Simulation Stochastique sous R

Prof. Mohamed El Merouani
https://elmerouani.jimdo.com
e-mail: m_merouani@yahoo.fr

1

Introduction

- Shareware : gratuit et installé facilement.
- Open source (on sait ce qui est réellement calculé).
- Développé par la communauté des chercheurs, contient énormément de fonctionnalités.
- Possibilité de programmer.
- Désavantage : pas très convivial.
- Manuel : http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf

2

Les graphiques

- Démarrer « R »,
- Enter les commandes suivantes:

  >demo(graphics)

  Appuyez sur ↵ plusieurs fois pour visionner les différents types de graphiques.

  >demo(persp)

  démonstration de graphiques en perspective

Les graphiques

- Après avoir donner une commande pour afficher un graphique, les graphiques, en R, sont affichés dans une fenêtre à part spécialement pour ça..
- Si aucune fenêtre graphique n'est ouverte lorsque la commande s’exécute, la commande ouvrira automatiquement une fenêtre graphique.

Les graphiques

- Si une fenêtre graphique est déjà ouverte, la commande affichera le graphique dans cette fenêtre tout en écrasant l'affichage précédent.
- Pour ouvrir une nouvelle fenêtre graphique, en plus de n'importe quelles fenêtres graphiques déjà ouverte, afin de garder le graphisme multiple exposé, on utilise la commande suivante :

  >dev.new()

Les fenêtres graphiques:

- La fenêtre graphique peut être divisée dans les panneaux qui prendront plusieurs graphiques . Par exemple:

  >par(mfrow=c(3,2))

  c-à-d « Divise la fenêtre dans 6 panneaux comme 3 lignes et 2 colonnes.
- Pour construire une fenêtre graphique de taille 2X2 en R, par exemple:

  >par(mfrow=c(2,2))

- Une fonction générique est:

```
>plot()
```

- Syntaxe générale: plot (x,y,…) les paramètres additionnels permettent de contrôler le type de graphique (diagramme de dispersion, en barres, etc.) et de gérer les annotations (axes en titre).
- Pour en savoir plus, taper ?plot ou ?par

Exemples:

```
>x<-rnorm(100)
>plot(x)
>y<-seq(-10,10,length=100)
>plot(y,y^2)
```

>plot(x,y)
Plus de contrôle peut être fait avec des arguments de plus:
>plot(x,y,'o') # relier les symboles point par des lignes
>plot(x,y,type='l') # relier le point coordonnées par des lignes, mais ne montre aucun symbole pour le point
>plot(x,y,col='blue') # spécifier la couleur pour le point
>plot(x,y,ylim=c(-1,2)) # spécifier la fenêtre de l'axe y
>plot(x,y,xlab="lbl") # spécifier le label de l'axe x
>plot(x,y,main="title") # spécifier le titre (d'entête)
Ces arguments peuvent être incorporé en plusieurs combinaisons.

- Autres fonctions graphiques: hist(), pie(), barplot(),dotchart()
- Certaines fonctions sont plus adaptées
  - à un certain type de variable (numérique vs. Catégorielle)
  - à l'information que l'on veut transmettre (distribution uni- ou bivariée)

Exemple d'application:

- Tracé de densités et de fonctions de répartitions
- Des tracés de densités de probabilité ou de fonctions de répartition des lois divers peuvent s'obtenir à l'aide de la fonction plot().
- Ainsi par exemple pour tracer la densité (répartition de masse) d'une loi binomiale avec n=10 et p=.25, on exécute en R:


```
x<-0:10
y<-dbinom(x,size=10,prob=.25)#évalue les
   probabilités
plot(x,y,type="h",lwd=30,main="Densité
   binomiale avec \nn=10,
   p=.25",ylab="p(x)",lend="square")
```

- Pour cet exemple, nous avons d'abord crée le vecteur x contenant les entiers allant de 0 à 10.
- Nous avons ensuite calculé les probabilités qu'une variable de loi binomiale B(10, 0.25) prenne pour chacune de ces valeurs, par dbinom.
- Le type de tracé est spécifié avec l'option type=h (lignes verticales d'un diagramme en bâtons), épaissies grâce à l'option lwd=30.
- L'option lend=«square» permet de tracer des barres rectangulaires.
- Nous avons ensuite ajouté des légendes.

Nombres aléatoires

- Pour générer « n » nombres aléatoires qui suivent une loi uniforme (0,1) la commande en R est:

  > runif(n, min=0, max=1)
- Si, nous voulons les représenter sur un graphique, on écrit:

  >x=runif(n, min=0, max=1)
- Alors R les garde le résultat dans x sans l'afficher. Puis on trace le nuage de points:

  >plot(x)
- Si on tape x tout simplement, R les affiche.

Nombres aléatoires

- Pour la reproductibilité et le contrôle, la graine pour les générateurs de nombre aléatoire devrait être mise explicitement, avant la première utilisation d'un générateur de nombre aléatoire dans le Script.
- La commande est :

  >set.seed(m)
- En changeant la graine et en lançant de nouveau le Script, on obtiendra de différents (indépendants) échantillons à partir des générateurs de nombres aléatoires.

Exemple: nombres aléatoires sous R

```
>set.seed(25)
>x=runif(1000,min=0,max=1)
>plot(x)
```

Simulation des lois de probabilités

Simulation des lois de probabilités

- Il est utile en statistique de pouvoir générer des données aléatoires, et R peut le faire pour un grand nombre de fonctions de densité de probabilité.
- Ces fonctions sont de la forme: rfunc(n,p1,p2,…) #où func indique la loi de probabilité, n le nombre de données générées et p1, p2,… sont les valeurs des paramètres de la Loi de probabilités.
- Pour certaines lois, les paramètres ont des valeurs par défaut. Parmi les plus utilisées, la loi uniforme porte par défaut sur l'intervalle (0,1), et la loi normale est centrée réduite par défaut.

loi	fonction
Gauss (normale)	rnorm(n, mean=0, sd=1)
exponentielle	rexp(n, rate=1)
gamma	rgamma(n, shape, scale=1)
Poisson	rpois(n, lambda)
Weibull	rweibull(n, shape, scale=1)
Cauchy	rcauchy(n, location=0, scale=1)
beta	rbeta(n, shape1, shape2)
'Student' (t)	rt(n, df)
Fisher–Snedecor (F)	rf(n, df1, df2)
Pearson (χ^2)	rchisq(n, df)
binomiale	rbinom(n, size, prob)
multinomiale	rmultinom(n, size, prob)
géométrique	rgeom(n, prob)
hypergéométrique	rhyper(nn, m, n, k)
logistique	rlogis(n, location=0, scale=1)
lognormale	rlnorm(n, meanlog=0, sdlog=1)
binomiale négative	rnbinom(n, size, prob)
uniforme	runif(n, min=0, max=1)
statistiques de Wilcoxon	rwilcox(nn, m, n), rsignrank(nn, n)

19

Exemples de lois continues

20

Simulation des lois continues

- Pour effectuer un calcul avec une loi de proba. sous R, il suffit d'utiliser comme fonction l'une des appellations R avec les préfixes:
- r pour un tirage aléatoire
- d pour une densité
- p pour une fonction de répartition
- q pour une fonction quantile

Exemples:

```
> x<-rnorm(100) #100tirages, loi N(0,1)
> w<-rexp(1000, rate=1) #1000 tirages loi
   exponentielle
> z<-dpois(0:2,lambda=4)#probabilités de 0,1,2
   pour la loi de Poisson(4)
> y<-pnorm(12,mean=10,sd=2)#P(X<12) pour la
   loi N(10,2)
>v<- qnorm(.75,mean=10,sd=2)#3ème quartile de
   la loi N(10,2)
```

Loi bêta

- Une v.a. continue X suit une loi bêta de paramètres α, β>0, si sa fonction de densité est:

$$f(x)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1};\quad 0<x<1$$

- Sur R on génère une loi bêta par:

>x=rbeta(n, a_1,a_2) #échantillon de n valeurs à partir d'une loi bêta de paramètres a_1 et a_2

>hist(x) #donne l'histogramme

Fonction Gamma Γ d'Euler:

La fonction Γ est définie sur]0,+∞[par:

$$\Gamma(\alpha)=\int_0^{\infty} e^{-t}t^{\alpha-1}dt$$

Propriétés:

1) $\Gamma(\alpha)=(\alpha-1)\Gamma(\alpha-1)$

2) $\Gamma(1)=1$

3) $\Gamma(n)=(n-1)!$

Simulation d'une v.a. beta B(3,4):

- Sa fonction de densité

$$f(x) = 60x^2 \left(1-x\right)^3 ; \quad 0 < x < 1$$

- Sur R on génère cette loi bêta par:

```
> x=rbeta(100, 3, 4)
> hist(x,prob=TRUE)
> curve(dbeta(x,3,4),add=TRUE,col="blue")
```


Simulation d'une loi normale:

>rnorm() #générer un échantillon aléatoire de loi normale.

Son utilisation en R sera:

>rnorm(n, mu, sigma) #générer un échantillon de taille n de la loi normale de moyenne « mu » et d'écart-type « sigma »

>dnorm(x,mean=mu,sd=sigma) #donne évaluation de la densité de prob. à x

On peut utiliser curve() de la façon suivante pour superimposer la densité de proba. sur l'histogramme

Simulation d’une loi normale:

```
>curve(dnorm(x,mean=mu,sd=sigma),add=TRUE,col="blue”)
```

- add=TRUE dans la fonction curve() superimpose la densité sur l’histogramme.
- Voici un exemple pour produire un histogramme avec une densité de probabilité superimposé

```
>x=rnorm(5000,35,5)
>hist(x,prob=TRUE)
>curve(dnorm(x,mean=35,sd=5),add=TRUE,col="blue”)
```



```
>n=1
> k=5000
> mu=35
> sigma=5
> SampleMeans<-c()
> for(i in( 1:k))
 {
>SampleMeans[i]<-mean(rnorm(n,mu,sigma))
}
>hist(SampleMeans,prob=TRUE,xlab="Moyenne(n=1)")
>curve(dnorm(x,mean=35,sd=5/sqrt(n)),add=TRUE,col="blue")
```

Méthode d'inversion-Exemples:

2. Simulation d'une v.a. exponentielle $\mathcal{E}xp(\lambda)$:

Sa fonction de répartition est:

$$F(x) = \begin{cases} 0 & si\, x < 0 \\ 1 - \exp(-\lambda x) & si\, x \geq 0 \end{cases}$$

Nous faisons $X = F^{-1}(U) = \left[-\frac{Ln(1-U)}{\lambda} \right]$

Ainsi pour générer un n-echantillon selon une $\mathcal{E}xp(\lambda)$ on pose puisque $(1-U) \to \mathcal{U}(0,1)$

$$(X_1, \cdots, X_n) = \left(-\frac{Ln\,U_1}{\lambda}, \cdots, -\frac{Ln\,U_n}{\lambda} \right)$$

Simulation d'une v.a. exponentielle Exp(λ) par inversion:

- Le code sur R sera:

```
> myrexp<-function(n,lambda) + return(-log(runif(n))/lambda)
> lambda<- 5 #Simulation d'une Exp(5)
> x<- myrexp(500, lambda)
> hist(x, freq=FALSE) #Comparaison histogramme / densité
> curve(dexp(x, lambda), xlim=c(0, max(x)), col="red", add=TRUE)
#freq=FALSE pour aire=1
```

Exercice:

- Écire un programme similaire pour les lois:
  - Weibull
  - Uniforme

Rappel: Loi de Cauchy

- La fonction de densité d'une v.a. de paramètres x_0 et α>0 de Cauchy est:

$$f(x) = \frac{\alpha}{\pi[\alpha^2 + (x - x_0)^2]};\ x \in IR$$

- La loi de Cauchy n'a ni moyenne ni variance.
- La moyenne de n v.a. indépendantes de Cauchy est une loi de Cauchy.

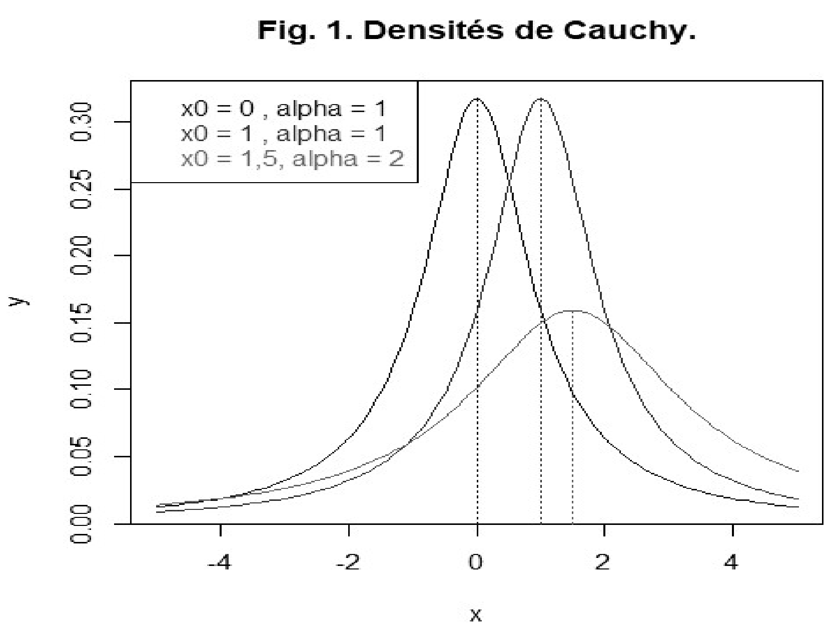
- Les commandes suivantes de R, séparées par des ; permettent de créer le graphique Fig. 1 ci-dessous.
- Nous y avons représenté plusieurs densités de lois de Cauchy pour illustrer l'allure de la densité en fonction des paramètres.


```
>plot(function(x) dcauchy(x,location=0,scale=1), -5, 5,
xlab="x",ylab="y", main="Fig. 1. Densités de Cauchy.",col="blue");
segments(0,0,0,dcauchy(0,location=0,scale=1),lty="dotted",col="blue");

>curve(dcauchy(x,location=1,scale=1),add=TRUE,col="red");
segments(1,0,1,dcauchy(1,location=1,scale=1),lty="dotted",col="red");

>curve(dcauchy(x,location=1.5,scale=2),add=TRUE,col="green4");
segments(1.5,0,1.5,dcauchy(1.5,location=1.5,scale=2),lty="dotted",col=
"green4");

>legend(x="topleft",y=NULL,text.col=c("blue","red","green4"),legend=
c("x0 = 0 , alpha = 1","x0 = 1 , alpha = 1","x0 = 1,5, alpha = 2"));
```



35

Exemples de lois discrètes

36

Vecteurs:

- Pour créer un vecteur on peut utiliser la commande c() de concaténation. Par exemple pour créer un vecteur appelé monvecteur de composantes 8, 6, 9, 10, et 5, on tape :

>monvecteur <- c(8,6,9,10,5)

- Pour examiner le contenu de la variable monvecteur il suffit de taper son nom monvecteur

>[1] 8 6 9 10 5

- Il existe plusieurs autres manières de définir un vecteur dans R. Une façon fréquente est d'utiliser des commandes d'itération : elle permettent de construire des vecteurs de nombres séparés par des pas positifs ou négatifs.
- La syntaxe de base, seq(deb,fin, by=pas), retourne un vecteur de valeurs allant de deb à fin par valeurs séparées par des multiples de pas. Les crochets sont facultatifs.
- Par défaut, pas vaut 1. Selon que fin-deb est ou non un multiple entier de pas, le vecteur se terminera ou non par fin.

- Si l’on souhaite un vecteur commençant par deb et se terminant par fin avec n coordonnées régulièrement espacées, il est préférable d’utiliser seq(deb,fin,length=n). La commande rep permet de dupliquer les éléments d’un vecteur. Par exemple, rep(5,6) duplique le nombre 5 six fois.


```
> v <- 0:10
> v <- seq(0:10)
> v <- seq(0,1,by=0.1)
> v <- seq(0,.99,by=0.1)
> v <- seq(0,1,by=0.15)
> v <- seq(1,0,by=-0.15)
> v <- rep(5,6)
```

- x:y nombres de x à y par pas de 1
- seq(x,y,by=p) nombres de x à y par pas de p
- seq(x,y,length=n) n nombres entre x et y
- v[i] i-ième coordonnée de v
- v[i1:i2] coordonnées i1 à i2 de v
- v[-(i1:i2)] supprimer les coordonnées i1 à i2 de v
- À

Exemple d'une loi discrète

- On peut utiliser sample() pour générer n valeurs aléatoires de la loi de probabilité suivante:

x_i	1	2	3	4	5	6	7
$P_i=P(X=x_i)$	0,1	0,4	0,3	0,1	0,05	0,03	0,02

- Son utilisation est:

>sample(x,n,replace=TRUE, prob=p)

où X contient l'image de la v.a. et p contient les probabilités.

Exemple d'une loi discrète

- Voici un exemple où nous générons 10 valeurs de la loi discrète précèdente:

```
>x=c(1,2,3,4,5,6,7)
>p=c(0.1,0.4,0.3,0.1,0.05,0.03,0.02)
>sample(x,10,replace=TRUE,p)
```


Exemple d'une loi discrète

- Nous allons, maintenant examiner par un exemple que $\overline{X} \sim N(\mu, \frac{\sigma^2}{n})$ approximativement, quand n est grand.
- Nous allons utiliser n=1, n=2, n=15 et n=30.
- Pour notre exemple (précédent),

$$\mu = E(X) = \sum xP(X = x) = 2.77$$

et $\sigma^2 = Var(X) = E\left((x-\mu)^2\right) = 1.6371$

Exemple d'une loi discrète

- Nous allons générer k=5000 échantillons de taille n. Pour chaque échantillon, on calcul la moyenne.

- On produit un histogramme des k moyennes et on superimpose une densité de probabilité pour une normale de moyenne μ=2.77 et d'écart-type $\sigma/\sqrt{n} = \sqrt{1.6371}/\sqrt{n}$

Prof. Mohamed El Merouani 45

Voici le programme pour n=1. Le resoumettre aussi pour n=2, n=15 et n=30

```
> n=1
> k=5000
> x=c(1,2,3,4,5,6,7)
> p=c(0.1,0.4,0.3,0.1,0.05,0.03,0.02)
> sampleMeans<-c()
> for(i in (1:k))
+ {
+ sampleMeans[i]<-mean(sample(x,n,replace=
+ TRUE,p))
+ }
>hist(sampleMeans,prob=TRUE,xlab="Moyenne(n=1)")
>curve(dnorm(x,mean=2.77,sd=sqrt(1.6371)/sqrt(n)),add=TRUE,col="blue")
```

46