tirage = N/n (entier par arrondi)
  - Tirer une unité au hasard au début du fichier entre 1 et **pas**
  - Sélectionner une unité tous les **pas**

  - Avantages: simplicité, N pas nécessairement connu a priori, peut être plus efficace que le tirage aléatoire si le fichier est trié selon une variable bien corrélée à la variable d'intérêt (cf cours sur le sondage en grappes)

- Inconvénients
  - Si périodicité dans le fichier (Ardilly)



Phénomène de périodicité dans un fichier

Probabilités inégales sans remise

- Infinité de plans de sondage pour des π_i fixés
  - Plus de 50 méthodes de tirage! Aucune ne satisfait tous les critères.
- Quelques techniques simples:
  - Tirage avec remise et conservation des unités distinctes mais taille non fixe
  - Rejet de l'échantillon si il y a des doublons mais probabilités d'inclusion non proportionnelles aux x_i

- Tirage successif sans remise:
  - On recalcule les probas d'inclusion après tirage de chaque individu. Si j est tiré: $\pi_i^{'} = \frac{\pi_i}{1-\pi_j}$
  - Ne respecte pas les probas d'inclusion d'ordre 1
- Tirage poissonnien: sélectionner i si $U_i < \pi_i$
  - $\pi_{ij} = \pi_i \pi_j$ variance simple
  - Mais taille non fixe

Tirage poissonnien

Exemple :

(S.Rousseau, 2004)

k	π_k	u_k	I_k
8	0,894	0,323	1
6	0,791	0,500	1
4	0,518	0,361	1
7	0,389	0,780	0
10	0,357	0,012	1
9	0,333	0,452	0
5	0,267	0,578	0
1	0,234	0,997	0
3	0,139	0,112	1
2	0,078	0,645	0
Total	**4**		**5**

Avantages : facilité de programmation
tirage i.i.d. des unités
N n'a pas besoin d'être connu a priori

Inconvénients: La taille de l'échantillon obtenu est aléatoire
s=ø est possible

↳ éviter ce plan de sondage surtout si n petit sauf si on ne peut faire autrement

Remarques

- $\pi_{kl} = \pi_k \pi_l$ par indépendance des tirages
- L'expression de la variance est assez simple
- Sondage d'entropie maximale (le plus aléatoire possible)

• Méthode de Sunter (généralisation de la méthode de sélection-rejet)

- au départ :
  - $k = 1$: *nombre d'unités de la population déjà examinées*
  - $j = 0$: *nombre d'unités de la population déjà sélectionnées*
  - $z = 0$: *cumul des probabilités d'inclusion*

    $z_k = V_{k-1} = \pi_1 + \pi_2 + \ldots + \pi_{k-1}$

- Puis tant que $j < n$ alors :
  - on génére u selon $U[0,1[$
  - si $u < \pi_k \times (n-j) / (n-z)$ alors :
    - sélectionner k
    - $j = j + 1$
  - $k = k + 1$
  - $z = z + \pi_k$

Exemple :

k	x_k	π_k	V_k	u_k	j	$(n - j) / (n - V_{k-1}) \times \pi_k$	I_k
1	10	0,8	0,8	0,375	0	0,8	1
2	10	0,8	1,6	0,624	1	0,75	1
3	8	0,64	2,24	0,045	2	0,533333333	1
4	6	0,48	2,72	0,517	3	0,272727273	0
5	6	0,48	3,2	0,632	3	0,375	0
6	4	0,32	3,52	0,246	3	0,4	1
7	2	0,16	3,68	0,927	4	0	0
8	2	0,16	3,84	0,325	4	0	0
9	1	0,08	3,92	0,645	4	0	0
10	1	0,08	4	0,178	4	0	0
Total	**50**	**4**					**4**

<u>*Avantages*</u> : une seule lecture de fichier suffit.

<u>*Inconvénient*</u> : il est possible que $\pi_k \times (n-j) / (n-z)$ dépasse 1 : ce cas est rare mais il amène à retenir $n-1$ unités et non n .

- **Méthode RHC (Rao, Hartley,Cochran)**

  - Pour un tirage à probabilités proportionnelles à la taille X
  - Trier les unités dans un ordre alétaoire
  - Tronçonner le fichier en n groupes successifs de N/n unités
  - Tirer dans chaque groupe **une** unité proportionnellement à la taille

  - Simple et performant
  - Remarque: procédé « inexactement proportionnel à la taille » car les groupes ne sont pas de même taille