

Metropolis-Hasting

Introducción

Uno de los grandes problemas en el mundo de la estadística es la existencia de distribuciones cuyas funciones de densidad son tan complejas que resultan difíciles de trabajar. Es por esto que muchas veces resulta de utilidad trabajar con muestras aleatorias, y a partir de estas responder determinadas preguntas. En la práctica, es común encontrarse con variables aleatorias con funciones de densidad cuyo muestreo directo no es simple.

Para estos casos existen distintas técnicas que permiten obtener muestras que, si bien no son tomadas de manera realmente independiente, se comportan de manera muy similar a como lo haría una muestra aleatoria independiente tomada de la función de densidad.

Particularmente en el campo de la *Estadística Bayesiana*, esto resulta útil ya que permite obtener muestras de la distribución a posteriori, la cual se puede utilizar por ejemplo para obtener un intervalo de credibilidad del parámetro bajo estudio.

Un método sencillo de emplear y que reporta buenos resultados es el algoritmo de *Metropolis-Hastings*.

Metodología

Si se quiere obtener una muestra de la función de densidad $f(x)$, de la cual no es sencillo obtener muestras, el algoritmo de Metropolis-Hastings plantea que proponiendo una función $q(x)$ conocida, de la cual es sencillo muestrear, y un punto inicial (θ) es posible obtener muestras de la función $f(x)$.

El método funciona de la siguiente forma:

- De la distribución $q(x)$ centrada en θ muestrear un nuevo punto (θ')
- Si la densidad de la función $f(x)$ es mayor en θ' que en θ , entonces θ' formará parte de la muestra, en caso contrario será parte de la muestra con probabilidad $\frac{f(\theta')}{f(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}$ o con probabilidad $1 - \frac{f(\theta')}{f(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}$ se repetirá θ en la muestra.
- Repetir el proceso, obteniendo nuevos puntos propuestos a partir de $q(x)$ centrada en el último valor muestreado.

En definitiva, la probabilidad de seleccionar θ' como nuevo valor de la muestra es $\alpha = \min \left\{ 1; \frac{f(\theta')}{f(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \right\}$

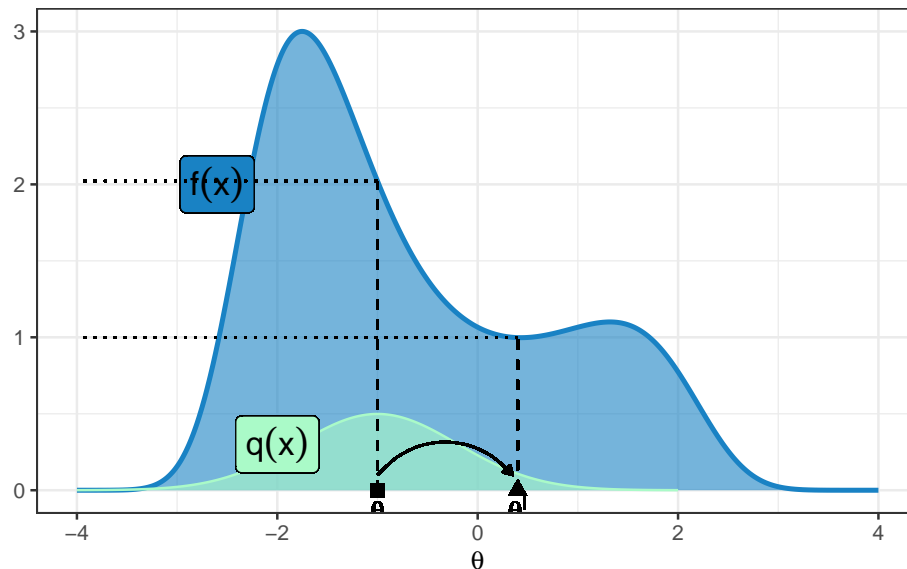


Figura 1: Ejemplificación de la obtención de una muestra por el método de Metropolis-Hastings

A continuación se presentan las funciones usadas para aplicar el método en los casos en los que $f(\mu)$ es unidimensional y bidimensional:

i Algoritmo Metrópolis-Hastings univariado

```
sample_mh <- function(n, d_objetivo, r_propuesta = NULL, d_propuesta = NULL,
                      p_inicial = NULL){

  if (is.null(r_propuesta) | is.null(d_propuesta)) {
    r_propuesta <- function(media) rnorm(n = 1, media, sd = 1)
    d_propuesta <- function(x, media) dnorm(x = x, media, sd = 1)
  }

  stopifnot(n > 0)
  contador <- 0
  muestras <- numeric(n)

  muestras[1] <- p_inicial

  for(i in 2:n) {

    p_actual <- muestras[i-1]
    p_propuesta <- r_propuesta(p_actual)

    q_actual <- d_propuesta(p_actual, p_propuesta)
    q_nuevo <- d_propuesta(p_propuesta, p_actual)

    f_actual <- d_objetivo(p_actual)
    f_nuevo <- d_objetivo(p_propuesta)

    if (f_actual == 0 || q_nuevo == 0) {
      alfa <- 1
    } else {
      alfa <- min(1, (f_nuevo/f_actual)*(q_actual/q_nuevo))
    }

    muestras[i] <- sample(c(p_propuesta, p_actual),
                         size = 1, prob = c(alfa, 1-alfa))

    if(muestras[i] != muestras[i-1]) {
      contador <- contador + 1
    }
  }
  return(list(cadena = data.frame(iteracion = 1:n, x = muestras),
            tasa_aceptacion = contador / n))
}
```

i Algoritmo Metrópolis-Hastings bivariado

```

sample_mh_mv <- function(n, d_objetivo, cov_propuesta = diag(2), p_inicial = numeric(2)) {

  if (length(p_inicial) != 2) {
    stop("El valor p_inicial debe ser bidimensional")
  }
  if ( n <= 0 || n %% 1 != 0) {
    stop("El tamaño de muestra n debe ser entero y mayor que 0")
  }
  if (any((dim(cov_propuesta) != c(2,2)))) {
    stop("La matriz de covariancia debe ser de 2x2")
  }

  contador <- 0

  r_propuesta <- function(media) rmvnorm(n = 1, mean = media, sigma = cov_propuesta)
  d_propuesta <- function(x, media) dmnorm(x = x, mean = media, sigma = cov_propuesta)

  muestras <- matrix(0, nrow = n, ncol = length(p_inicial))
  muestras[1, ] <- p_inicial

  for(i in 2:n) {
    p_actual <- muestras[i-1,]
    p_propuesta <- r_propuesta(p_actual)

    q_actual <- d_propuesta(p_actual, p_propuesta)
    q_nuevo <- d_propuesta(p_propuesta, p_actual)

    f_actual <- d_objetivo(p_actual)
    f_nuevo <- d_objetivo(p_propuesta)

    if (f_actual == 0 || q_nuevo == 0) {
      alfa <- 1
    } else {

      alfa <- min(1, (f_nuevo/f_actual)*(q_actual/q_nuevo))
    }

    aceptar <- rbinom(1, 1, alfa)

    if (aceptar) {
      muestras[i,] <- p_propuesta
    } else {
      muestras[i,] <- p_actual
    }

    if(!any(muestras[i,] != muestras[i-1,])) {
      contador <- contador + 1
    }
  }

  salida <- data.frame(iteracion = 1:n, x5= muestras)
  colnames(salida) <- c("iteracion", paste0("dim_", 1:length(p_inicial)))
}

```

Discusiones

Distribución de Kumaraswamy

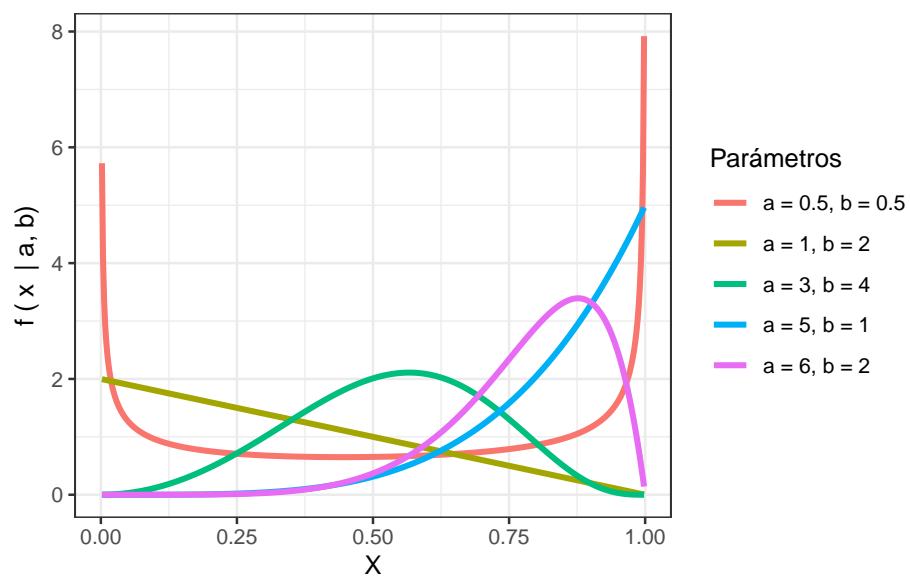


Figura 2: Función de densidad de la distribución de Kumaraswamy

Metropolis-Hastingsen una dimensión

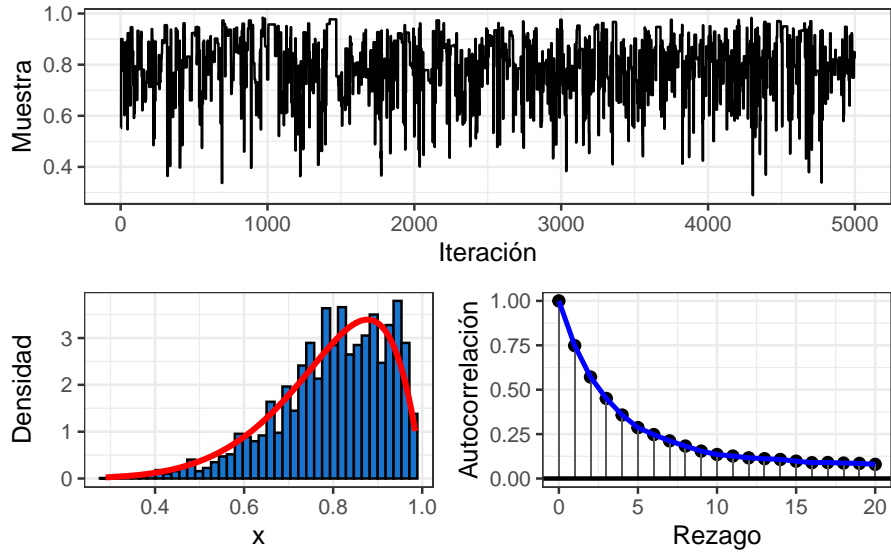


Figura 3: Muestreo por Metropolis-Hastings con $\kappa = 1$

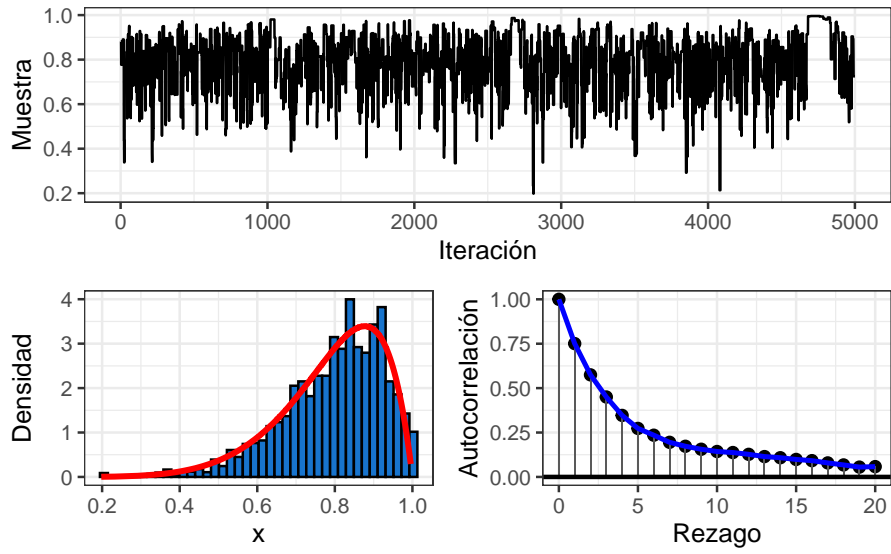


Figura 4: Muestreo por Metropolis-Hastings con $\kappa = 2$

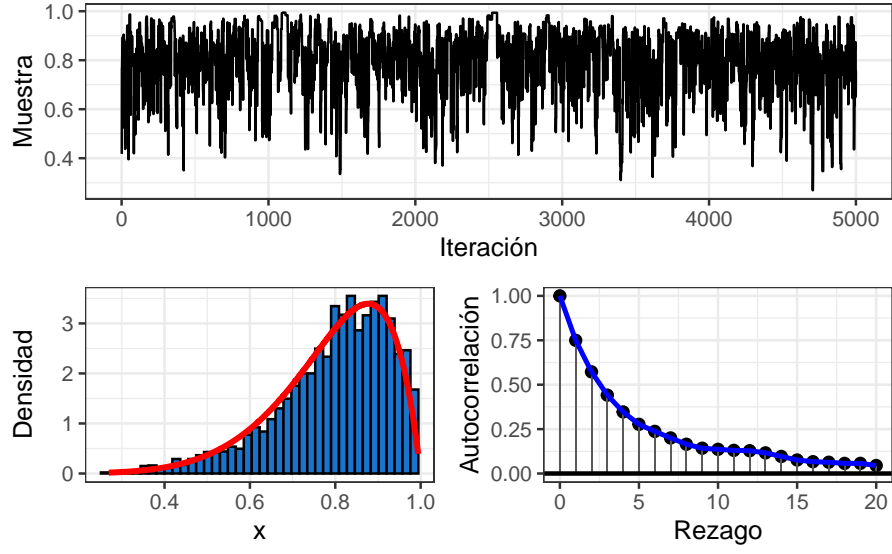


Figura 5: Muestreo por Metropolis-Hastings con $\kappa = 5$

Tabla 1: Número efectivo de muestras para cadenas de 5000 muestras y distintos valores de κ

Cadena	κ	n_{eff}
Cadena 1	1	485
Cadena 2	2	358
Cadena 3	5	503

Tabla 2: Media y quantiles estimadas para las funciones X y $Logit(X)$

$f(x)$	κ	$\hat{E}(x)$	$\hat{q}_{0.05}$	$\hat{q}_{0.95}$
X	1	0.802	0.571	0.959
	2	0.797	0.545	0.971
	5	0.799	0.534	0.969
$Logit(X)$	1	1.615	0.287	3.165
	2	1.636	0.182	3.528
	5	1.632	0.137	3.432

Tabla 3: Media y quantiles para las funciones X y $Logit(X)$

$f(x)$	$E(x)$	$q_{0.05}$	$q_{0.95}$
X	0.791	0.542	0.959
$Logit(X)$	1.547	0.168	3.145

Metropolis-Hastings en dos dimensiones

2. 9 10 11
3. 12 13 14

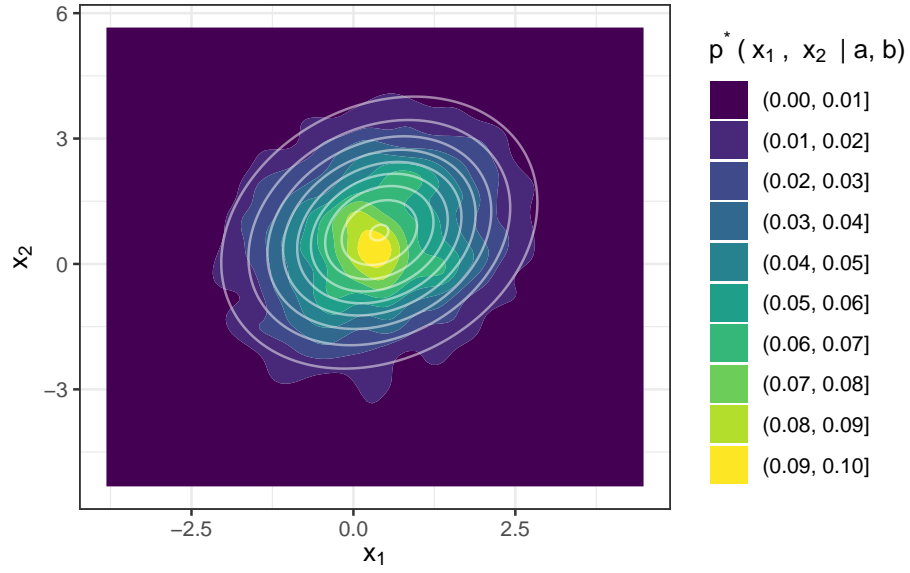


Figura 6: Distribución de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_1 como propuesta

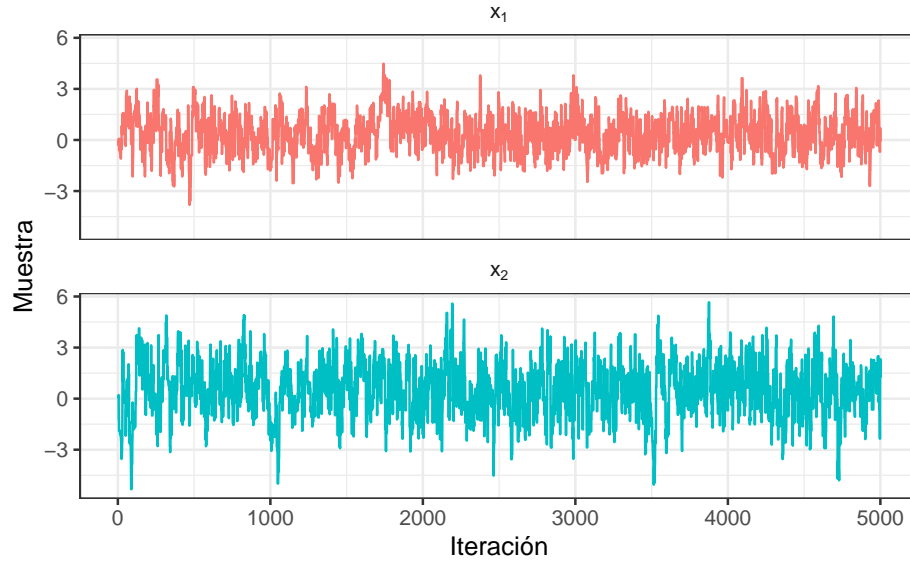


Figura 7: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_1 como propuesta

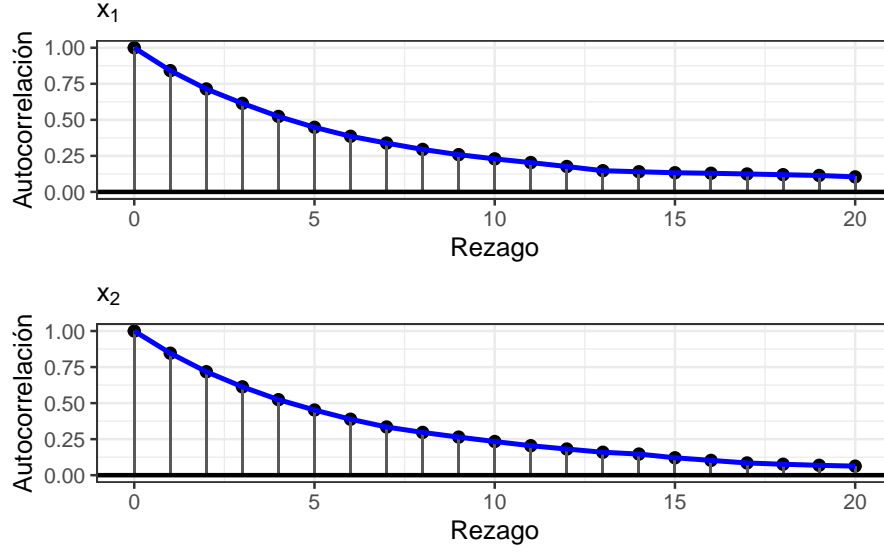


Figura 8: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_1 como propuesta

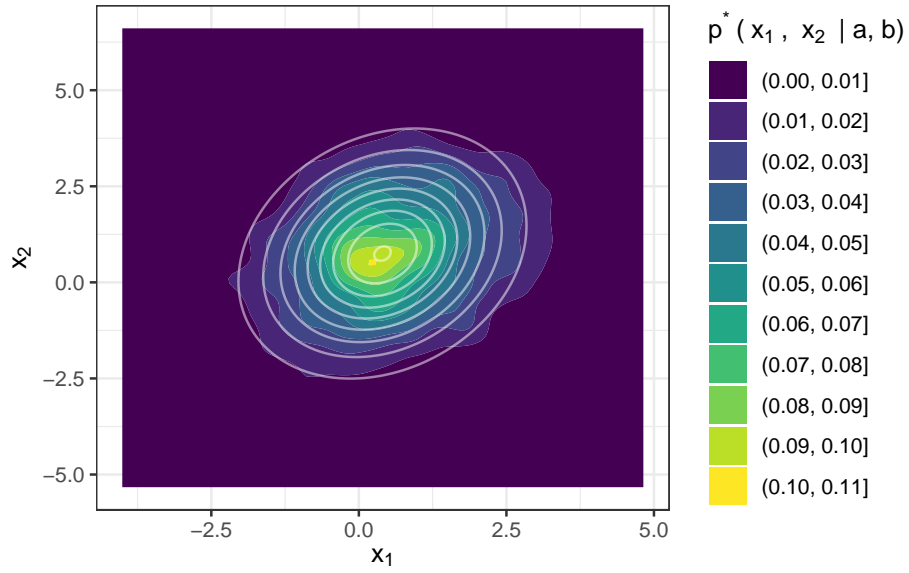


Figura 9: Distribución de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_2 como propuesta

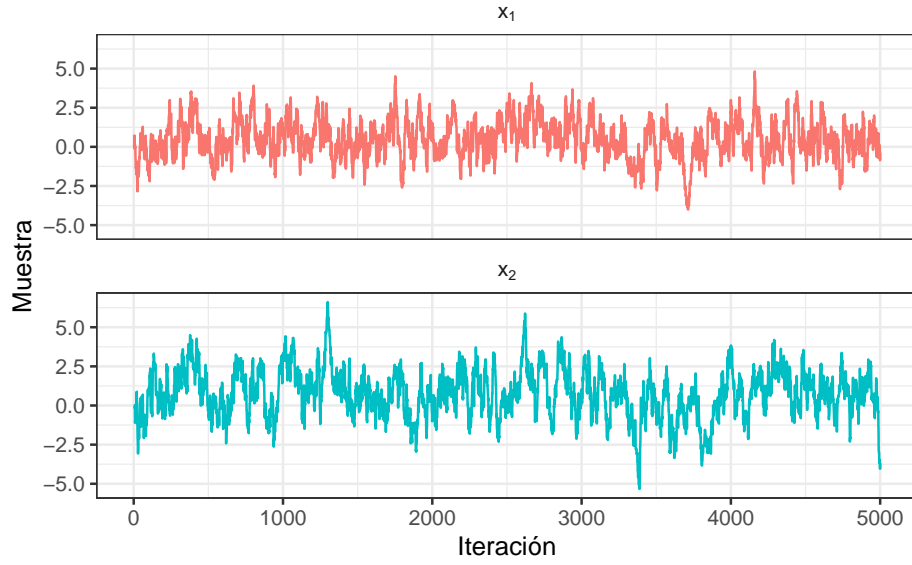


Figura 10: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_2 como propuesta

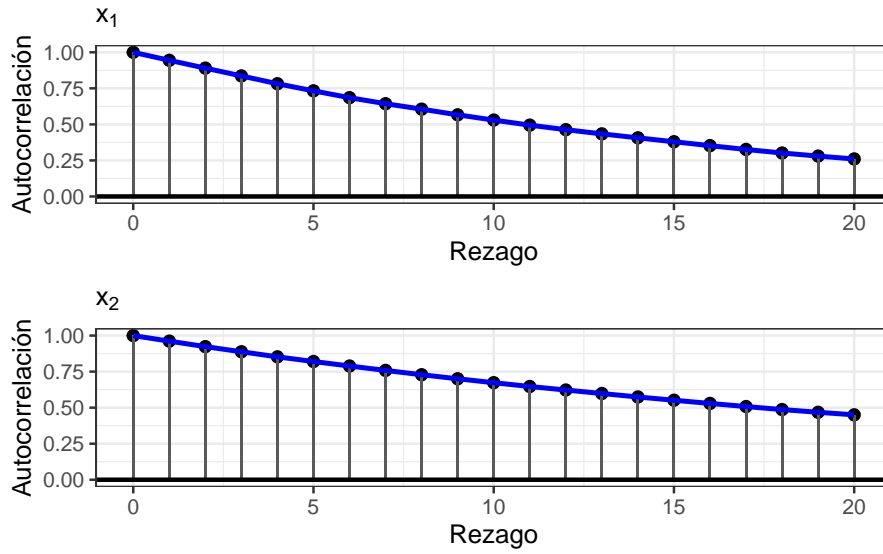


Figura 11: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_2 como propuesta

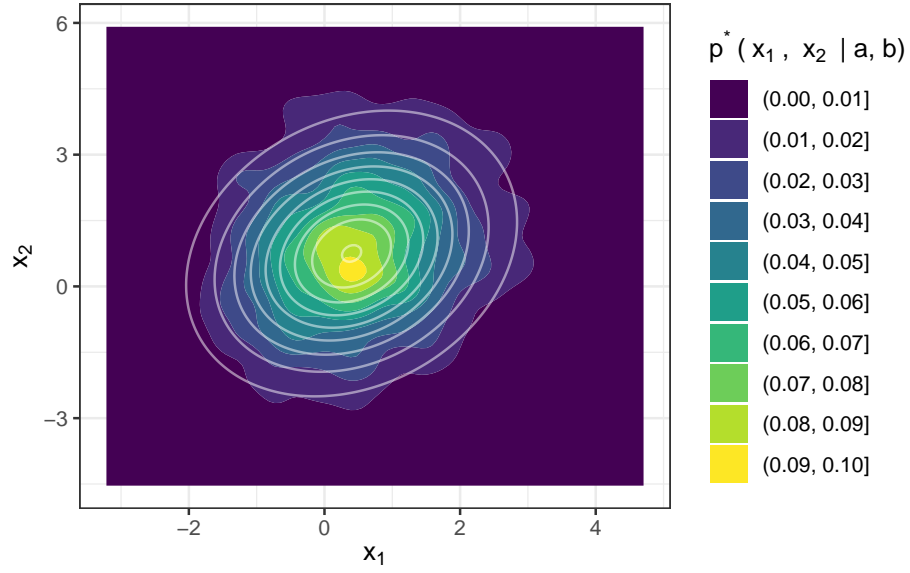


Figura 12: Distribución de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_3 como propuesta

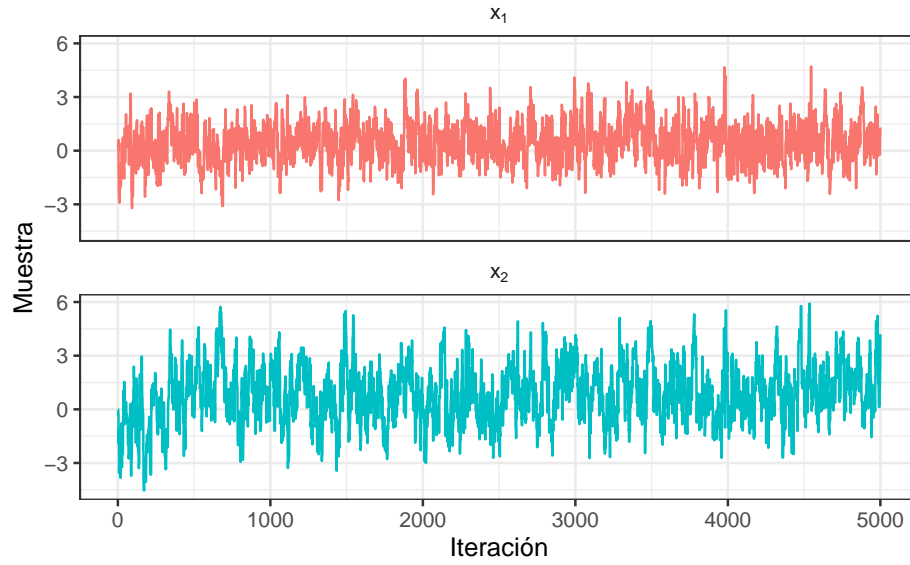


Figura 13: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_3 como propuesta

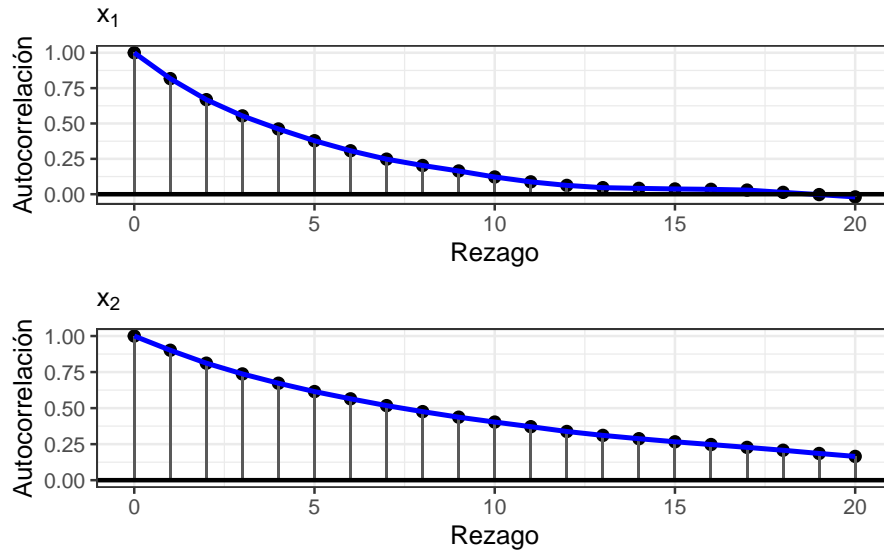


Figura 14: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de una $\mathcal{N}_2(\mu^*, \Sigma^*)$ con Σ_1 como propuesta

Tabla 4: Número efectivo de muestras para cadenas de 5000 muestras y distintos valores de Σ

Cadena	x_1	x_2
Cadena 1	340	377
Cadena 2	171	85
Cadena 3	523	235

Tabla 5: Cálculo de probabilidades con distintos métodos

Método	$P(X_1 > 1, X_2 < 0)$	$P(X_1 > 1, X_2 > 2)$	$P(X_1 > 0.4, X_2 > 0.75)$
M-H	0.0858	0.0654	0.2332
Función de distribución	0.0683	0.0870	0.2857
MC	0.0754	0.0910	0.2808

Función de Rosenbrock

1.	15	16
	17	
2.	18	19
	20	
3.	21	22
	23	

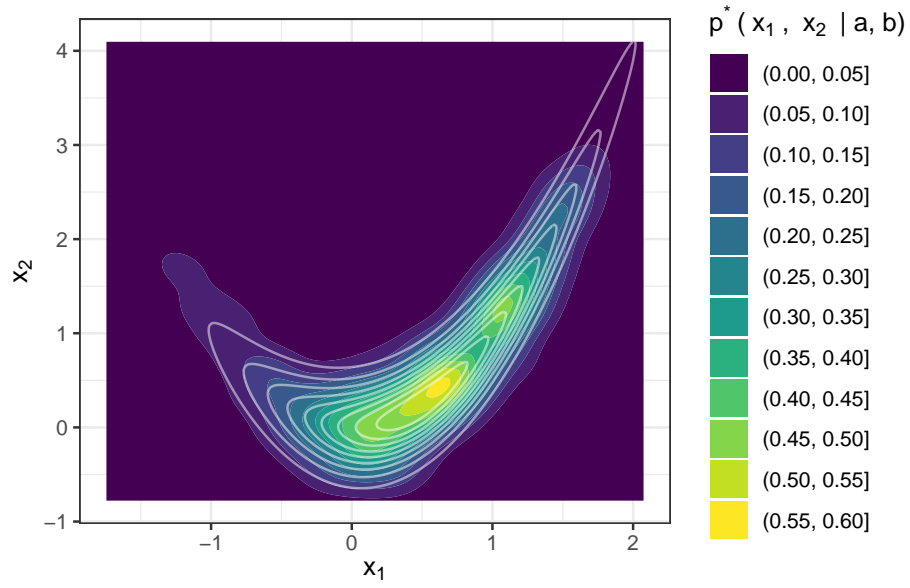


Figura 15: Distribución de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_1 como propuesta

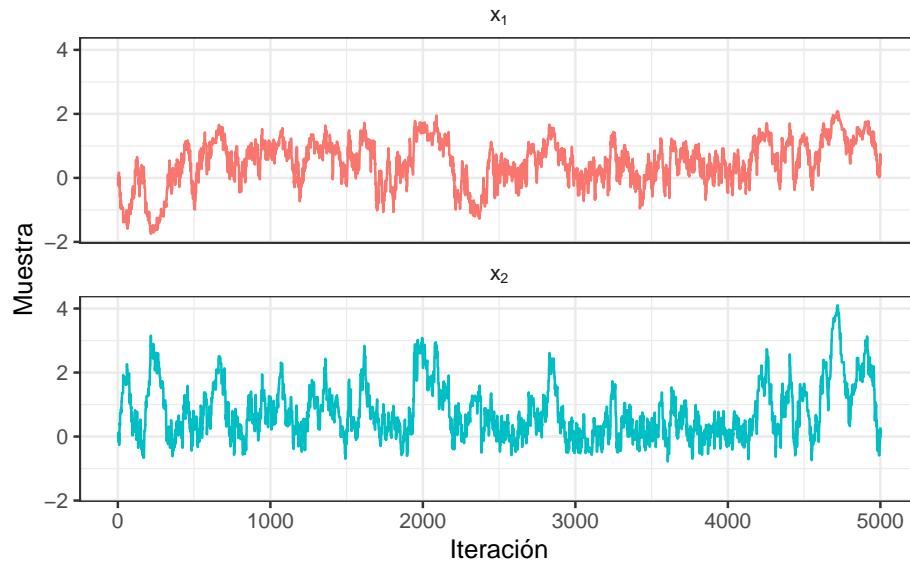


Figura 16: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_1 como propuesta

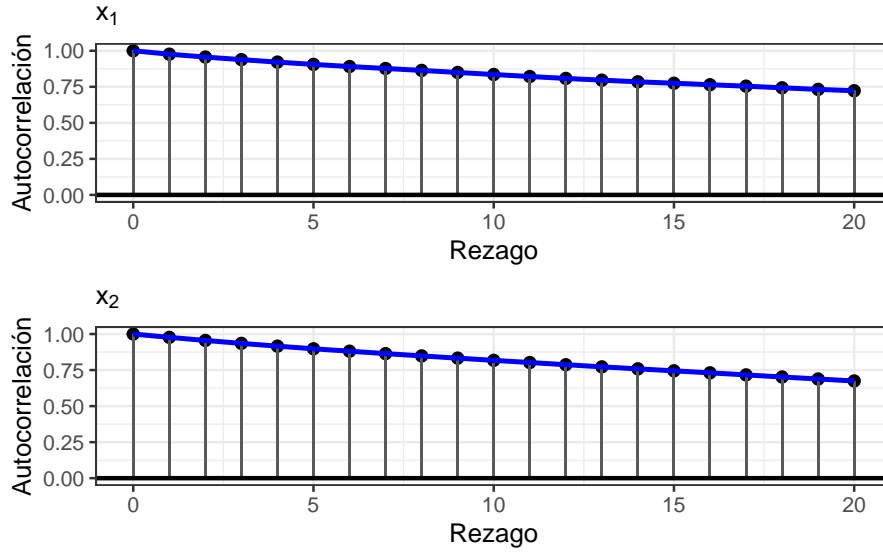


Figura 17: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_1 como propuesta

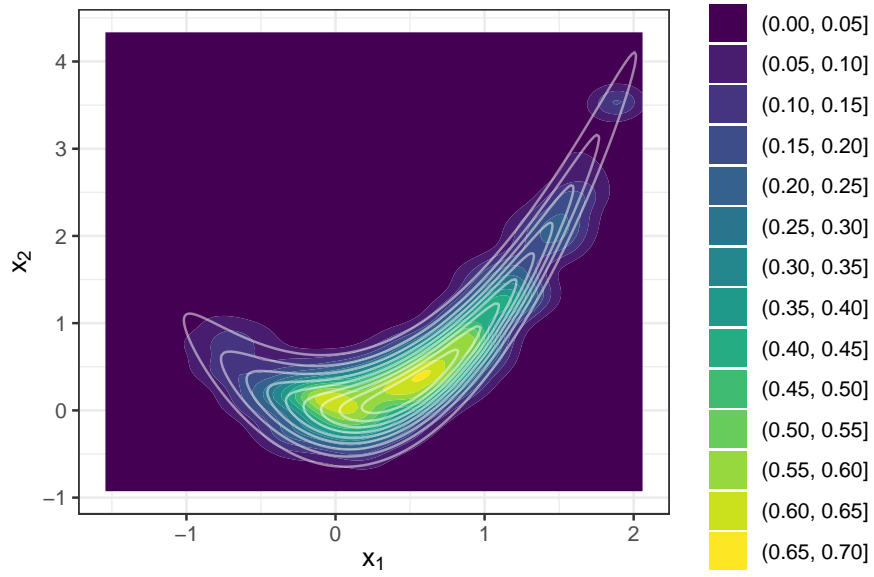


Figura 18: Distribución de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_2 como propuesta

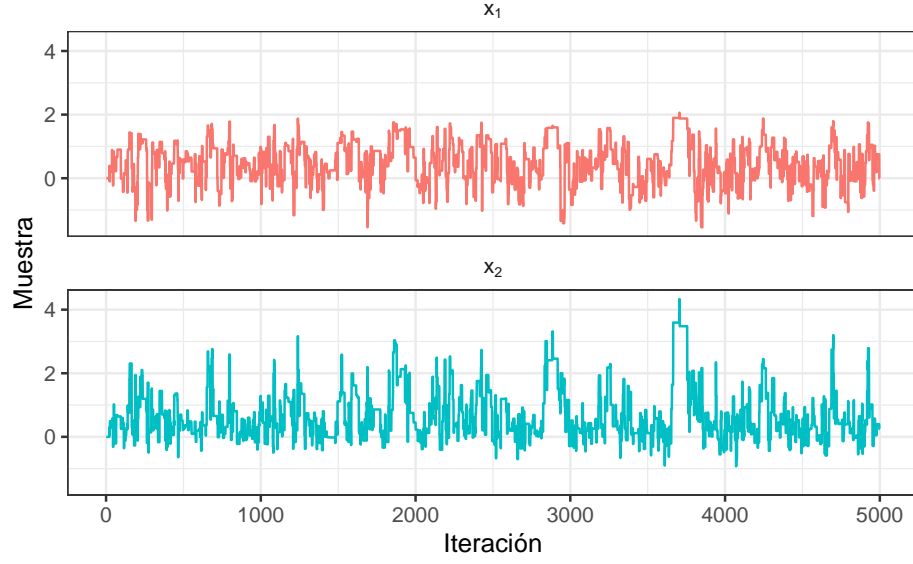


Figura 19: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_2 como propuesta

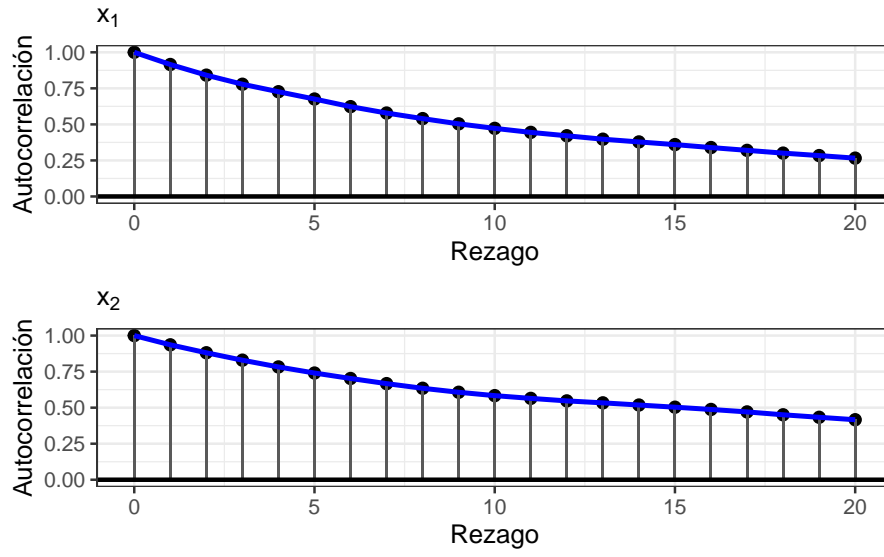


Figura 20: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_1 como propuesta

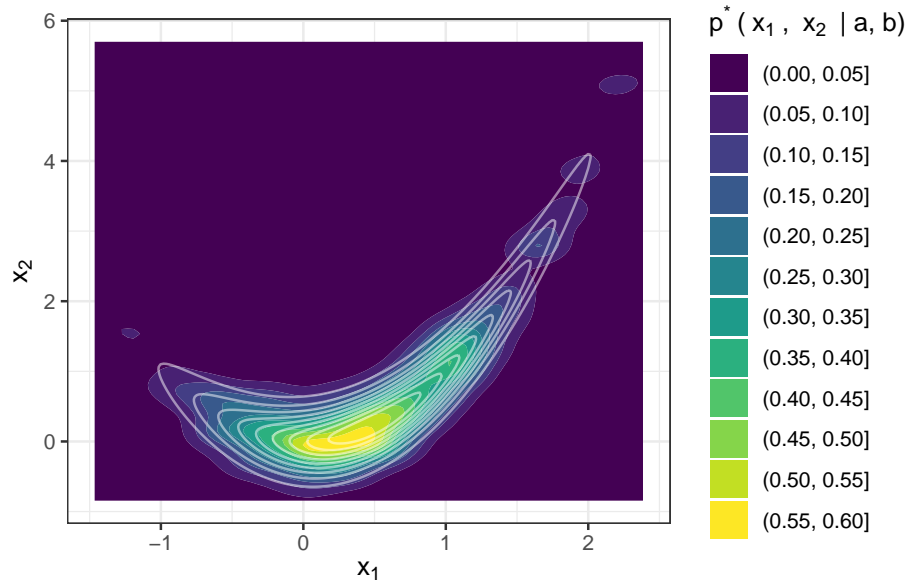


Figura 21: Distribución de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_3 como propuesta

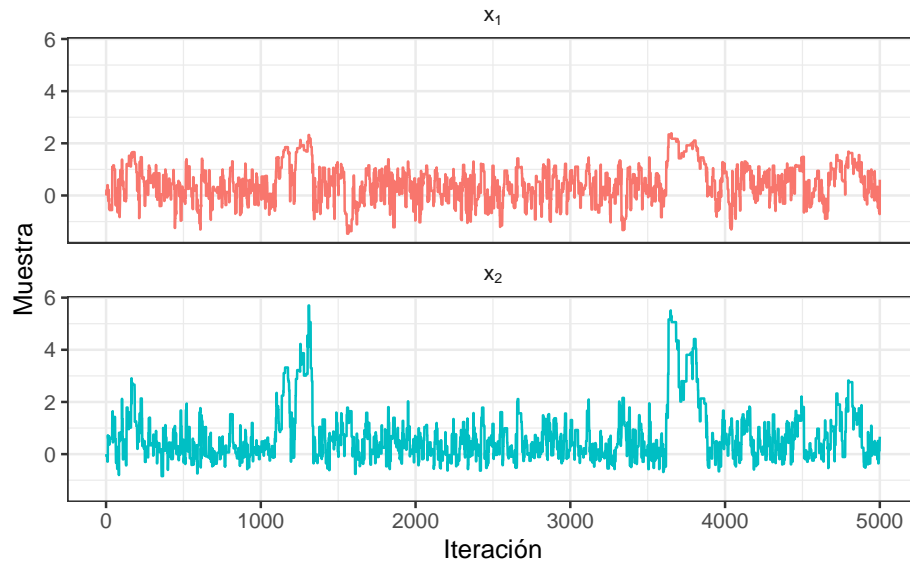


Figura 22: Cadena de Markov de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_3 como propuesta

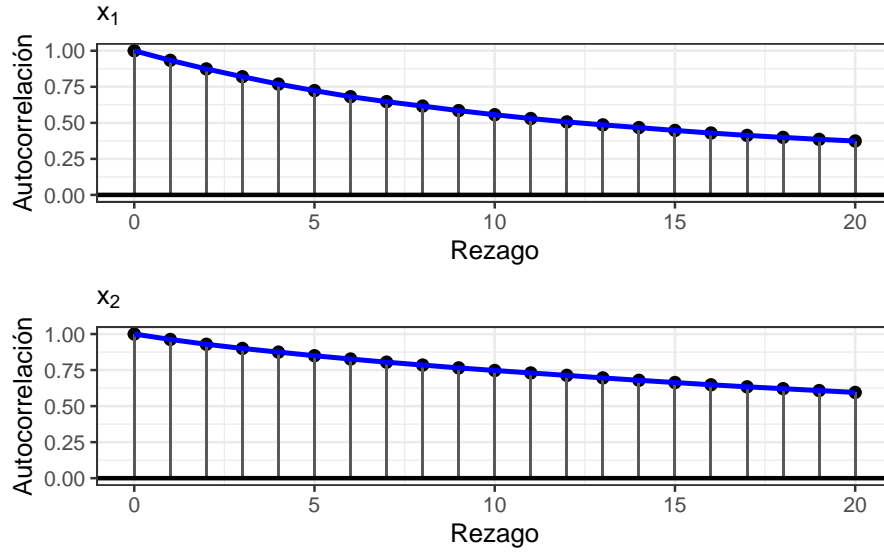


Figura 23: Autocorrelación de las muestras obtenidas por Metropolis-Hastings de p^* con Σ_1 como propuesta

Tabla 6: Número efectivo de muestras para cadenas de 5000 muestras y distintos valores de Σ

Cadena	x_1	x_2
Cadena 1	30	46
Cadena 2	149	94
Cadena 3	67	43

Tabla 7: Cálculo de probabilidades con distintos métodos

Método	$P(A, B)^1$	$P(C, D)^2$	$P(E, F)^3$
M-H	0.3954	0.1602	0.0588
Integración numérica	0.5137	0.2022	0.0838

¹ $A = 0 < X_1 < 1, B = 0 < X_2 < 1$

² $C = -1 < X_1 < 0, D = 0 < X_2 < 1$

³ $E = 1 < X_1 < 2, F = 2 < X_2 < 3$

Conclusiones