UNIVERSIDAD NACIONAL DE ROSARIO

Facultad de Ciencias Económicas y Estadística

"Metropolis-Hastings"

Estadística Bayesiana - Trabajo Práctico Nº2

Alumnas: Agustina Mac Kay, Ailén Salas y Rocío Canteros

Año 2024

Introducción

El algoritmo de Metropolis-Hastings (MH) permite generar muestras (pseudo-)aleatorias a partir de una distribución de probabilidad P que no necesariamente pertence a una familia de distribuciones conocida. El único requisito es que se pueda evaluar la función de densidad (o de masa de probabilidad) $p^*(\theta)$ en cualquier valor de θ, incluso cuando $p^*(\theta)$ sea impropia (es decir, incluso aunque sea desconocida la constante de normalización que hace que la integral en el soporte de la función sea igual a uno).

Los pasos del algoritmo son:

1. Durante la iteración i, se encuentra en el valor del parámetro $\theta^{(i)}$.
2. En función del valor de parámetro actual $\theta^{(i)} = \theta$, se propone un nuevo valor θ' en función de $q(\theta'|\theta)$.
3. Se decide si se vá a la nueva ubicación $\theta^{(i+1)} = \theta'$ o si se queda $\theta^{(i+1)} = \theta$:

- Se calcula la probabilidad de salto:

$$\alpha_{\theta \to \theta'} = \min \left\{ 1, \frac{f(\theta')}{f(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \right\}$$

- Pasar a θ' con probabilidad $\alpha_{\theta \to \theta'}$:

$$\theta^{(i+1)} = \begin{cases} \theta & con \quad probabilidad \quad \alpha_{\theta \to \theta'} \\ \theta & con \quad probabilidad \quad (1 - \alpha_{\theta \to \theta'}) \end{cases}$$

A continuación, se presenta la función que implementa el algoritmo de Metropolis-Hastings para tomar muestras de una distribución de probabilidad a partir de su función de densidad. Se otorga flexibilidad al algoritmo permitiendo elegir entre un punto de inicio arbitrario o al azar, utilizar distribuciones de propuesta de transición arbitrarias (por defecto, se utiliza una distribución normal estándar), y posibilitando el muestreo de funciones en más de una dimensión.

```r
# Función de Metropolis-Hastings

cant_saltos <- 0  # se inicia en 0 el contador de saltos

sample_mh <- function(d_objetivo, r_propuesta, d_propuesta, p_inicial, n) {
  muestras <- matrix(nrow = n, ncol = length(p_inicial))
  muestras[1, ] <- p_inicial

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

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

    q_actual <- d_propuesta(p_actual, mean = p_nuevo)
    q_nuevo <- d_propuesta(p_nuevo, mean = p_actual)

    alpha <- min(1, (f_nuevo/f_actual)*(q_actual/q_nuevo))
    aceptar <- rbinom(1, 1, alpha)

    if(aceptar) {
      muestras[i,] <- p_nuevo
      cant_saltos <- cant_saltos + 1

    } else {
      muestras[i,] <- p_actual
    }
  }

  if (ncol(muestras) == 1) {
    muestras <- as.vector(muestras)
  }
  return(list(muestras=muestras,cant_saltos=cant_saltos))
}
```

Metropolis-Hastings en 1D

Distribución de Kumaraswamy

La distribución de Kumaraswamy es una distribución de probabilidad continua que se utiliza para modelar variables aleatorias con soporte en el intervalo $(0, 1)$. Si bien gráficamente la forma de su función de densidad puede hacer recordar a la distribución beta, vale mencionar que la distribución de Kumaraswamy resulta en una expresión matemática cuyo cómputo es más sencillo:

$$p(x|a, b) = abx^{a-1}(1 - x^a)^{b-1}$$

con $a, b > 0$

A continuación, se grafica la función de densidad de la distribución de Kumaraswamy para las combinaciones de los parámetros:

Combinación	a	b
1	0.2	0.2
2	0.1	0.7
3	3	3
4	4	9
5	10	5

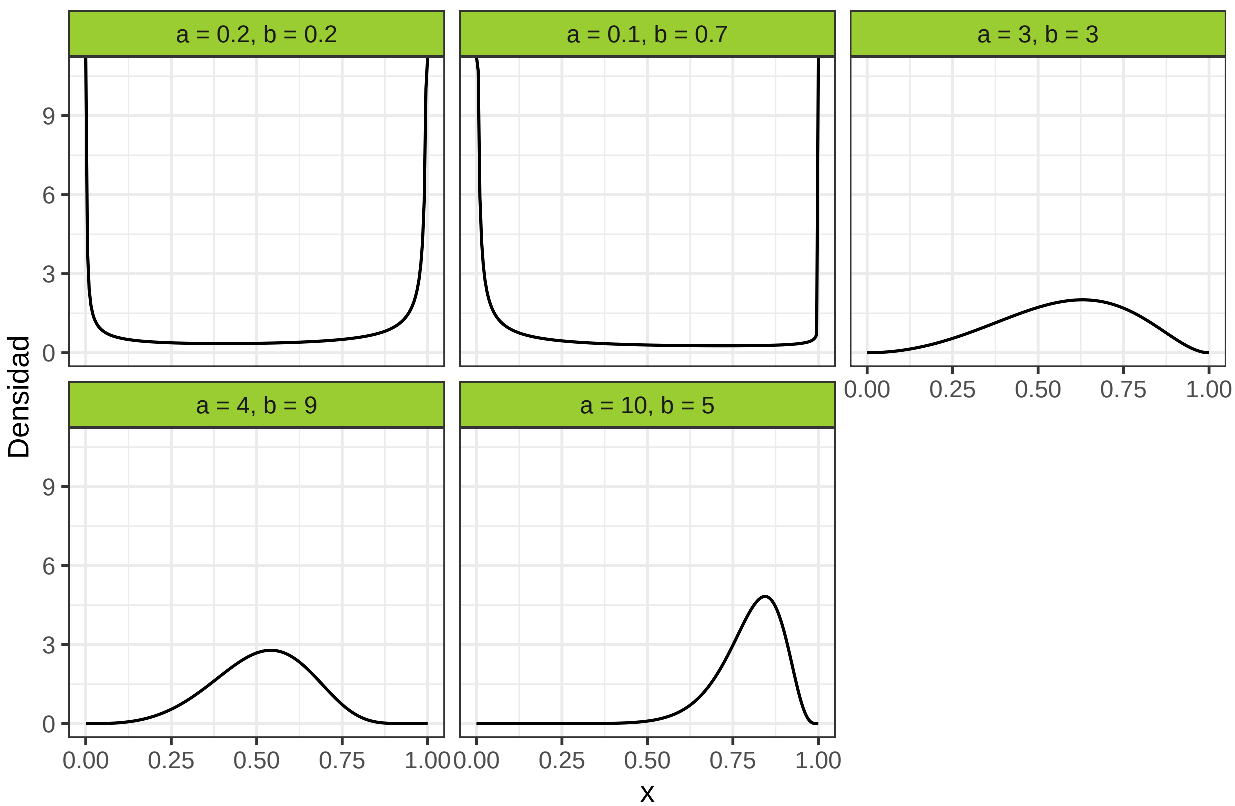


Gráfico 1: Distribución Kumaraswamy con distintas combinaciones de los parámetros a y b

En el gráfico 1 se puede apreciar las distintas formas que toman las curvas de la distribución Kumaraswamy dependiendo de los parámetros a y b que se elijan. El parámetro a controla la asimetría de la curva y el parámetro b controla la curvatura de la gráfica. Se espera que si $a = b$, la curva sea simétrica. Si $a > b$, la curva se inclina hacia la derecha. Si $a < b$, la curva se inclina hacia la izquierda. Se observa que:

- Si los parámetros son iguales y menores a 1, la curva es simétrica y tiene forma de U.
- Si los dos parámetros son menores a 1 y $a < b$, la curva tiene forma de U y es más aplastada del lado derecho.
- Si los parámetros son iguales y mayores a 1, la curva es simétrica y tiene forma de campana.
- Si ambos parámetros son mayores a 1 y $a < b$, la curva tiene forma de campana.
- Si ambos parámetros son mayores a 1 y $a > b$, la curva es asimétrica a la izquierda y tiene forma de campana.

Conocer las distintas formas que puede tomar la curva de la distribución de Kumaraswamy según los parámetros a y b es útil en Estadística Bayesiana porque:

- Facilita la elección de un prior que refleje adecuadamente las creencias previas sobre los parámetros del modelo. Esto es crucial para obtener inferencias precisas y robustas.
- Permite adapar el modelo a diferentes tipos de datos, ya que la distribución puede variar ampliamente de forma dependiendo de los valores de a y b.
- Ayuda a comprender cómo los valores de los parámetros afectan el posterior y, por lo tanto, las conclusiones que se pueden extraer del análisis.

Utilizando la función de MH construída al comienzo, se obtienen 5000 muestras de una distribución Kumaraswamy con parámetros $a = 6$ y $b = 2$. Como distribución propuesta se utiliza una *Beta* con los siguientes grados de concentración:

$$Concentración \quad 4 \quad 10 \quad 20$$

Como punto inicial del algoritmo de MH, se obtiene un valor aleatorio de una distribución $beta(2, 2)$

La tasa de aceptación en el algoritmo de Metropolis-Hastings indica qué tan frecuentemente se aceptan los nuevos θ propuestos, en relación al total de θ propuestos.

	Tabla 1:Tasa de aceptación para cada concentración
Concentracion	Tasa
4	0.45
10	0.63
20	0.77

En la tabla 1 se observa que, a mayor concentración de la distribución propuesta beta, mayor es la tasa de aceptación.

A continuación, una representación gráfica que muestra cómo evolucionan las muestras generadas por el algoritmo a lo largo del tiempo.

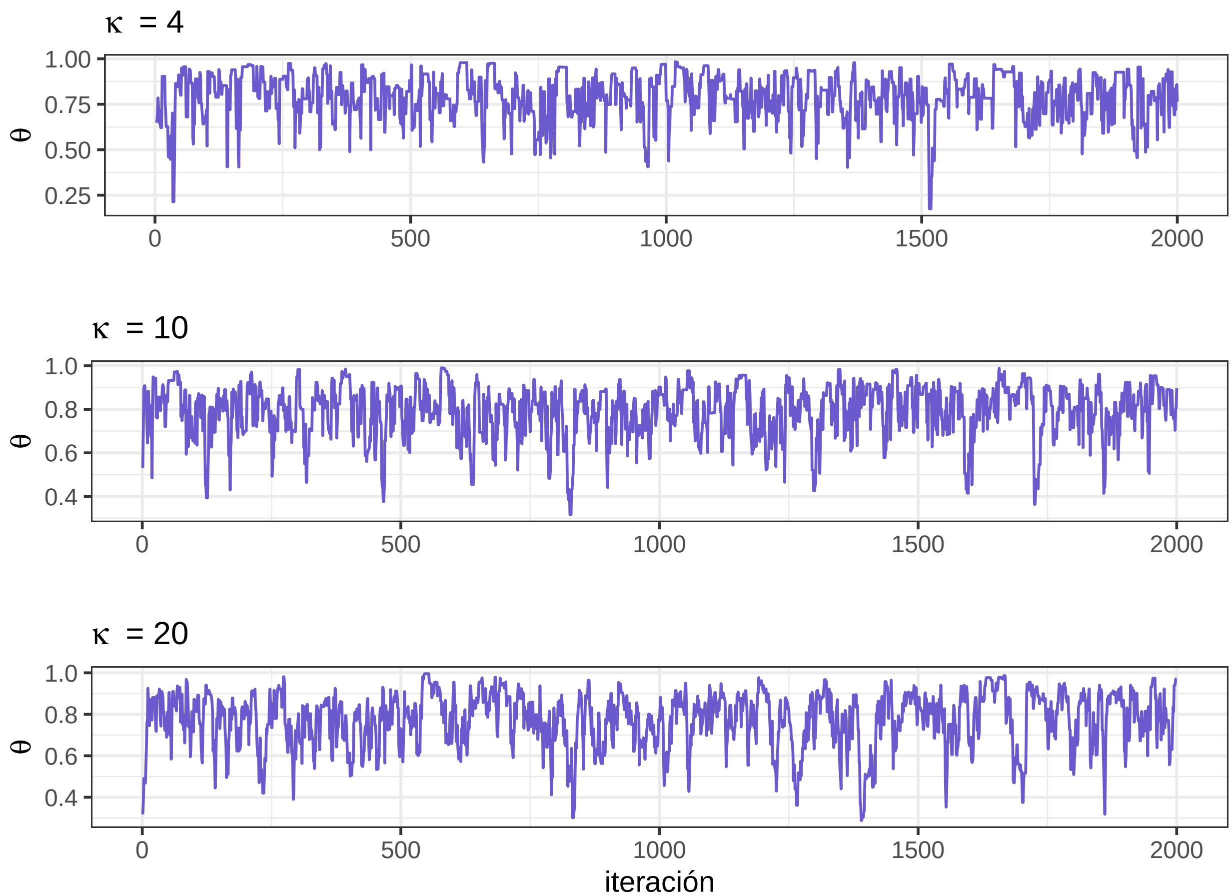


Gráfico 2: Plot trace según concentración

El gráfico 2 muestra que para las 3 concentraciones elegidas, el trace plot NO resulta ser un ruido blanco, pues no oscila alrededor del cero. Además, aunque es dificil apreciar de manera detallada el comportamiento del algoritmo debido a la cantidad de muestras, pareciera ser que las 3 concentraciones presentan algunos estancamientos.

Para evaluar la convergencia de las muestras a la distribución objetivo se presenta:

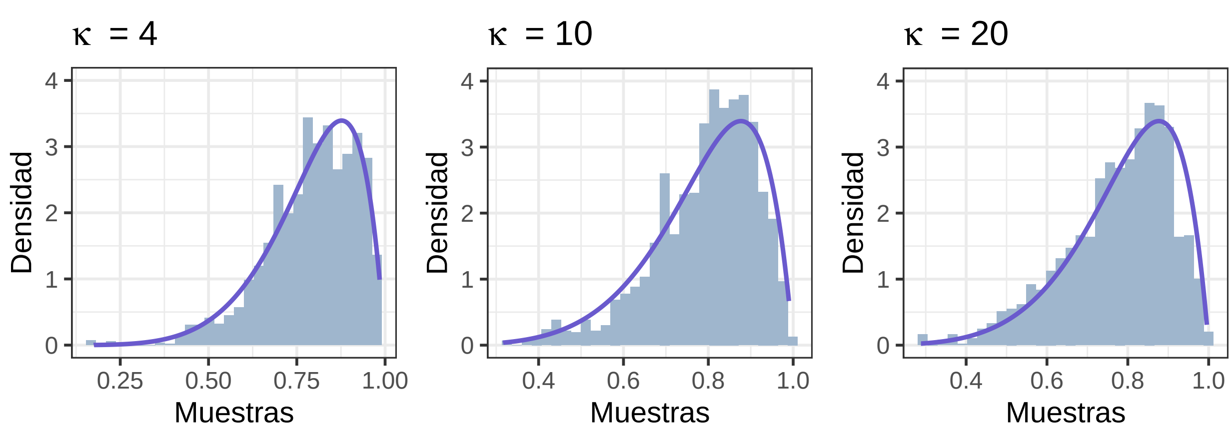


Gráfico 3: Muestras obtenidas y distribución de kumaraswamy según concentración

En el gráfico 3 se observa que las muestras generadas para los 3 valores de concentración se ajustan bastante bien a la distribución objetivo. Las 3 muestras exploran el rango completo de la distribución a posteriori. Sin embargo, pareciera que las concentraciones $k = 4$ y $k = 20$ ajustan mejor.

Se calcula la correlación de la serie para cada valor de lag k consigo misma originando la función de autocorrelación:

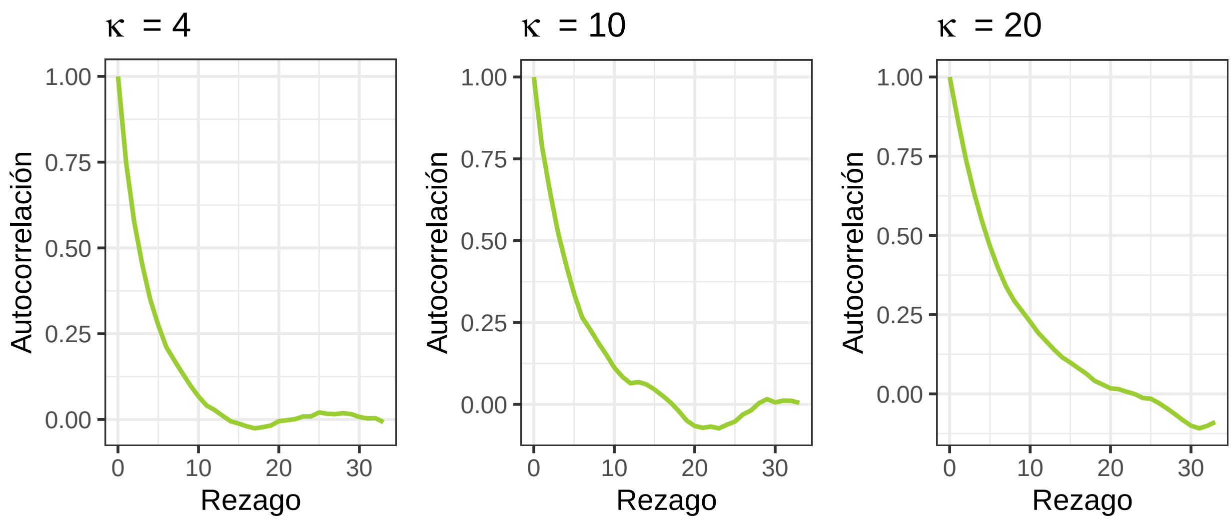


Gráfico 4: Autocorrelación según concentración

Las muestras tienen que ser independientes. La dependencia de valores anteriores tiene que desaparecer rápido. En el gráfico 4 se observa que esto ocurre para los 3 valores de concentración. Este gráfico sugiere que el valor más adecuado para la concentración es $k = 4$, teniendo una correlación de 0.25 en el rezago 5. Cabe destacar que la diferecia con las otras concentraciones no es grande, siendo éstas de 0.25 en los rezagos 6 y 9 aproximadamente.

Para cada una de las cadenas anteriores, se presentan la media de la distribución y los percentiles de X:

		Tabla 2: Media y percentiles de X		
Muestra	Concentración	Media	Percentil del 5%	Percentil del 95%
1	k=4	0.79	0.55	0.96
2	k=10	0.79	0.57	0.95
3	k=20	0.77	0.52	0.95

Para los 3 valores de concentración, se obtienen resultados muy similares para la media (0.79) y para los percentiles 5 y 95 de la distribución, siendo éstos aproximados por 0.55 y 0.95 respectivamente.

Para cada una de las concentraciones, se presentan la media y los percentiles de la distribución de $logit(X)$:

		Tabla 3: Media y percentiles de logit(X)		
Muestra	Concentración	Media	Percentil del 5%	Percentil del 95%
1	k=4	1.54	0.22	3.12
2	k=10	1.53	0.27	3.03
3	k=20	1.42	0.06	2.92

Al hacer uso de todo el eje real mediante el $logit(x)$, se observa que la media y los percentiles para $k = 20$ difieren un poco de estos valores para las otras 2 concentraciones.

Conclusión

Al analizar las distintas muestras de distribución Kumaraswamy de parámetros $a = 6$ y $b = 2$ con propuesta beta de distintas concentraciones, no se encontraron grandes diferencias entre éstas. No es posible decidir con certeza cuál de los valores de concentración es mejor, debido a que las diferencias obtenidas han sido muy pequeñas. Sin embargo, si habría que sugerir un valor de concentración, se sugiere el 4. Esto es debido a que tiene una tasa de aceptación moderada (0.45), el algoritmo no se estanca demasiado en los mismos valores de θ, la muestras explora el rango completo de la distribución a posteriori y se ajusta bien a la curva. Además, para este valor de concentración, la autocorrelación decrece rápidamente y es menor a 0.25 a partir del rezago 5. Esto último no ocurre para las otras dos concentraciones.

Metropolis-Hastings en 2D

La verdadera utilidad del algoritmo de Metropolis-Hastings se aprecia cuando se obtienen muestras de distribuciones en más de una dimensión, incluso cuando no se conoce la constante de normalización. En esta sección se trabaja con ejemplos que permitirán advertir las limitaciones del algoritmo y motivarán la búsqueda de mejores alternativas.

Normal multivariada

La distribución normal multivariada es la generalización de la distribución normal univariada a múltiples dimensiones (o mejor dicho, el caso en una dimensión es un caso particular de la distribución en múltiples dimensiones). La función de densidad de la distribución normal en k dimensiones es:

$$p(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right)$$

donde $\boldsymbol{\mu}$ es el vector de medias y $\boldsymbol{\Sigma}$ la matriz de covarianza.

Utilizando la función de Metropolis-Hastings descrita al principio del informe, se obtienen muestras de una distribución normal bivariada con media $\boldsymbol{\mu}^*$ y matriz de covarianza $\boldsymbol{\Sigma}^*$, donde:

$$\boldsymbol{\mu}^* = \begin{bmatrix} 0.4 \\ 0.75 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^* = \begin{bmatrix} 1.35 & 0.4 \\ 0.4 & 2.4 \end{bmatrix}$$

Con el objetivo de analizar cuál es la matriz de covarianza para la distribución propuesta más óptima, se prueba con las siguientes:

$$\boldsymbol{\Sigma}^{\mathbf{1}} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 31.35 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{\mathbf{2}} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{\mathbf{3}} = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{\mathbf{4}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{\mathbf{5}} = \begin{bmatrix} 5 & 0 \\ 0 & 6 \end{bmatrix}$$

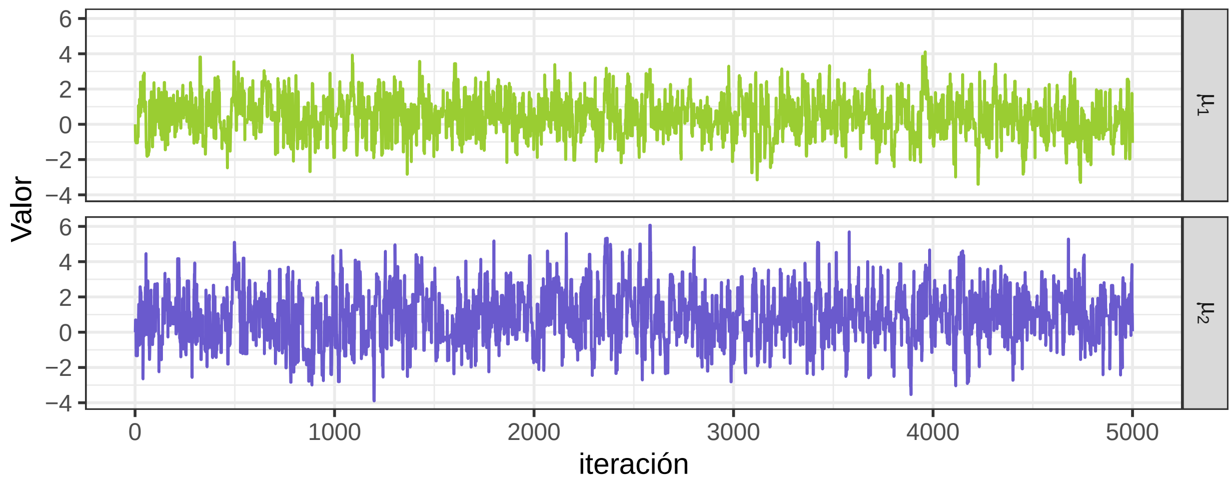


Gráfico 5: Trace plot de la matriz 1

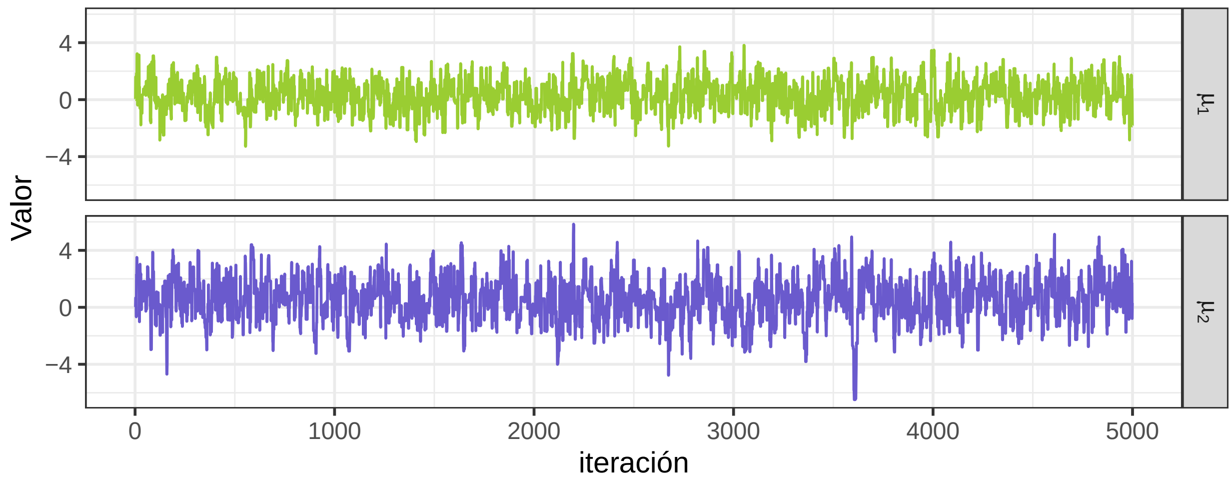


Gráfico 6: Trace plot de la matriz 2

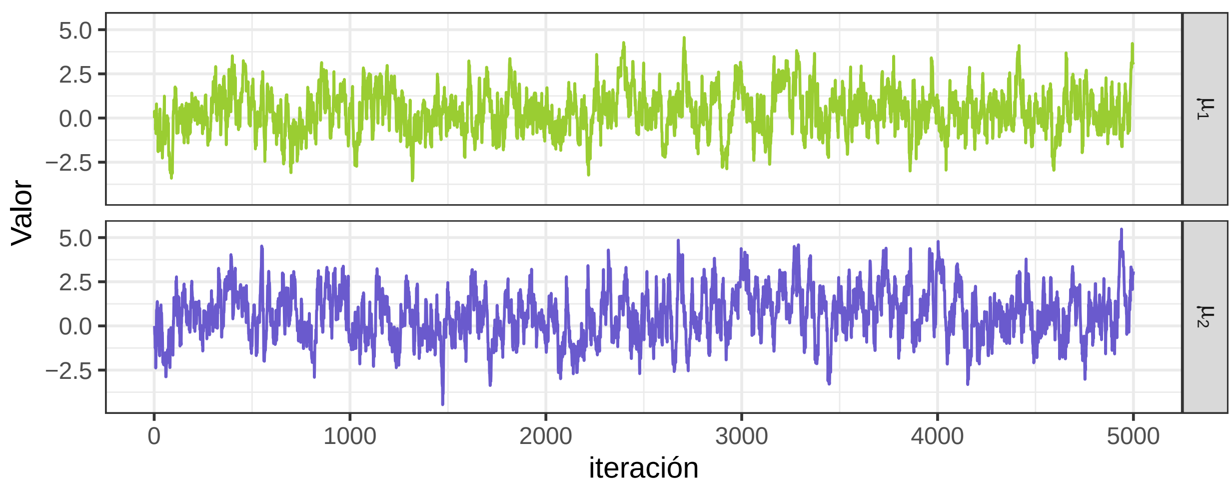


Gráfico 7: Trace plot de la matriz 3

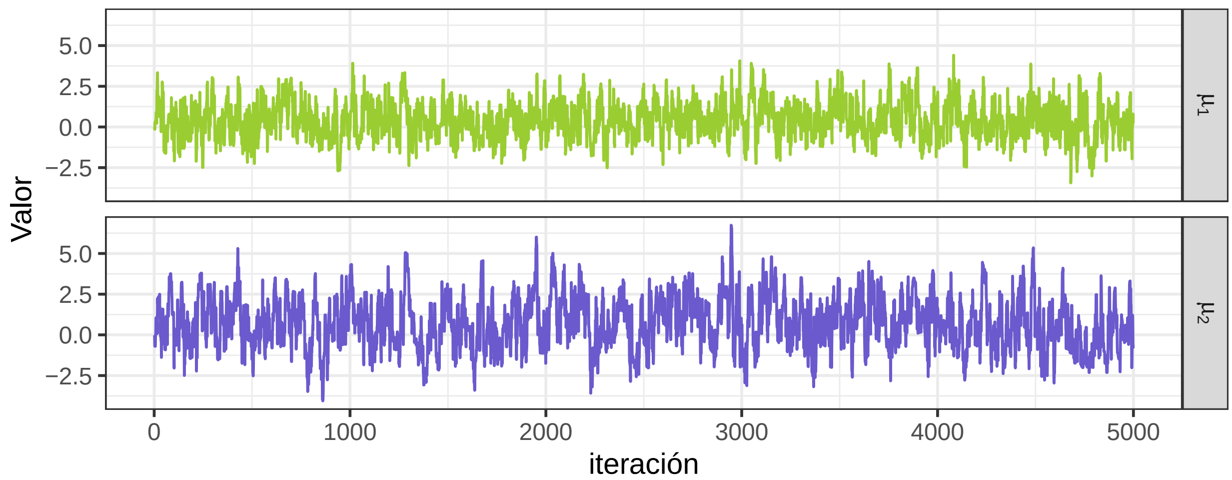


Gráfico 8: Trace plot de la matriz 4

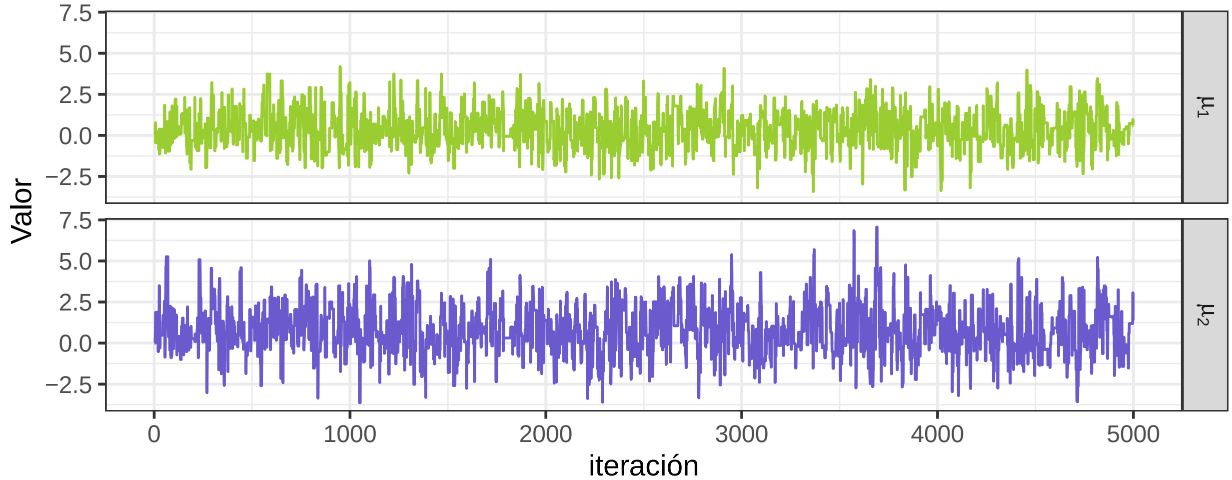


Gráfico 9: Trace plot de la matriz 5

En los gráficos 5 a 9 se observa que, para los dos parámetros, las iteraciones oscilan alrededor del cero. Puesto que la cantidad de observaciones es elevada, no se aprecian con claridad posibles estancamientos de los valores de los parámetros muestreados. Sin embargo, se puede observar que en los gráficos 5 y 9 se tiene un rango de valores de los parámetros más amplio, principalmente de μ_2. Esto se debe a que las matrices utilizadas en estos casos son las que asignan mayores variacias para $\boldsymbol{\mu}$. Pareciera que con la matriz 3 es con la cual más se estanca el método.

Tabla 4: Números efectivos de muestras para cada matriz y cada parámetro

Matriz	mu_1	mu_2
1	603.09	570.95
2	613.45	531.96
3	205.30	141.53
4	455.25	278.06
5	808.04	608.99

En la tabla 4 se observa que la matriz de variancias y covarianzas de la distribución propuesta que devuelve un valor más óptimo de muestras efectivas es la matriz 5, y por eso se continúa el análisis con ella.

Se concluye con la tabla que 5000 muestras correlacionadas con la mejor matriz $\boldsymbol{\Sigma}$, equivalen a 808 y 608 muestras independientes para μ_1 y μ_2, respectivamente. Este número de muestras efectivas es muy bajo, lo cual representa un aspecto negativo del método de MH.

A partir de las muestras obtenidas con la matriz 5, se estiman las siguientes probabilidades:

i. $Pr(X_1 > 1, X_2 > 0)$

ii. $Pr(X_1 > 1, X_2 > 2)$

iii. $Pr(X_1 > 0.4, X_2 > 0.75)$

Luego, mediante la función de la distribución normal bivariada se obtienen las probabilidades reales con el objetivo de compararlas y ver la calidad de la muestra obtenida.

	Tabla 5: Probabilidades estimadas y reales	
Probabilidad	Estimada	Real
i	0.07	0.07
ii	0.08	0.09
iii	0.29	0.29

Según lo observado en la tabla 5, se podría considerar que se tiene una buena muestra de una normal bivariada con media $\boldsymbol{\mu}^*$ y matriz de covarianza $\mathbf{\Sigma}$ a través del algoritmo de Metrópolis-Hastings en 2D, con la matriz de covarianza para la distribución propuesta $\mathbf{\Sigma}^5$.

Conclusión

Utilizando el algoritmo de Metrópolis-Hastings en 2D para generar muestras de una distribución normal bivariada, se observa que la elección de la matriz de varianzas y covarianzas de la distribución propuesta ($\mathbf{\Sigma^5}$) influye en la eficiencia del muestreo. Una mala elección de dicha matriz puede conducir a una exploración ineficiente del espacio paramétrico, y por consecuencia, a la obtención de muestras poco informativas.

Función de Rosenbrock

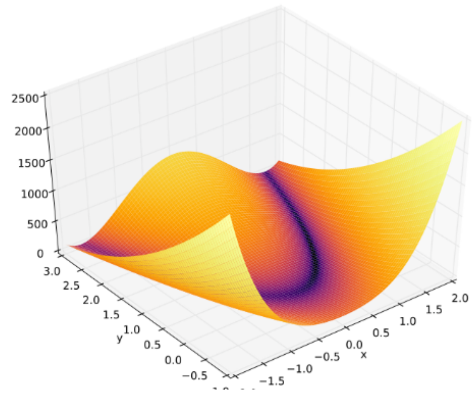
La función de Rosenbrock, a veces llamada el "valle de Rosenbrock", y comunmente conocida como la "banana de Rosenbrock", es una función matemática utilizada frecuentemente como un problema de optimización y prueba para algoritmos de optimización numérica.

La función está definida por:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

y cuenta con un mínimo global en $(x, y) = (a, a^2)$, que satisface $f(a, a^2) = 0$.

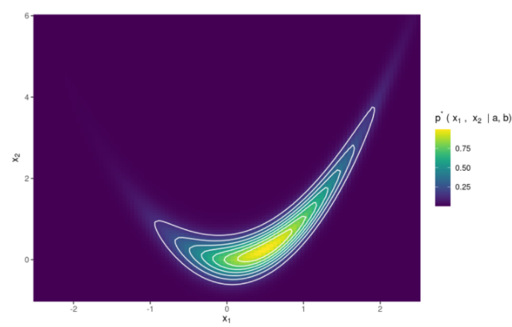
Debido a su forma peculiar, la función de Rosenbrock presenta desafíos particulares para los algoritmos de optimización, ya que tiene un valle largo y estrecho en el que la convergencia puede ser lenta.



Función de Rosenbrock

Esta forma de banana popularizada por Rosenbrock es también muy conocida en el campo de la estadística bayesiana, ya que en ciertos escenarios, la densidad del posterior toma una forma que definitivamente se asemeja a la banana de Rosenbrock. Un ejemplo de este fenómeno es la función p^*:

$$p^*(x_1, x_2|a, b) = exp(-[(a - x_1)^2 + b(x_2 - x_1^2)^2])$$



Función de densidad de la que se desean obtener muestras con $a = 0.5$ y $b = 5$

A continuación se obtienen muestras de la distribución a posteriori determinada por p^* con $a = 0.5$ y $b = 5$ utilizando la función del algoritmo de Metrópolis-Hastings. Para la distribución de propuesta se utilizan las siguientes matrices de variancias y covarianzas:

$$\mathbf{\Sigma^1} = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}$$

$$\mathbf{\Sigma^2} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\mathbf{\Sigma^3} = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$$

	Tabla 6: Tasa de aceptación según matriz de variancias y covariancias
Matriz	Tasa
1	0.41
2	0.09
3	0.05

Se comparan las trayectorias seguidas por las cadenas en el proceso de muestreo para las matrices $\mathbf{\Sigma}^1$ y $\mathbf{\Sigma}^3$:

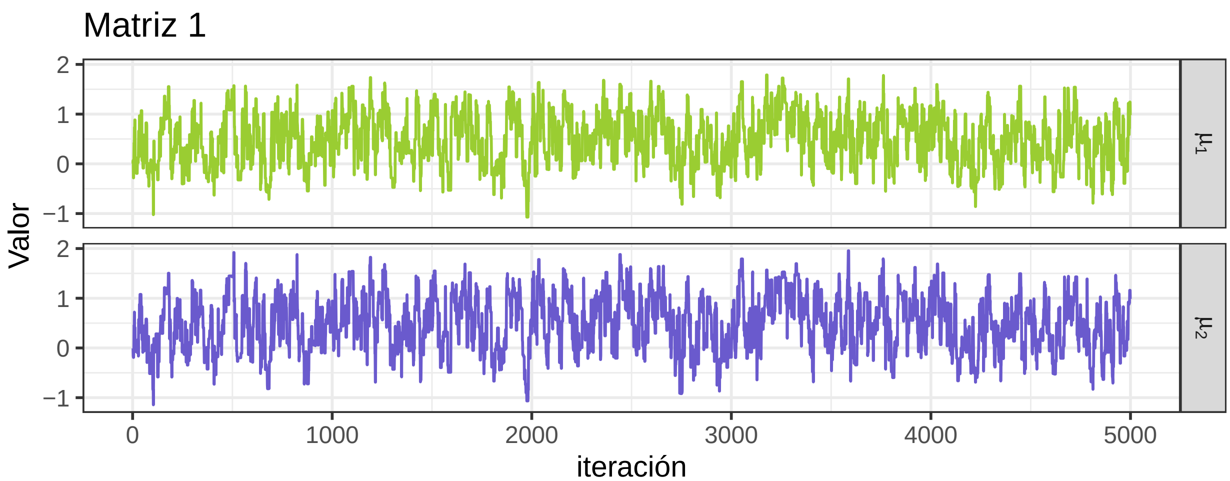


Gráfico 10: Plot trace de la matriz 1

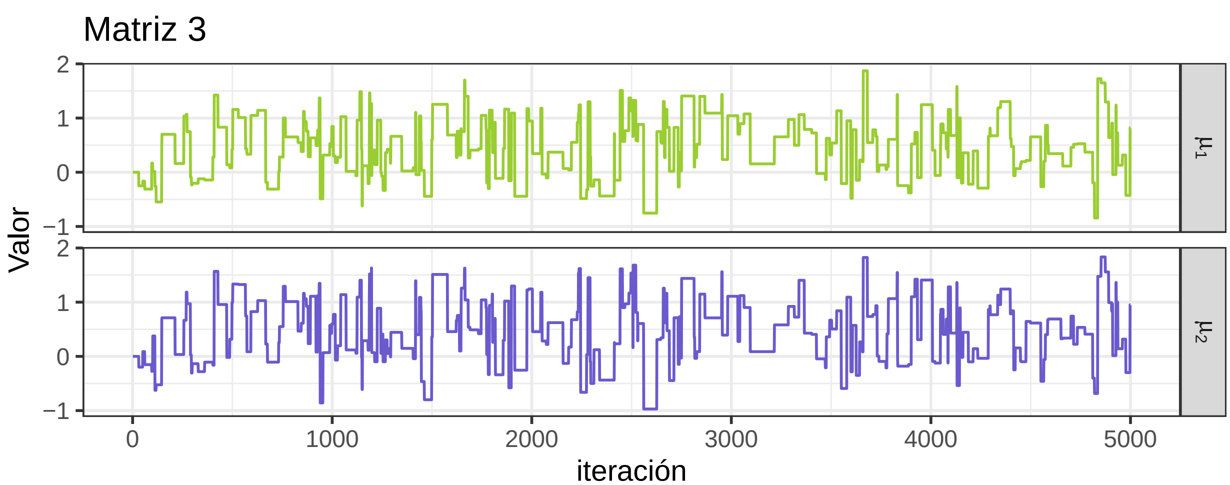


Gráfico 11: Plot trace de la matriz 3

En los gráficos 10 y 11 se observa que usando la matriz $\mathbf{\Sigma}^3$ el algoritmo se estanca bastante más que al utilizar la matriz $\mathbf{\Sigma}^1$.

En la tabla 6 se aprecia de manera clara que el algoritmo tiene una tasa de aceptación muy baja para las matrices de covarianza $\mathbf{\Sigma}^2$ y $\mathbf{\Sigma}^3$. Para la matriz $\mathbf{\Sigma}^1$ la probabilidad de aceptación resulta ser mucho más alta.

A continuación, las funciones de autocorrelación para cada parámetro y cada matriz $\mathbf{\Sigma}$ de la distribución propuesta:

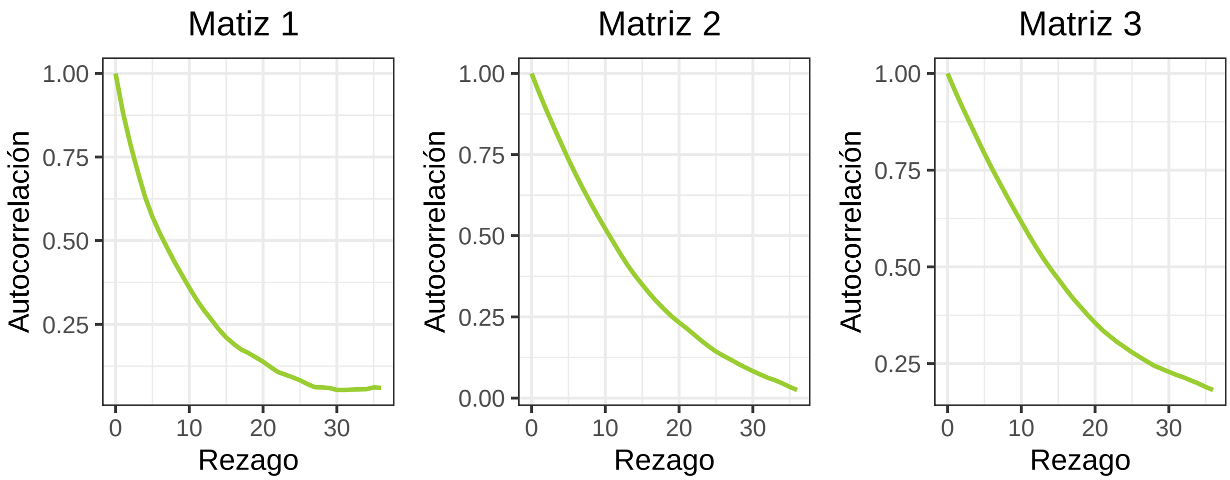


Gráfico 12: Autocorrelación de las muestras del parámetro a

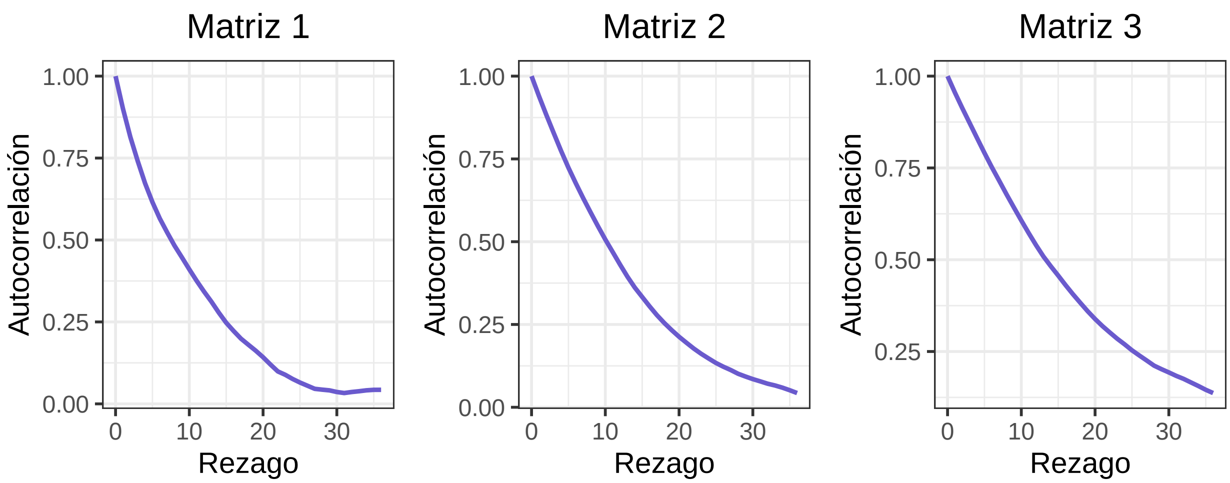


Gráfico 13: Autocorrelación de las muestras del parámetro b

En los gráficos 12 y 13 se puede ver que la dependencia de las muestras a los valores anteriores desciende más rápido al utilizar la matriz $\mathbf{\Sigma}^1$.

A partir de estos resultados, se elige utilizar el conjunto de muestras correspondiente a la matriz $\mathbf{\Sigma}^1$ con el objetivo de obtener las estimaciones de las probabilidades:

i. $Pr(0 < X_1 < 1, 0 < X_2 < 1)$

ii. $Pr(-1 < X_1 < 0, 0 < X_2 < 1)$

iii. $Pr(1 < X_1 < 2, 2 < X_2 < 3)$,

para luego compararlas con las probabilidades estimadas obtenidas a través de la integración por Monte Carlo y a través de la función *integral2* del paquete de R *pracma*.

```
## [1] NA
```

Tabla 7: Probabilidades estimadas por distintos métodos

Probabilidad	Muestra	Monte Carlo	Función 'integral2'
i	0.48	0.61	0.61
ii	0.04	0.05	0.05
iii	0.00	0.00	0.00

En la tabla 7 se puede ver que las probabilidades estimadas son similares con los 3 métodos, pero Metrópolis-Hastings es el menos acertado de los 3.

Conclusión

Al emplear el algoritmo de Metrópolis-Hastings para generar muestras de la función de Rosenbrock, la matriz de variancias y covarianzas $\mathbf{\Sigma}^1$ produjo los mejores resultados, en comparación con las otras 2 opciones. Esto se basó en los criterios del análisis del trace plot, que mostró que el algoritmo no se estanca en valores de θ, y tuvo una tasa de aceptación favorable en comparación con $\mathbf{\Sigma}^2$ y $\mathbf{\Sigma}^3$. Además, la autocorrelación de la serie disminuyó rápidamente.

En cuanto al cálculo de probabilidades, los resultados demostraron que, si bien MH es un algoritmo sencillo para obtener muestras de diversas distribuciones, puede no ser el método más efectivo para el cálculo de integrales. Hay que prestar especial atención a la definición de la distribución de salto propuesta y tomar un número grande de muestras