

Simulação Estocástica via Cadeias de Markov III

Ricardo Ehlers
ehlers@icmc.usp.br

Departamento de Matemática Aplicada e Estatística
Universidade de São Paulo

Espaço de estados contínuo

Amostrador de Gibbs

Metropolis-Hastings

Espaço de Estados Contínuo

Seja $\pi(x)$ a função de densidade da distribuição estacionária de uma cadeia de Markov com espaço de estados contínuo. Esta é a distribuição alvo, da qual queremos gerar valores.

Seja $q(x, y)$ a densidade da distribuição proposta, ou seja o processo está no estado x e um valor y é gerado desta distribuição.

O algoritmo de Metropolis-Hastings pode ser descrito em termos da probabilidade de aceitação,

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \frac{q(x, y)}{q(y, x)} \right\}$$

As densidades de transição ficam,

$$p(x, y) = \begin{cases} q(x, y) \alpha(x, y), & y \neq x \\ 1 - \int q(x, t) \alpha(x, t) dt, & y = x \end{cases}$$

e as condições de reversibilidade são,

$$\pi(x)p(x, y) = \pi(y)p(y, x)$$

- ▶ Se estas condições são satisfeitas e $p(x, y)$ define uma cadeia de Markov irredutível e aperiódica, então $\pi(\cdot)$ será a distribuição limite.
- ▶ Tais condições são em geral satisfeitas se $q(x, y) > 0$ no mesmo suporte de $\pi(x)$.
- ▶ $\pi(x)$ pode ser conhecida a menos de uma constante,

$$\pi(x) = Ch(x)$$

sendo $h(x)$ uma função conhecida e

$$C^{-1} = \int_{-\infty}^{\infty} h(x) dx$$

desconhecida.

Amostrador de Gibbs

Lembrando que o amostrador de Gibbs é uma cadeia de Markov onde as transições no espaço de estados são feitas de acordo com as *distribuições condicionais completas*.

Seja $\mathbf{X} = (X_1, \dots, X_d)$ um vetor aleatório contínuo e defina,

$$\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)'.$$

Então,

$$\begin{aligned}\pi(x_i | \mathbf{x}_{-i}) &= \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}_{-i})} \\ &= \frac{\pi(\mathbf{x})}{\int \pi(\mathbf{x}) dx_i} \\ &\propto \pi(\mathbf{x}), \quad i = 1, \dots, d,\end{aligned}$$

Exemplo. Sejam X e Y variáveis aleatórias contínuas com densidade de probabilidade conjunta,

$$f(x, y) = kx^4 \exp(-x(2 + y)), \quad x > 0, y > 0.$$

As densidades condicionais completas são,

$$f(y|x) \propto kx^4 \exp(-x(2 + y)) \propto \exp(-xy), \quad y > 0.$$

$$f(x|y) \propto kx^4 \exp(-x(2 + y)) \propto x^4 \exp(-x(2 + y)), \quad x > 0.$$

Note que,

$$\int_0^{\infty} f(y|x) dy = C \int_0^{\infty} \exp(-xy) dy = 1 \Rightarrow C = x,$$

portanto, $f(y|x) = x \exp(-xy)$ e,

$$Y|X = x \sim \text{Exponencial}(x).$$

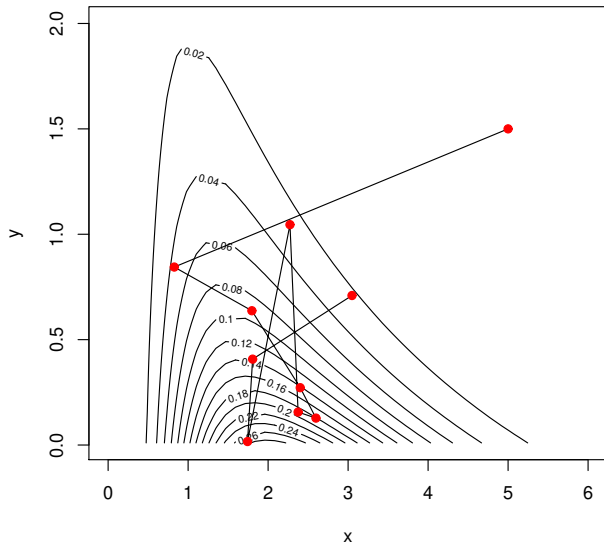
Analogamente pode-se verificar que,

$$X|Y = y \sim \text{Gama}(5, 2 + y).$$

Função em R para o amostrador de Gibbs do exemplo.

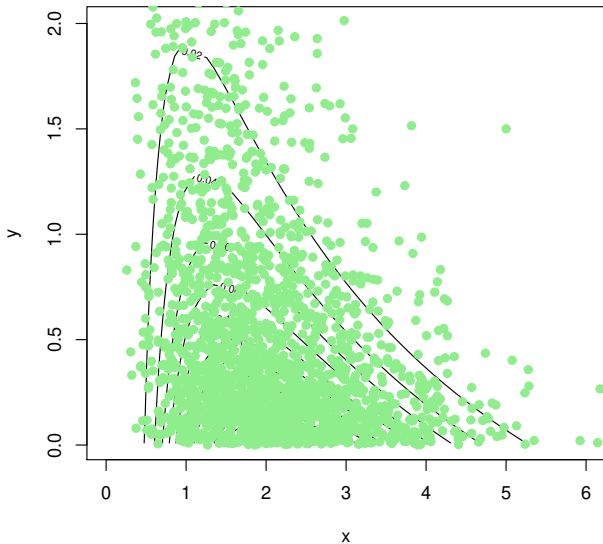
```
> gibbs <- function(x0,y0,niter) {  
+   x = y = rep(0, niter)  
+   x[1] = x0  
+   y[1] = y0  
+   for (i in 2:niter) {  
+       x[i] = rgamma(1,5,2+y[i-1])  
+       y[i] = rexp(1,x[i])  
+   }  
+   return(cbind(x,y))  
+ }
```

10 valores gerados com ponto inicial $x = 5$ e $y = 1.5$.



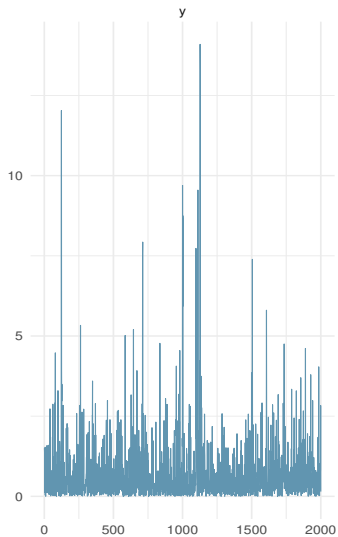
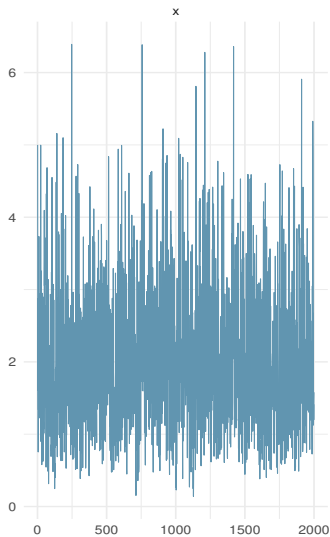
```
> contour.xy <- function(M=50) {  
+   x=seq(0,6,length=M)  
+   y=seq(0.01,2,length=M)  
+   f= matrix(0,M,M)  
+   for (i in 1:M) {  
+     for (j in 1:M) {  
+       f[i,j]= x[i]^4 * exp(-x[i]*(2 +y[j]))  
+     }  
+   }  
+   contour(x,y,f,nlevels=20,xlab="x",ylab="y")  
+ }  
  
> contour.xy(M=50)  
> set.seed(134); m = gibbs(x0=5,y0=1.5,niter=10)  
> lines(m,type="l")  
> points(m,col="red",pch=19)
```

2000 valores gerados com ponto inicial $x = 5$ e $y = 1.5$.



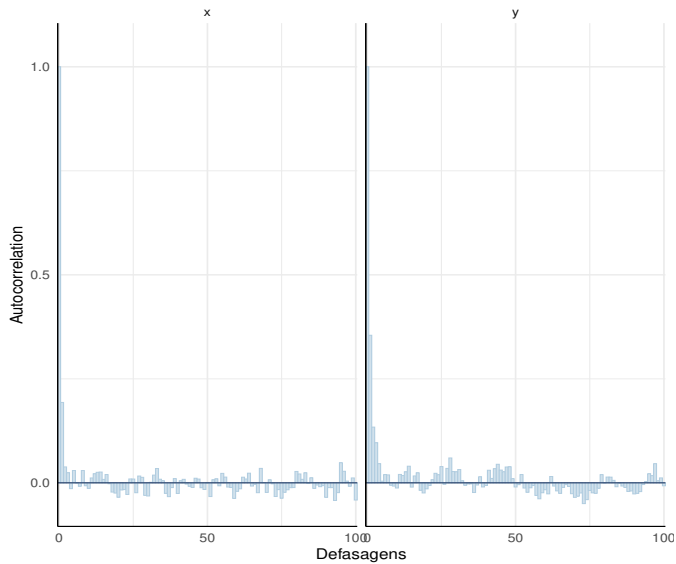
```
> contour.xy(M=50)
> m = gibbs(x0=x0,y0=y0,niter=niter)
> points(m,col="lightgreen",pch=19)
```

Traços das cadeias simuladas



```
> library(bayesplot)
> library(ggplot2)

> plot_title <- ggtitle("Traços das cadeias simuladas")
> bayesplot_theme_set(theme_minimal())
> color_scheme_set("blue")
> mcmc_trace(as.matrix(m)) + plot_title
```



Exemplo. Modelo autoexponencial (Besag, 1974). Seja o vetor aleatório $\mathbf{y} = (y_1, y_2, y_3)$ cuja função de densidade conjunta é,

$$f(\mathbf{y}) \propto \exp[-(y_1 + y_2 + y_3 + \theta_{12}y_1y_2 + \theta_{23}y_2y_3 + \theta_{31}y_3y_1)],$$

com $y_1, y_2, y_3 > 0$ e $\theta_{12}, \theta_{23}, \theta_{31}$ conhecidos.

Verifique que,

$$f(y_2|y_1) \propto \frac{\exp[-(y_1 + y_2 + \theta_{12}y_1y_2)]}{1 + \theta_{23}y_2 + \theta_{31}y_1},$$
$$f(y_1) \propto \exp(-y_1) \int_0^\infty \frac{\exp[-(y_2 + \theta_{12}y_1y_2)]}{1 + \theta_{23}y_2 + \theta_{31}y_1} dy_2$$

que são difíceis de serem simuladas.

No entanto as distribuições condicionais completas são todas exponenciais. Por exemplo,

$$\begin{aligned} f(y_3|y_1, y_2) &\propto \exp[-(y_1 + y_2 + y_3 + \theta_{12}y_1y_2 + \theta_{23}y_2y_3 + \theta_{31}y_3y_1)] \\ &\propto \exp[-(y_3 + \theta_{23}y_2y_3 + \theta_{31}y_3y_1)] \\ &\propto \exp[-y_3(1 + \theta_{23}y_2 + \theta_{31}y_1)], \quad y > 0. \end{aligned}$$

Portanto,

$$Y_3|y_1, y_2 \sim \text{Exponencial}(1 + \theta_{23}y_2 + \theta_{31}y_1).$$

Verifique as outras!

Exemplo. Sejam Y_1, \dots, Y_n independentes e identicamente distribuídos sendo,

$$Y_i|X \sim \text{Poisson}(X), \quad i = 1, \dots, n$$

$$X|Z \sim \text{Exponencial}(Z)$$

$$Z \sim \text{Gama}(c, d)$$

com c e d conhecidos.

Temos então que,

$$\begin{aligned} p(\mathbf{y}|x) = p(y_1, \dots, y_n|x) &= \prod_{i=1}^n \frac{x^{y_i} e^{-x}}{y_i!} \\ p(x|z) &= ze^{-zx} \\ p(z) &= \frac{d^c}{\Gamma(c)} z^{c-1} e^{-dz}. \end{aligned}$$

Suponha que foram observados os valores y_1, \dots, y_n de Y_1, \dots, Y_n e deseja-se simular valores de X e Z condicionalmente em y_1, \dots, y_n .

Note que,

$$\begin{aligned} p(x, z | \mathbf{y}) &= \frac{p(x, z, \mathbf{y})}{p(\mathbf{y})} \propto p(\mathbf{y} | x) p(x | z) p(z) \\ &\propto \prod_{i=1}^n \frac{x^{y_i} e^{-x}}{y_i!} p(x | z) p(z) \\ &\propto x^t e^{-nx} z e^{-zx} z^{c-1} e^{-dz}, \quad t = \sum_{i=1}^n y_i. \end{aligned}$$

Portanto,

$$\begin{aligned} p(x|z, \mathbf{y}) &\propto x^t e^{-nx} z e^{-zx} z^{c-1} e^{-dz} \\ &\propto x^t e^{-(z+n)x} \end{aligned}$$

$$\begin{aligned} p(z|x, \mathbf{y}) &\propto x^t e^{-nx} z e^{-zx} z^{c-1} e^{-dz} \\ &\propto z^c e^{-(x+d)z} \end{aligned}$$

e as distribuições condicionais completas são,

$$\begin{aligned} X|z, \mathbf{y} &\sim \text{Gama}(t+1, z+n) \\ Z|x, \mathbf{y} &\sim \text{Gama}(c+1, x+d). \end{aligned}$$

```
> Gibbs2 <- function(c,d,y,niter) {  
+   N = length(y)  
+   lambda = matrix(0, nrow=niter)  
+   beta    = matrix(0, nrow=niter)  
+   lambda[1]= 1  
+   beta   [1]= 1  
+   t1 = sum(y)  
+   for (i in 2:niter) {  
+     lambda[i]= rgamma(1, 1+t1, beta[i-1]+N)  
+     beta   [i]= rgamma(1, 1+c, lambda[i]+d)  
+   }  
+   return(theta = list(lambda=lambda,beta=beta))  
+ }
```

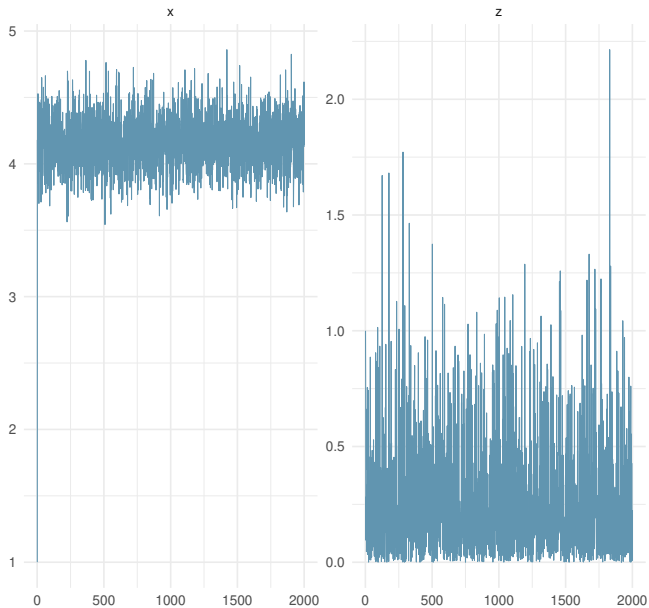
Exemplo. Aplicando o algoritmo anterior com dados simulados $Y_1, \dots, Y_n \sim \text{Poisson}(4)$, $n = 100$, $c = d = 0.01$ e 2000 iterações.

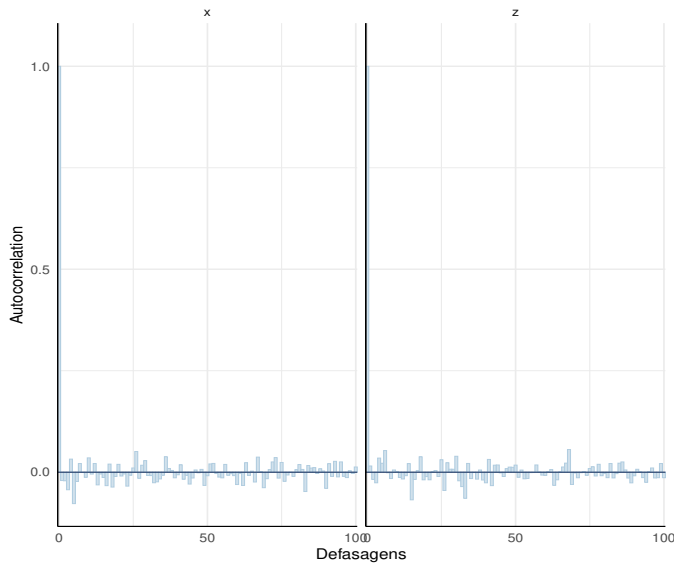
```
> y= rpois(100,lambda=4)
> niter = 2000
> q = Gibbs2(c = 0.01, d = 0.01, y, niter= niter)
> library(coda)
> q1= cbind(q$lambda, q$beta)
> colnames(q1)= c("x","z")
```

```
> y
```

[1]	3	5	1	5	3	4	7	5	2	2	7	6	2	1	7	4	5	6	3	5	6	4
[26]	4	3	3	6	5	2	3	6	7	2	2	5	2	3	3	3	8	5	7	2	9	3
[51]	5	10	7	4	5	7	1	7	2	4	6	3	3	5	5	3	4	3	2	3	4	5
[76]	9	3	4	2	1	4	4	5	2	4	4	4	5	2	5	4	3	4	0	5	7	4

Traços das cadeias simuladas





Exemplo. Em um processo de contagem no qual foram observados Y_1, \dots, Y_n suspeita-se que houve um ponto de mudança m tal que

$$\begin{aligned} Y_i | \lambda &\sim \text{Poisson}(\lambda), & i = 1, \dots, m \\ Y_i | \phi &\sim \text{Poisson}(\phi), & i = m + 1, \dots, n. \end{aligned}$$

sendo m , λ e ϕ desconhecidos.

Suponha que m , λ e ϕ são variáveis aleatórias independentes com as seguintes distribuições,

$$\begin{aligned} \lambda &\sim \text{Gama}(a, b), & a > 0, b > 0 \\ \phi &\sim \text{Gama}(c, d), & c > 0, d > 0 \\ p(m) &= 1/n. \end{aligned}$$

Deseja-se simular valores de (m, λ, ϕ) dados y_1, \dots, y_n observados.

Note que,

$$\begin{aligned}
 p(\lambda, \phi, m | \mathbf{y}) &= \frac{p(\mathbf{y} | \lambda, \phi, m) p(\lambda) p(\phi) p(m)}{p(\mathbf{y})} \\
 &\propto \prod_{i=1}^m e^{-\lambda} \lambda^{y_i} \prod_{i=m+1}^n e^{-\phi} \phi^{y_i} \lambda^{a-1} e^{-b\lambda} \phi^{c-1} e^{-d\phi} \frac{1}{n} \\
 &\propto \lambda^{a+t_1-1} e^{-(b+m)\lambda} \phi^{c+t_2-1} e^{-(d+n-m)\phi} \frac{1}{n}
 \end{aligned}$$

sendo

$$t_1 = \sum_{i=1}^m y_i \quad \text{e} \quad t_2 = \sum_{i=m+1}^n y_i.$$

.

Portanto, as densidades condicionais completas ficam,

$$\begin{aligned}p(\lambda|\phi, m, \mathbf{y}) &\propto \lambda^{a+t_1-1} e^{-(b+m)\lambda} \\p(\phi|\lambda, m, \mathbf{y}) &\propto \phi^{c+t_2-1} e^{-(d+n-m)\phi} \\p(m|\lambda, \phi, \mathbf{y}) &\propto \lambda^{t_1} e^{-m\lambda} \phi^{t_2} e^{-(n-m)\phi}, \quad m = 1, \dots, n.\end{aligned}$$

ou seja,

$$\begin{aligned}\lambda|\phi, m, \mathbf{y} &\sim \text{Gama}(a + t_1, b + m) \\ \phi|\lambda, m, \mathbf{y} &\sim \text{Gama}(c + t_2, d + n - m)\end{aligned}$$

```

> Gibbs <- function(a,b,c,d,y,niter){
+ N = length(y)
+ lambda = phi = m = matrix(0, nrow=niter)
+ lambda[1] = phi[1] = 1; m[1] = 10
+ for (i in 2:niter) {
+   t1 = sum(y[1:m[i-1]]); t2 = 0; prob = NULL
+   if (m[i-1] < N) t2 = sum(y[(m[i-1]+1):N])
+   lambda[i] = rgamma(1,(a + t1), (b + m[i-1]))
+   phi[i] = rgamma(1,(c + t2), (d + N-m[i-1]))
+   for (j in 1:N){
+     t1 = sum(y[1:j])
+     t2 = 0
+     if (j < N) t2 = sum(y[(j+1):N])
+     aux=(lambda[i-1]^t1)*exp(-j*lambda[i-1])*(phi[i-1]^t2)*exp(-(N-j))
+     prob = c(prob,aux)
+   }
+   soma = sum(prob)
+   probm = prob/soma
+   m[i] = sample(x=N, size=1, prob=probm)
+ }
+ return(list(lambda=lambda, phi=phi, m=m))}

```

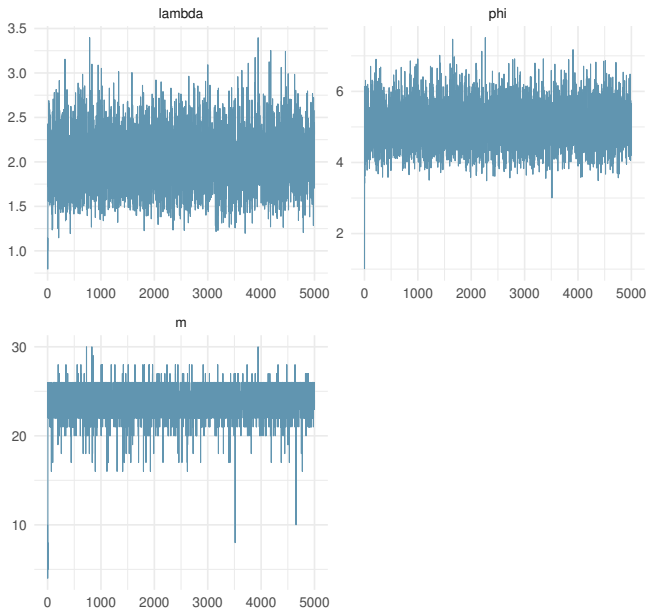
Exemplo. Testando a função Gibbs com 40 dados simulados de processos com médias 2 e 5 e ponto de mudança 23.

```
> set.seed(124)
> y= c(rpois(n=22, lambda=2),rpois(n=18, lambda=5))
> y

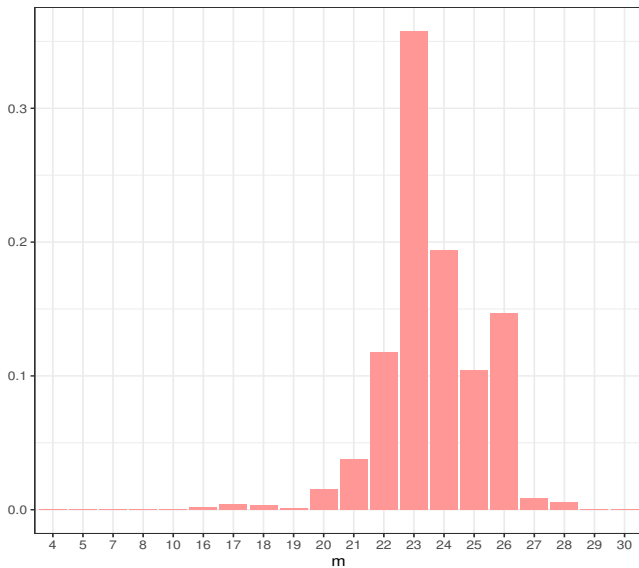
 [1]  0  2  2  1  1  1  2  2  4  1  3  3  3  3  2  0  2  3  4  0
[26]  3  7  4  7  4  7  4  4  7  2 10  3  9  3  4

> niter= 5000
> x = Gibbs(a=0.1, b=0.1, c=0.1, d=0.1, y=y, niter=niter)
```

Traços das cadeias simuladas



Distribuição aproximada de $m|y$.



Algoritmo de Metropolis-Hastings

1. Especifique um valor inicial da cadeia X_0 .
2. Se $X_t = x$,
 - ▶ Gere y da distribuição proposta $q(x, \cdot)$ e $u \sim U(0, 1)$.
 - ▶ Calcule $\alpha(x, y)$
 - ▶ Se $u < \alpha(x, y)$ faça $X_{t+1} = y$, caso contrário faça $X_{t+1} = x$.
 - ▶ Faça $t = t + 1$ e retorne ao passo 2.
3. Retorne os valores $\{X_0, X_1, \dots, X_N\}$.

Exemplo. Deseja-se gerar valores de uma variável aleatória X cuja distribuição é uma mistura de distribuições normais.

$$X \sim p N(\mu, \tau^2) + (1 - p)N(\theta, \sigma^2)$$

Ou seja X tem distribuição,

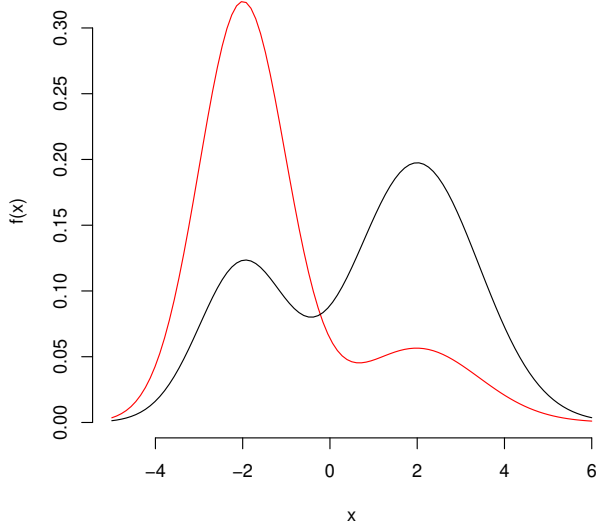
$N(\mu, \tau^2)$, com probabilidade p , ou
 $N(\theta, \sigma^2)$, com probabilidade $1 - p$.

A função de densidade de probabilidade de X é dada por,

$$f(x) = p \frac{1}{\sqrt{2\pi\tau^2}} \exp \left\{ -\frac{(x - \mu)^2}{2\tau^2} \right\} + (1-p) \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x - \theta)^2}{2\sigma^2} \right\}$$

sendo $\mu \in \mathbb{R}$, $\theta \in \mathbb{R}$, $\sigma^2 > 0$, $\tau^2 > 0$ e $p \in (0, 1)$ os parâmetros da distribuição.

Mistura de densidades $N(-2, 1)$ e $N(2, 2)$ com pesos $(0.3, 0.7)$ e $(0.8, 0.2)$.



Para aplicar o algoritmo Metropolis-Hastings será usada uma distribuição proposta normal centrada no estado atual da cadeia.

Ou seja, se a cadeia está no estado $X_t = x$ um valor $y \sim N(x, s^2)$ será proposto.

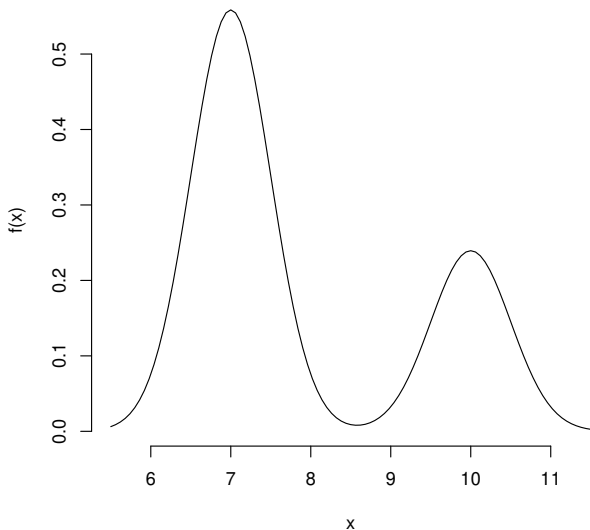
Note que neste caso,

$$q(x, y) = \frac{1}{\sqrt{2\pi s^2}} \exp \left\{ -\frac{1}{2s^2} (x - y)^2 \right\} = q(y, x),$$

Portanto a probabilidade de aceitação fica,

$$\alpha(x, y) = \min \left\{ 1, \frac{f(y)}{f(x)} \right\}.$$

Exemplo. Deseja-se gerar valores de X com $p = 0.7$, $\mu = 7$, $\tau^2 = 0.5^2$, $\theta = 10$ e $\sigma^2 = 0.5^2$.

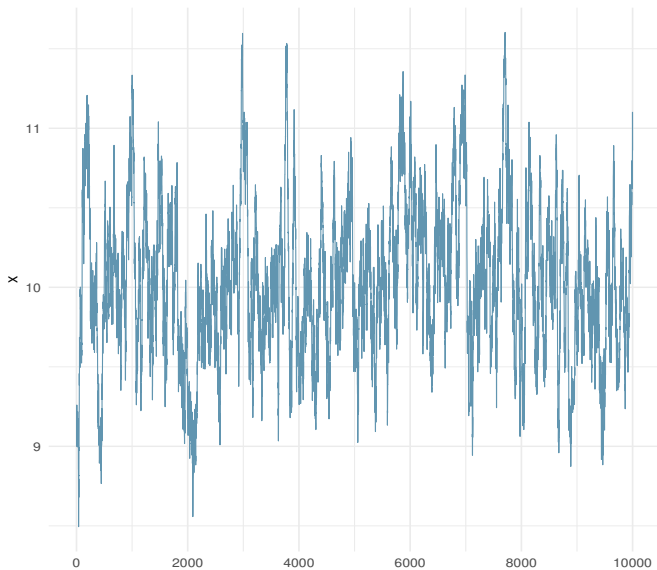


```

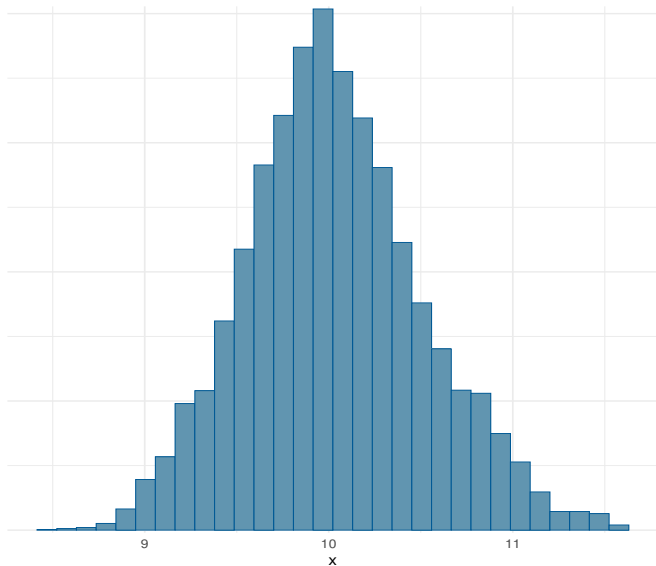
> dmixnorm <- function(x,p,mu,tau,theta,sigma) {
+   p*dnorm(x,mu,tau)+(1-p)*dnorm(x,theta,sigma)
+ }
> normal.metro <- function(x0,n,sd,p,mu,tau,theta,sigma) {
+   set.seed(1234,"Mersenne-Twister")
+   r= rep(0,n)
+   r[1]= x0
+   for(k in 2:n) {
+     u= runif(1)
+     x= r[k-1]
+     y= rnorm(1,x,sd)
+     a= dmixnorm(y,p,mu,tau,theta,sigma)/dmixnorm(x,p,mu,tau,theta,sigma)
+     alpha = min(1,a)
+     if (u < alpha) {
+       r[k]=y
+     } else {
+       r[k]=x
+     }
+   }
+   return(r)
+ }

```

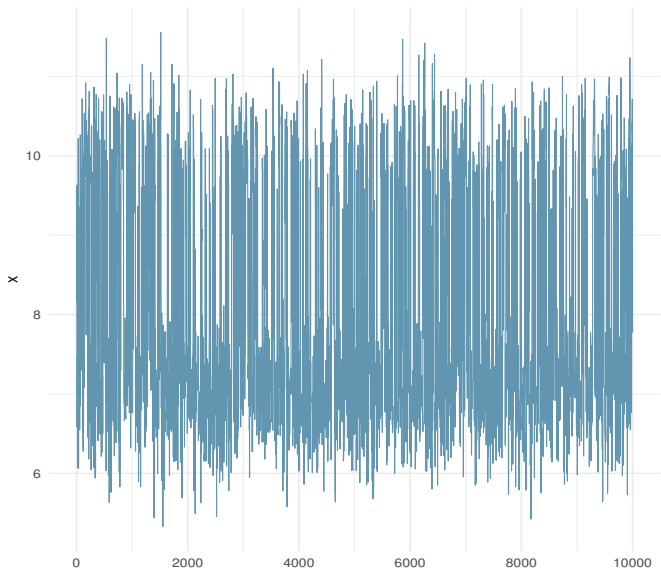
Valores simulados com $X_0 = 9$ e $s = 0.1$.



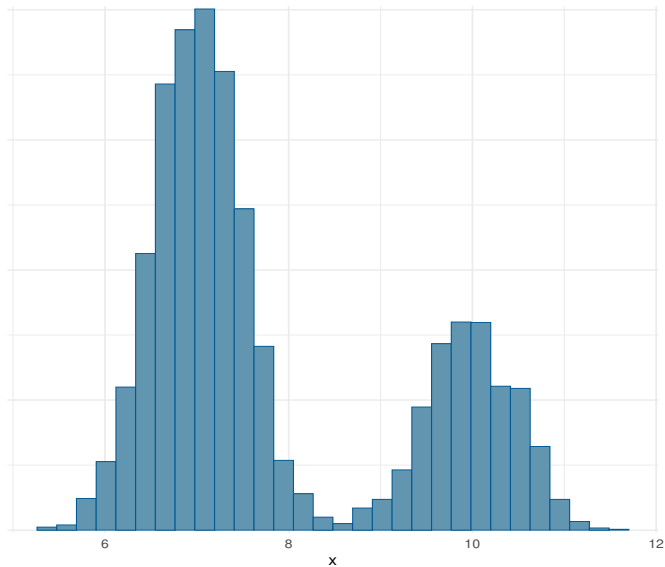
Histograma dos valores simulados com $X_0 = 9$ e $s = 0.1$.



Valores simulados com $X_0 = 9$ e $s = 1.5$.



Histograma dos valores simulados com $X_0 = 9$ e $s = 1.5$.



- ▶ A partir destes gráficos com valores simulados, o algoritmo parece ser mais eficiente quando $s = 1.5$ do que quando $s = 0.1$.
- ▶ Na prática é preciso experimentar alguns valores de s .
- ▶ Se s for muito pequeno, a cadeia simulada pode ser uma representação pobre da distribuição alvo.

Escolha da distribuição proposta

- ▶ A distribuição proposta é arbitrária mas na prática deve-se tomar alguns cuidados para garantir a eficiência do algoritmo.
- ▶ Na maioria das aplicações, a escolha de $q(x, y)$ é crucial.
- ▶ As escolhas mais comuns são propostas do tipo passeio aleatório e propostas independentes.

Passeio Aleatório

- ▶ A ideia é usar o valor atual da cadeia de Markov para propor um novo valor.
- ▶ Se $X_t = x$ podemos propor um novo valor y como,

$$y = x + \epsilon$$

sendo ϵ uma perturbação aleatória simétrica em torno de zero.

- ▶ Se $E(\epsilon) = 0$ e $Var(\epsilon) = s^2$ então,

$$E(Y|X_t = x) = x \quad \text{e} \quad Var(Y|X_t = x) = s^2.$$

- ▶ Neste caso, a densidade proposta é simétrica e depende apenas da distância entre x e y , i.e. $q(x, y) = h(|y - x|)$.
- ▶ Consequentemente $q(x, y) = q(y, x)$ e,

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\}.$$

Esta proposta é muito utilizada na prática mas é preciso ajustar a variância da distribuição.

Se $q(\cdot)$ tem variância s^2 duas situações extremas podem ocorrer,

1. s^2 muito pequena. Valores gerados estarão próximos do atual e quase sempre serão aceitos. Levará muitas iterações para cobrir todo o espaço de X ;
2. Valores grandes de s^2 . Taxa de rejeição excessivamente alta e a cadeia se movimenta muito pouco.

Uma possível distribuição proposta é $y \sim N(x, s^2)$, ou equivalentemente,

$$y = x + \epsilon, \quad \epsilon \sim N(0, s^2).$$

Propostas independentes

A distribuição proposta não depende do estado atual da cadeia,
 $q(x, y) = q(y)$.

Neste caso a probabilidade de aceitação fica,

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \frac{q(x)}{q(y)} \right\}.$$

- ▶ Em geral, $q(\cdot)$ deve ser uma boa aproximação de $\pi(\cdot)$, mas é mais seguro se $q(\cdot)$ tiver caudas mais pesadas do que $\pi(\cdot)$.
- ▶ A cadeia de Markov resultante continua a ser correlacionada pois a probabilidade de aceitação ainda depende de x .

Exemplo. No exemplo da mistura de distribuições normais considere o algoritmo de Metropolis-Hastings com propostas independentes.

- ▶ Considere novamente uma distribuição proposta normal porém independente do valor atual da cadeia.
- ▶ Se $X_t = x$ um novo valor y é proposto como

$$y \sim N(m, s^2)$$

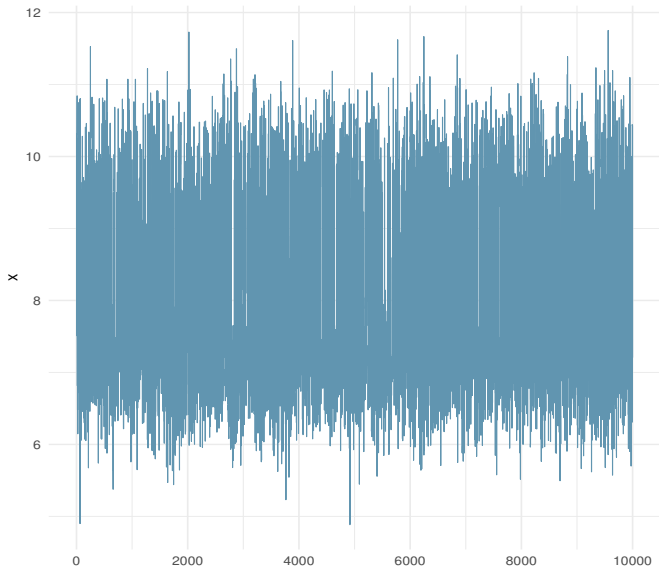
sendo $m = p\mu + (1 - p)\theta$.

```

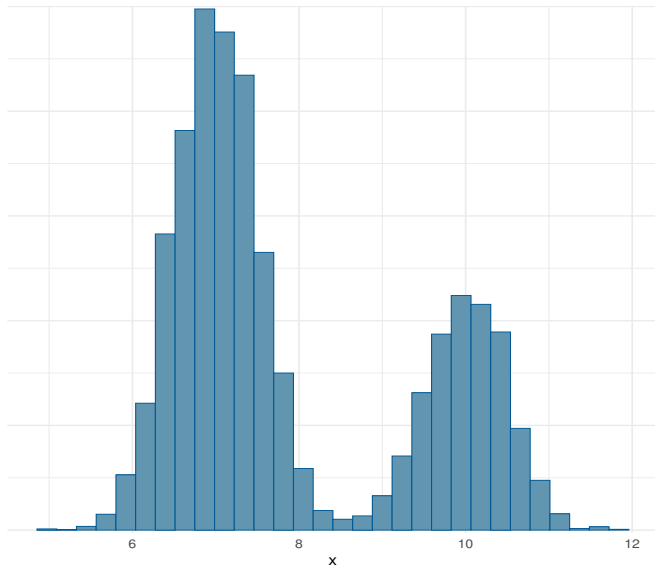
> normal.metro1 <- function(x0,n,sd,p,mu,tau,theta,sigma) {
+   r= rep(0,n)
+   r[1]= x0
+   m= p*mu+(1-p)*theta
+   for (k in 2:n) {
+     u= runif(1)
+     x= r[k-1]
+     y= rnorm(1,m,sd)
+     propratio= dnorm(x,m,sd)/dnorm(y,m,sd)
+     a= dmixnorm(y,p,mu,tau,theta,sigma)/dmixnorm(x,p,mu,tau,theta,sig
+     a= a*propratio
+     alpha = min(1,a)
+     if (u < alpha) {
+       r[k]=y
+     } else {
+       r[k]=x
+     }
+   }
+   return(r)
+ }

```

Valores simulados da mistura de normais com $X_0 = 8$ e $s = 1.5$.



Histograma dos valores simulados da mistura de normais.



Exemplo. Deseja-se simular valores de uma variável aleatória $X \sim \text{Gama}(a, b)$, $a > 0$, $b > 0$.

Usaremos o algoritmo de Metropolis-Hastings com propostas independentes.

A distribuição proposta será normal com média e variância iguais a da distribuição Gama.

Ou seja, se $X_t = x$ um novo valor y é proposto como

$$y \sim N(m, s^2)$$

sendo

$$m = \frac{a}{b} \quad \text{e} \quad s^2 = \frac{a}{b^2}.$$

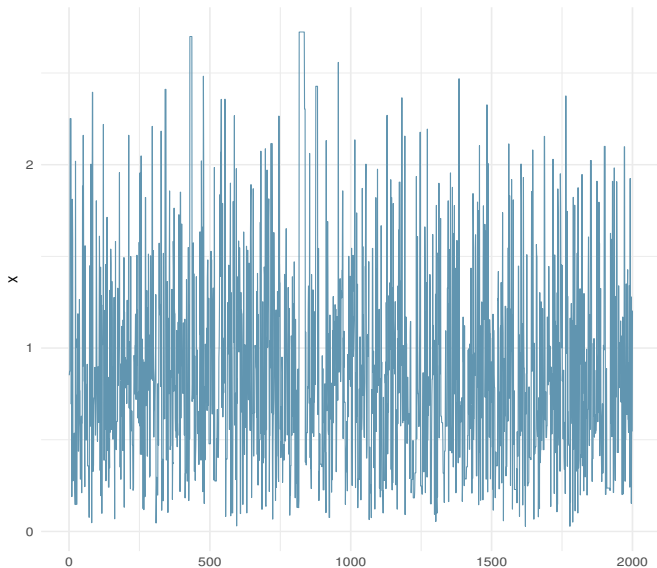
Neste caso a probabilidade de aceitação fica,

$$\alpha(x, y) = \min \left\{ 1, \frac{f_G(y|a, b)}{f_G(x|a, b)} \frac{f_N(x|m, s^2)}{f_N(y|m, s^2)} \right\}.$$

sendo $f_G(\cdot|a, b)$ e $f_N(\cdot|m, s^2)$ as funções de densidade Gama com parâmetros a e b e Normal com parâmetros m e s^2 .

```
> gamm <- function (n, a, b) {  
+ mu <- a/b  
+ sig <- sqrt(a/(b * b))  
+ vec <- vector("numeric", n)  
+ x <- a/b  
+ vec[1] <- x  
+ for (i in 2:n) {  
+   can <- rnorm(1, mu, sig)  
+   r <- (dgamma(can,a,b)/dgamma(x,a,b))/(dnorm(can,mu,sig)/dnorm(x,mu,  
+   aprob <- min(1,r)  
+   u <- runif(1)  
+   if (u < aprob) x <- can  
+   vec[i] <- x  
+ }  
+ vec  
+ }
```


Valores simulados da distribuição Gama(2.3,2.7).



Histograma dos valores simulados da Gama.

