

# Parcial 1

Juan M Karawcki

2025-09-29

## Ejercicio 1 (Algoritmo de Metropolis-Hastings: cambio del modelo de propuesta) (2.5 puntos)

Se considera la función de R `one_mh_iteration()`, presentada en la sección 7.2 “*The Metropolis-Hastings algorithm*” del libro *Bayes Rules!*.

Crear una nueva función de R `one_mh_iteration_normal()`, la cual utiliza un modelo de propuesta Normal simétrica, centrada en el valor actual de la cadena  $\mu$ , con desviación estándar  $s$ :

$$\mu' \mid \mu \sim N(\mu, s^2)$$

Comenzando desde un valor inicial de  $\mu = 3$  y con `set.seed(1)`, ejecutar la nueva función `one_mh_iteration_normal` bajo cada una de las siguientes configuraciones:

Comentar sobre los valores obtenidos de *proposal*, *alpha* y *next\_stop*.

```
one_mh_iteration_normal <- function(sigma, current){  
  proposal <- rnorm(1, mean = current, sd = (sigma**2))  
  proposal_plaus <- dnorm(proposal, 0, 1) * dnorm(6.25, proposal, 0.75)  
  current_plaus <- dnorm(current, 0, 1) * dnorm(6.25, current, 0.75)  
  alpha <- min(1, proposal_plaus / current_plaus)  
  next_stop <- sample(c(proposal, current),  
                      size = 1, prob = c(alpha, 1-alpha))  
  return(data.frame(proposal, alpha, next_stop))  
}
```

1. `one_mh_iteration_normal(s = 0.01, current = 3)`

Primero seteamos la semilla en 1 y aplicamos la función `one_mh_iteration_normal()` al valor inicial  $\mu = 3$  con un desvío estándar  $\sigma = 0.01$  para la propuesta normal.

proposal	alpha	next_stop
2.999937	0.999826	2.999937

En este caso, el algoritmo partió de  $\mu = 3$  y, dado que el desvío estándar de la propuesta era muy pequeño ( $\sigma = 0.01$ ), la propuesta resultó ser prácticamente igual al valor actual:  $\mu' = 2.999937$ . Al comparar la plausibilidad del valor actual y de la propuesta bajo la posterior, ambas resultaron casi idénticas, por lo que la razón de aceptación fue prácticamente 1 ( $\alpha = 0.999826$ ). Esto implica que la propuesta se acepta con una probabilidad altísima y, en efecto, la cadena se mueve al nuevo valor, quedando en  $\mu = 2.999937$ .

Este resultado es consistente: cuando la propuesta es muy cercana al valor actual, tanto la prior como la verosimilitud cambian muy poco, lo que hace que la posterior apenas varíe y la aceptación sea prácticamente segura.

```
2. one_mh_iteration_normal(s = 0.5, current = 3)
```

Ahora aplicamos la función `one_mh_iteration_normal()` al mismo valor inicial  $\mu = 3$ , pero aumentando el desvío estándar de la propuesta a  $\sigma = 0.5$ .

proposal	alpha	next_stop
3.33245	1	3.33245

En esta ocasión, la propuesta se aleja un poco más del valor inicial, resultando en  $\mu' = 3.33245$ . Al evaluar la plausibilidad del valor actual y de la propuesta bajo la posterior, la razón de aceptación fue máxima,  $\alpha = 1$ . Esto significa que la propuesta se acepta sin ninguna duda y la cadena se mueve directamente al nuevo valor, quedando en  $\mu = 3.33245$ . Aunque la propuesta se encuentra más lejos que en el caso con  $\sigma = 0.01$ , la posterior sigue otorgándole una plausibilidad igual o mayor que al valor actual, lo que justifica la aceptación total. Este comportamiento muestra cómo, al permitir pasos más amplios, la cadena puede explorar nuevas regiones del espacio de parámetros sin perder consistencia con la distribución posterior.

```
3. one_mh_iteration_normal(s = 1, current = 3)
```

A continuación, usamos `one_mh_iteration_normal()` con el mismo valor inicial  $\mu = 3$ , pero aumentando aún más el desvío estándar de la propuesta a  $\sigma = 1$ .

proposal	alpha	next_stop
4.595281	1	4.595281

En esta ejecución, la propuesta  $\mu' = 4.595281$  se aleja bastante del valor inicial  $\mu = 3$ . Sin embargo, al evaluar la plausibilidad bajo la posterior, resultó ser mayor o igual a la del valor actual, por lo que la razón de aceptación alcanzó su valor máximo,  $\alpha = 1$ . Esto implica que la propuesta fue aceptada sin ninguna duda, y la cadena se movió directamente al nuevo valor, quedando en  $\mu = 4.595281$ . Este resultado muestra cómo, aunque el salto sea más grande al aumentar  $\sigma$ , el algoritmo puede aceptar propuestas lejanas si los datos y la prior las respaldan, lo que contribuye a una exploración más amplia del espacio de parámetros.

```
4. one_mh_iteration_normal(s = 3, current = 3)
```

Finalmente, aplicamos `one_mh_iteration_normal()` con el valor inicial  $\mu = 3$  y un desvío estándar aún mayor para la propuesta,  $\sigma = 3$ .

proposal	alpha	next_stop
-10.85955	2.604854e-133	3

En esta iteración, la propuesta  $\mu' = -10.85955$  se aleja de manera extrema del valor actual. Como consecuencia, la plausibilidad bajo la posterior cae prácticamente a cero, y la razón de aceptación se vuelve enormemente pequeña ( $\alpha \approx 2.6 \times 10^{-133}$ ). Esto hace que la propuesta no se acepte y la cadena permanezca en  $\mu = 3$ . El resultado es consistente con el funcionamiento del algoritmo: cuando se proponen valores

muy alejados que no son compatibles ni con la prior ni con los datos, la probabilidad de aceptación es prácticamente nula.

En conclusión, al aumentar  $\sigma$  de forma excesiva, las propuestas pueden explorar regiones muy lejanas, pero la gran mayoría de ellas son rechazadas. Este comportamiento ilustra la necesidad de un equilibrio: valores pequeños de  $\sigma$  generan pasos seguros pero lentos, mientras que valores demasiado grandes producen propuestas inviables que rara vez se aceptan.

---

## Ejercicio 2 (Recorrido de Metropolis-Hastings con propuestas Normales) (3 puntos)

Emplear la nueva función `one_mh_iteration_normal()`, creada resolviendo el ejercicio anterior, con el fin de modificar la función de R `mh_tour()`, presentada en la sección 7.3 “*Implementing the Metropolis-Hastings*” del libro *Bayes Rules!*, para crear una nueva función de R `mh_tour_normal()`, la cual construye una cadena de valores de  $\mu$  utilizando un modelo de propuestas Normal con desviación estándar  $s$ .

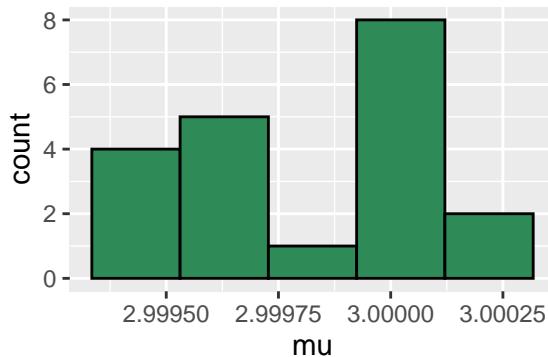
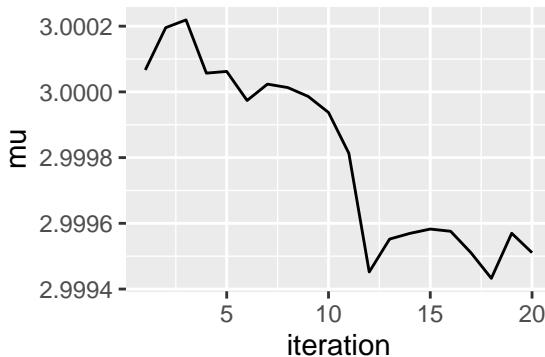
Utilizando `set.seed(84735)`, ejecutar la nueva función `mh_tour_normal()` bajo cada una de las siguientes configuraciones y construir un gráfico de la traza (*trace plot*) de cada cadena:

- a) 20 iteraciones,  $s = 0.01$
- b) 20 iteraciones,  $s = 10$
- c) 1000 iteraciones,  $s = 0.01$
- d) 1000 iteraciones,  $s = 10$

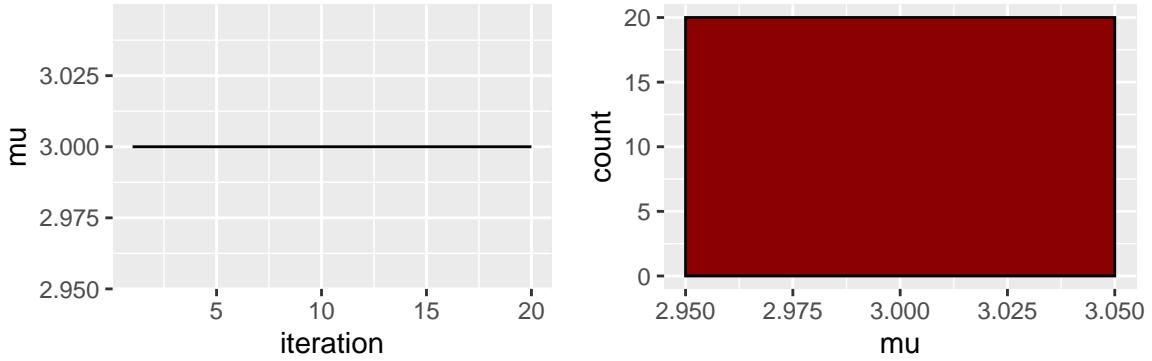
Comparar los gráficos de traza de los incisos a) y b). Explique en términos sencillos por qué cambiar la desviación estándar del modelo de propuestas Normal causa estas diferencias.

Reflexionando sobre los resultados anteriores, ajustar su algoritmo de Metropolis-Hastings. Es decir, identificar un valor razonable de la desviación estándar  $s$  y proporcionar un gráfico de traza como evidencia.

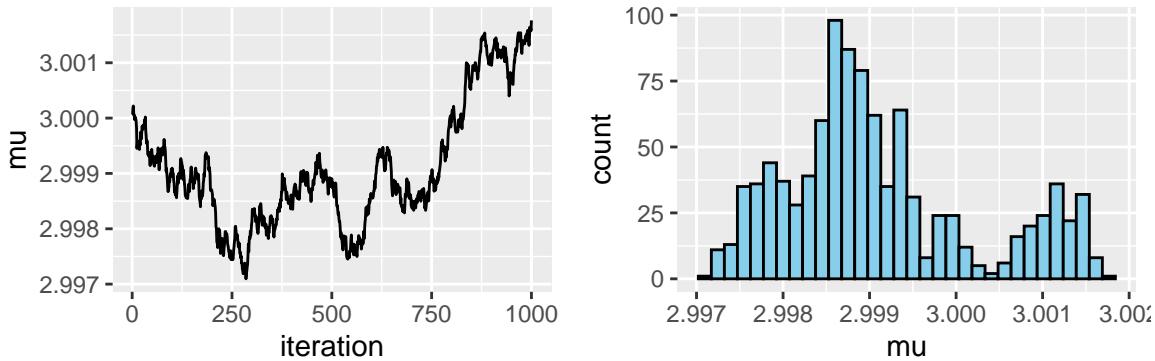
- a) 20 iteraciones,  $s = 0.01$



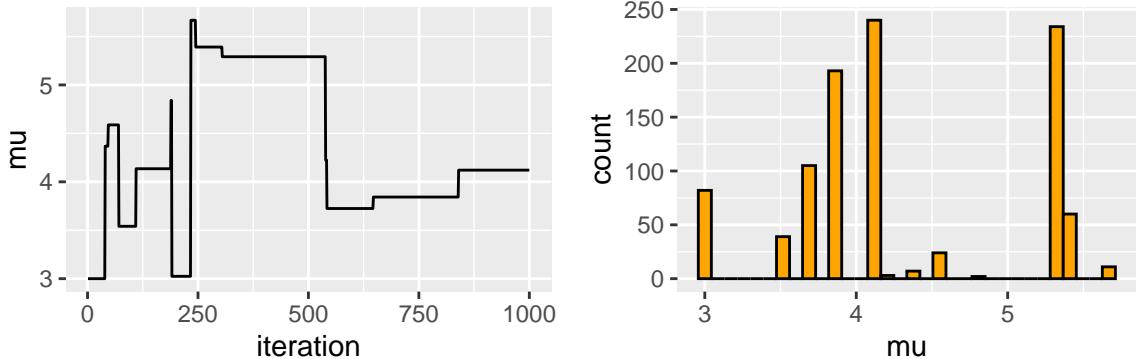
- b) 20 iteraciones,  $s = 10$



- c) 1000 iteraciones,  $s = 0.01$



- d) 1000 iteraciones,  $s = 10$



### Comparación entre los casos a) y b)

Al comparar los gráficos de traza de los apartados a) ( $s = 0.01$ ) y b) ( $s = 10$ ), se observan comportamientos muy distintos:

- Con  $s = 0.01$ , las propuestas se generan extremadamente cerca del valor actual, por lo que casi todas son aceptadas. El resultado es una traza muy “suave” donde la cadena avanza lentamente, con pasos diminutos, y apenas logra explorar la distribución en pocas iteraciones.

- Con  $s = 10$ , las propuestas se generan muy lejos del valor actual. La gran mayoría son rechazadas, y la cadena queda estancada durante largos períodos en un mismo valor, mostrando un gráfico en forma de “escalones”.

En términos sencillos: un  $s$  demasiado pequeño hace que la cadena se mueva, pero de forma muy lenta y local; un  $s$  demasiado grande hace que casi todas las propuestas sean descartadas y la cadena quede trabada. El valor de  $s$  controla el tamaño de los pasos y marca el balance entre explorar y aceptar propuestas.

---

### Resultados con 1000 iteraciones (casos c y d)

- En el apartado c) ( $s = 0.01$ , 1000 iteraciones), la cadena sigue avanzando con pasos minúsculos. Aunque con más iteraciones logra moverse un poco más, la exploración de la distribución sigue siendo muy lenta.
- En el apartado d) ( $s = 10$ , 1000 iteraciones), la cadena da saltos enormes que son en su mayoría rechazados, por lo que se queda atrapada mucho tiempo en ciertos valores. El histograma resultante refleja acumulaciones en pocos puntos, en lugar de una buena aproximación de la distribución objetivo.

Esto muestra que, aunque aumentar el número de iteraciones ayuda, una mala elección de  $s$  sigue afectando la eficiencia: con  $s$  demasiado chico la mezcla es muy lenta, y con  $s$  demasiado grande la cadena pasa la mayor parte del tiempo rechazando propuestas.

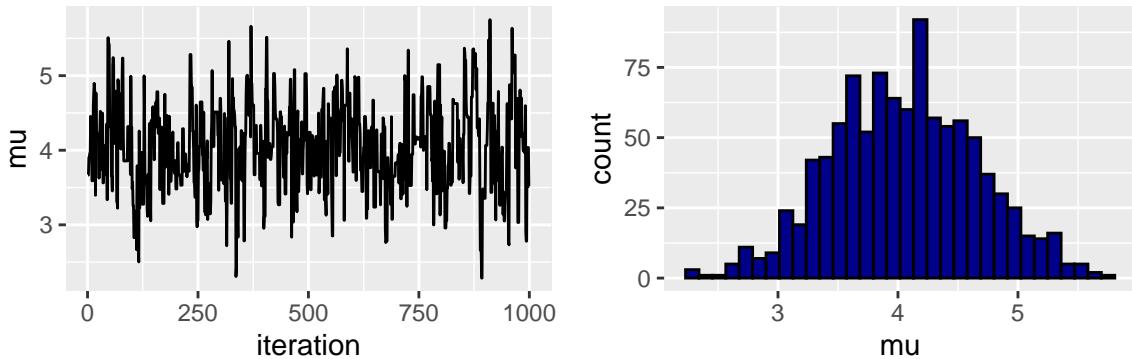
---

### Ajuste del algoritmo

Reflexionando sobre estos resultados, un valor razonable de  $s$  debería estar entre los extremos, de modo que:

- las propuestas no sean tan pequeñas que limiten la exploración,
- ni tan grandes que resulten casi siempre rechazadas.

En la práctica, un valor intermedio (por ejemplo,  $s \approx 1$  en este caso) logra un mejor equilibrio: la cadena acepta una fracción suficiente de propuestas y, al mismo tiempo, explora de manera eficiente la distribución objetivo. Un gráfico de traza con este ajuste mostraría una cadena que se mueve de manera fluida, recorriendo la distribución sin quedarse estancada ni avanzar con pasos imperceptibles.



### Ejercicio 3 (Modelo Normal-Normal) (3 puntos)

Se considera el modelo Normal-Normal de  $\mu$  con:

$$Y_i \mid \mu \sim N(\mu, 8^2) \quad \text{y} \quad \mu \sim N(-14, 2^2).$$

Suponer que con  $n = 5$  observaciones independientes se obtienen los siguientes datos:

$$(Y_1, Y_2, Y_3, Y_4, Y_5) = (-10.1, 5.5, 0.1, -1.4, 11.5).$$

1. Simular el modelo a posteriori de  $\mu$  (`set.seed(84735)`) utilizando RStan con 4 cadenas y 10000 iteraciones por cadena.

```
# Modelo Normal-Normal en Stan (media desconocida, conocida)

nn_model <- "
  data {
    int<lower=1> N;                      // número de observaciones
    real y[N];                            // datos normales
    real mu0;                             // media de la prior de mu
    real<lower=0> tau0;                  // desvío estándar de la prior de mu
    real<lower=0> sigma;                 // desvío estándar conocido de los datos
  }
  parameters {
    real mu;                                // parámetro de interés
  }
  model {
    // Prior
    mu ~ normal(mu0, tau0);

    // Verosimilitud
    y ~ normal(mu, sigma);
  }
"
```

```
# Datos y prior
y      <- c(-10.1, 5.5, 0.1, -1.4, 11.5)
N      <- length(y)
mu0   <- -14    # media de la prior
tau0  <- 2      # desvío estándar de la prior
sigma <- 8      # desvío estándar conocido de los datos
```

```
# PASO 2: Simulación de la distribución a posteriori con rstan::stan()
gp_sim <- stan(
  model_code = nn_model,
  data = list(N = N, y = y, mu0 = mu0, tau0 = tau0, sigma = sigma),
  chains = 4, iter = 10000, warmup = 1000, seed = 84735
)
```

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install the appropriate version of Rtools for 4.4.3 from
## https://cran.r-project.org/bin/windows/Rtools/.
```

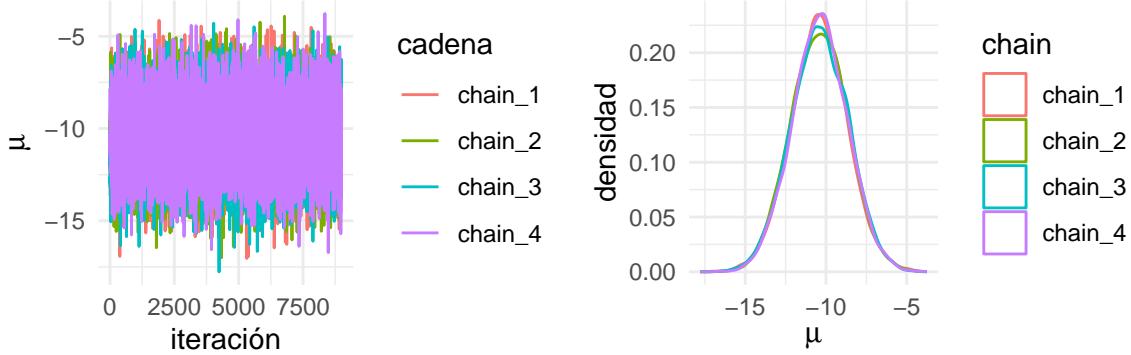
```

## Trying to compile a simple C file

## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install the appropriate version of Rtools for 4.4.3 from
## https://cran.r-project.org/bin/windows/Rtools/.

```

2. Generar los gráficos de traza (*trace plots*) y de densidad (*density plots*) para las cuatro cadenas.



3. A partir de los gráficos de densidad, ¿cuál parece ser el valor más plausible a posteriori de  $\mu$ ?

### 3. Valor más plausible a posteriori de $\mu$

A partir de los gráficos de densidad obtenidos para las cuatro cadenas, se observa que las distribuciones posteriores son muy similares entre sí, lo que indica una buena convergencia. Todas presentan un pico máximo en torno al mismo rango de valores.

El valor más plausible a posteriori de  $\mu$  (es decir, el de mayor densidad posterior) se encuentra aproximadamente en  $-10$ . Este resultado refleja el punto en el que las cadenas asignan la mayor probabilidad de densidad al parámetro bajo el modelo considerado.

4. Especificar el modelo a posteriori exacto de  $\mu$  (sección 5.3.3 “*Normal-Normal conjugacy*” del libro *Bayes Rules!*).

¿Cómo se compara su aproximación MCMC?

### 4. Posterior exacto de $\mu$ y comparación con MCMC

**Modelo.** Sean  $y_1, \dots, y_N \mid \mu \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$  con  $\sigma^2$  conocida. La previa es  $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ .

**Resultado de conjugación (sección 5.3.3 de *Bayes Rules!*).** La distribución a posteriori es nuevamente normal:

$$\mu \mid \mathbf{y} \sim \mathcal{N}(\mu_N, \tau_N^2),$$

donde, escribiendo

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i,$$

$$\tau_N^2 = \left( \frac{1}{\tau_0^2} + \frac{N}{\sigma^2} \right)^{-1}, \quad \mu_N = \tau_N^2 \left( \frac{\mu_0}{\tau_0^2} + \frac{N \bar{y}}{\sigma^2} \right).$$

### Cómo se compara la aproximación MCMC.

La aproximación por MCMC debería reproducir muy de cerca esta  $\mathcal{N}(\mu_N, \tau_N^2)$ :

- La **media posterior** estimada por las cadenas debe estar muy próxima a  $\mu_N$ .
- La **dispersión** (desvío estándar posterior) de los draws debe coincidir con  $\tau_N$ .
- Diagnósticos como  $\hat{R} \approx 1$ , tamaño efectivo grande y error Monte Carlo pequeño respecto de  $\tau_N$  confirman la concordancia.

En las densidades por cadena, el pico común alrededor de  $-10$  indica que la media posterior estimada por MCMC es coherente con la solución cerrada; superponer la curva  $\mathcal{N}(\mu_N, \tau_N^2)$  sobre las densidades de MCMC debería ser prácticamente indistinguible.

---

### Ejercicio 4 (El modelo Beta-Geométrico) (3.5 puntos)

Se considera el siguiente modelo bayesiano:

$$Y \mid \theta \sim \text{Geométrica}(\theta), \quad \theta \sim \text{Beta}(\alpha, \beta),$$

donde el modelo Geométrico tiene función de masa de probabilidad:

$$f(y \mid \theta) = \theta(1 - \theta)^{y-1}, \quad y \in \{1, 2, \dots\}.$$

Hallar el modelo a posteriori de  $\theta$  dada una muestra de  $n$  observaciones independientes  $(Y_1, \dots, Y_n) = (y_1, \dots, y_n)$ .

Identificar el nombre de la familia de la distribución a posteriori de  $\theta$  y cuáles son sus parámetros.  
¿Es el modelo Beta una distribución a priori conjugada para el modelo de datos Geométrico?

Suponer que  $\theta \sim \text{Beta}(2, 5)$ , y que con  $n = 6$  observaciones independientes se obtienen los siguientes datos:

$$(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6) = (3, 5, 1, 3, 4, 5).$$

- Simular el modelo a posteriori de  $\theta$  (`set.seed(84735)`) utilizando RStan con 4 cadenas y 10000 iteraciones por cadena.

```
# Modelo Geométrica-Beta en Stan (theta desconocida, y enteros >= 1)
# Modelo Geométrica-Beta en Stan con verosimilitud manual
gb_model <- "
  data {
    int<lower=1> N;                      // número de observaciones
    int<lower=1> y[N];                    // datos geométricos: 1,2,3,...
    real<lower=0> alpha;                  // parámetro alpha de la prior Beta
    real<lower=0> beta;                   // parámetro beta de la prior Beta
  }
  parameters {
    real<lower=0, upper=1> theta;        // prob. de éxito (0<theta<1)
  }
  model {
    // Prior
    theta ~ beta(alpha, beta);
```

```

    // Verosimilitud: P(Y=y) = theta * (1-theta)^(y-1), y>=1
    target += N * log(theta) + (sum(y) - N) * log1m(theta);
}
"

# Datos y prior (tu ejemplo)
y      <- c(3, 5, 1, 3, 4, 5)
N      <- length(y)
alpha  <- 2
beta   <- 5

# Simulación MCMC de la a posteriori de theta
fit_gb <- stan(
  model_code = gb_model,
  data = list(N = N, y = y, alpha = alpha, beta = beta),
  chains = 4, iter = 10000, warmup = 1000, seed = 84735
)

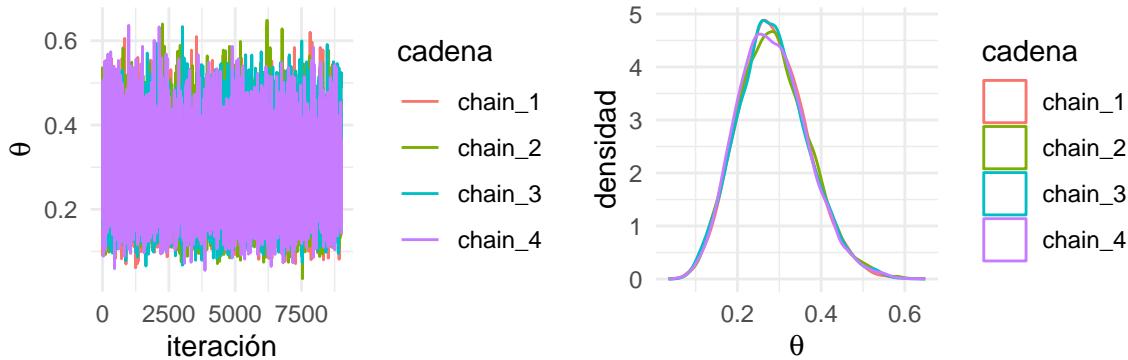
```

## WARNING: Rtools is required to build R packages, but is not currently installed.  
##  
## Please download and install the appropriate version of Rtools for 4.4.3 from  
## <https://cran.r-project.org/bin/windows/Rtools/>.

## Trying to compile a simple C file

## WARNING: Rtools is required to build R packages, but is not currently installed.  
##  
## Please download and install the appropriate version of Rtools for 4.4.3 from  
## <https://cran.r-project.org/bin/windows/Rtools/>.

- Generar los gráficos de traza (*trace plots*) y de densidad (*density plots*) para las cuatro cadenas.



- A partir de los gráficos de densidad, ¿cuál parece ser el valor más plausible a posteriori de  $\theta$ ?

### Valor más plausible a posteriori de $\theta$

A partir de los gráficos de densidad por cadena, la masa posterior se concentra con un pico alrededor de  $\theta \approx 0.27$ –0.30.

Bajo la conjugación Beta(2, 5) con los datos (3, 5, 1, 3, 4, 5), la posterior exacta es

$$\theta | y \sim \text{Beta}(\alpha + n, \beta + \sum y - n) = \text{Beta}(8, 20).$$

Para esta distribución:

- **Moda (MAP):**

$$\frac{\alpha - 1}{\alpha + \beta - 2} = \frac{8 - 1}{8 + 20 - 2} = \frac{7}{26} \approx 0.269$$

- **Media:**

$$\frac{\alpha}{\alpha + \beta} = \frac{8}{28} \approx 0.286$$

Ambas coinciden con el máximo de las densidades MCMC, por lo que el **valor más plausible a posteriori (modo)** es

$$\theta \approx 0.27$$

y la media posterior se ubica cerca de

$$\theta \approx 0.29.$$

- Especificar el modelo a posteriori exacto de  $\theta$ .  
¿Cómo se compara su aproximación MCMC?

### A posteriori exacto de $\theta$ y comparación con MCMC

Partiendo del modelo

$$Y_i | \theta \sim \text{Geométrica}(\theta), \quad \theta \sim \text{Beta}(\alpha, \beta),$$

con función de masa de probabilidad

$$f(y_i | \theta) = \theta (1 - \theta)^{y_i - 1}, \quad y_i \in \{1, 2, \dots\},$$

la verosimilitud para una muestra independiente  $y_1, \dots, y_n$  es

$$L(\theta | \mathbf{y}) \propto \theta^n [1 - \theta]^{\sum_{i=1}^n y_i - n}.$$

Multiplicando por la previa Beta( $\alpha, \beta$ ) se obtiene

$$p(\theta | \mathbf{y}) \propto \theta^{\alpha+n-1} [1 - \theta]^{\beta+\sum y_i - n - 1}.$$

Por lo tanto, la **distribución a posteriori exacta** es

Por lo tanto, la **distribución a posteriori exacta** es

$$\theta \mid \mathbf{y} \sim \text{Beta}(\alpha^*, \beta^*), \quad \alpha^* = \alpha + n, \quad \beta^* = \beta + \sum y_i - n.$$

Para el caso numérico con  $\alpha = 2$ ,  $\beta = 5$ ,  $n = 6$  y  $(3, 5, 1, 3, 4, 5)$ :

$$\theta \mid \mathbf{y} \sim \text{Beta}(8, 20).$$

- **Media posterior:**

$$\mathbb{E}[\theta \mid \mathbf{y}] = \frac{8}{8+20} = \frac{8}{28} \approx 0.286.$$

- **Moda (MAP):**

$$\frac{\alpha^* - 1}{\alpha^* + \beta^* - 2} = \frac{8 - 1}{8 + 20 - 2} = \frac{7}{26} \approx 0.269.$$

### ¿Cómo se compara la aproximación MCMC?

Los gráficos de densidad por cadena muestran curvas prácticamente superpuestas y con pico cerca de  $\theta \approx 0.27$ , y masa distribuida en torno a  $\theta \approx 0.29$ , lo que coincide con la posterior exacta Beta(8, 20) (modo  $\approx 0.269$ , media  $\approx 0.286$ ).

Las pequeñas discrepancias visibles se explican por **ruido de Monte Carlo** y desaparecen al aumentar las iteraciones o el adelgazamiento. En síntesis, la aproximación MCMC **reproduce fielmente** la a posteriori conjugada esperada.