

Algoritmo de Metropolis

Distribuição normal padrão

2026

O algoritmo de Metropolis é aplicado para gerar uma amostra da distribuição $\text{normal}(0, 1)$. A distribuição proposta é $\text{normal}(x, \sigma_0^2)$, sendo que a variância σ_0^2 é fixada. Este exemplo está no livro Gilks *et al.* (1996, *Markov Chain Monte Carlo in Practice*, Cap. 1. London: Chapman & Hall).

```
# Separador decimal: ","
options(OutDec = ",")

# Número de iterações
M <- 500

# Desvio padrão da distribuição proposta
sigprop <- sqrt(0.1)

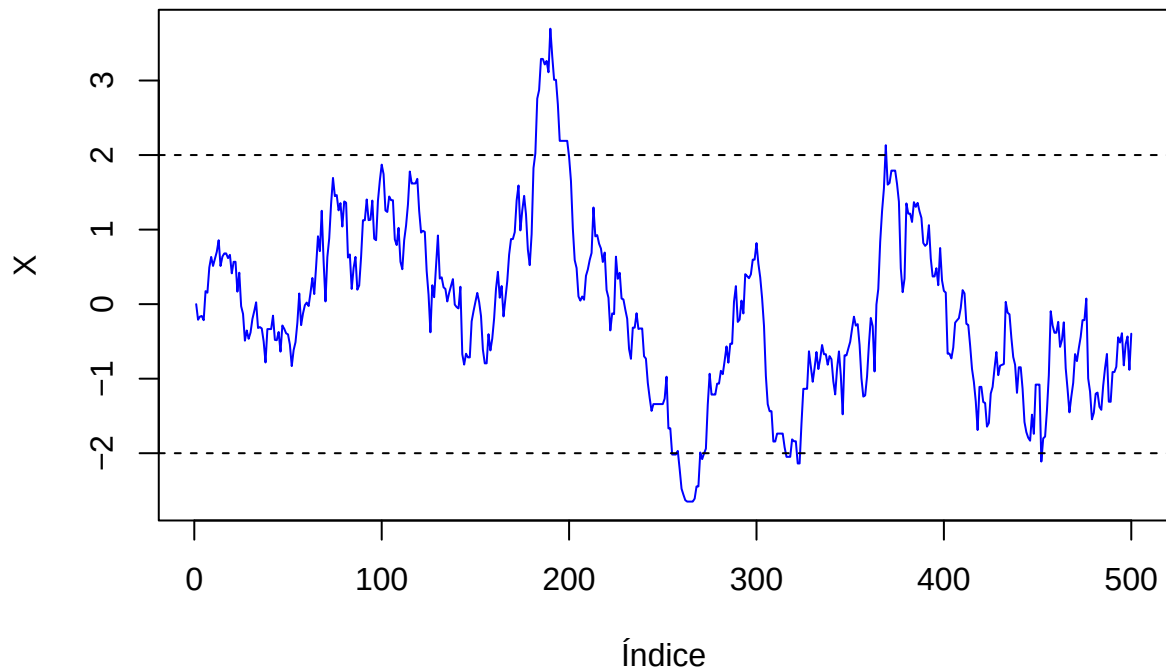
# Algoritmo de Metropolis
set.seed(14)
x <- c()
x[1] <- 0
for (j in 2:M) {
  # candidato
  y <- rnorm(1, mean = x[j - 1], sd = sigprop)

  alfa <- min(1, dnorm(y, 0, 1) / dnorm(x[j - 1], 0, 1))

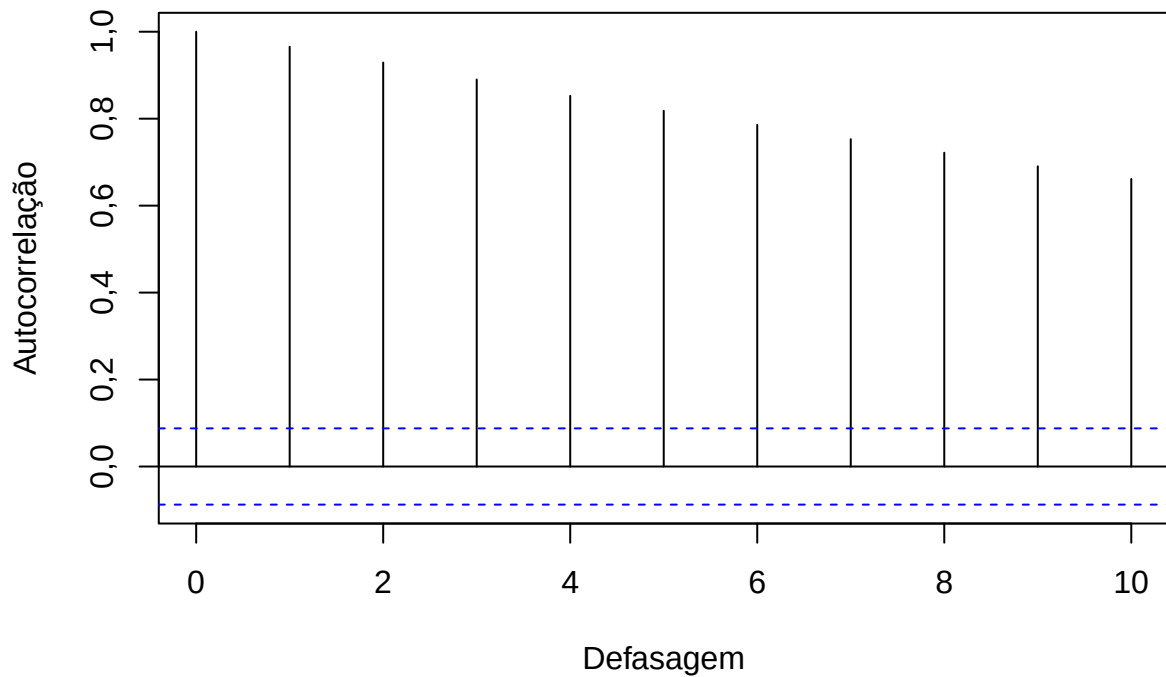
  # geração
  if (rbinom(1, 1, alfa) == 1) {
    x[j] <- y
  } else {
    x[j] <- x[j - 1]
  }
}
```

Em seguida as sequências geradas são apresentadas graficamente.

```
plot(x, type = "l", xlab = "Índice", ylab = "X", col = "blue",
     ylim = range(x, -2, 2))
abline(h = c(-2, 2), lty = 2)
```



```
acf(x, lag.max = 10, main = "", xlab = "Defasagem",
    ylab = "Autocorrelação")
```



```
# Desvio padrão da distribuição proposta
sigprop <- sqrt(10)
```

```
# Algoritmo de Metropolis
x <- c()
x[1] <- 0
for (j in 2:M) {
  # candidato
  y <- rnorm(1, mean = x[j - 1], sd = sigprop)
```

```

alfa <- min(1, dnorm(y, 0, 1) / dnorm(x[j - 1], 0, 1))

# geração
if (rbinom(1, 1, alfa) == 1) {
  x[j] <- y
} else {
  x[j] <- x[j - 1]
}
}

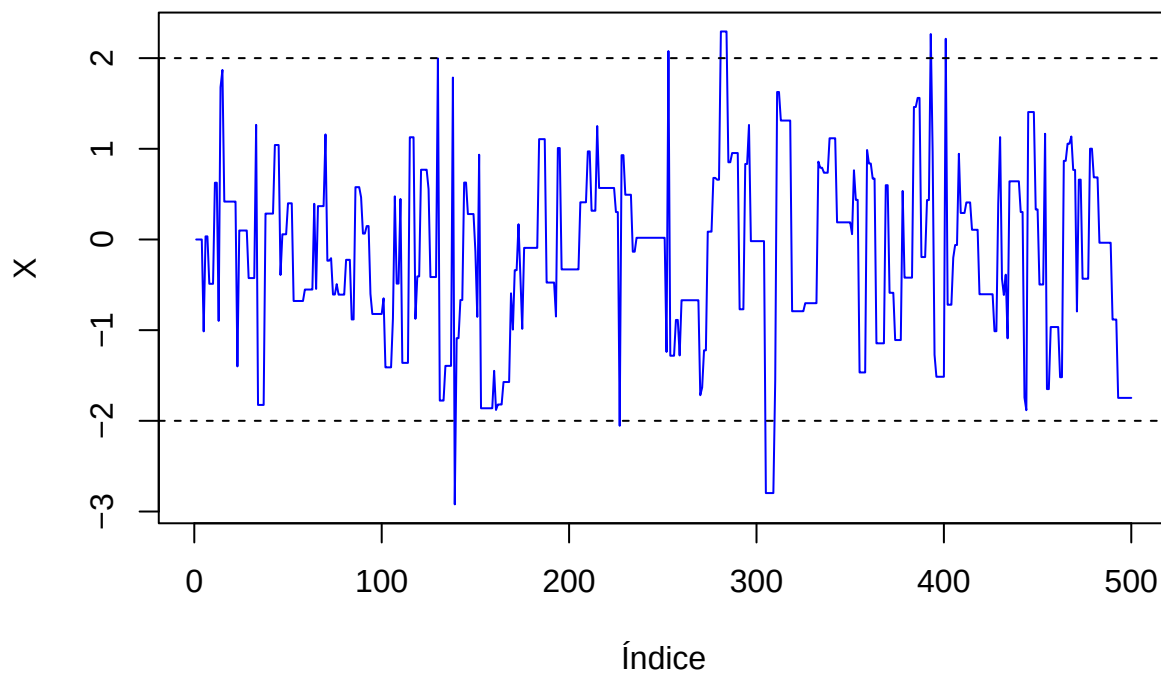
```

Em seguida as sequências geradas são apresentadas graficamente.

```

plot(x, type = "l", xlab = "Índice", ylab = "X", col = "blue",
     ylim = range(x, -2, 2))
abline(h = c(-2, 2), lty = 2)

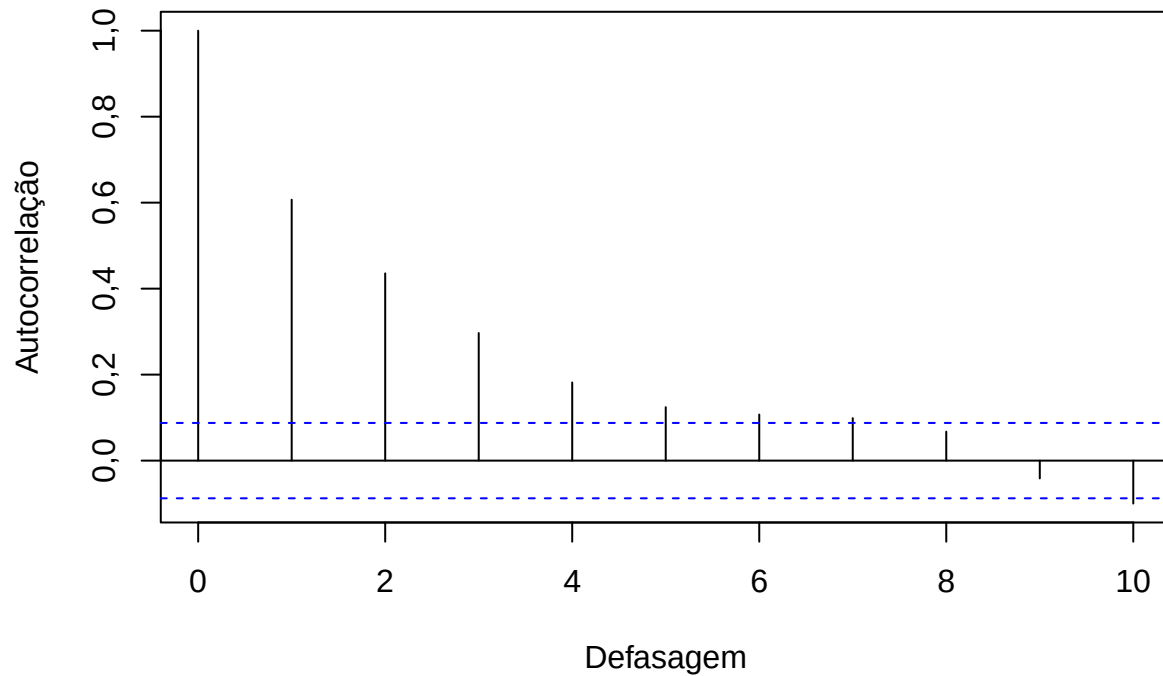
```



```

acf(x, lag.max = 10, main = "", xlab = "Defasagem",
    ylab = "Autocorrelação")

```



```
# Desvio padrão da distribuição proposta
sigprop <- sqrt(0.8)

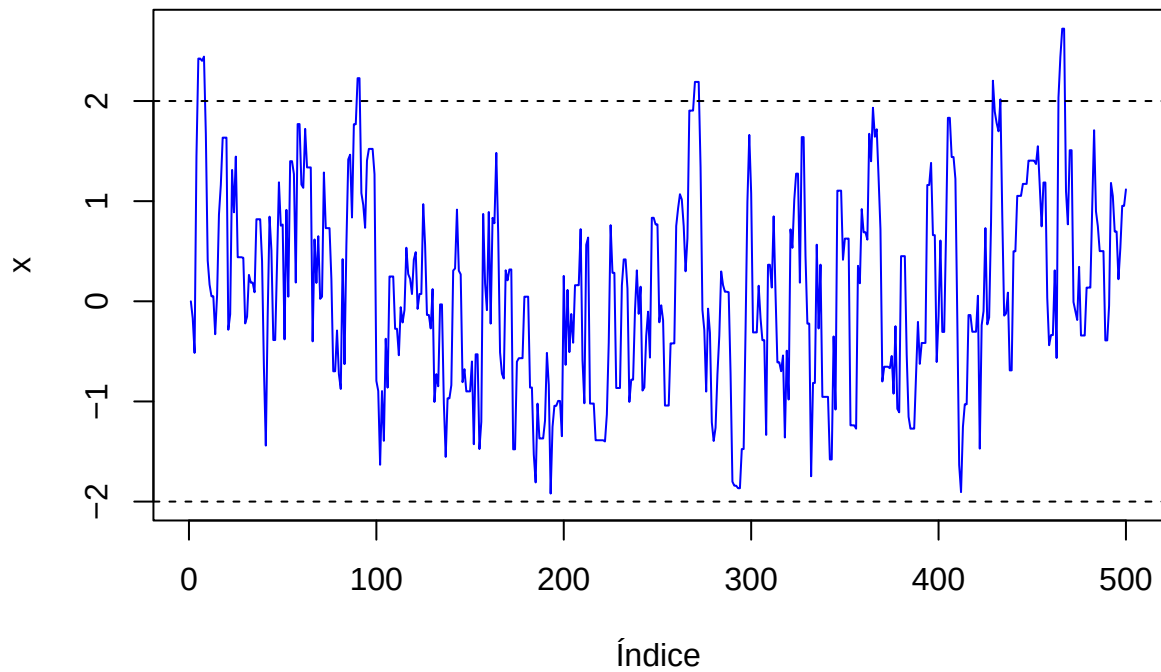
# Algoritmo de Metropolis
x <- c()
x[1] <- 0
for (j in 2:M) {
  # candidato
  y <- rnorm(1, mean = x[j - 1], sd = sigprop)

  alfa <- min(1, dnorm(y, 0, 1) / dnorm(x[j - 1], 0, 1))

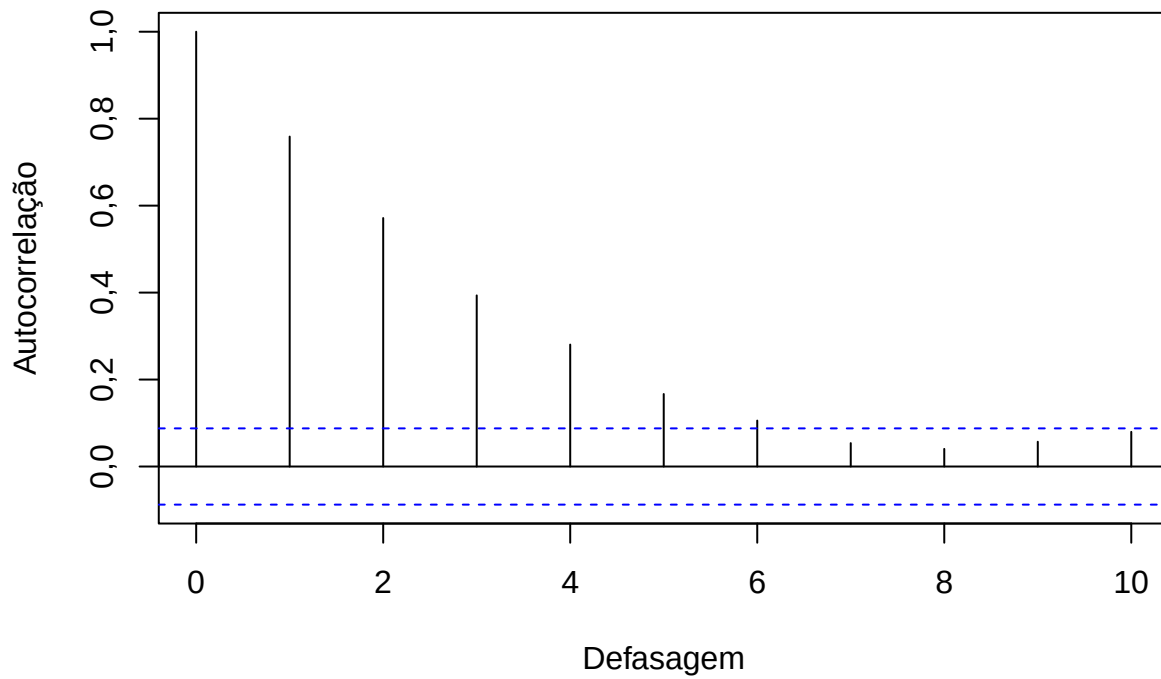
  # geração
  if (rbinom(1, 1, alfa) == 1) {
    x[j] <- y
  } else {
    x[j] <- x[j - 1]
  }
}
```

Em seguida as sequências geradas são apresentadas graficamente.

```
plot(x, type = "l", xlab = "Índice", ylab = "x", col = "blue",
     ylim = range(x, -2, 2))
abline(h = c(-2, 2), lty = 2)
```



```
acf(x, lag.max = 10, main = "", xlab = "Defasagem",
    ylab = "Autocorrelação")
```



Nota 1 Refaça os exemplos calculando a taxa de aceitação.