Algoritmo EM

Mistura de duas distribuições normais

2021

Os dados correspondem ao tempo de erupção (em min) do gêiser Old Faithful no Parque Nacional Yellowstone, EUA. Estes dados estão no pacote `datasets` da linguagem R. Mais informações podem ser obtidas com o comando `?faithful`.

```
# Dados
mydata <- faithful$eruptions
```

As funções densidade e distribuição acumulada do modelo de mistura de duas distribuições normais são apresentadas abaixo.

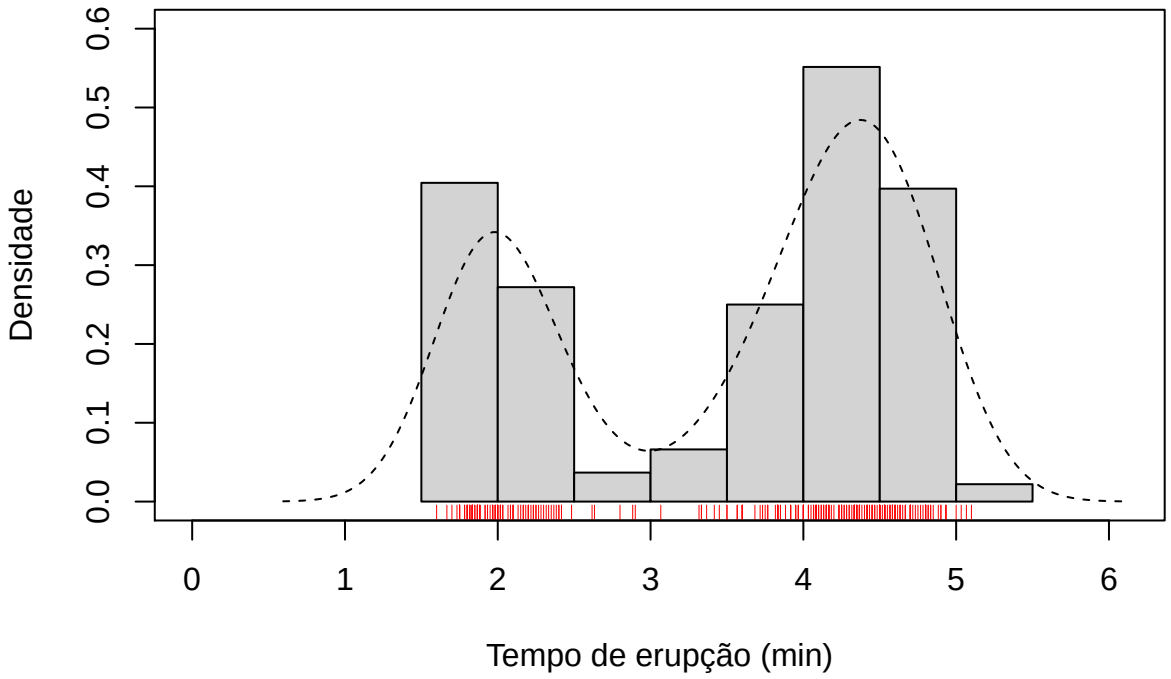
```
## Função densidade
d2norm <- function(x, alfa, mu1, sig1, mu2, sig2) {
   dens <- (1 - alfa) * dnorm(x, mu1, sig1) + alfa * dnorm(x, mu2, sig2)
   return(dens)
}
```

```
## Função distribuição acumulada
p2norm <- function(x, alfa, mu1, sig1, mu2, sig2) {
   fda <- (1 - alfa) * pnorm(x, mu1, sig1) + alfa * pnorm(x, mu2, sig2)
   return(fda)
}
```

```
n <- length(mydata)
cat("\n Tamanho da amostra:", n)
```

```
##
##  Tamanho da amostra: 272
```

```
hist(mydata, main = "", freq = FALSE, xlim = c(0, 1.2 * max(mydata)),
 xlab = "Tempo de erupção (min)", ylab = "Densidade", ylim = c(0, 0.6))
rug(mydata, col = "red")
lines(density(mydata), lty = 2)
box()
```




```
# Diferença relativa máxima (convergência)
eps <- 1e-4

# Iniciando ...
alfa0 <- 0.5
mu10 <- quantile(mydata, 0.25)
mu20 <- quantile(mydata, 0.75)
sig10 <- sig20 <- sd(mydata)
iter <- 0
dif <- 1  # dif > eps

cat("\n Algoritmo EM \n Tolerância:", eps)
```

```
##
##  Algoritmo EM
##  Tolerância: 1e-04
```

```
cat("\n Estimativas iniciais (alfa, mu1, sig1, mu2, sig2): \n",
    c(alfa0, mu10, sig10, mu20, sig20))
```

```
##
##  Estimativas iniciais (alfa, mu1, sig1, mu2, sig2):
##  0.5 2.16275 1.141371 4.45425 1.141371
```

No código abaixo os passos do algoritmo EM são executados até que

$$\text{dif} = \max\left(|\frac{\widehat{\alpha}^{(k+1)} - \widehat{\alpha}^{(k)}}{\widehat{\alpha}^{(k)}}|, |\frac{\widehat{\mu_1}^{(k+1)} - \widehat{\mu_1}^{(k)}}{\widehat{\mu_1}^{(k)}}|, |\frac{\widehat{\sigma_1}^{(k+1)} - \widehat{\sigma_1}^{(k)}}{\widehat{\sigma_1}^{(k)}}|, |\frac{\widehat{\mu_2}^{(k+1)} - \widehat{\mu_2}^{(k)}}{\widehat{\mu_2}^{(k)}}|, |\frac{\widehat{\sigma_2}^{(k+1)} - \widehat{\sigma_2}^{(k)}}{\widehat{\sigma_2}^{(k)}}| \right) \leq \text{eps}.$$

```
# Passos E e M
while (dif > eps) {
   iter <- iter + 1

   # Passo E
```

```
   gama <- alfa0 * dnorm(mydata, mu20, sig20) /
            d2norm(mydata, alfa0, mu10, sig10, mu20, sig20)
   # Passo M
   mu1 <- weighted.mean(mydata, w = 1 - gama)
   sig1 <- sqrt(weighted.mean((mydata - mu1)^2, w = 1 - gama))
   mu2 <- weighted.mean(mydata, w = gama)
   sig2 <- sqrt(weighted.mean((mydata - mu2)^2, w = gama))
   alfa <- mean(gama)

   # Critério de parada
   dif <- max(abs((alfa - alfa0) / alfa0), abs((mu1 - mu10) / mu10),
            abs((sig1 - sig10) / sig10), abs((mu2 - mu20) / mu20),
            abs((sig2 - sig20) / sig20))

   alfa0 <- alfa
   mu10 <- mu1
   sig10 <- sig1
   mu20 <- mu2
   sig20 <- sig2
}
```

Nota 1 Modifique o código acima para que o processo iterativo tenha um número máximo de iterações.

```
# Resulados
cat("\n Estimativas após", iter, "iterações")
```

```
##
##  Estimativas após 21 iterações
```

```
cat("\n alfa, mu1, sig1, mu2, sig2: \n",
    c(alfa, mu1, sig1, mu2, sig2))
```

```
##
##  alfa, mu1, sig1, mu2, sig2:
##  0.6515892 2.018622 0.2356447 4.273357 0.4370426
```

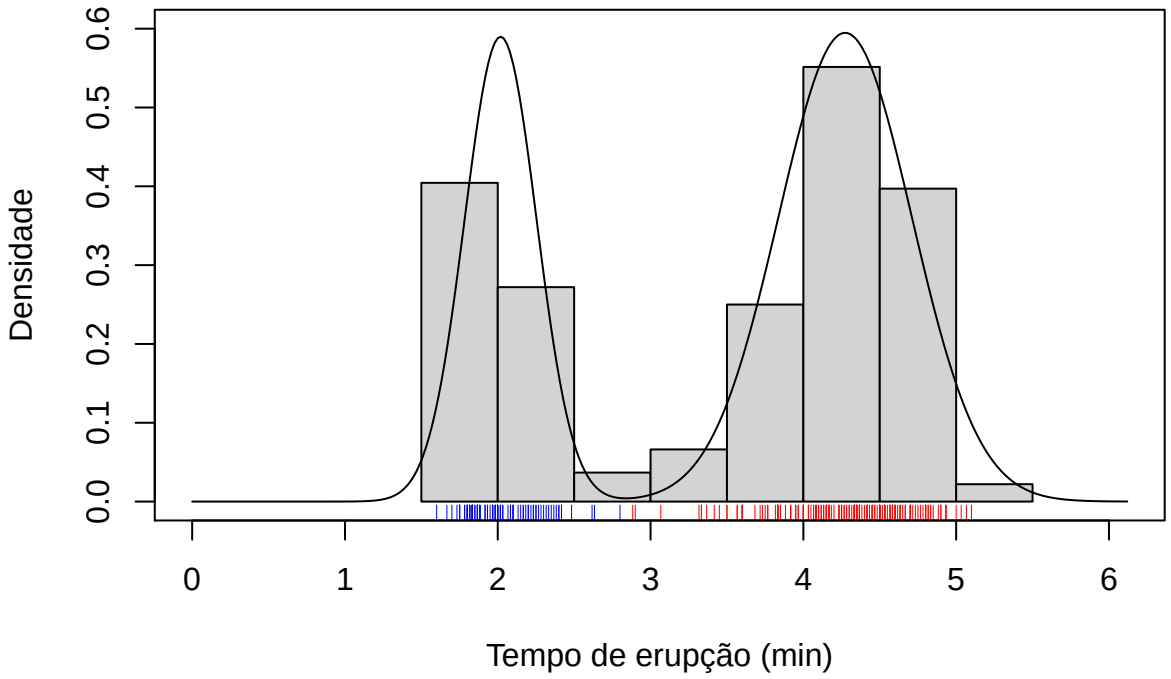
```
cat("\n Critério de parada:", dif)
```

```
##
##  Critério de parada: 6.790204e-05
```

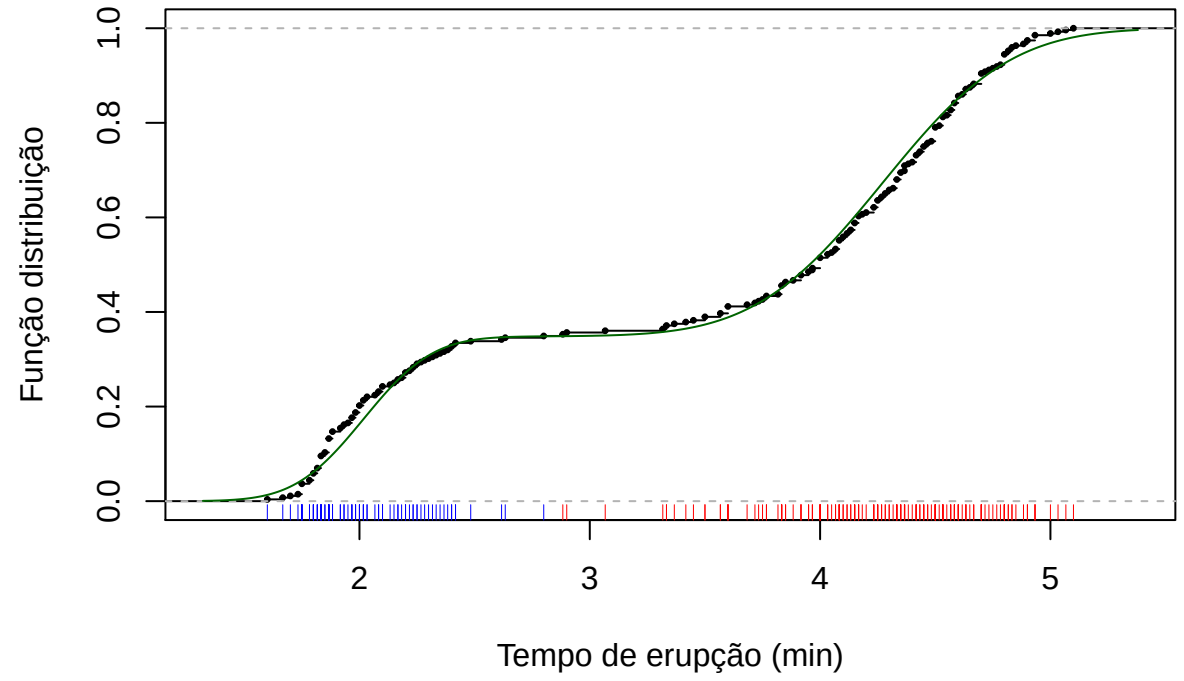
```
g1 <- which(gama < 0.5)
g2 <- which(gama >= 0.5)
cat("\n Número de obervações nos grupos 1 e 2:",
    c(length(g1), length(g2)))
```

```
##
##  Número de obervações nos grupos 1 e 2: 95 177
```

```
# Distribuição ajustada
hist(mydata, main = "", freq = FALSE, xlim = c(0, 1.2 * max(mydata)),
 xlab = "Tempo de erupção (min)", ylab = "Densidade", ylim = c(0, 0.6))
rug(mydata[g1], col = "blue")
rug(mydata[g2], col = "red")
box()
curve(d2norm(x, alfa, mu1, sig1, mu2, sig2), add = TRUE, n = 301)
```




```
plot(ecdf(mydata), pch = 20, main = "", ylab = "Função distribuição",
     xlab = "Tempo de erupção (min)", cex = 0.5)
curve(p2norm(x, alfa, mu1, sig1, mu2, sig2), add = TRUE,
      col = "darkgreen", n = 301)
rug(mydata[g1], col = "blue")
rug(mydata[g2], col = "red")
```



Nota 2 Escolha diferentes pontos iniciais e repita o processo de estimação dos parâmetros.