Exemplo de implementação computacional - Newton-Raphson

Modelo normal

AUTOR
Paulo Cerqueira Jr

AFILIAÇÕES
Faculdade de Estatística - FAEST
Universidade Federal do Pará - UFPA

Considerações iniciais

Seja $X_1, X_2, \dots, X_n$ uma amostra aleatória de uma distribuição normal $\mathcal{N}(\mu, \sigma^2)$, com função densidade de probabilidade:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

1. **Função de Verossimilhança**

A função de verossimilhança é o produto das densidades individuais:

$$L(\mu, \sigma^2) = \prod_{i=1}^{n} f(X_i \mid \mu, \sigma^2).$$

Substituindo $f(x \mid \mu, \sigma^2)$:

$$L(\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(X_i-\mu)^2}{2\sigma^2}\right).$$

Reescrevendo:

$$L(\mu, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\sum_{i=1}^{n} \frac{(X_i-\mu)^2}{2\sigma^2}\right).$$

2. **Log-Verossimilhança**

A log-verossimilhança é dada por:

$$\ell(\mu, \sigma^2) = \log L(\mu, \sigma^2).$$

Tomando o logaritmo:

$$\ell(\mu, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \sum_{i=1}^{n} \frac{(X_i-\mu)^2}{2\sigma^2}.$$

Reescrevendo:

$$\ell(\mu, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i-\mu)^2.$$

3. **Vetor Escore**

O vetor escore é o gradiente da log-verossimilhança, ou seja,

$$U(\mu, \sigma^2) = \left(\frac{\partial \ell}{\partial \mu}, \frac{\partial \ell}{\partial \sigma^2}\right).$$

Derivando em relação a μ:

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i - \mu).$$

Derivando em relação a σ^2:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(X_i - \mu)^2.$$

Assim, o vetor escore é:

$$U(\mu, \sigma^2) = \begin{bmatrix} \frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i - \mu) \\ -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(X_i - \mu)^2 \end{bmatrix}.$$

4. **Matriz Hessiana**

A matriz Hessiana é a matriz das segundas derivadas da log-verossimilhança:

$$H(\mu, \sigma^2) = \begin{bmatrix} \frac{\partial^2 \ell}{\partial \mu^2} & \frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} \\ \frac{\partial^2 \ell}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 \ell}{\partial (\sigma^2)^2} \end{bmatrix}.$$

Calculando as segundas derivadas:

$$\frac{\partial^2 \ell}{\partial \mu^2} = -\frac{n}{\sigma^2}.$$

$$\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} = \frac{\partial}{\partial \sigma^2}\left(\frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i - \mu)\right) = -\frac{1}{\sigma^4}\sum_{i=1}^{n}(X_i - \mu).$$

$$\frac{\partial^2 \ell}{\partial (\sigma^2)^2} = \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\sum_{i=1}^{n}(X_i - \mu)^2.$$

Assim, a matriz Hessiana é:

$$H(\mu, \sigma^2) = \begin{bmatrix} -\frac{n}{\sigma^2} & -\frac{1}{\sigma^4}\sum_{i=1}^{n}(X_i - \mu) \\ -\frac{1}{\sigma^4}\sum_{i=1}^{n}(X_i - \mu) & \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\sum_{i=1}^{n}(X_i - \mu)^2 \end{bmatrix}.$$

Implementação computacional

A implementação computacional do método de Newton-Raphson para o modelo normal. A função `U` representa a implementação do vetor escore:

```
## Vetor escore:

U <- function(theta, dados){
  U_mu    <- (sum(dados-theta[1])/theta[2])
  U_sigma <-  (-0.5*length(dados)/theta[2]) + sum((dados-theta[1])^2)/(2*(theta[2]^2)

  return(c(U_mu, U_sigma))
}
```

A matriz hessiana através da função `H`:

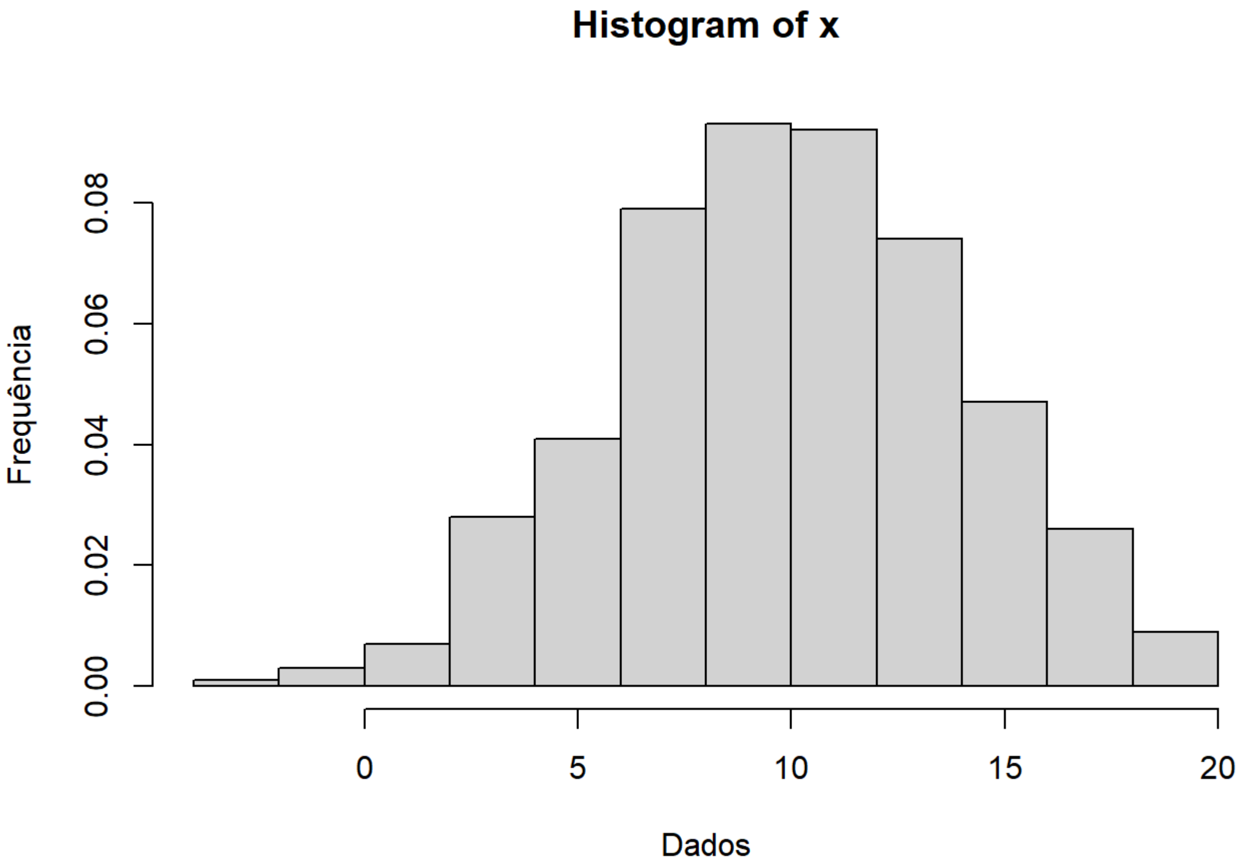
```
## Matriz Hessiana:
H <- function(theta, dados){
  m_2    <- -length(dados)/theta[2]
  sig_2  <- (0.5*length(dados)/(theta[2]^2)) - sum((dados-theta[1])^2)/(theta[2]^3)
  mu_sig <- -(1/(theta[2]^2))*sum(dados-theta[1])
  sig_mu <- -(1/(theta[2]^2))*sum(dados-theta[1])

  return(matrix(c(m_2, mu_sig, sig_mu, sig_2), nrow=2, ncol = 2))
}
```

Para a implementação, vamos gerar uma amostra de tamanho 100, com média

```
set.seed(1234567890) # Semente
n <- 500 # Tamanho amostral
x <- rnorm(n, mean = 10, sd = 4) # Amostra gerada
```

Os dados gerados são

```
hist(x, ylab="Frequência", xlab="Dados", freq = FALSE)
```



A implementação do método:

```
## Iniciando NR
theta0 <- c(5, 5) # Chute inicial
dif <- 1 #Diferença
erro <- 10^(-6) # Tolerancia
i <- 1 # contador

while( dif>erro ){

  H0 <- H(theta = theta0, dados = x)
  U0 <- U(theta = theta0, dados = x)
  prodHU <- solve(H0, U0)
  theta1 <- theta0 - prodHU
  dif <- max(abs(theta1-theta0))

  theta0 <- theta1
  i<- i+1
  cat("Iter:", i, "est:", theta1, "\n")
  #if(i==10)break
}
```

```
Iter: 2 est: 12.48741 2.51244
Iter: 3 est: 10.62756 3.14639
Iter: 4 est: 10.27176 4.509212
Iter: 5 est: 10.11327 6.389607
Iter: 6 est: 10.04269 8.801475
```

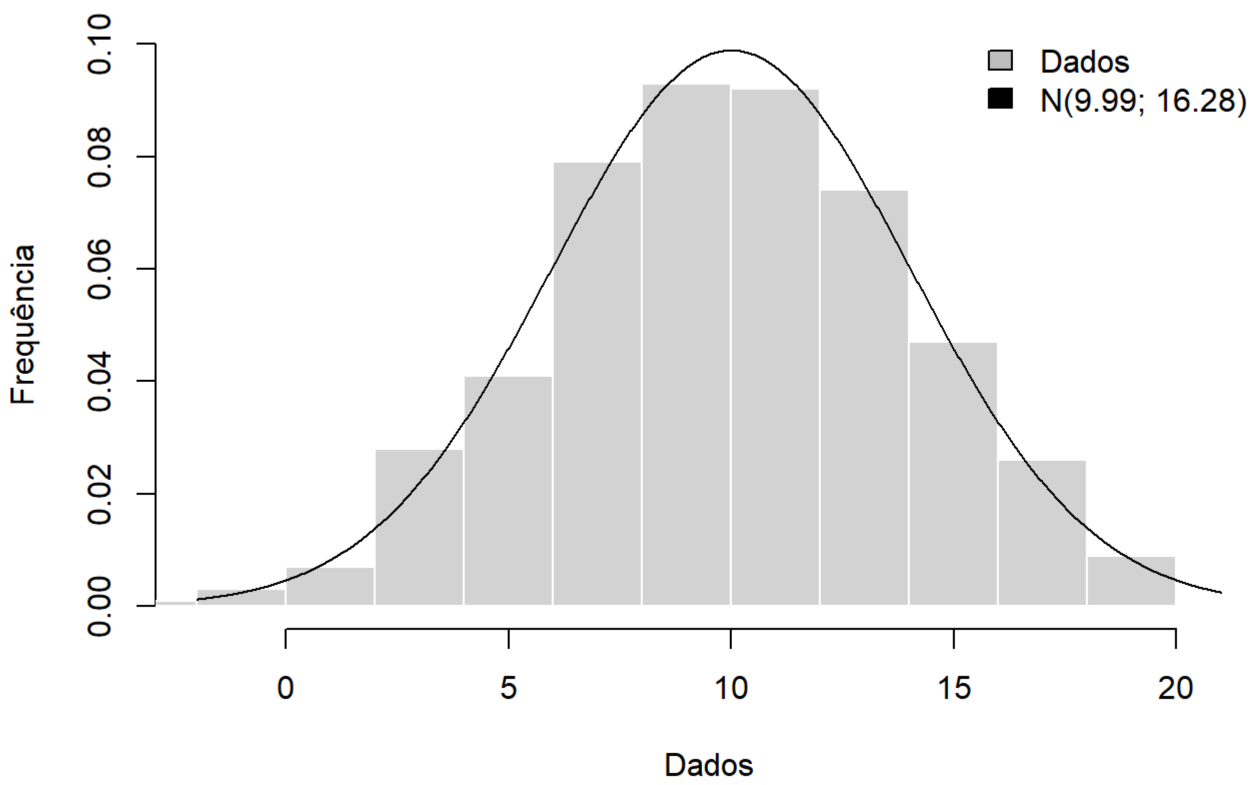
```
Iter: 7 est: 10.01337 11.57178
Iter: 8 est: 10.00292 14.16869
Iter: 9 est: 10.00025 15.79749
Iter: 10 est: 9.999909 16.25559
Iter: 11 est: 9.999899 16.28369
Iter: 12 est: 9.999899 16.28379
Iter: 13 est: 9.999899 16.28379
```

```r
## Estimativa de máxima verossimilhança final:

cat("mu:", theta1[1],"-", "sigma2:", theta1[2])
```

```
mu: 9.999899 - sigma2: 16.28379
```

A comparação dos dados com o modelo estimado:

```r
x.seq <- seq(-2, 21, length.out=1000)
hist(x, ylab="Frequência", xlab="Dados", freq = FALSE,
     main="", border="white", ylim = c(0,0.1), xlim=c(-2,21))
lines(x.seq, dnorm(x.seq, mean = theta1[1], sd=sqrt(theta1[2])))
legend("topright", c("Dados","N(9.99; 16.28)"),
       fill=c("gray","black"), col=c("gray","black"), bty="n")
```




```r
logNormal <- function(theta, dados){

  m <- theta[1] # média
  s <- theta[2] # variância
```

```r
  n <- length(dados)
  x <- dados

  l <-  -(n/2)*log(2*pi*s) - (1/(2*s))*sum((x-m)^2)
  return(-l)
}
```

Usando a função `optim`:

```r
fit <- optim(par = theta0, fn = logNormal, gr =NULL , method ="BFGS" ,
      hessian = TRUE, dados=x)
fit
```

```
$par
[1]  9.999899 16.283791

$value
[1] 1407.012

$counts
function gradient 
       5        1 

$convergence
[1] 0

$message
NULL

$hessian
         [,1]      [,2]
[1,] 30.70538 0.0000000
[2,]  0.00000 0.9428205
```