Année : 2008-2009 1er semestre
Niveau : MASTER IS 1ère année
Cours : Logiciel R
Enseignant : A. Illig

TP5 : Simulations en R, inverse de la fonction de répartition, méthode acceptation-rejet

Table des matières

1 Simulation de variables aléatoires

Dans certaines études statistiques, il est nécessaire de disposer d'échantillons de taille prescrite pour une loi de probabilité connue. Lorsque l'expérimentation physique ne le permet pas, on fabrique ces échantillons à l'aide d'un algorithme adapté à une implémentation informatique. La création d'un échantillon de loi donnée repose alors sur la génération de réalisations de loi uniforme $\mathcal{U}[0,1]$ obtenues à l'aide de générateurs de nombres pseudo-aléatoires.

1.1 Nombres pseudo-aléatoires

Un générateur de nombres pseudo-aléatoires est un algorithme qui génère une suite de nombres ayant en apparence certaines propriétés du hasard. Les méthodes les plus employées reposent sur une congruence linéaire (qui fournit malheureusement des suites périodiques). Ainsi, Derrick Henry Lehmer propose en 1948 la formule de récurrence suivante :

$$x_{n+1} = a\, x_n \mod m$$

avec x_0 la "graine" (généralement un nombre premier) utilisée pour initialiser l'algorithme. La période du générateur étant au maximum égale à m, en pratique on choisit m le plus grand possible. Par exemple, l'algorithme *Standard minimal* utilise $m = 2^{31} - 1$ et $a = 16807$.

Si la période est égale à d, les nombres $\frac{x_i}{d-1}$ sont des nombres compris entre 0 et 1 que l'on considère comme aléatoires. Cependant la qualité des nombres générés varie d'un générateur à l'autre et peut être étudiée au moyen de tests statistiques. L'algorithme *Mersenne Twister* (par défaut en R) inventé par Makoto Matsumoto et Takuji Nishimura en 1997 est particulièrement réputé pour sa qualité (période $d = 2^{19937} - 1$) et sa rapidité.

1.2 Inverse de la fonction de répartition

Remarquons que si X est une variable aléatoire de fonction de répartition F inversible, alors la variable $U = F(X)$ suit une loi uniforme $\mathcal{U}[0,1]$. En effet, pour tout $y \in [0,1]$:

$$\begin{aligned} \mathbb{P}(U \leq y) &= \mathbb{P}(F^{-1}(U) \leq F^{-1}(y)), \\ &= \mathbb{P}(X \leq F^{-1}(y)), \\ &= F(F^{-1}(y)) = y. \end{aligned}$$

Par conséquent, si $u_1, \ldots, u_n$ sont des réalisations de variables aléatoires indépendantes de loi $\mathcal{U}[0,1]$, $F^{-1}(u_1), \ldots, F^{-1}(u_n)$ constitue un échantillon de réalisations de la loi de X.

Exemple d'application : Cette méthode permet de simuler un échantillon de loi exponentielle $\mathcal{E}(\lambda)$ car $F^{-1}(y) = -\frac{Log(y)}{\lambda}$ pour $y \in]0,1[$. En effet, si u est une réalisation d'une loi uniforme $\mathcal{U}[0,1]$, alors $-4\, Log(u)$ est une réalisation d'une loi exponentielle $\mathcal{E}(\frac{1}{4})$.

1.3 Méthode acceptation-rejet

Soit f une fonction de densité et (X, U) une variable aléatoire bidimensionnelle de loi uniforme sur le sous-graphe :

$$\mathcal{G}_f = \{(x, u) \,|\, 0 \leq u \leq f(x)\}.$$

Alors la variable aléatoire unidimensionnelle X à pour densité f. En effet, la densité de X est la densité marginale :

$$\begin{aligned} f_X(x) &= \int_0^\infty 1_{\mathcal{G}_f}(x, u)du, \\ &= \int_0^{f(x)} du, \\ &= f(x). \end{aligned}$$

Pour obtenir un échantillon de la loi de densité f, on considère une autre densité g que l'on sait simuler et telle que pour tout x du support de f :

$$f(x) \leq Mg(x).$$

La méthode d'acceptation-rejet consiste à simuler des variables aléatoires $Y_1, Y_2 \ldots$ selon la loi g et des variables aléatoires $U_1, U_2, \ldots$ selon la loi uniforme $\mathcal{U}[0, 1]$ et de considérer $X = Y_k$ avec

$$k = \inf\{n \,|\, U_n \leq \frac{f(Y_n)}{M\, g(Y_n)}\}.$$

Si l'on note $V = M\, g(Y_k)\, U_k$, alors le couple (X, V) est uniforme sur $\mathcal{G}_f$ et d'après ce qui précède, X a pour densité f.

Exemple d'application : Soit une variable aléatoire X dont la densité f est proportionnelle à $\tilde{f}$:

$$\tilde{f}(x) = (2 + \sin^2(x)) \exp(-(2 + \cos^3(3x) + \sin^3(2x))x) 1_{[0,+\infty[}(x).$$

Puisque, pour tout x de $[-\pi, \pi]$,

$$\cos^3(3x) + \sin^3(2x) > -\frac{7}{4},$$

on en déduit que :

$$\begin{aligned} \tilde{f}(x) &\leq (2 + \sin^2(x)) \exp((-2 + \tfrac{7}{4})x) 1_{[0,\infty[}(x) \\ &\leq 3 \exp(-\tfrac{1}{4}x) 1_{[0,\infty[}(x) \\ &\leq 12 \underbrace{\frac{1}{4} \exp(-\frac{1}{4}x) 1_{[0,\infty[}(x)}_{\mathcal{E}(\frac{1}{4})} \end{aligned}$$

Notons g la densité d'une loi exponentielle $\mathcal{E}(\frac{1}{4})$, $\tilde{M} = 12$ la constante telle que $\tilde{f} \leq \tilde{M}g$, c la constante (inconnue) de proportionnalité entre f et $\tilde{f}$ telle que $f = c\tilde{f}$ et $M = c\,\tilde{M}$. Ainsi, pour tout x de $[0, \infty[$,

$$\frac{f(x)}{g(x)} = \frac{c\,\tilde{f}(x)}{g(x)} \leq c\,\tilde{M} = M.$$

Pour obtenir une réalisation de X, on construit des réalisations u_i d'une loi $\mathcal{U}[0,1]$ et des réalisations y_i d'une loi de densité g tant que :

$$u_i > \frac{f(y_i)}{M\,g(y_i)} = \frac{\tilde{f}(y_i)}{\tilde{M}\,g(y_i)} \text{ (Rejet)}$$

On obtient une réalisation $x = y_k$ de X au premier instant k où

$$u_k \leq \frac{f(y_k)}{M\,g(y_k)} = \frac{\tilde{f}(y_i)}{\tilde{M}\,g(y_i)} \text{ (Acceptation).}$$

On remarque sur cet exemple qu'il n'est pas nécessaire de connaitre la constante de proportionnalité c entre f et $\tilde{f}$ pour simuler des réalisations de X. Il suffit d'appliquer l'algorithme avec $\tilde{f}$ et $\tilde{M}$ même si $\tilde{f}$ n'est pas une densité ! Le programme ci-dessous permet de produire une réalisation de X (c.f. figure FIG.1 pour une illustration graphique).

```
> # Fonction de calcul de ftilde
> tildef=function(x){
> x=(2+(sin(x))^2)*exp(-(2+(cos(3*x))^3+(sin(2*x))^3)*x)
> ;x}
> # Fonction d'attente
> delay=function(n){
> for (i in 1:(n*10000)){
> a=(log(4))^3}}
> # Représentation de ftilde et de mtilde*g
> tildeM=12
> par(bg="lightblue")
> abs=seq(0,10,0.1)
> ord=tildef(abs)
> ord2=tildeM*dexp(abs,1/4)
> plot(abs,ord,type="l",col=3,main="Acceptation-Rejet")
> lines(abs,ord2,type="l",col=2)
> # Sur ce graphique apparaîssent successivement
> # les points rejetés (+) puis le point accepté (o)
> # dont l'abscisse est une réalisation de X
> # (enregistrée dans la variable x)
> u=runif(1)
> y=rexp(1,1/4)
> K=tildef(y)/(tildeM*dexp(y,1/4))
```

```
> while (u>K){
> points(y,u*tildeM*dexp(y,1/4),pch=1,col=2)
> delay(20)
> u=runif(1)
> y=rexp(1,1/4)
> K=tildef(y)/(tildeM*dexp(y,1/4))}
> x=y
> points(y,u*tildeM*dexp(y,1/4),pch=3,col=3)
> dev.copy2eps(file="AR.eps")
```

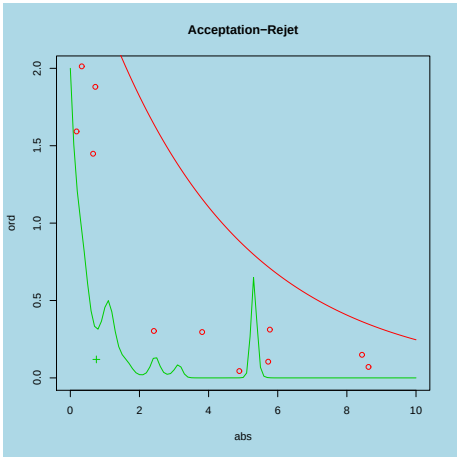

FIG. 1 – Illustration de la méthode d'acceptation-rejet

1.4 Test de Kolmogorov-Smirnov

Le test de Kolmogorov permet sur la base d'un échantillon $X_1, \dots X_n$ de fonction de répartition F de tester l'adéquation à une loi de fonction de répartition F_0 donnée. Une extension de ce résultat est le test de Kolmogorov-Smirnov qui permet de tester si deux échantillons indépendants $X_1, \dots X_n$ et $Y_1, \dots, Y_m$ sont issus d'une même loi. Notons F_n et G_m les fonctions de répartition des deux échantillons et considérons la variable aléatoire :

$$D_{n\,m} = \sup_x |F_n(x) - G_m(x)|.$$

Si les deux échantillons sont issus d'une même loi,

$$\mathbb{P}(\sqrt{\frac{n\,m}{n+m}} D_{n\,m} \leq y) \xrightarrow[n,\,m\to\infty]{} K(y)$$

où

$$K(y) = \sum_{k=-\infty}^{+\infty} (-1)^k \exp(-2k^2y^2).$$

Pour tester

$$H_0 : F = G \text{ contre } H_1 : F \neq G,$$

au niveau asymptotique α, on rejette H_0 lorsque $\sqrt{\frac{n\,m}{n+m}} D_{n\,m} > C$ avec C le quantile d'ordre $1-\alpha$ de la loi de fonction de répartition K.

En R, la commande `ks.test` appliquée à deux vecteurs de données permet également d'effectuer le test de Kolmogorov-Smirnov.

2 Distributions de probabilité en R

Le logiciel R permet pour un certain nombre de loi de probabilité (c.f. tableau TAB. 1) d'évaluer en un point la fonction de répartition et la fonction de densité, de calculer les quantiles et de générer des réalisations. Plus précisément, pour chacune des distributions du tableau TAB. 1, quatre commandes R préfixées par les lettres `d, p, q, r` sont disponibles. Ainsi pour la distribution `xxx` :

- `dxxx` donne la fontion de densité $f(x)$ pour une loi continue et la fonction de probabilité $\mathbb{P}(X = k)$ pour une loi discrète.
- `pxxx` donne la fonction de répartition $F(x) = \mathbb{P}(X \leq x)$.
- `qxxx` renvoie le quantile q tel que $F(q) = \alpha$ pour une distribution continue et le plus petit entier u tel que $F(u) \geq \alpha$ pour une distribution discrète.
- `rxxx` génère des réalisations aléatoires indépendante de loi `xxx`.

Ci-dessous, un petit exemple d'utilisation de ces commandes :

```
> # Calcul de la densité d'une loi normale centrée réduite
> # aux points d'abscisses x
> x=seq(-4,4,0.1)
> y=dnorm(x)
> plot(x,y, type=''l'', col=3)
> # Calcul de la fonction de répartition d'une loi
> # exponentielle de paramètre 3 aux points d'abscisses a
> a=seq(0,2,0.1)
> b=pexp(a,3)
> plot(a,b,type=''l'',col=4)
> # Quantile d'ordre 95% d'une loi de Student à 10 degrés
> # de liberté
> qt(0.95,10)
> # Création d'un vecteur X de taille 50 de réalisations d'une loi
> # Gamma(2,3)
> X=rgamma(50,2,3)
```

Distribution	Nom	Paramètre(s)	Valeurs par défaut
Beta	`beta`	`shape1, schape2`	
Binomiale	`binom`	`size, prob`	
Binomiale Négative	`nbinom`	`size, prob`	
Cauchy	`cauchy`	`location, scale`	0, 1
Khi-Deux	`chisq`	`df`	
Exponentielle	`exp`	`rate=1/mean`	1
Fischer	`f`	`df1, df2`	
Gamma	`gamma`	`shape, rate=1/scale`	-, 1
Géométrique	`geom`	`prob`	
Hypergéométrique	`hyper`	`m, n, k`	
Log-Normale	`lnorm`	`meanlog, sdlog`	0, 1
Logistique	`logis`	`location, scale`	0, 1
Normale	`norm`	`mean, sd`	0, 1
Poisson	`pois`	`lambda`	
Student	`t`	`df`	
Uniforme	`unif`	`min, max`	0, 1
Weibull	`weibull`	`shape, scale`	
Wilcoxon	`wilcox`	`m, n`	

Tab. 1 – Distributions en R

3 Exercices

3.1 Exercice 1

1. Attacher les données `faithful` et afficher les noms des variables.
2. Afficher les paramètres de position et de dispersion des données `eruptions`.
3. Tracer sur le même graphique l'histogramme d'aire égale à 1 des données `eruptions` et l'estimation par noyau gaussien aux points abscisses définis par `abs=seq(min(eruptions),max(eruptions),0.1)` (utiliser la selection automatique de la taille de la fenêtre `bw=''SJ''`, `help(bw.nrd)` affiche les différentes méthodes de sélection automatique).
4. Le tracé de l'histogramme fait apparaître deux sous-populations. Enregistrer dans un vecteur `longtime` les données `eruptions` strictement supérieures à 3. Représenter sur un autre graphique la fonction de répartition empirique de `longtime` (utiliser la commande `ecdf` pour calculer la fonction de répartition empirique et la commande `plot` avec les options `verticals=TRUE` et `do.points=FALSE`). Sur le même graphique tracer à l'aide de la fonction `lines` la fonction de répartition d'une loi normale de moyenne `mean(longtime)` et d'écart-type `sd(longtime)` aux points d'abscisses `longabs=abs[abs>3]`.
5. Tester au moyen du test de Kolmogorov l'adéquation de l'échantillon à une loi normale de moyenne `mean(longtime)` et d'écart-type `sd(longtime)`.
6. Utiliser les commandes `qqnorm` et `qqline` pour tester la normalité de l'échantillon `longtime` au moyen de la droite de Henry.
7. Détacher les données `faithful`.

3.2 Exercice 2

Nous simulons d'abord un échantillon de loi exponentielle par inverse de la fonction de répartition. Ensuite, nous simulons un échantillon de réalisations positives d'une loi $\mathcal{N}(0,1)$ en majorant la densité gaussienne sur $[0,\infty[$ par la densité exponentielle. Enfin, nous en déduisons un échantillon de réalisations d'une loi $\mathcal{N}(0,1)$ en choisissant ensuite le signe pour chaque réalisation positive à l'aide d'une réalisation de Bernoulli.

1. Pour $n = 100$ générer un vecteur `U` de réalisations d'une loi uniforme $\mathcal{U}[0,1]$. Puis, créer un vecteur `B` de taille n dont la composante i est égale à 1 si `U[i]>0.5` et 0 sinon. Tracer l'histogramme de densité de `B`.
2. Générer un vecteur `V` de taille $N = 1000$ de réalisations d'une loi uniforme $\mathcal{U}[0,1]$. En déduire, un échantillon `E` de taille N de réalisations d'une loi exponentielle de paramètre $\lambda = 1$. Tracer sur le même graphique l'histogramme de densité de `E` et la fonction de densité d'une loi exponentielle $\mathcal{E}(1)$.

3. Remarquer que si $\tilde{f}$ est la densité d'une loi $\mathcal{N}(0,1)$ et g la densité d'une loi exponentielle $\mathcal{E}(1)$, alors

$$\sup_{x\geq 0} \frac{\tilde{f}(x)}{g(x)} \leq \frac{\sqrt{e}}{\sqrt{2\pi}}.$$

   Notons f la densité d'une loi $\mathcal{N}(0,1)$ restreinte à $[0,+\infty[$. La fonction f dont le support est égal à $[0,+\infty[$ est proportionnelle à $\tilde{f}$ sur $[0,+\infty[$. Par conséquent, nous pouvons obtenir des réalisations d'une loi de densité f en appliquant l'algorithme d'acceptation-rejet à la fonction $\tilde{f}$.

   Générer un vecteur `W` de taille N de réalisations d'une loi uniforme $\mathcal{U}[0,1]$. A l'aide des vecteurs `E` et `W`, générer un vecteur `Normplus` de taille n de réalisations positives d'une loi $\mathcal{N}(0,1)$. Former ensuite un vecteur `Norm` à partir du vecteur `Normplus` tel que si `B[i]=1`, `Norm[i]=-Normplus[i]` et si `B[i]=0`, `Norm[i]=Normplus[i]`.

4. Créer un vecteur `NormR` de taille n de réalisations d'une loi normale centrée réduite avec la commande `rnorm`. Effectuer le test de Kolmogorov-Smirnov pour les échantillons `Norm` et `NormR`.