Année : 2008-2009 1er semestre
Niveau : MASTER IS 2ème année
Cours : Remise à niveau en R
Enseignant : A. Illig

TP4 : Remise à niveau en R

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 :

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

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 Simulation par inversion

La méthode que nous décrivons ici repose sur la connaissance de l'inverse ou de l'inverse généralisée de la fonction de répartition de la variable à simuler. Bien qu'elle s'applique à de nombreuses lois, elle ne permet pas de simuler les lois normales par exemple. Pour cela, il existe d'autres méthodes de simulation (voir méthode d'acceptation-rejet) que nous aborderons plus tard dans l'année.

1.2.1 Inverse de la fonction de répartition

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

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

Plus généralement, on a la proposition suivante :

Proposition 1. *Soit X une variable aléatoire réelle de fonction de répartition $F_X(t) = \mathbb{P}(X \leq t)$. On définit l'inverse généralisée F_X^{-1} de F_X sur $]0,1[$ par*

$$F_X^{-1} = \inf\{t \in \mathbb{R} \,|\, F_X(t) \geq x\}.$$

Alors, si U est une variable aléatoire uniforme sur $]0,1[$, $F_X^{-1}(U)$ a même loi que X.

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

1.2.2 Cas d'une variable aléatoire continue

En général, dans le cas d'une variable aléatoire continue et plus particulièrement pour les variables à densité, il faut d'abord calculer F_X^{-1} et simuler X à partir d'une loi uniforme sur $[0,1]$.

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

1.2.3 Cas d'une variable aléatoire discrète

Dans le cas d'une variable aléatoire discrète, la méthode est canonique. En effet, si X est une variable aléatoire discrète à valeurs dans $\{x_1, \ldots, x_r\}$ telle que pour tout $j \in 1 \ldots r$, $\mathbb{P}(X = x_j) = p_j$, alors l'inverse de la fonction de répartition est

$$F_X^{-1}(u) = \sum_{j=1}^{r} x_j \mathbb{1}_{[p_0+\cdots+p_{j-1}, p_1+\cdots+p_j[}(u)$$

où $p_0 = 0$.

Exemple 2. Appliquons cette méthode pour simuler une variable aléatoire X de loi de Bernoulli $\mathcal{B}(p)$. D'après ce qui précède, l'inverse généralisée est $F_X^{-1} = \mathbb{1}_{[1-p,1[}$. Ainsi, si u est une réalisation d'une loi uniforme $\mathcal{U}[0,1]$, alors $\mathbb{1}_{[1-p,1[}(u)$ est une réalisation d'une loi de Bernoulli $\mathcal{B}(p)$.

2 Distributions de probabilité en R

Le logiciel R permet pour un certain nombre de lois 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.

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

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 N(0,1) aux points 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 E(3)
> # aux points 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)
```

3 Exercices

3.1 Exercice 1

1. Considérons un étudiant qui tente de deviner les réponses aux questions d'un test.
   (a) Pour chaque question, supposons qu'il a 20% de chances de répondre correctement. A l'aide de la procédure de simulation des lois discretes et de la commande `runif`, simuler les réponses (correcte/incorrecte) d'un tel étudiant à un test comportant 20 questions.
   (b) Ecrire une fonction R permettant de simuler les réponses d'un étudiant qui tente de deviner les réponses aux n questions d'un test avec une probabilité de bonne réponse égale à p.
2. Supposons qu'une classe de 100 étudiants passe un test "vrai ou faux" comportant 20 questions et que tous les étudiants répondent au hasard à chaque question.
   (a) Au moyen d'une simulation, estimer la note moyenne et l'écart-type des notes d'une telle classe. Comparer les résultats obtenus aux valeurs théoriques.
   (b) Estimer la proportion d'étudiants qui obtiendraient un pourcentage de bonne réponses supérieur à 30%.

3.2 Exercice 2

1. Supposons que 10% des tubes métalliques produit par une machine sont défectueux et 15 tubes sont produits par heure. On suppose que chaque tube est indépendant des autres. Si plus de 4 tubes defectueux sont produits dans une seule heure, le processus de fabrication est déclaré hors de contrôle.
   (a) Simuler au moyen de la commande `rbinom` le nombre de tubes défectueux par heure sur une journée de 24 heures.
   (b) Decider, après chaque heure écoulée, si le processus de fabrication est hors de contrôle ou pas.
2. Si maintenant le taux de tubes défectueux est égal à 15%, simuler la production sur une période de 24 heures avec une production horaire de 25 unités.
   (a) Controler après chaque heure écoulée si le processus de fabrication est hors de contrôle ou pas.
   (b) Quelle est la probabilité que le nombre de tubes defectueux soit de 5 unités par heure ? supérieur ou égal à 5 unités par heure ?

3.3 Exercice 3

1. La loi de Poisson $\mathcal{P}(\lambda)$ permet de modéliser simplement le nombre d'événements survenant au cours d'un intervalle de temps donné : nombre de tembrements de terre dans une région donnée sur une période de un an, nombre d'individus se présentant au guichet d'une banque pendant une heure... Le paramètre λ est appelé le taux de la loi de Poisson.
   (a) Simuler, au moyen de la commande `rpoiss`, le nombre d'accidents annuels sur une période de 15 ans pour une loi de Poisson de taux d'accidents annuels de 2.8.
   (b) Estimer la moyenne d'une loi de Poisson de moyenne 7.2 sur la base d'un échantillon de taille 10000.
2. Une application du théorème de la limite centrale implique que si X suit une loi de Poisson $\mathcal{P}(\lambda)$, alors la variable $Z = \frac{X-\lambda}{\sqrt{\lambda}}$ suit approximativement de loi $\mathcal{N}(0,1)$ lorsque le paramètre λ est grand.
   (a) Simuler un échantillon de taille 10000 de la variable Z pour plusieurs valeurs de λ.
   (b) Pour chaque valeurs de λ, effectuer un QQ-plot afin de voir si l'approximation donnée par le TCL est acceptable.
   (c) A partir de quelle valeur de λ l'approximation donnée par le TCL est acceptable ?