Université de Strasbourg
M1 - Magistère
Statistique - études de cas

Ségolen Geffray
geffray@math.unistra.fr
Année 2014/2015

Sujet 10: statistique et nombres aléatoires

On parle de **nombres pseudo-aléatoires** lorsqu'ils sont engendrés de façon déterministe par un programme informatique. Ces générateurs sont généralement périodiques. La qualité du générateur est évaluée au moyen du critère qui stipule que le générateur est un bon générateur de nombres pseudo-aléatoires si on ne parvient pas à distinguer les nombres qu'il engendre d'une suite de nombres réellement aléatoires.

On souhaite pouvoir générer des suites $(u_n)_{n\geq 1}$ de nombres dans $[0,1]$ qui soient des réalisations d'une suite $(U_n)_{n\geq 1}$ de variables aléatoires i.i.d. uniformément distribuées sur $[0,1]$. Il se trouve qu'en réalité les représentations informatiques des nombres ne permettent pas d'accéder à tous les nombres réels mais seulement à un nombre fini d'entre eux. Par exemple, si l'on code les nombres sur N bits, on peut obtenir 2^N nombres différents. Si l'on ne peut pas accéder à tous les nombres réels de l'intervalle $[0,1]$, on souhaite néanmoins accéder à un grand nombre d'entre eux régulièrement espacés afin d'approcher autant que possible une répartition uniforme sur $[0,1]$. On considère pour cela l'ensemble des nombres $\left\{0, \frac{1}{m}, ..., \frac{(m-1)}{m}\right\}$ où m est un entier suffisamment grand. Ainsi, au lieu de considérer une suite (u_n) à valeurs dans $[0,1]$, on s'est ramené aux suites (x_n) à valeurs dans $\{0,1,...,m-1\}$. La suite (x_n) est donnée de manière algorithmique en général par une fonction $\phi : \{0,1,...,m-1\}^k \to \{0,1,...,m-1\}$, une valeur initiale $(x_1,...,x_k)$ qui déterminent une récurrence $x_{n+1} = \phi(x_n,...,x_{n-k+1})$. Notons que la suite obtenue est à valeurs dans un ensemble fini à savoir $\{0,1,...,m-1\}^k$ et, par conséquent, elle reviendra nécessairement à un k-uplet déjà atteint. Autrement écrit, la suite (x_n) aura un comportement périodique. Notons que plus la période est grande, plus le générateur a de bonnes qualités. On exige que chaque k-uplet soit atteint au cours d'une période, ce qui signifie que la suite (x_n) passe par chaque élément de $\{0,1,....,m-1\}$ avec la même fréquence au cours de chacune des périodes. Ceci correspond à l'hypothèse selon laquelle chaque U_n est uniformément distribuée sur $[0,1]$.
La fonction ϕ définit le générateur de nombres aléatoires. La valeur initiale est appelée germe ou graine (*seed*). Sa valeur est souvent modifiable par l'utilisateur afin d'obtenir de nouvelle suites ou de reproduire une suite donnée.

Pour évaluer la qualité d'équi-répartition de l'algorithme d'échantillonnage, on peut dans un premier temps tracer l'histogramme des valeurs $(u_1,...,u_n)$ générées. On s'attend à ce qu'il soit uniformément proche de 1. Dans un second temps, une procédure classique consiste à comparer la fonction de répartition empirique de l'échantillon $(u_1,...,u_n)$ généré avec la fonction de répartition théorique de la loi $\mathcal{U}(0,1)$. Rappelons que la fonction de répartition théorique de la loi $\mathcal{U}(0,1)$ est définie sur $\mathbb{R}$ par

$$F : t \to \begin{cases} 0 & \text{si } t < 0, \\ t & \text{si } 0 \leq t < 1, \\ 1 & \text{si } t \geq 1. \end{cases}$$

Rappelons également que la fonction de répartition empirique de l'échantillon $(U_1, ..., U_n)$ est définie pour $t \in \mathbb{R}$ par

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} I(U_i \leq t) .$$

Notons G la loi des variables (U_n) dont (u_n) est une réalisation.
Considérons le problème de test H_0: $G = F$ (adéquation à F) contre H_1: $G \neq F$ (non-adéquation à F).

La statistique de test de Kolmogorov-Smirnov associée à ce problème de test est

$$K_n = \sup_{t \in \mathbb{R}} \left\{ \sqrt{n} |F(t) - F_n(t)| \right\} .$$

Intuitivement, lorsque la valeur de K_n calculée à partir de l'échantillon généré, notée k_n, est petite, l'hypothèse d'uniformité est plausible. Dans le cas contraire, il faut remettre en cause la méthode. Déterminons la p-valeur du test de Kolmogorov-Smirnov:

$$p = 1 - F_{K_n}(k_n)$$

en notant F_{K_n} la fonction de répartition (théorique) de la variable aléatoire K_n. Si la p-valeur obtenue est inférieure à 5%, on rejette l'hypothèse d'adéquation à la loi uniforme.
Le test de Kolmogorov-Smirnov est implémenté dans le logiciel `R` dans la fonction `ks.test`. Lorsque les valeurs générées sont stockées dans un vecteur u, le test d'adéquation à la loi uniforme peut être effectué au moyen de l'instruction:
`ks.test(u,"punif")`

La statistique de test de Cramer-Von Mises (bien pour évaluer l'ajustement global) est:

$$C_n = \int (F(x) - F_n(x))^2 \, dF(x) = \frac{1}{12n} \sum_{i=1}^{n} \left(\frac{2i-1}{2n} - F(X_{(i,n)} \right)^2$$

en notant $X_{(1,n)} \leq X_{(2,n)} \leq ... \leq X_{(n,n)}$.

Le test de Cramer-Von Mises est implémenté dans le logiciel `R` dans le package `goftest` dans la fonction `cvm.test`. Lorsque les valeurs générées sont stockées dans un vecteur u, le test d'adéquation à la loi uniforme peut être effectué au moyen de l'instruction:
`cvm.test(u,"punif")`

La statistique de test d'Anderson-Darling (sensible aux écarts dans les queues de distribution) est:

$$A_n = n \int \frac{(F_n(x) - F(x))^2}{F(x)(1 - F(x))} dF(x) = -n - \frac{1}{n} \sum_{i} \left(\log(F(X_{(i,n)}) + \log(1 - F(X_{(n-i+1,n)})) \right)$$

Le test d'Anderson-Darling est implémenté dans le logiciel `R` dans le package `goftest` dans la fonction `ad.test`. Lorsque les valeurs générées sont stockées dans un vecteur u, le test d'adéquation à la loi uniforme peut être effectué au moyen de l'instruction:
`ad.test(u,"punif")`

Abordons maintenant la question de l'indépendance mutuelle des termes de la suite de nombres générés. Pour mémoire, les nombres réellement aléatoires, étant indépendants, ne présentent aucune corrélation entre eux. Un premier diagnostic graphique très simple consiste à représenter les éléments x_{n+1} en fonction des éléments x_n pour identifier d'éventuelles corrélations. On pourra également représenter les corrélations entre trois observations successives en traçant cette fois les points (x_k, x_{k+1}, x_{k+2}) dans l'espace (avec la fonction `plot3d` du package `rgl` du logiciel `R`).

Dans un second temps, on pourra estimer la fonction d'autocorrélation de la suite générée.
Admettons que cela a un sens[1] de définir les coefficients d'autocorrélation de la suite générées par

$$\rho(k) = \frac{\mathrm{Cov}(X_i, X_{i+k})}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_{i+k})}}, \quad k \in \mathbb{N}, \forall i \in \mathbb{N}.$$

On a toujours $\rho(0) = 1$ et $|\rho(k)| \leq 1$. On appelle (auto-)corrélogramme de la suite (X_i) le tracé des $\rho(k)$ en fonction de k pour $k \in \mathbb{N}$. Lorsque les termes de la suite (X_i) sont indépendants, la corrélogramme est nul au-delà de 0.

L'estimateur naturel de $\rho(k)$ pour $k \in \mathbb{N}$ est le coefficient d'autocorrélation empirique

$$\widehat{\rho}_n(k) = \frac{\sum_{i=1}^{n-k}(X_i - \overline{X}_n)(X_{i+k} - \overline{X}_n)}{\sum_{i=1}^{n}(X_i - \overline{X}_n)^2}.$$

NB: Au sens strict, le coefficient de corrélation empirique entre la série $(X_1, ..., X_{n-k})$ et la série décalée de k instants $(X_{k+1}, ..., X_n)$ est définie par:

$$\widetilde{\rho}_n(k) = \frac{\sum_{i=1}^{n-k}(X_i - \overline{X}_n^{(1)})(X_{i+k} - \overline{X}_n^{(2)})}{\sqrt{\sum_{i=1}^{n-k}(X_i - \overline{X}_n^{(1)})^2 \sum_{i=1}^{n-h}(X_{i+k} - \overline{X}_n^{(2)})^2}}$$

avec $\overline{X}_n^{(1)} = \dfrac{1}{n-k}\displaystyle\sum_{i=1}^{n-k} X_i$ et $\overline{X}_n^{(2)} = \dfrac{1}{n-k}\displaystyle\sum_{i=1}^{n-k} X_{i+k}$. Si k reste petit devant n, alors $\widetilde{\rho}_n(k) \approx \widehat{\rho}_n(k)$.

Lorsque les valeurs générées sont stockées dans un vecteur u, la fonction `acf` du logiciel `R` permet de calculer et de tracer les coefficients d'autocorrélation empiriques:

```
acf(u,type = "correlation")
```

Une première possibilité pour évaluer graphiquement l'hypothèse H_0 d'indépendance des X_i consiste à exploiter le fait que 95% des $\widehat{\rho(k)}_n$ devraient se situer entre les bornes $\pm 1.96/\sqrt{n}$ si H_0 était valide.
Box et Pierce (1970) ont développé la statistique de test suivante (appelé test de Portmanteau):

$$Q = n\sum_{k=1}^{k_0} \widehat{\rho}_n(k)^2.$$

Sous H_0, Q suit asymptotiquement une loi $\chi^2(k_0)$. Intuitivement, on rejette l'hypothèse H_0 si certains des $\widehat{\rho}_n(k)$ sont trop grands ie si Q est trop grande, ce qui fournit la région de rejet au

[1] en réalité, cette définition a un sens du moment que la suite (X_i) est stationnaire, notion non abordée en cours

niveau α suivante $\{Q > F^{-1}_{\chi^2(k_0)}(1-\alpha)$ ou de manière équivalente la p-valeur=$\mathbb{P}(\chi^2(k_0) \geq Q)$ (auquel cas on rejette H_0 si p-valeur $< \alpha$).

La loi de la statistique de test de Box-Pierce n'est valable sous H_0 qu'asymptotiquement. Box et Ljung (1978) ont donc proposé une modification pour petits échantillons

$$Q^* = n(n+2)\sum_{k=1}^{k_0} \frac{\widehat{\rho}_n(k)^2}{n-k_0}.$$

Sous H_0, Q^* suit asymptotiquement une loi $\chi^2(k_0)$. Comme précédemment, on rejette H_0 lorsque Q^* est trop grande, ce qui fournit la région de rejet suivante $\{Q > F^{-1}_{\chi^2(k_0)}(1-\alpha)$ ou de manière équivalente la p-valeur=$\mathbb{P}(\chi^2(k_0) \geq Q)$ (auquel cas on rejette H_0 si p-valeur $< \alpha$).

En pratique, on recommande que k_0 reste petit devant n et de ne pas dépasser $n/4$.

Les tests de Box-Pierce et de Ljung-Box sont implémentés dans le logiciel `R` dans la fonction `Box.test`. Lorsque les valeurs générées sont stockées dans un vecteur u, le test d'adéquation à la loi uniforme peut être effectué au moyen de l'instruction:

```
Box.test(u, lag = h, type = c("Box-Pierce"))
Box.test(u, lag = h, type = c("Ljung-Box"))
```

Vous devez bien sûr avoir au préalable déclarer la valeur de `h`.

Exercice 1.
Générer des suites de n variables selon les différents générateurs proposés ci-dessous, ceci pour différentes valeurs de n. Effectuer les diagnostics détaillés auparavant. Commenter les résultats obtenus.

1. La classe de générateurs la plus simple est celle des générateurs par congruence linéaire obtenue en fixant
$$x_{n+1} = (a\, x_n + b),\ \text{mod}\ m$$
où a et b sont des entiers. On obtient alors $u_n = \dfrac{x_n}{m}$.
   (a) utiliser $m = 81$, $a = 1$, $c = 8$,
   (b) utiliser $m = 1024$, $a = 401$, $c = 101$,
   (c) utiliser $m = 2^{32}$, $a = 1664$, $c = 1\,013\,904\,223$,

2. utiliser
$$\begin{cases} x_n = (1\,403\,580\, x_{n_2} - 810\,728\, x_{n-3}),\ \text{mod}\ m_1 \\ y_n = (527\,612\, y_{n-1} - 1\,370\,589\, y_{n-3}),\ \text{mod}\ m_2 \end{cases}$$
avec $m_1 = 2^{32} - 209$ et $m_2 = 2^{32} - 22853$. On obtient alors
$$u_n = \begin{cases} \dfrac{x_n + y_n + m_1}{m_1 + 1} & \text{si } x_n \leq y_n, \\ \dfrac{x_n + y_n}{m_1 + 1} & \text{si } x_n > y_n. \end{cases}$$

3. utiliser le générateur non-linéaire (L'Ecuyer, Hellekalek, 1998)

$$\begin{cases} x_{n+1} = a_1 x_n^3 + 1, \ \text{mod } m_1 \\ y_{n+1} = a_2 y_n^3 + 1, \ \text{mod } m_2 \end{cases}$$

avec $m_1 = 65\,519$, $a_1 = 512$, $m_2 = 65\,447$ et $a_2 = 27\,076$. On obtient alors

$$u_n = \frac{x_n}{m_1} + \frac{y_n}{m_2}, \ \text{mod } 1.$$