

Método da Rejeição

Isabelle Fernandes de Oliveira

2025-06-04

```
mi_real = 5 # valor desconhecido que quero estimar
sigma_2_real = 3 # valor conhecido
n = 50 # tamanho da amostra
x = rnorm(n, mi_real, sqrt(sigma_2_real)) # minha amostra
v = 10 # variancia da priori theta ~ N(m, v)
m = 8 # media da priori
theta = rnorm(1000, m, sqrt(v)) # gerei valores de theta a partir da priori
m_g = 10 # parametros da g(theta) ~ N(m_g, v_g)
v_g = 10

#nucleo e ver se há valor negativo
pi_theta = exp((-1/(2*sigma_2_real)) * (-2*theta*sum(x) + n*theta^2))*
  exp((-1/(2*v)) * (theta^2 - 2*theta*m)) # avaliando o nucleo

cat("menor valor encontrado: ", min(pi_theta)) # nao há valor negativo

## menor valor encontrado: 0

w_theta = pi_theta / dnorm(theta, m_g, sqrt(v_g))
# importance ratio
M2 = max(w_theta)

n_samp = 1000 # tamanho amostral desejado
samp = NULL
i = 1

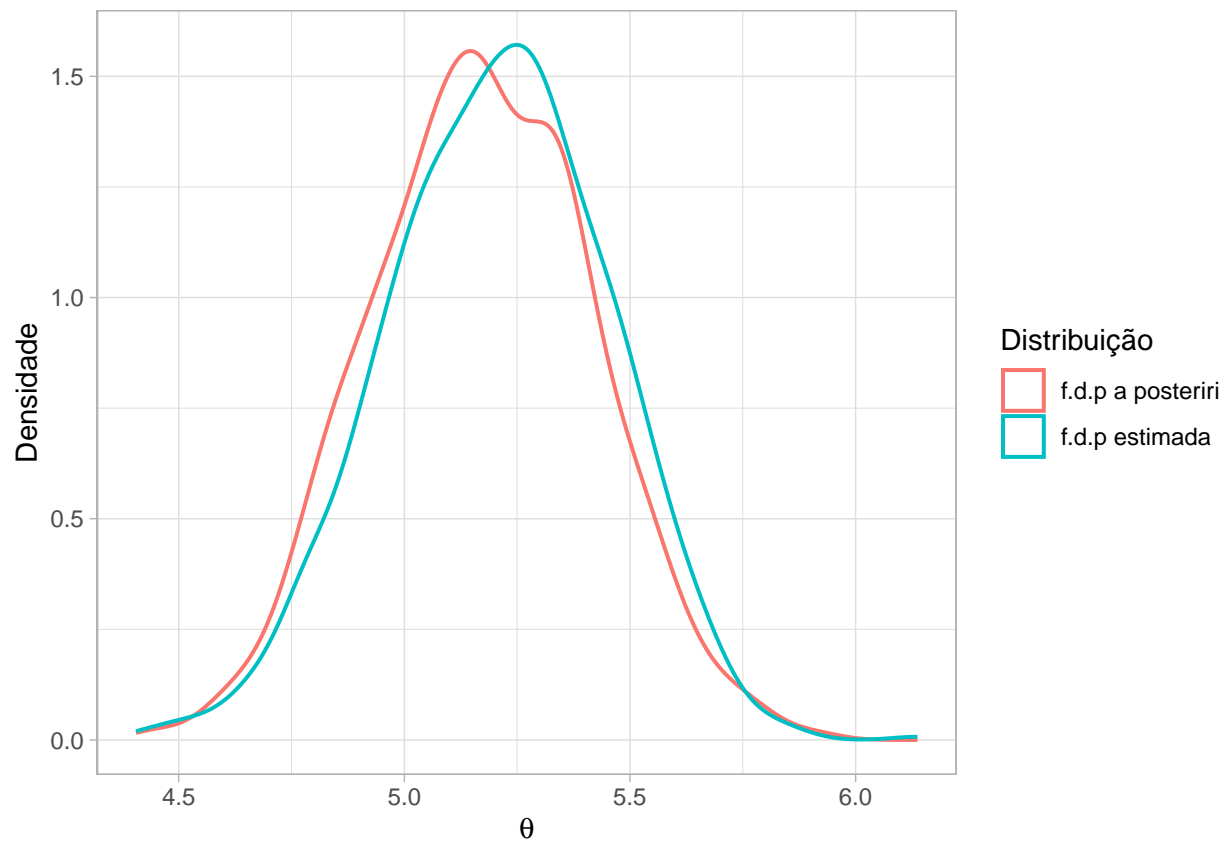
while (length(samp) < n_samp) {
  theta = rnorm(1, m_g, sqrt(v_g)) # gera candidato
  pi_theta = exp((-1/(2*sigma_2_real)) * (-2*theta*sum(x) + n*theta^2))*
    exp((-1/(2*v)) * (theta^2 - 2*theta*m)) # avaliando o nucleo
  w_theta = pi_theta / dnorm(theta, m_g, v_g) # importance ratio
  if(runif(1,0,1) < (w_theta/M2)){
    samp[i] = theta
    i = i+1
  }
}

v_star = 1/((n/sigma_2_real) + 1/v) # parametros da posteriori real
m_star = v_star * ((sum(x)/sigma_2_real) + m/v) # parametros da posteriori real

posteriori_real = rnorm(n_samp, m_star, sqrt(v_star))
```

```
df <- bind_rows(
  data.frame(theta = posteriori_real, tipo = "f.d.p a posteriri"),
  data.frame(theta = samp, tipo = "f.d.p estimada")
)

ggplot(df, aes(x = theta, color = tipo)) +
  geom_density(alpha = 0.1, linewidth = 0.7) +
  labs(
    x = expression(theta),
    y = "Densidade",
    color = "Distribuição",
    fill = "Distribuição"
  ) +
  theme_light()
```



```
media_post = m_star
var_post = v_star

cat("Média f.d.p estimada: ", mean(samp))
```

```
## Média f.d.p estimada: 5.204433
```

```
cat("Média f.d.p posteriori: ", media_post)
```

```
## Média f.d.p posteriori: 5.160942
```

```
cat("Variancia f.d.p estimada: ", var(samp))
```

```
## Variancia f.d.p estimada: 0.06053764
```

```
cat("Variancia f.d.p posteriori: ", var_post)
```

```
## Variancia f.d.p posteriori: 0.05964215
```

```
rm(list = ls(all = TRUE))
```

```
mi_real = 13 # valor conhecido
```

```
sigma_2_real = 5 # valor desconhecido que quero estimar
```

```
n = 50 # tamanho da amostra
```

```
x = rnorm(n, mi_real, sqrt(sigma_2_real)) # minha amostra
```

```
a = 12 # variancia da priori  $\theta \sim GI(a, b)$ 
```

```
b = 110 # media da priori
```

```
theta = 1/rgamma(1000, shape = a, rate = b) # gerei valores de  $\theta$  a partir da priori
```

```
a_g = 10 # parametros da  $g(\theta) \sim N(m_g, v_g)$ 
```

```
b_g = 1
```

```
#nucleo e ver se há valor negativo
```

```
pi_theta = theta^(-a - n/2 - 1)*
```

```
exp(-1/theta *(b + 1/2 * sum((x - mi_real)^2))) # avaliando o nucleo
```

```
cat("menor valor encontrado: ", min(pi_theta)) # não há valor negativo
```

```
## menor valor encontrado: 1.466558e-60
```

```
w_theta = pi_theta / dgamma(theta, shape = a_g, rate = b_g)
```

```
# importance ratio
```

```
M2 = max(w_theta)
```

```
n_samp = 1000 # tamanho amostral desejado
```

```
samp = NULL
```

```
i = 1
```

```
while (length(samp) < n_samp) {
```

```
  theta = rgamma(1, a_g, b_g) # gera candidato
```

```
  pi_theta = theta^(-a - n/2 - 1)*
```

```
  exp(-1/theta *(b + 1/2 * sum((x - mi_real)^2))) # avaliando o nucleo
```

```
  w_theta = pi_theta / dgamma(theta, a_g, b_g) # importance ratio
```

```
  if(runif(1,0,1) < (w_theta/M2)){
```

```
    samp[i] = theta
```

```
    i = i+1
```

```
  }
```

```
}
```

```

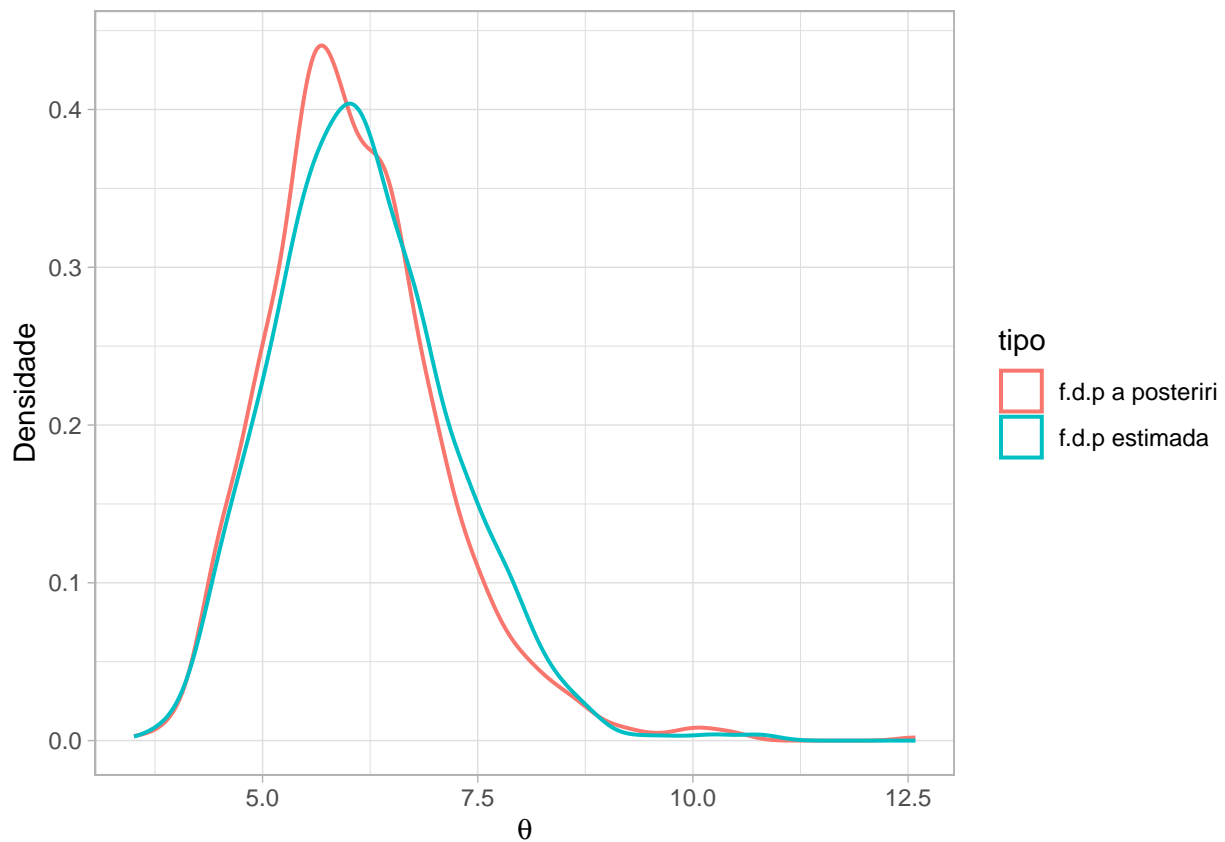
a_star = a + n/2 # parametros da posteriori real
b_star = b + 0.5*(sum(x^2) - 2*mi_real*sum(x) + n*mi_real^2) # parametros da posteriori real

posteriori_real = 1/rgamma(n_samp, shape = a_star, rate = b_star)

df <- bind_rows(
  data.frame(theta = posteriori_real, tipo = "f.d.p a posteriri"),
  data.frame(theta = samp, tipo = "f.d.p estimada")
)

ggplot(df, aes(x = theta, color = tipo)) +
  geom_density(alpha = 0.9, linewidth = 0.7) +
  labs(
    x = expression(theta),
    y = "Densidade",
  ) +
  theme_light()

```



```

media_post = b_star/(a_star-1)
var_post = b_star^2/((a_star-1)^2 * (a_star-2))

cat("Média f.d.p estimada: ", mean(samp))

```

```
## Média f.d.p estimada: 6.173493
```

```
cat("Média f.d.p posteriori: ", media_post)
```

```
## Média f.d.p posteriori: 6.090455
```

```
cat("Variancia f.d.p estimada: ", var(samp))
```

```
## Variancia f.d.p estimada: 1.052789
```

```
cat("Variancia f.d.p posteriori: ", var_post)
```

```
## Variancia f.d.p posteriori: 1.059818
```

```
rm(list = ls(all = TRUE))
```

```
mi = 0 # parâmetros da  $g(\theta) \sim N(mi, \sigma_{2\_real})$ 
```

```
sigma_2_real = 1
```

```
n = 10000 # tamanho da primeira amostra
```

```
theta = rnorm(n, mi, sqrt(sigma_2_real))
```

```
cauchy = 1 / (pi * (1 + theta^2))
```

```
w_theta = cauchy / dnorm(theta)
```

```
# importance ratio
```

```
M2 = max(w_theta)
```

```
n_samp = 1000 # tamanho amostral desejado
```

```
samp = NULL
```

```
i = 1
```

```
while (length(samp) < n_samp) {
```

```
  theta = rnorm(1, mi, sqrt(sigma_2_real))
```

```
  # gera candidato
```

```
  pi_theta = 1 / (pi * (1 + theta^2)) # minha  $\pi(\theta)$  é a própria distribuição desejada  
  # a estimar
```

```
  w_theta = pi_theta / dnorm(theta, mi, sqrt(sigma_2_real)) # importance ratio
```

```
  if(runif(1,0,1) < (w_theta/M2)){
```

```
    samp[i] = theta
```

```
    i = i+1
```

```
  }
```

```
}
```

```
theta_seq <- seq(-10, 10, length.out = 1000)
```

```
# Densidade teórica da Cauchy(0,1)
```

```
densidade_real <- 1 / (pi * (1 + theta_seq^2))
```

```
# Densidade estimada a partir da amostra
```

```
densidade_estimada <- density(samp, from = -10, to = 10, n = 1000)$y
```

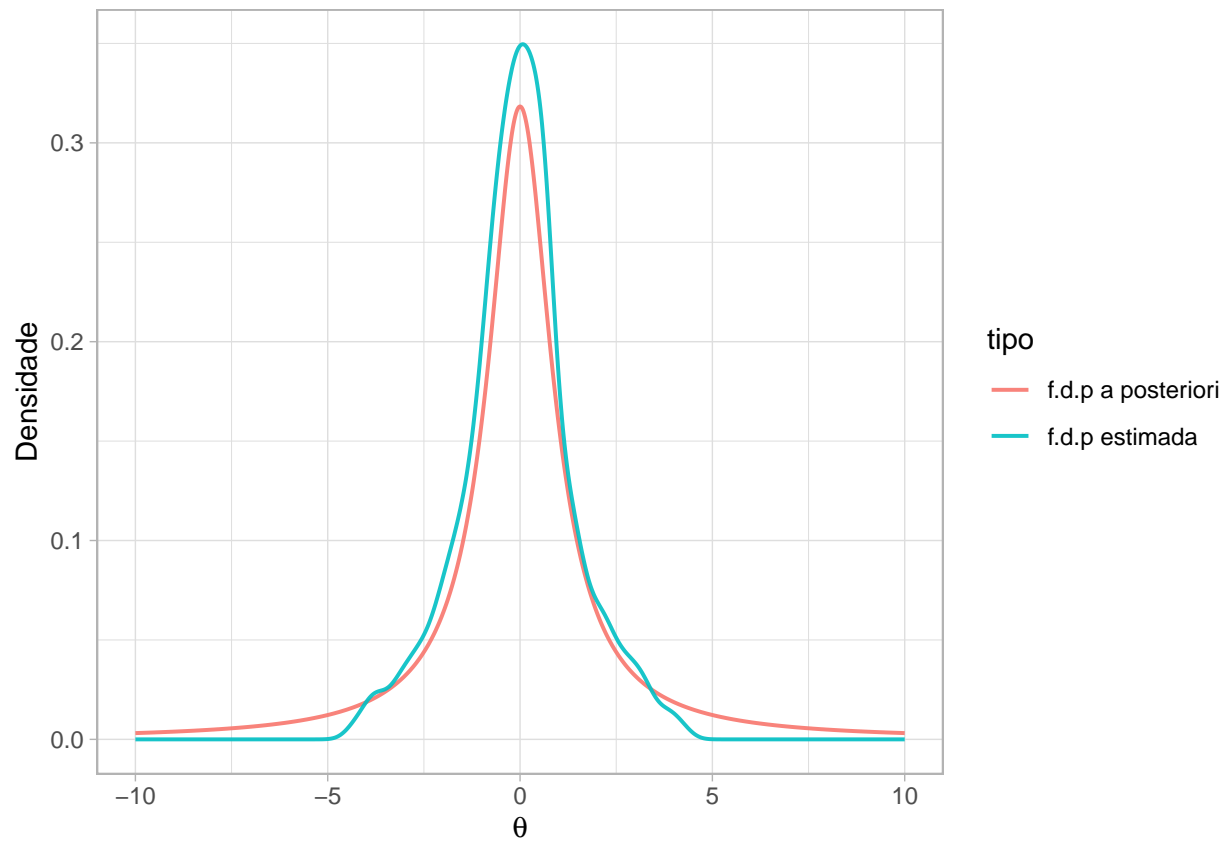
```
df <- bind_rows(
```

```
  data.frame(theta = theta_seq, densidade = densidade_real, tipo = "f.d.p a posteriori"),
```

```
  data.frame(theta = theta_seq, densidade = densidade_estimada, tipo = "f.d.p estimada")
```

```
)

# Gera o gráfico de densidade
ggplot(df, aes(x = theta, y = densidade, color = tipo, fill = tipo,)) +
  geom_line(alpha = 0.9, linewidth = 0.7) +
  labs(
    x = expression(theta),
    y = "Densidade",
  ) +
  theme_light()
```



```
cat("Média f.d.p estimada: ", mean(samp))
```

```
## Média f.d.p estimada: -0.04925599
```

```
cat("Média f.d.p posteriori: ", mean(rcauchy(n_samp)))
```

```
## Média f.d.p posteriori: -1.03906
```

```
cat("Variancia f.d.p estimada: ", var(samp))
```

```
## Variancia f.d.p estimada: 2.058666
```

```

cat("Variancia f.d.p posteriori: ", var(rcauchy(n_samp)))

## Variancia f.d.p posteriori: 4031.705

rm(list = ls(all = TRUE))

mi = 0 # parâmetros da  $g(\theta) \sim N(mi, \sigma_{2\_real})$ 
sigma_2_real = 10
n = 1000 # tamanho da primeira amostra
theta = rnorm(n, mi, sqrt(sigma_2_real))
cauchy = 1 / (pi * (1 + theta^2))

w_theta = cauchy / dnorm(theta)
# importance ratio
M2 = 1000 # percebi que pegar o máximo era uma constante muito grande, então na
# divisão  $w_{\theta}/M2$  dava um número muito próximo de zero

n_samp = 1000 # tamanho amostral desejado
samp = NULL
i = 1

while (length(samp) < n_samp) {
  theta = rnorm(1, mi, sqrt(sigma_2_real))
  # gera candidato
  pi_theta = 1 / (pi * (1 + theta^2)) # minha  $\pi(\theta)$  é a própria distribuição desejada
  # a estimar
  w_theta = pi_theta / dnorm(theta, mi, sqrt(sigma_2_real)) # importance ratio
  if(runif(1,0,1) < (w_theta/M2)){
    samp[i] = theta
    i = i+1
  }
}

theta_seq <- seq(-10, 10, length.out = 1000)

# Densidade teórica da Cauchy(0,1)
densidade_real <- 1 / (pi * (1 + theta_seq^2))

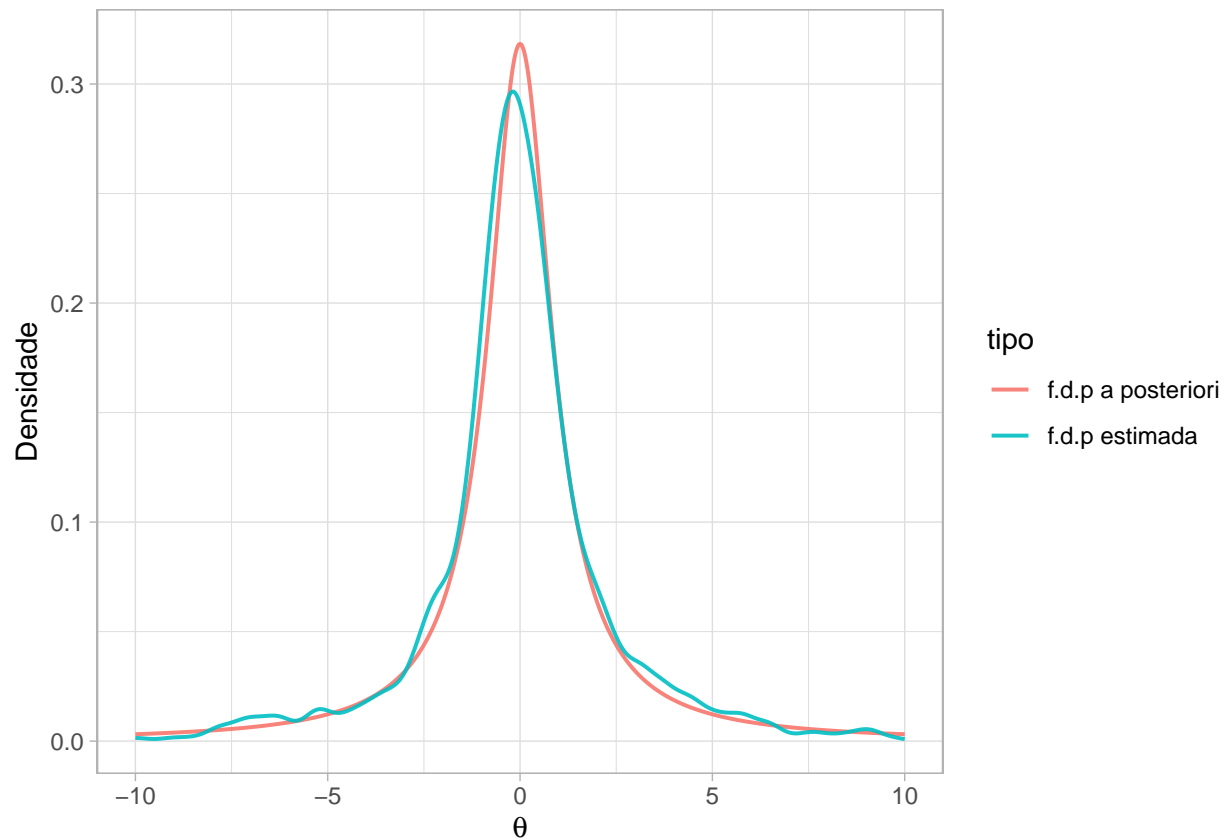
# Densidade estimada a partir da amostra
densidade_estimada <- density(samp, from = -10, to = 10, n = 1000)$y

df <- bind_rows(
  data.frame(theta = theta_seq, densidade = densidade_real, tipo = "f.d.p a posteriori"),
  data.frame(theta = theta_seq, densidade = densidade_estimada, tipo = "f.d.p estimada")
)

# Gera o gráfico de densidade
ggplot(df, aes(x = theta, y = densidade, fill = tipo, color = tipo)) +
  geom_line(alpha = 0.9, linewidth = 0.7) +
  labs(
    x = expression(theta),
    y = "Densidade",
  ) +

```

```
theme_light()
```



```
cat("Média f.d.p estimada: ", mean(samp))
```

```
## Média f.d.p estimada: 0.0839758
```

```
cat("Média f.d.p posteriori: ", mean(rcauchy(n_samp)))
```

```
## Média f.d.p posteriori: -5.467885
```

```
cat("Variancia f.d.p estimada: ", var(samp))
```

```
## Variancia f.d.p estimada: 9.21637
```

```
cat("Variancia f.d.p posteriori: ", var(rcauchy(n_samp)))
```

```
## Variancia f.d.p posteriori: 913.1812
```