Universidad Santo Tomás

División de Ciencias Económicas y Administrativas

Departamento de Estadística

Simulación y cálculo numérico

By: **Hanwen Zhang**

hanwenzhang@usantotomas.edu.co

Índice

1. Clase 25. Algoritmo de Metropolis Hastings

1.1. Cadena de Markov

Una cadena de Markov es una secuencia de variables dependientes: X_0, X_1, $\cdots$, tales que la distribución de X_t dadas las variables pasadas depende sólo en X_{t-1}. Esto es

$$p(X_t|X_{t-1}, \cdots, X_1, X_0) = p(X_t|X_{t-1})$$

y $p(X_t|X_{t-1})$ se conoce como el kernel de transición.

Hay muchas propiedades importantes para una cadena de Markov como: distribución límite, distribución estacionaria, ergodicidad, cadena recurrente, cadena irreducible, entre otras.

En particular, decimos que f es la distribución estacionaria de la cadena de Markov al cumplir: si $X_{t-1} \sim f$, entonces $X_t \sim f$. En caso de una cadena recurrente, la distribución estacionaria también es la distribución límite de la cadena, lo anterior es importante desde el punto de vista de simulaciones puesto que al simular un número suficiente grande de valores, se puede garantizar que los valores provienen de la distribución estacionaria.

1.2. Método de Markov chain Monte Carlo (MCMC)

Dada una distribución objetiva f, se construye un kernel de transición con distribución estacionaria f, se obtiene una cadena de Markov $\{X_t\}$ usando el kernel tal que la distribución límite sea f.

Existen métodos que provee kernel asociado a una densidad objetiva f, tales como MH y el muestreador de Gibbs.

1.3. Algoritmo MH

Igual que el método de aceptación y rechazo, el algoritmo MH también es un método de muestreo indirecto, en vez de muestrear valores de la distribución objetiva f, muestrea valores de una distribución candidata (*proposal*, instrumental), y se acepta.

El algoritmo consiste en los siguientes pasos:

1. Para un valor dado x_t, generar un valor y_t de la distribución candidata $q(y|x_t)$

2. Calcular
$$\rho(x_t, y_t) = \text{mín}\left\{\frac{f(y_t)q(x_t|y_t)}{f(x_t)q(y_t|x_t)}, 1\right\}$$

3. Tomar
$$x_{t+1} = \begin{cases} y_t & \text{con probabilidad } \rho(x_t, y_t) \\ x_t & \text{con probabilida } 1 - \rho(x_t, y_t) \end{cases}$$

Observación:

1. El valor de x_{t+1} solo depende de x_t y no de valores pasados (Markov chain).

2. Como en el cálculo de ρ aparece $f(y_t)/f(x_t)$, el algoritmo MH puede ser aplicado a f sólo conociendo su kernel.

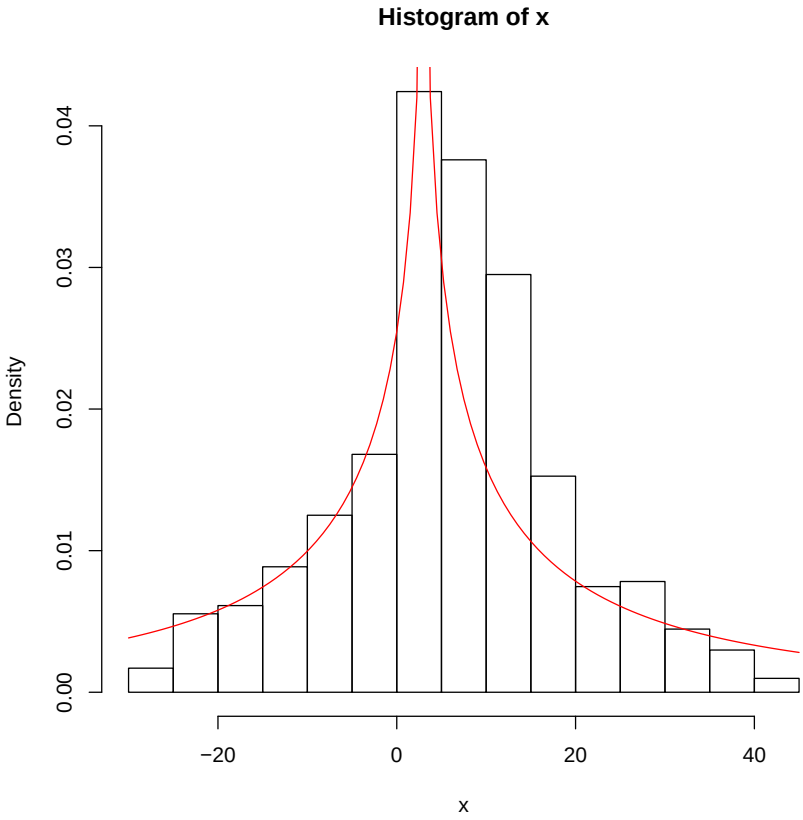
Example 1 Suponga que queremos simular valores de

$$f(x) \propto e^{-(|x-3|/2)^{0.4}}, \quad \text{para } -\infty < x < \infty$$

En primer lugar, debemos pensar en la distribución candidata q, como los valores de x están en todos los números reales, la opción natural para q es de una distribución normal; pero también hay que tener en cuenta que esta distribución q depende del valor anterior x_t, entonces podemos pensar en $q(\cdot|x_t) \sim N(x_t, 1)$. Y tenemos los siguientes códigos:

```
set.seed(1234)
N.sim<-10000
f<-function(x){exp(-(abs(x-3)/2)^0.4)}
x<-NULL
x[1]<-0
for(i in 2:N.sim){
  y<-rnorm(1,x[i-1],1)
  prob<-min(1,f(y)*dnorm(x[i-1],y,1)/(f(x[i-1])*dnorm(y,x[i-1],1)))
```

```
  u<-runif(1)
  if(u<prob){x[i]<-y}
  if(u>=prob){x[i]<-x[i-1]}
}
K<-1/integrate(f,-100,100)$val
K

## [1] 0.08254714

f<-function(x){exp(-(abs(x-3)/2)^0.4)*K}
hist(x,freq=FALSE)
curve(f,add=T,col=2)
```



1.4. Clase 26. MH independiente

El algoritmo se llama MH independiente cuando el kernel de transición $q(\cdot|\cdot) = q(\cdot)$. Es decir, en cada iteración, la generación del valor candidato y no depende del valor x de la iteración anterior. Sin embargo, eso no implica que los valores muestreados sean independientes, puesto que cuando un valor candidato no es aceptado, se deja el valor x de la iteración anterior.

El algoritmo de MH independiente entonces es:

1. Para un valor dado x_t, generar un valor y_t de la distribución candidata $q(y)$

2. Calcular
$$\rho(x_t, y_t) = \text{mín}\left\{\frac{f(y_t)q(x_t)}{f(x_t)q(y_t)}, 1\right\}$$

3. Tomar
$$x_{t+1} = \begin{cases} y_t & \text{con probabilidad } \rho(x_t, y_t) \\ x_t & \text{con probabilida } 1 - \rho(x_t, y_t) \end{cases}$$

Example 2 Para la distribución $f(x)$ con

$$f(x) \propto 1 + \cos\left(\frac{x-2}{5}\pi\right), \quad \text{para } -3 < x < 7$$

La distribución candidata natural sería $U(-3, 7)$

```
set.seed(123)
N.sim<-1000
f<-function(x){1+cos(pi*(x-2)/5)}
x_MH<-x_AR<-NULL
x_MH[1]<-0
j_MH<-0
# Metropolis Hastings
for(i in 2:N.sim){
  y<-runif(1,-3,7)
```

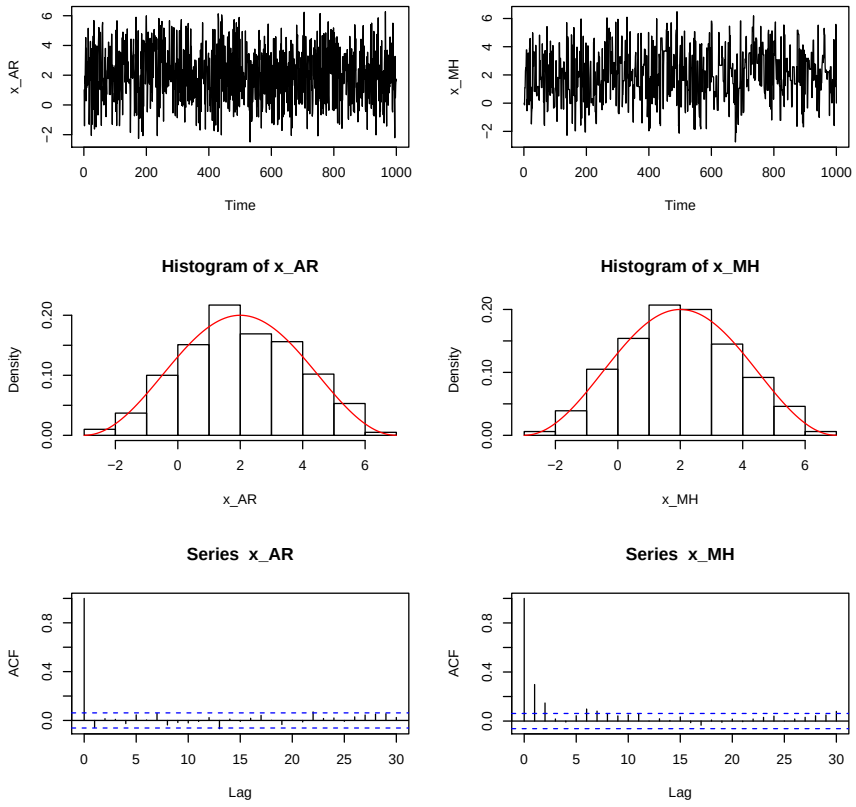
```
  prob<-min(1,f(y)/f(x_MH[i-1]))
  u<-runif(1)
  if(u<prob){
    x_MH[i]<-y
    j_MH<-j_MH+1
  }
  if(u>=prob){x_MH[i]<-x_MH[i-1]}
}
# Aceptación y rechazo
g<-function(x){0.1}
f.g<-function(x){f(x)/g(x)}
M<-optimize(f.g,interval=c(-3,7),maximum=T)$objective
j_AR<-0
while(length(x_AR)<N.sim){
  cand<-runif(1,-3,7)
  u<-runif(1)
  if(u<=(f.g(cand)/M)){x_AR<-c(x_AR,cand)}
  j_AR<-j_AR+1
}
j_MH/N.sim

## [1] 0.61

N.sim/j_AR
```

```
## [1] 0.5030181
```

```
K<-1/integrate(f,-3,7)$val
K
```

```
## [1] 0.1
```

```
f1<-function(x){f(x)*K}
# Gráficas de comparación
par(mfrow=c(3,2))
ts.plot(x_AR)
ts.plot(x_MH)
hist(x_AR,freq=FALSE)
curve(f1,add=T,col=2)
hist(x_MH,freq=FALSE)
curve(f1,add=T,col=2)
acf(x_AR)
acf(x_MH)
```



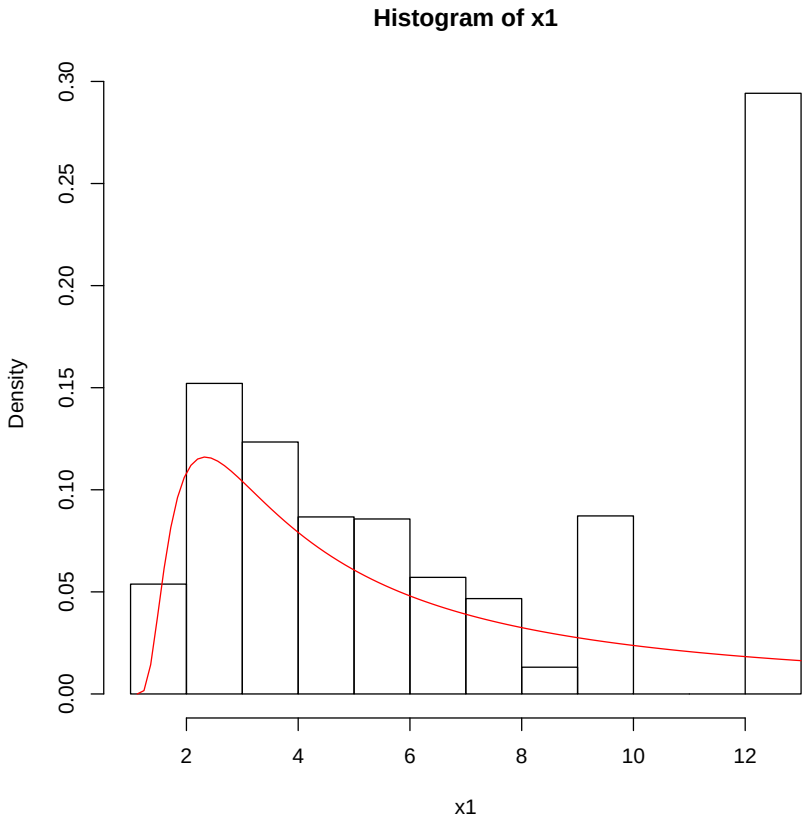
Cuando podemos usar el método de aceptación y rechazo, generalmente no usamos el MH.

Example 3 Retomemos un ejercicio anterior con

$$f(x) \propto \exp\{-2/(x-1)\}/((x-1)^{1.5})$$

```r
set.seed(123456)
x1<-NULL
x1[1]<-2
N.sim<-10000
f<-function(x){exp(-2/(x-1))/((x-1)^1.5)}
q<-function(x){exp(-(x-1))}
#f<-function(x){exp(0.5*x)/(1+exp(0.5*x))}
j<-0
for(i in 2:N.sim){
  y<-rexp(1,1)+1
  u<-runif(1)
  prob<-min(1,f(y)*q(x1[i-1])/(f(x1[i-1])*q(y)))
  if(u<prob){
    x1[i]<-y
    j<-j+1
  }
  if(u>=prob){x1[i]<-x1[i-1]}
}
j/N.sim

## [1] 0.1537

K<-1/integrate(f,1,200000)$val
f1<-function(x){f(x)*K}
```

```r
hist(x1,freq=FALSE)

curve(f1,add=T,col=2)
```



¿Qué pasó con esa barra tan alta?

1.5. Verificando la calidad de MH

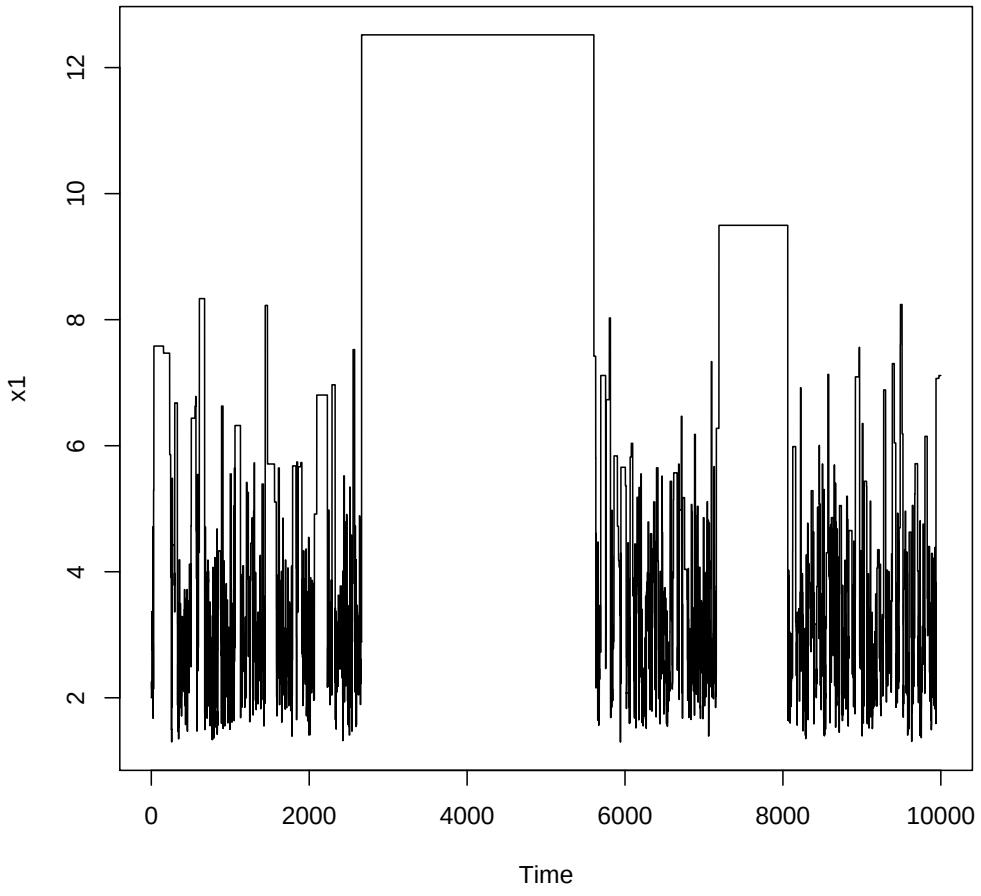
1.5.1. Ajuste de la distribución objetiva a los datos simulados

Histograma vs. función de densidad objetiva

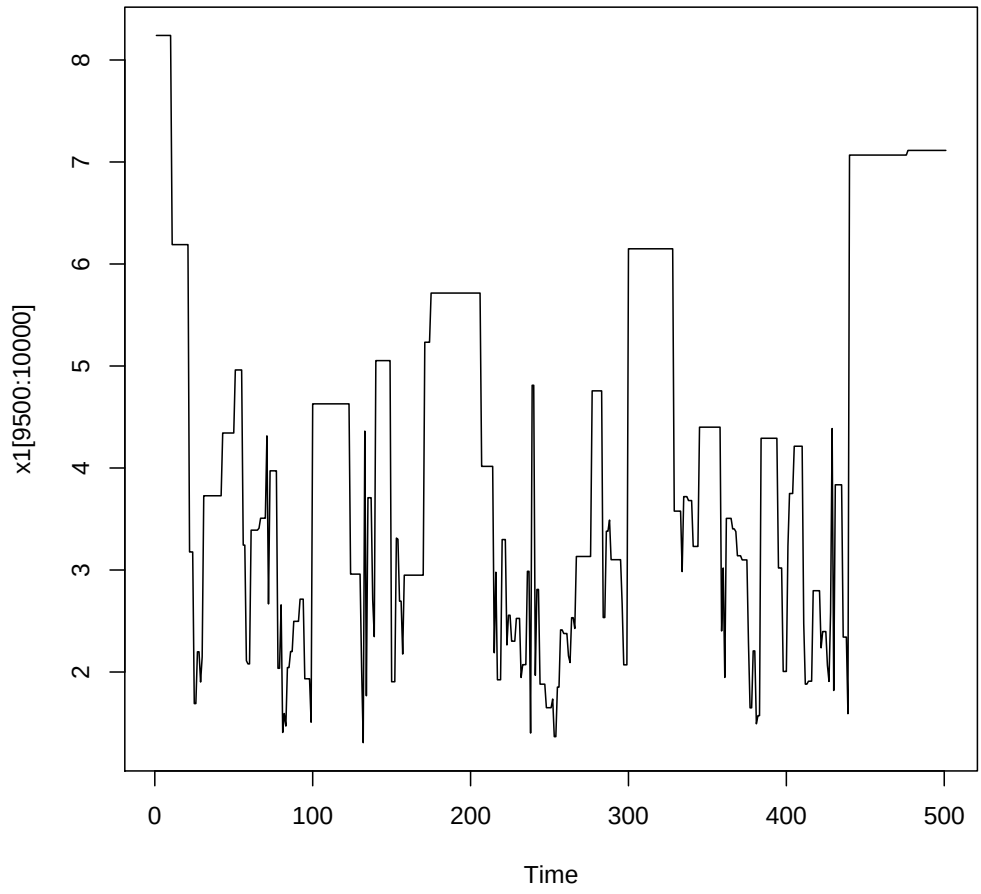
1.5.2. Eficiencia

La forma más simple de verificar la convergencia de la cadena de Markov es observando la tendencia de los datos generados

```
ts.plot(x1)
```



```r
ts.plot(x1[9500:10000])
```



Se debe enfocar que no haya un periodo largo donde la cadena se quede en un sólo valor.

1.5.3. Tasa de aceptación

Queremos que el método sea eficiente, en el sentido de que una porción grande de los valores candidatos sean aceptados.

1.5.4. Baja correlación

Los resultados de MH produce valores correlacionados pues forman una cadena de Markov. Sin embargo, entre más fuerte sea la correlación entre los valores, más lenta es la convergencia hacia la distribución estacionaria. Por lo tanto, siempre queremos escoger una distribución candidata que no produzca fuertes correlaciones entre los valores obtenidos.

La herramienta útil para cuantificar la correlación en datos ordenados es la autocorrelación.

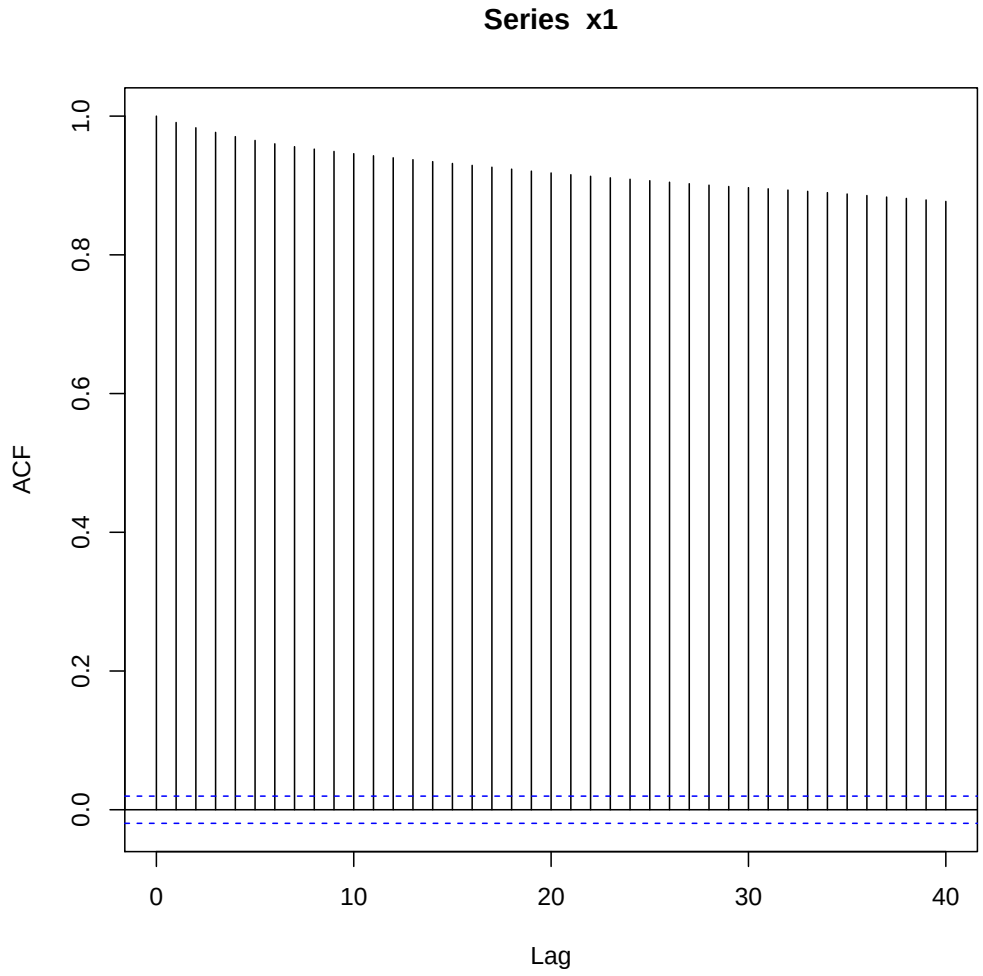
Autocorrelación de rezago k se define como

$$\rho_k = \frac{\sum_{t=1}^{n-k} x_t x_{t+k}}{\sum_{t=1}^{n} x_t^2}$$

ρ_k mide la corelación entre x_t y x_{t+k}. Valores pequeños de ρ_k indica la débil correlación de rezago k. En `R`, el comando `acf` calcula y grafica los valores de ρ_k para $k = 0, 1, 2, \cdots$ (llamado el correlograma).

Chequeemos qué tan fuertes son las correlaciones del ejemplo 1

```
acf(x1)

acf(x1)$acf
```




```
## , , 1
##
##             [,1]
##  [1,] 1.0000000
##  [2,] 0.9907930
##  [3,] 0.9831737
```

```
##  [4,] 0.9765292
##  [5,] 0.9704273
##  [6,] 0.9648937
##  [7,] 0.9600597
##  [8,] 0.9558877
##  [9,] 0.9523927
## [10,] 0.9490563
## [11,] 0.9458206
## [12,] 0.9427817
## [13,] 0.9399029
## [14,] 0.9371749
## [15,] 0.9343642
## [16,] 0.9317085
## [17,] 0.9289677
## [18,] 0.9263173
## [19,] 0.9235272
## [20,] 0.9206916
## [21,] 0.9180285
## [22,] 0.9154925
## [23,] 0.9132311
## [24,] 0.9110091
## [25,] 0.9089586
## [26,] 0.9068643
```

```
## [27,] 0.9046891
## [28,] 0.9025342
## [29,] 0.9004445
## [30,] 0.8985057
## [31,] 0.8968688
## [32,] 0.8951960
## [33,] 0.8933629
## [34,] 0.8915507
## [35,] 0.8896851
## [36,] 0.8876947
## [37,] 0.8854834
## [38,] 0.8832964
## [39,] 0.8813340
## [40,] 0.8791140
## [41,] 0.8769877
```

Vemos que las correlaciones de todos los rezagos son importantes, lo cual indica que la calidad de la muestra obtenida no es buena.

1.6. Ejercicios

1. Grafique el correlograma de la muestra obtenida en el ejemplo 1. Cambie el valor de la desviacón estándar del kernel de transición y mire cómo cambia el correlograma.

2. Implemente el algoritmo de MH para obtener una muestra de tamaño 1000 de una distribución t-student con 3 grados de libertad. Utiliza como el kernel de transición una distribución normal (que dependa del valor x_t) con desviación estándar 10.

3. Repite el punto anterior utilizando el algoritmo de MH independiente, usando como el kernel de transición una distribución normal con media 0 y desviación estándar 10. Obtenga gráficas que comparen los resultados obtenidos con MH y MH independiente.

4. Implemente el algoritmo de MH para simular 1000 valores de la distribución $f(x)$ con

$$f(x) = \frac{1}{\pi(1 + (x-4)^2)}, \quad \text{con } -\infty < x < \infty$$

   Compare los resultados obtenidos con lo obtenido usando el método de la grilla.

5. Suponga que la función de densidad de una variable X

2. Clase 28. Muestreador de Gibbs

El muestreador de Gibbs es una herramienta supremamente poderosa para obtener muestras desde una distribución multivariada.

2.1. Muestreador de Gibbs para el caso bivariado

Suponga que las variables X y Y tienen distribución conjunta $f(x, y)$, y suponga que la distribución condicional de X dada la variable Y está dada por $f(x|y)$, y la de Y dada X está dada por $f(y|x)$, el muestreador de Gibbs consiste en:

1. Fijar el valor inicial de X, denotado por $x^{(0)}$
2. Obtener un valor para la variable Y desde $f(y|x^{(0)})$, denotado por $y^{(0)}$.
3. Obtener un valor para X desde $f(x|y^{(0)})$, denotado por $x^{(1)}$
4. Obtener un valor para Y desde $f(y|x^{(1)})$, denotado por $y^{(1)}$
5. Así sucesivamente.

Example 4 Suponga que se quiere simular valores de un vector $(X, Y)'$ con distribución normal bivariada con el vector de medias $(\mu_X, \mu_Y)'$ y matriz de varianzas y covarianzas $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$. Esto es: $Var(X) = \sigma_1^2$, $Var(Y) = \sigma_2^2$, $Cov(X, Y) = \sigma_{12}$, de esta forma $Cor(X, Y) = \frac{\sigma_{12}}{\sigma_1 \sigma_2}$.

La función de densidad conjunta del vector (X, Y) está dada por

$$
\begin{aligned}
f(x,y) &= \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_X, y-\mu_Y) \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}^{-1} \begin{pmatrix} x-\mu_X \\ y-\mu_Y \end{pmatrix} \right\} \\
&= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\
&\qquad \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_1} \right)^2 - 2\rho \frac{(x-\mu_X)(y-\mu_Y)}{\sigma_1\sigma_2} + \left(\frac{y-\mu_Y}{\sigma_2} \right)^2 \right] \right\}
\end{aligned}
$$

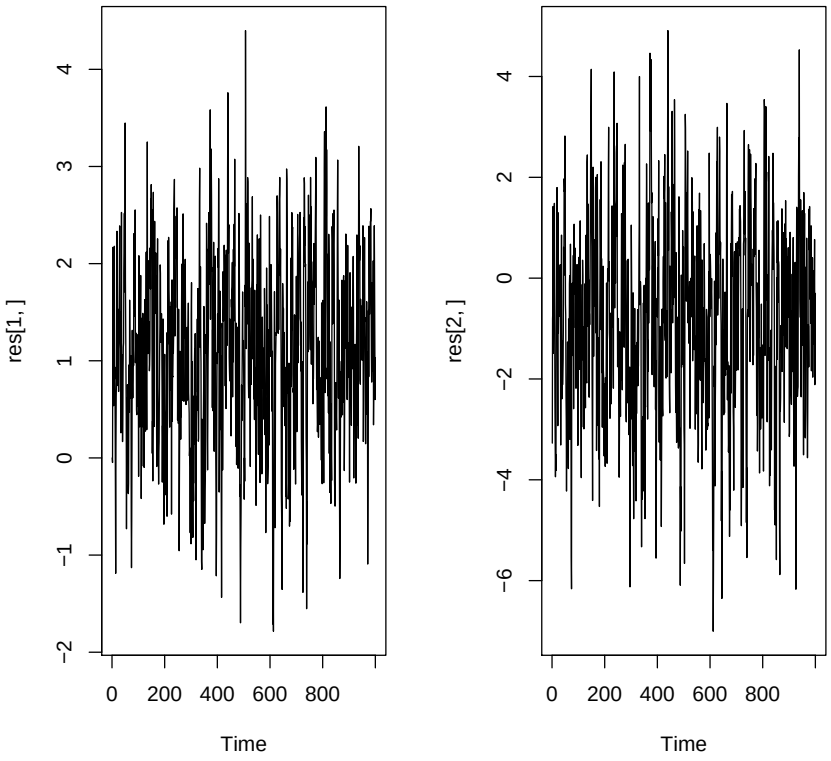
Las distribuciones condicionales están dadas por

$$
\begin{aligned}
X|Y &\sim N\left(\mu_X + \frac{(y-\mu_Y)\rho\sigma_1}{\sigma_2}, \sigma_1^2(1-\rho^2) \right) \\
Y|X &\sim N\left(\mu_Y + \frac{(x-\mu_X)\rho\sigma_2}{\sigma_1}, \sigma_2^2(1-\rho^2) \right)
\end{aligned}
$$

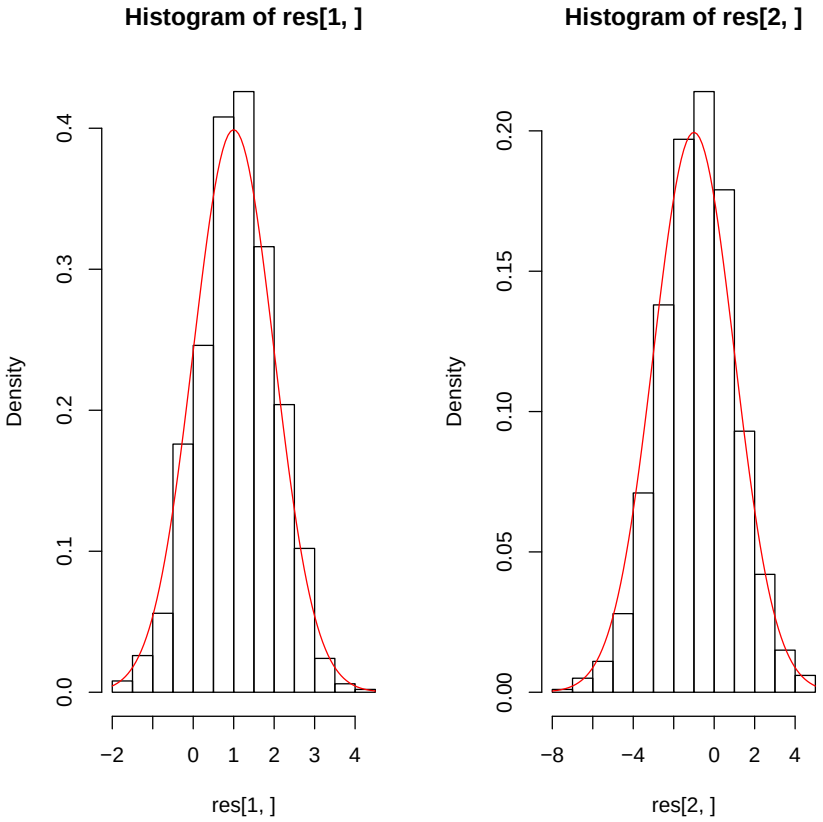
Dado que las distribuciones condicionales corresponden a la distribución normal que son fáciles de simular, un muestreador de Gibbs sería muy conveniente. Implementemos el método con $\mu_X = 1$, $\mu_Y = -1$, $\sigma_1 = 1$, $\sigma_2 = 2$, $\rho = 0.8$

```
set.seed(123)
n<-1000
res<-matrix(NA,2,n)
mu<-c(1,-1)
sigma<-c(1,2)
```

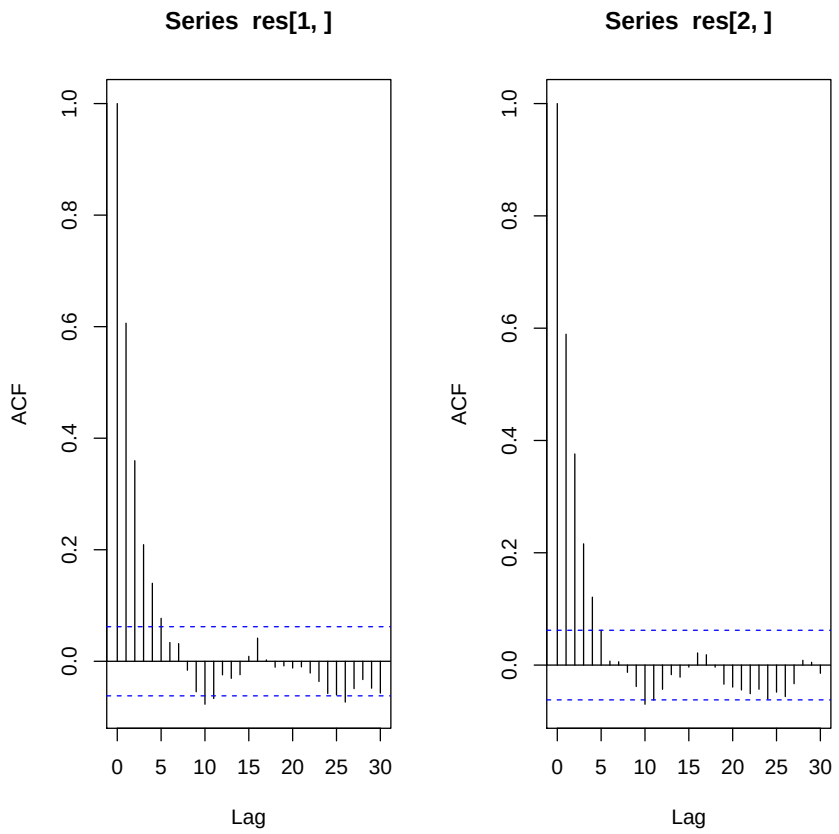
```r
rho<-0.8
# en la fila 1 guarda los x, y en la fila 2 guarda los y
res[1,1]<-0 # Valor inicial para x
res[2,1]<-rnorm(1,mu[2]+(res[1,1]-mu[1])*rho*sigma[2]/sigma[1],
                sigma[2]*sqrt(1-rho^2))
res[,1]

## [1]  0.000000 -3.272571

for(i in 2:n){
  res[1,i]<-rnorm(1,mu[1]+(res[2,i-1]-mu[2])*rho*sigma[1]/sigma[2],sigma[1]*s
  res[2,i]<-rnorm(1,mu[2]+(res[1,i]-mu[1])*rho*sigma[2]/sigma[1],
                  sigma[2]*sqrt(1-rho^2))
}
par(mfrow=c(1,2))
ts.plot(res[1,])
ts.plot(res[2,])
```



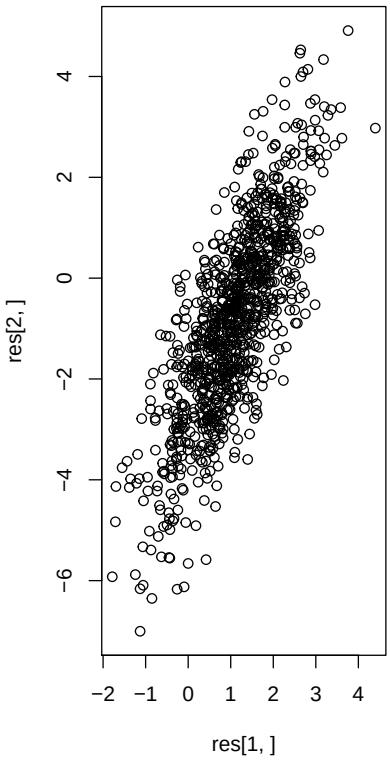

```
par(mfrow=c(1,2))
hist(res[1,],freq=F)
curve(dnorm(x,mu[1],sigma[1]),add=T,col=2)
hist(res[2,],freq=F)
curve(dnorm(x,mu[2],sigma[2]),add=T,col=2)
```




```
par(mfrow=c(1,2))

acf(res[1,])

acf(res[2,])
```




```
plot(res[1,],res[2,])

cor(res[1,],res[2,])


## [1] 0.7855984
```



En el ejemplo anterior, como se trata de la distribución normal multivariada, las distribuciones condicionales ya han sido desarrolladas. En general, cuando se tiene $f(x, y)$ (puede ser en su forma completa o sólo el kernel), las densidades condicionales pueden ser encontradas como

$$f(x|y) \propto f(x, y) \quad \text{(visto a } y \text{ como fijo)}$$
$$f(y|x) \propto f(x, y) \quad \text{(visto a } x \text{ como fijo)}$$

Example 5 Suponga que $(\theta, \tau)'$ es un vector aleatorio con fun-

ción de densidad dada por

$$f(\theta, \tau) \propto \tau^3 \exp\{-\tau(\theta - 1)^2/2\} \exp\{-\tau/5\}$$

donde $-\infty < \theta < \infty$ y $\tau > 0$.

Para encontrar la distribución condicional de $\theta|\tau$, tenemos que

$$\begin{aligned} f(\theta|\tau) &\propto f(\theta, \tau) \quad \text{con } \tau \text{ fijo} \\ &\propto \tau^3 \exp\{-\tau(\theta - 1)^2/2\} \exp\{-\tau/5\} \\ &\propto \exp\{-\tau(\theta - 1)^2/2\} \end{aligned}$$

de donde podemos concluir que $\theta|\tau \sim N(1, 1/\tau)$.

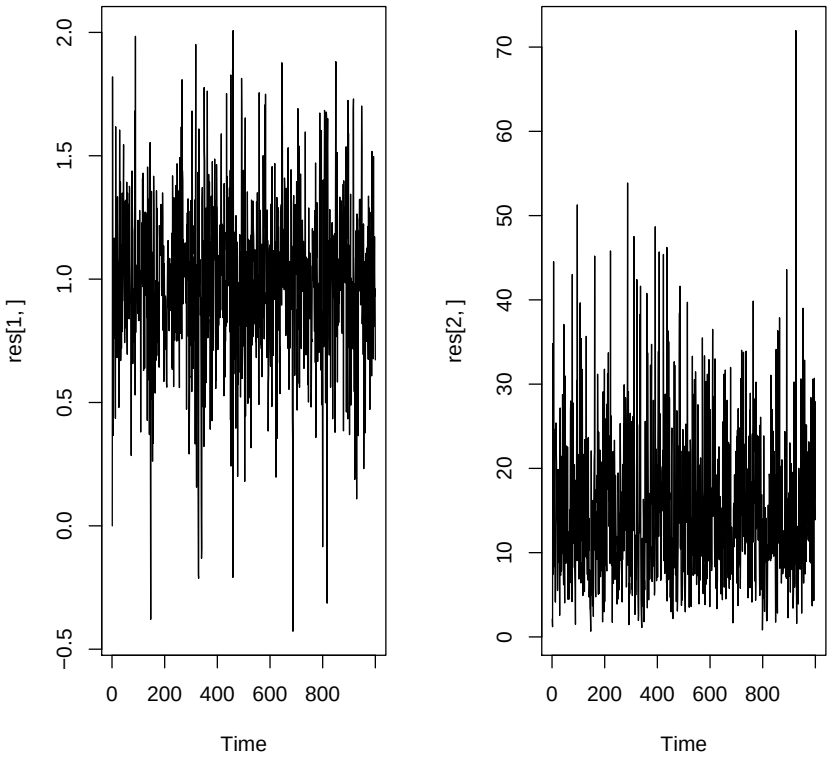
Ahora, encontramos la distribución de $\tau|\theta$, tenemos

$$\begin{aligned} f(\tau|\theta) &\propto f(\theta, \tau) \quad \text{con } \theta \text{ fijo} \\ &\propto \tau^3 \exp\{-\tau(\theta - 1)^2/2\} \exp\{-\tau/5\} \\ &\propto \tau^3 \exp\{-\tau[(\theta - 1)^2/2 + 1/5]\} \end{aligned}$$

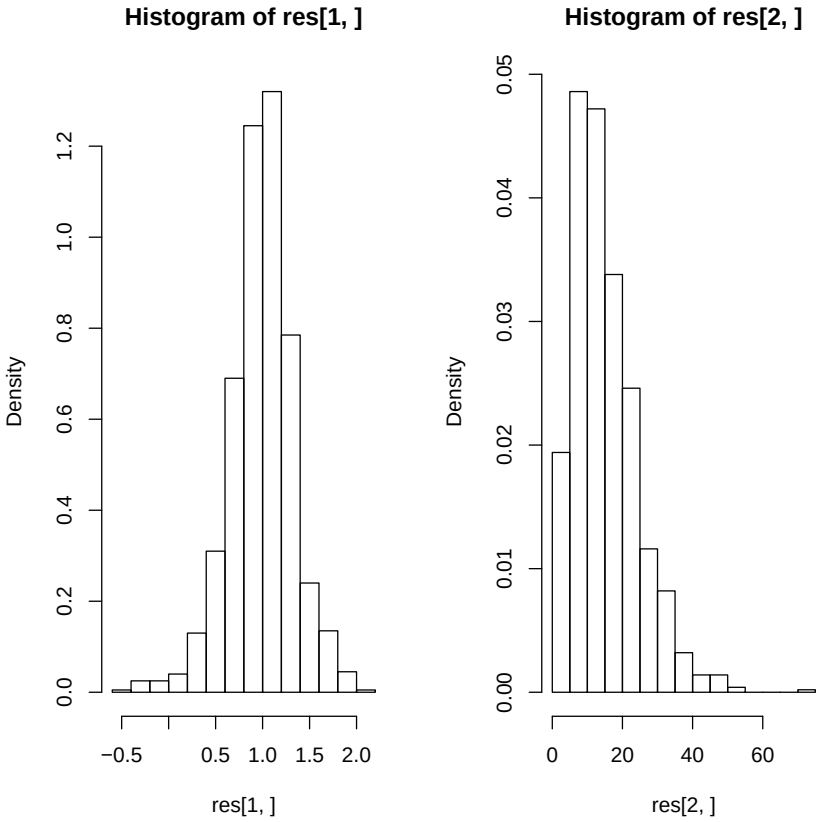
de donde podemos concluir que $\tau|\theta \sim Gamma(4, 1/[(\theta - 1)^2/2 + 1/5])$.

Con estas dos distribuciones condicionales, podemos implementar el muestreador de Gibbs de la siguiente forma

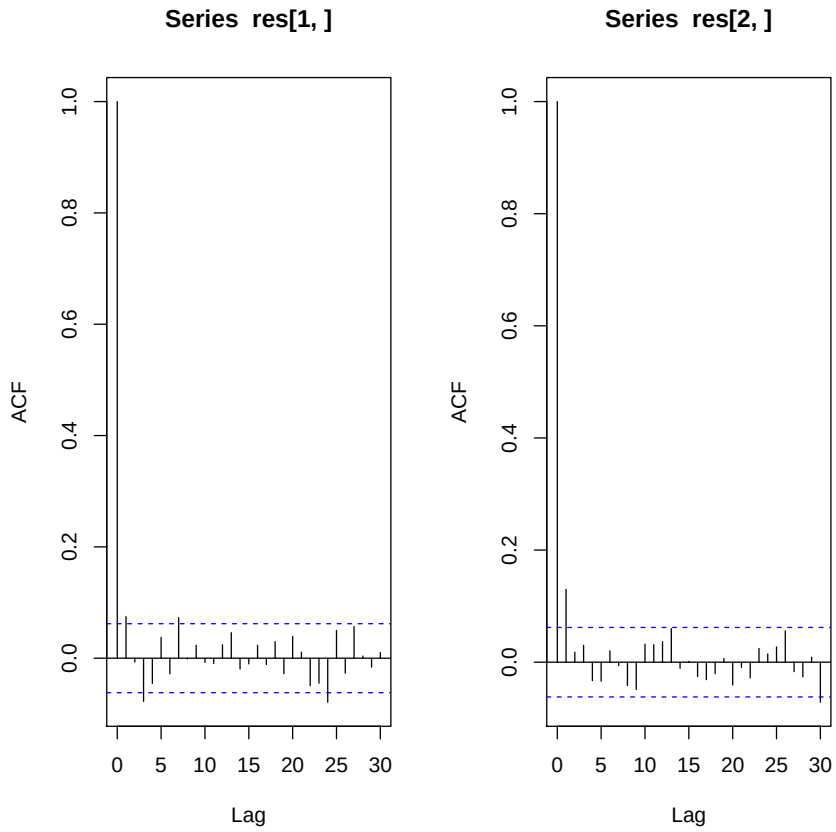
```r
set.seed(123)
n<-1000
res<-matrix(NA,2,n)
# en la fila 1 guarda los theta, y en la fila 2 guarda los tau
res[1,1]<-0 # Valor inicial para theta
res[2,1]<-rgamma(1,shape=4,scale=1/((res[1,1]-1)^2+1/5))
for(i in 2:n){
  res[1,i]<-rnorm(1,1,sqrt(1/res[2,i-1]))
  res[2,i]<-rgamma(1,shape=4,scale=1/((res[1,i]-1)^2+1/5))
}
par(mfrow=c(1,2))
ts.plot(res[1,])
ts.plot(res[2,])
```




```
par(mfrow=c(1,2))

hist(res[1,],freq=F)

hist(res[2,],freq=F)
```




```
par(mfrow=c(1,2))

acf(res[1,])

acf(res[2,])
```



2.2. Ejercicios

1. Suponga que X es una variable que toma valores en $0, 1, \cdots, n$, y θ es una variable que toma valores en $(0, 1)$, y la función de densidad conjunta de X y θ es

$$f(x, \theta) \propto \binom{n}{x} \theta^{x+3}(1-\theta)^{n-x+5}$$

   - Encuentra las distribuciones condicionales $f(x|\theta)$ y $f(\theta|x)$ y diga

a cuáles distribuciones corresponden.

- Usando las anteriores distribuciones condicionales, implemente un muestreo de Gibbs de longitud 1000 para muestrear valores para X y para θ, use $n = 20$.
- Elabore las gráficas pertinentes para evaluar la calidad de los valores obtenidos.

2. Para la densidad del punto anterior,

   - Halle la densidad marginal de $f(\theta)$ de θ, y diga a qué distribución corresponde.
   - Usando las densidades $f(\theta)$ y $f(x|\theta)$, piense cómo sería la variación del muestreador de Gibbs para simular valores para X y θ.
   - Implemente la metodología propuesta en el punto anterior.
   - Elabore las gráficas necesarias para comparar la metodología de este punto con el punto anterior.
   - (*) La distribución marginal de X se llama la distribución Beta binomial, halle la función de densidad de esta distribución.

3. Combinando el muestreador de Gibbs y otros métodos de muestreo

El muestreador de Gibbs consiste en muestrear valores desde las densidades condicionales $f(x|y)$ y $f(y|x)$. El kernel de estas densidades siempre se puede hallar desde la densidad conjunta $f(x, y)$. En la sección anterior, se discutió el caso cuando ambas densidades condicionales tienen formas cerrada y son fáciles de muestrear. Cuando alguna o ambas densidades condicionales no corresponde a una distribución estándar, se necesita combinar el muestreador de Gibbs con otros algoritmos como el de la Grilla, el de aceptación y rechazo o el de MH.

Example 6 Suponga que la variable $h > 0$ y $n > 0$ son variables aleatorias con función de densidad conjunta

$$f(h, n) \propto \prod_{t=1}^{T} \left(1 + \frac{e_t^2}{h^2 n}\right)^{-(n+1)/2}$$

El kernel de las densidades condicionales $f(h|n)$ y $f(n|h)$ tienen la misma forma que $f(h, n)$ y claramente ninguna corresponde a una distribución estándar que sea fácil de simular.

4. *Burn in* y *thinning*

En los métodos de MCMC, se puede presentar el problema de que el valor inicial sea muy diferente al valor real (y así afectar esetadísticas como la media y la desviación estándar), y también el problema de la convergencia lenta debido a la alta correlación entre los valores muestreados.

4.1. *Burn in*

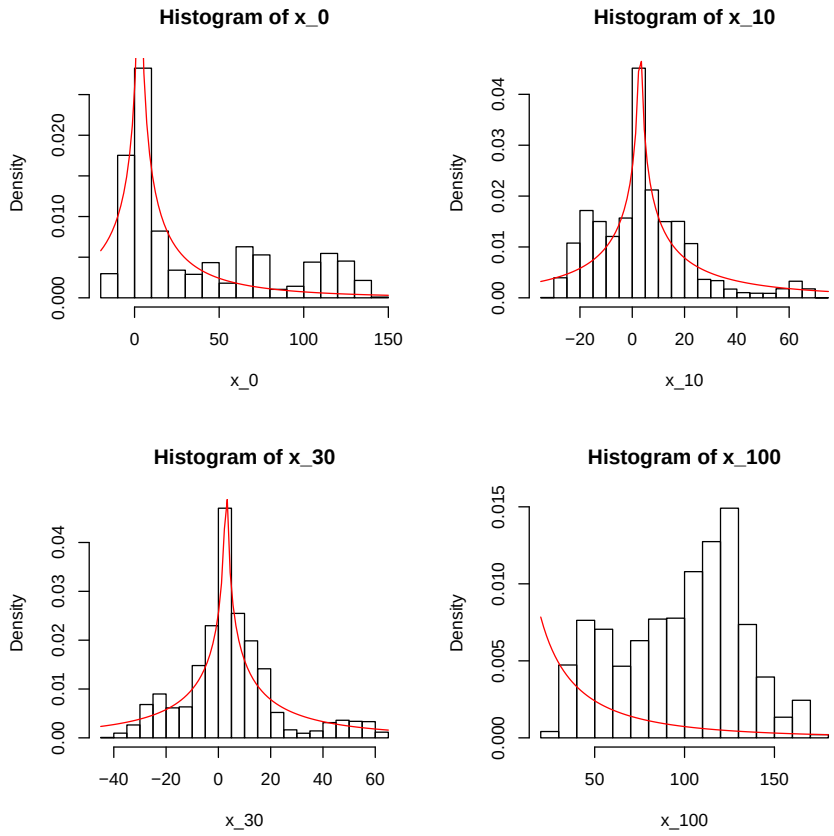
Se refiere a quemar una porción inicial de valores muestreados para minimiar el efecto de los valores iniciales. Ilustramos el efecto del *burn in* en el siguiente ejemplo.

Example 7 Para la densidad del ejemplo 1

```
set.seed(12341)
N.sim<-10000
f<-function(x){exp(-(abs(x-3)/2)^0.4)}
x_0<-x_10<-x_30<-x_100<-NULL
x_0[1]<-0
x_10[1]<-10
x_30[1]<-30
x_100[1]<-100
for(i in 2:N.sim){
  y_0<-rnorm(1,x_0[i-1],1)
```

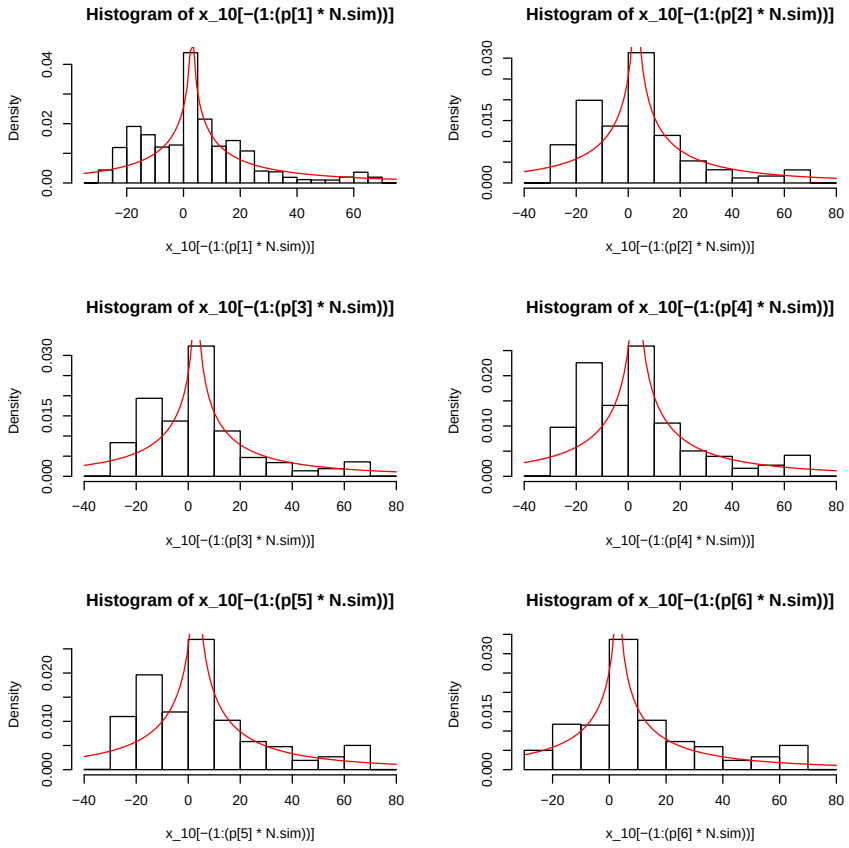
```
  y_10<-rnorm(1,x_10[i-1],1)
  y_30<-rnorm(1,x_30[i-1],1)
  y_100<-rnorm(1,x_100[i-1],1)
  prob_0<-min(1,f(y_0)*dnorm(x_0[i-1],y_0,1)/(f(x_0[i-1])*dnorm(y_0,x_0[i-1],
  prob_10<-min(1,f(y_10)*dnorm(x_10[i-1],y_10,1)/(f(x_10[i-1])*dnorm(y_10,x_1
  prob_30<-min(1,f(y_30)*dnorm(x_30[i-1],y_30,1)/(f(x_30[i-1])*dnorm(y_30,x_3
  prob_100<-min(1,f(y_100)*dnorm(x_100[i-1],y_100,1)/(f(x_100[i-1])*dnorm(y_1
  u_0<-runif(1);u_10<-runif(1);u_30<-runif(1);u_100<-runif(1)
  if(u_0<prob_0){x_0[i]<-y_0}
  if(u_0>=prob_0){x_0[i]<-x_0[i-1]}
  if(u_10<prob_10){x_10[i]<-y_10}
  if(u_10>=prob_10){x_10[i]<-x_10[i-1]}
  if(u_30<prob_30){x_30[i]<-y_30}
  if(u_30>=prob_30){x_30[i]<-x_30[i-1]}
  if(u_100<prob_100){x_100[i]<-y_100}
  if(u_100>=prob_100){x_100[i]<-x_100[i-1]}
}
K<-1/integrate(f,-100,100)$val
f<-function(x){exp(-(abs(x-3)/2)^0.4)*K}
par(mfrow=c(2,2))
hist(x_0,freq=FALSE)
curve(f,add=T,col=2)
hist(x_10,freq=FALSE)
```

```r
curve(f,add=T,col=2)
hist(x_30,freq=FALSE)
curve(f,add=T,col=2)
hist(x_100,freq=FALSE)
curve(f,add=T,col=2)
```



En las anteriores gráficas, vemos que el valor inicial sí puede afectar sobre los valores muestreados. Ahora miramos el efecto del *burn in* para la muestra de `x_30`

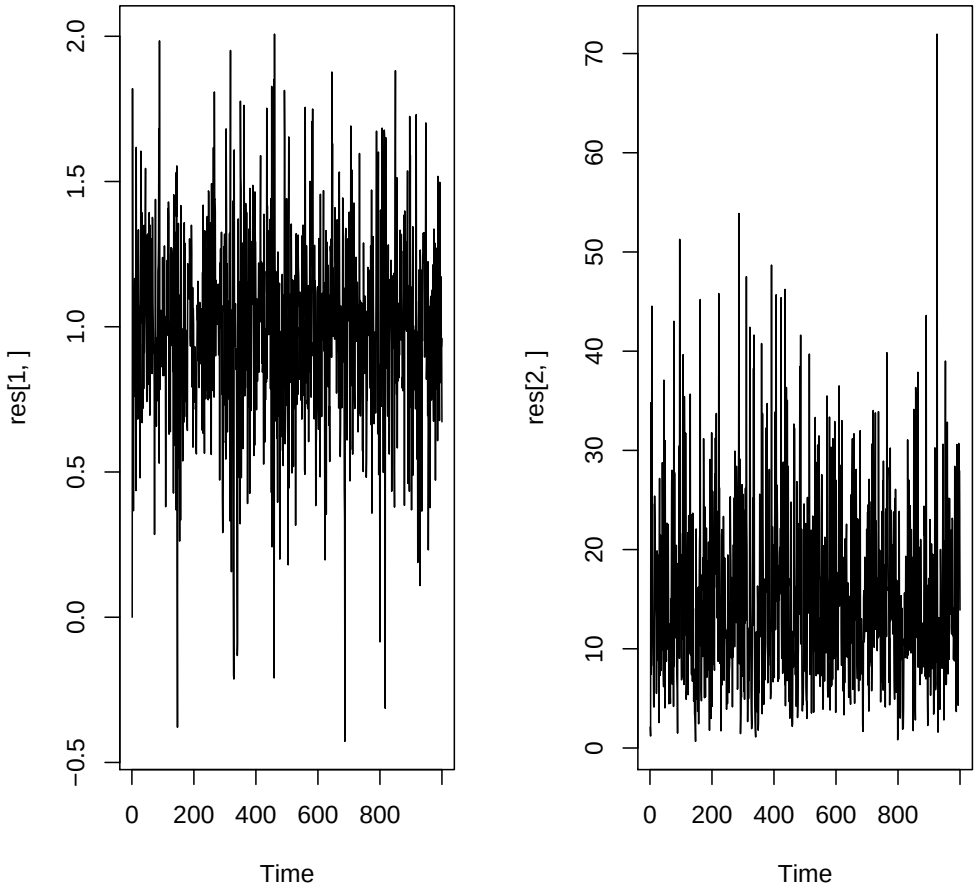
```
p<-c(0.1,0.2,0.3,0.4,0.5,0.6)
par(mfrow=c(3,2))
hist(x_10[-(1:(p[1]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
hist(x_10[-(1:(p[2]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
hist(x_10[-(1:(p[3]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
hist(x_10[-(1:(p[4]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
hist(x_10[-(1:(p[5]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
hist(x_10[-(1:(p[6]*N.sim))],freq=FALSE)
curve(f,add=T,col=2)
```



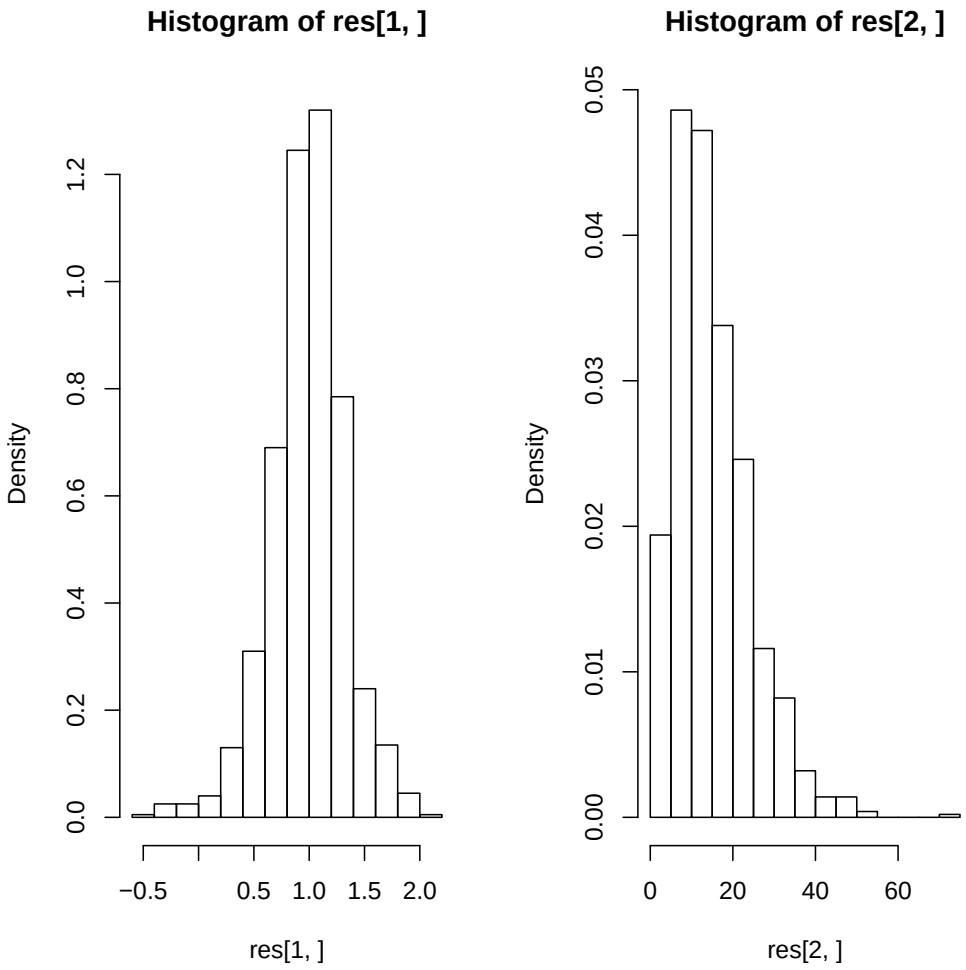
4.2. *Thinning*

La técnica *thinning* consiste en dejar solamente los valores muestreados cada k, por ejemplo, dejar solamente las observaciones $1,\ k+1,\ 2k+1, \cdots$. Para ilustrar el procedimiento, tomamos el muestreador de Gibbs del ejemplo 5

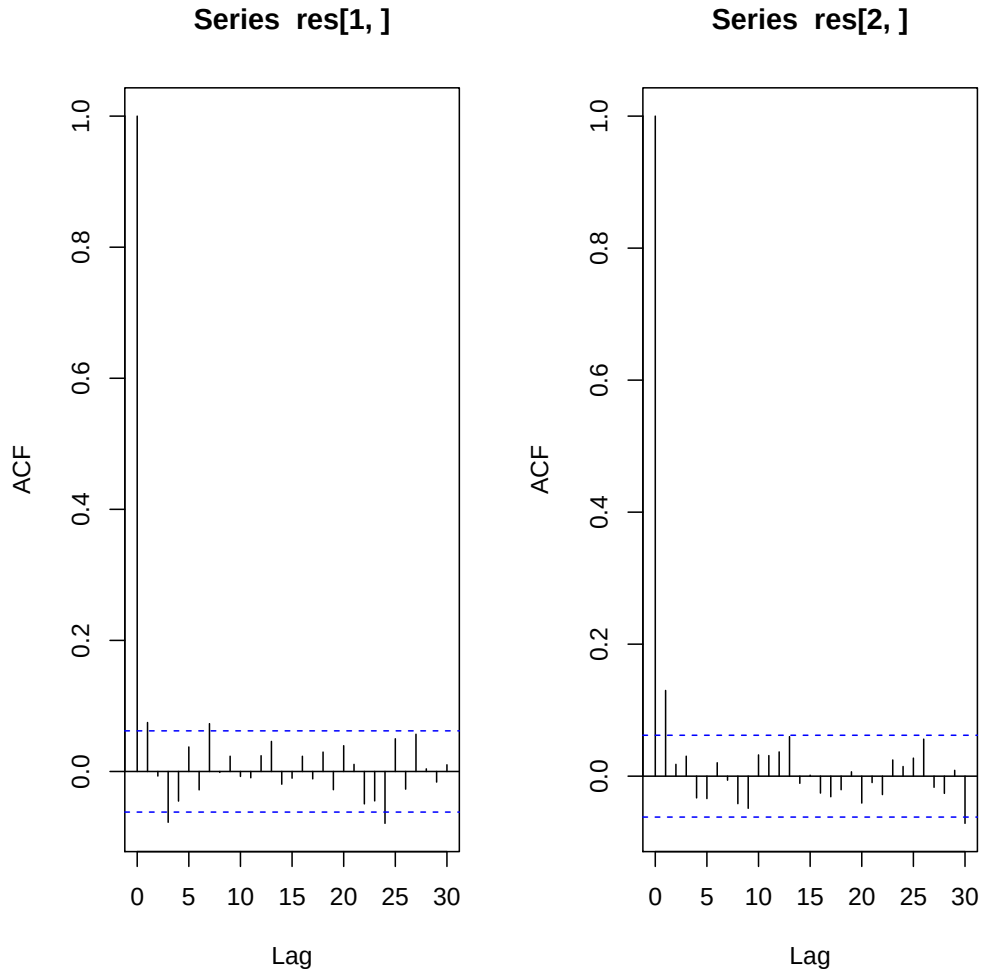
```r
set.seed(123)

n<-1000

res<-matrix(NA,2,n)

# en la fila 1 guarda los theta, y en la fila 2 guarda los tau

res[1,1]<-0 # Valor inicial para theta

res[2,1]<-rgamma(1,shape=4,scale=1/((res[1,1]-1)^2+1/5))

for(i in 2:n){

  res[1,i]<-rnorm(1,1,sqrt(1/res[2,i-1]))

  res[2,i]<-rgamma(1,shape=4,scale=1/((res[1,i]-1)^2+1/5))

}

par(mfrow=c(1,2))

ts.plot(res[1,])

ts.plot(res[2,])
```



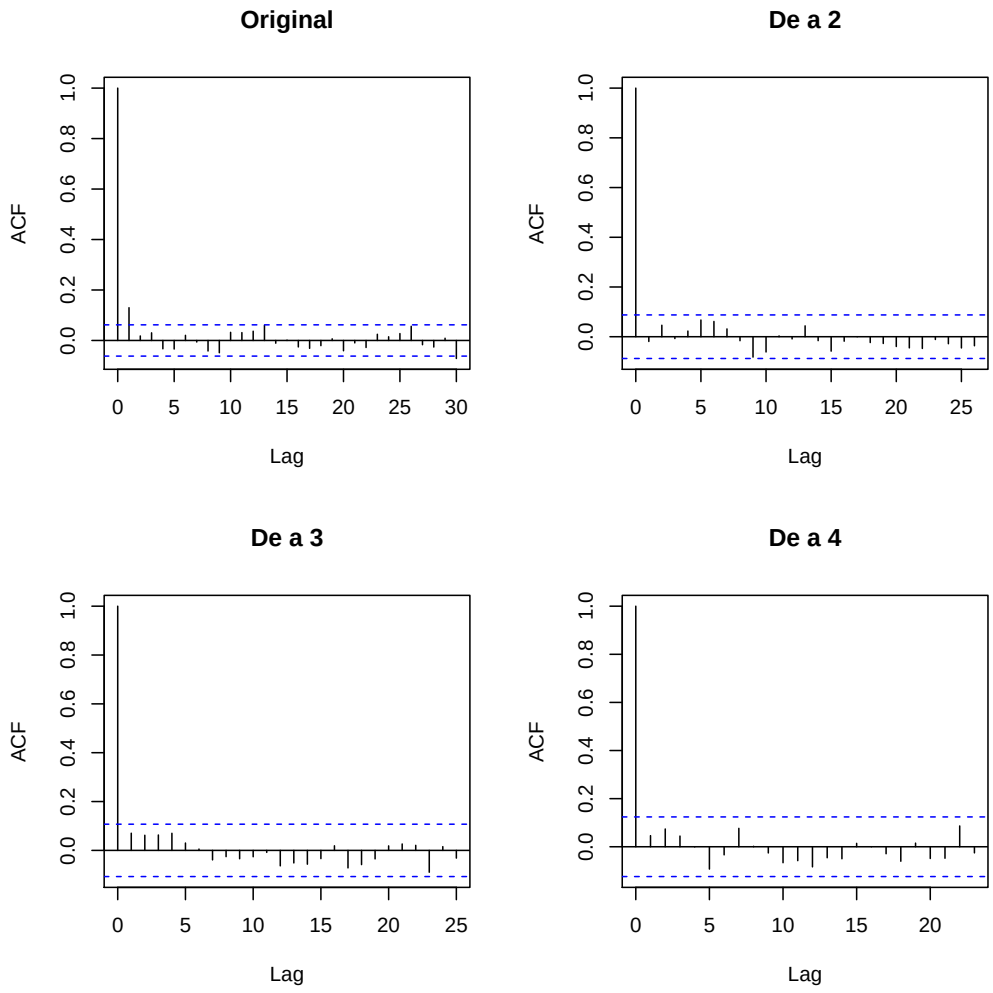

```
par(mfrow=c(1,2))

hist(res[1,],freq=F)

hist(res[2,],freq=F)
```




```
par(mfrow=c(1,2))

acf(res[1,])

acf(res[2,])
```




```
# Tomar los resultados de las iteraciones 1, 3, 5, ...
par(mfrow=c(2,2))
acf(res[2,],main="Original")
acf(res[2,seq(1,n,by=2)],main="De a 2")
acf(res[2,seq(1,n,by=3)],main="De a 3")
acf(res[2,seq(1,n,by=4)],main="De a 4")
```



Conclusión, si se presenta correlación en el rezago k, entonces se deja las observación cada $k+1$.

4.3. Recomendaciones finales

- Generalmente se utiliza el *burn in* y *thinning* conjuntamente.

- En la práctica, también se acostumbra a simular varias cadenas paralelas, y se obtienen criterios de convergencia. Ver por ejemplo Gelman, A and Rubin, DB (1992) Inference from iterative simulation using multiple sequences, Statistical Science, 7, 457-511, y Brooks, SP. and Gelman, A. (1998) General methods for monitoring convergence of iterative simu- lations. Journal of Computational and Graphical Statistics, 7, 434-455.

- En `R`, el paquete `coda` provee muchas herramientas para el diagnóstico de la convergencia en MCMC.

4.4. Ejercicios

1. Para el ejercicio del muestreador de Gibbs con

$$f(x, \theta) \propto \binom{n}{x} \theta^{x+3}(1-\theta)^{n-x+5}$$

   - Por medio de gráficas, evalue el efecto de *burn in* eliminando diferentes porciones de valores iniciales de la cadena obtenida para θ. Diga cuál es la porción ideal para el *burn in*.

   - Por medio de gráficas, evalue el efecto de *thinning* con diferentes valores de k para la cadena obtenida para θ. Diga cuál es el valor ideal para k.