

Método escore de Fisher com a distribuição gama

```
## Escore de Fisher

# Função log-verossimilhança (a: alfa e b: beta)
logver <- function(teta) {
  a <- teta[1]
  b <- teta[2]
  n * a * log(b) - n * lgamma(a) + (a - 1) * T1 - b * T2
}
```

Nota 1. Reescreva a função log-verossimilhança utilizando a função `digamma`.
Verifique a parametrização adotada na linguagem R.

Em seguida são apresentadas a função escore e a inversa da informação de Fisher. As funções `digama` (ψ) e `trigama` (ψ') estão implementadas como `digamma` e `trigamma`.

```
# Escore
U <- function(teta) {
  a <- teta[1]
  b <- teta[2]
  c(n * log(b) - n * digamma(a) + T1, n * a / b - T2)
}

# Inversa da informação de Fisher
IF1 <- function(teta) {
  a <- teta[1]
  b <- teta[2]
  det1 <- (trigamma(a) * a - 1) / b^2
  matrix(c(a / b^2, 1 / b, 1 / b, trigamma(a)), ncol = 2) / (n * det1)
}

# Tolerância
eps <- 1e-6

# Dados e estatísticas suficientes
dados <- scan()
49.85 64.35 43.44 40.17 80.56 76.67 57.42 133.39
45.40 40.35 59.80 24.35 58.99 29.72 65.29 107.18
76.15 37.27 61.38 54.02 81.58 76.44 45.19 60.26
53.90 38.09 58.59 26.80 20.21 37.66 45.40 78.45
45.34 40.84 27.82 60.73 34.13 39.07 30.27 107.85
61.13 32.05 31.06 46.21 22.21 44.70 47.06

n <- length(dados)
cat("\n n =", n)

n = 47

T1 <- sum(log(dados))
T2 <- sum(dados)
```

```

# Estimativas iniciais (método dos momentos)
alfa0 <- mean(dados)^2 / var(dados)
beta0 <- mean(dados) / var(dados)
teta0 <- c(alfa0, beta0)

# Escore de Fisher
niter <- 0
dif <- 1
while (dif > eps) {
  niter <- niter + 1
  passo <- IF1(teta0) %*% U(teta0)
  while(logver(teta0 + passo) <= logver(teta0)) {
    passo <- 0.5 * passo
  }
  tetac <- teta0 + passo

  # Critério de parada
  dif <- max(abs(passo / teta0))
  teta0 <- tetac
}

```

Nota 2. Modifique a atualização da estimativa de θ (tetac) de modo a garantir que a estimativa pertença ao espaço paramétrico, lembrando que $\alpha > 0$ e $\beta > 0$.

Nota 3. Modifique o critério de parada de modo que o algoritmo seja encerrado após um certo número de iterações ter sido executado, mesmo que a diferença máxima (dif) ainda seja maior do que ϵ (eps).

```

cat("\n Estimativas iniciais (alfa, beta):", c(alfa0, beta0))
cat("\n Função log-verossimilhança:", logver(c(alfa0, beta0)))
cat("\n Tolerância:", eps)
cat("\n Número de iterações:", niter)
cat("\n Diferença relativa máxima:", dif)
cat("\n EMV:", tetac)
cat("\n Função log-verossimilhança:", logver(tetac))
cat("\n Escore:", U(tetac))
cat("\n Erros padrão da EMV:", sqrt(diag(IF1(tetac))))

```

```

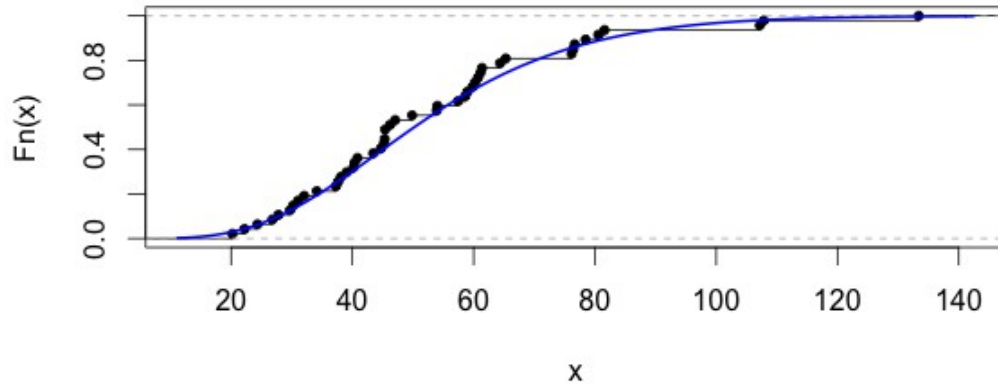
Estimativas iniciais (alfa, beta): 5.140039 0.09667952
Função log-verossimilhança: -209.0958
Tolerância: 1e-06
Número de iterações: 4
Diferença relativa máxima: 1.191503e-07
EMV: 5.930438 0.1115462
Função log-verossimilhança: -208.853
Escore: 8.526513e-14 -4.547474e-13
Erros padrão da EMV: 1.190536 0.02336842

```

Nota 4. Compare com os resultados obtidos com a função `fitdistr` do pacote MASS.

Nota 5. Refaça o exemplo partindo de diferentes estimativas iniciais.

```
# Distribuição empírica e modelo ajustado
plot(ecdf(dados), main = "", pch = 20)
curve(pgamma(x, shape = tetac[1], rate = tetac[2]), col = "blue",
      add = TRUE, lwd = 2)
```



Nota 6. Pode ser afirmado que a distribuição gama faz um bom ajuste dos dados?