

Estimação por máxima verossimilhança

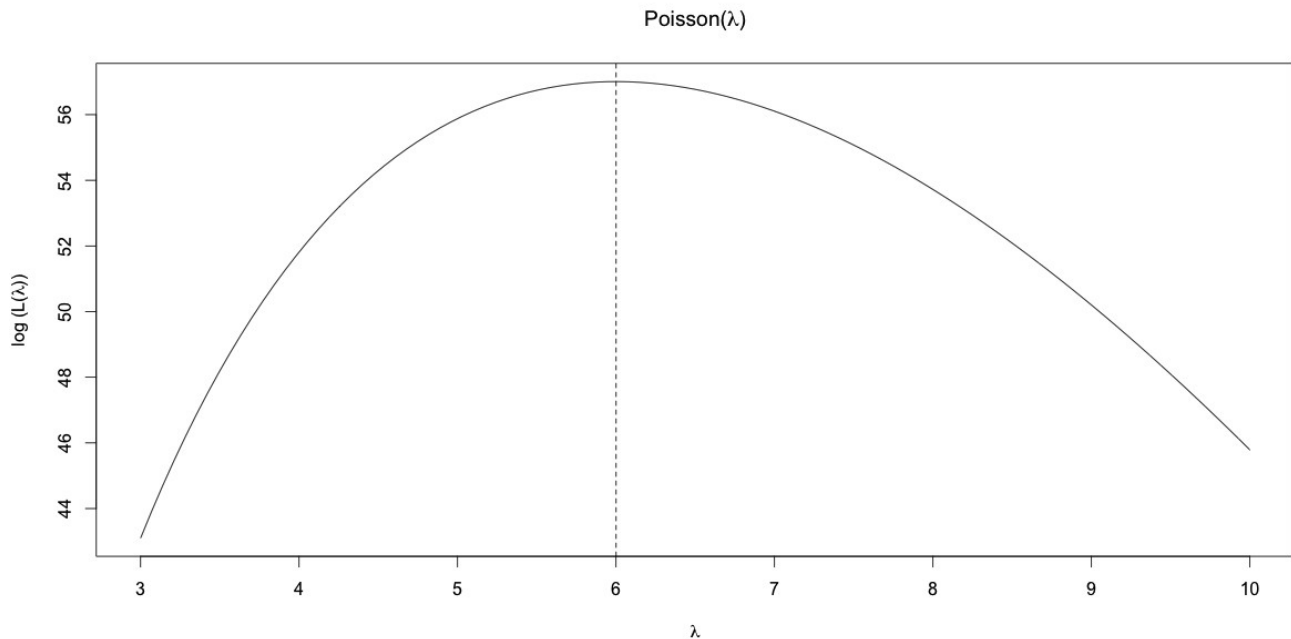
```
# Função fitdistr
library(MASS)

# 1. Poisson
logL <- function(teta, amostra) {
  n <- length(amostra)
  sx <- sum(amostra)
  return(-n * teta + sx * log(teta))
}

dados <- c(4, 4, 3, 3, 6, 6, 13, 3, 8, 7, 5, 10)
lambdac <- mean(dados) # EMV
cat("\n n = ", length(dados), "\n EMV de lambda = ", lambdac)

n = 12
EMV de lambda = 6

curve(logL(x, amostra = dados), 3, 10, xlab = expression(lambda),
      ylab = expression(paste('log (L(', lambda, '))')),
      main = expression(paste('Poisson(', lambda, '))'))
abline(v = lambdac, lty = 2)
```



```
fitdistr(dados, "poisson")
```

```
lambda
6.0000000
(0.7071068)
```

O número entre parênteses é uma estimativa do desvio padrão do estimador (erro padrão da estimativa).

Nota 1. Utilize o erro padrão para determinar os limites do eixo horizontal do gráfico.

2. Exponencial

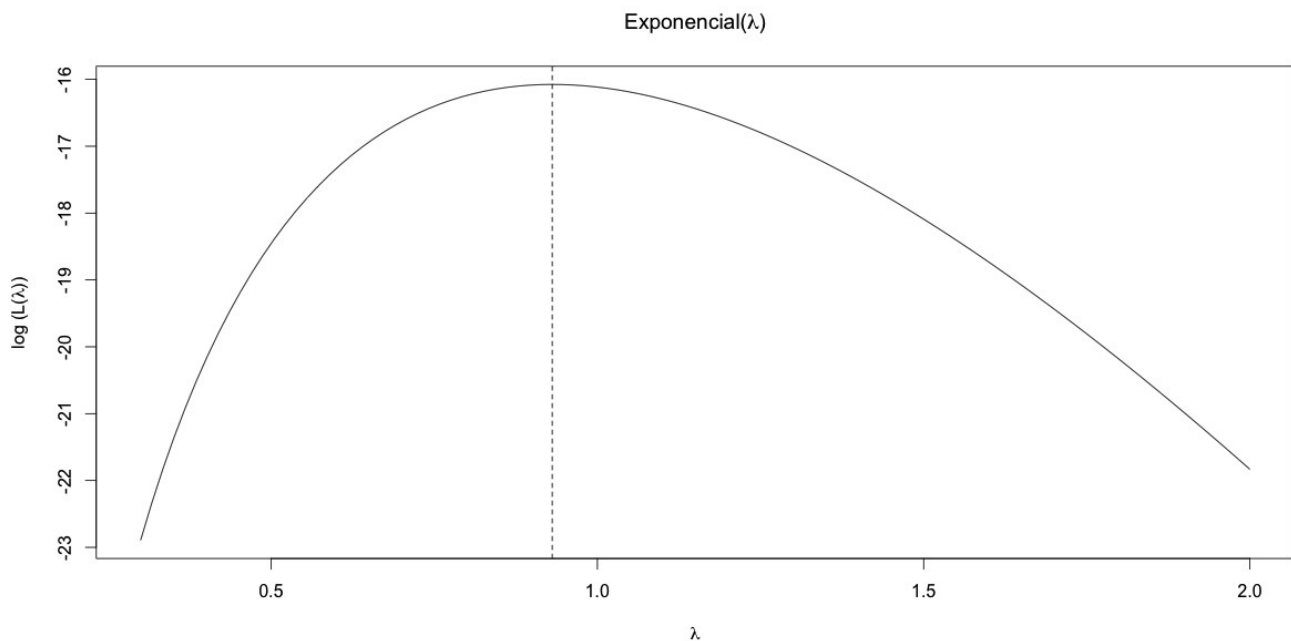
```
logL <- function(teta, amostra) {  
  n <- length(amostra)  
  sx <- sum(amostra)  
  return(n * log(teta) - sx * teta)  
}
```

```
dados <- c(0.792, 2.057, 0.631, 2.323, 0.103, 0.276, 2.081, 0.206, 0.462,  
          1.243, 0.304, 0.548, 2.909, 1.280, 0.900)  
lambdac <- 1 / mean(dados) # EMV  
cat("\n n = ", length(dados), "\n EMV de lambda = ", lambdac)
```

n = 15

EMV de lambda = 0.9308098

```
curve(logL(x, amostra = dados), 0.3, 2, xlab = expression(lambda),  
      ylab = expression(paste('log (L(', lambda, '))'),  
      main = expression(paste('Exponencial(', lambda, '))'),  
      abline(v = lambdac, lty = 2)
```



```
fitdistr(dados, "exponential")
```

```
rate  
0.9308098  
(0.2403341)
```

Nota 2. O parâmetro rate corresponde a λ , conforme pode ser visto com o comando ?dexp.

```

# 3. Normal (mu, sigma^2)
logL <- function(teta1, teta2, amostra) {
  n <- length(amostra)
  sx <- sum(amostra)
  sx2 <- sum(amostra^2)
  return(-0.5 * (n * log(teta2) + (sx2 - 2 * teta1 * sx + n *
    teta1^2) / teta2))
}

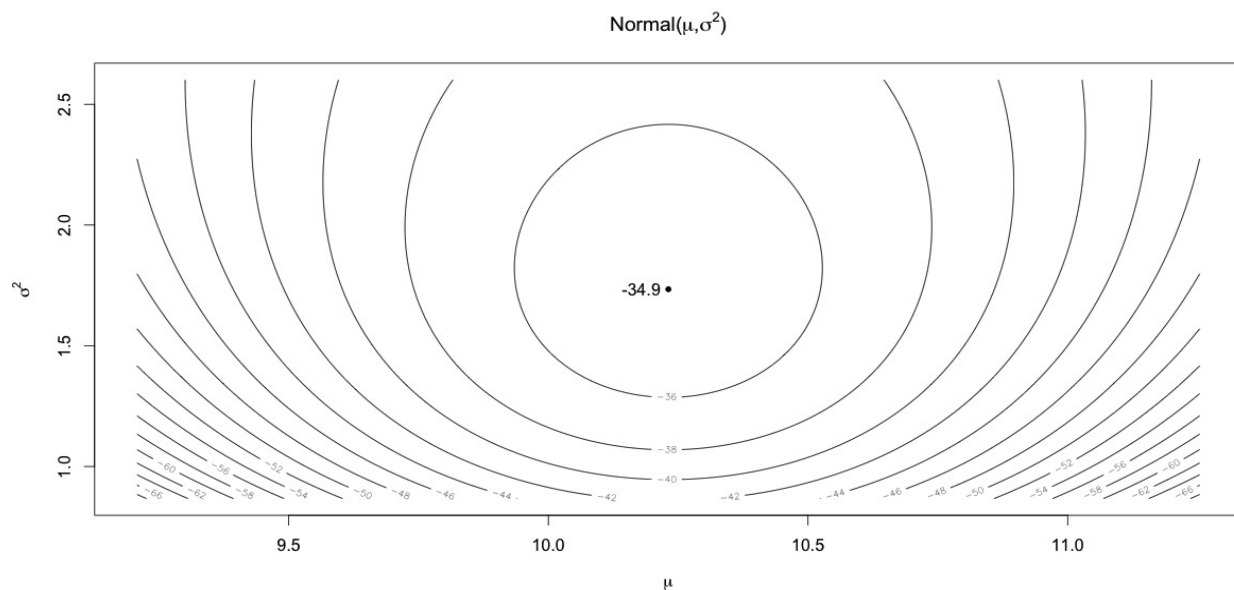
dados <- c(8.6, 9.0, 11.2, 8.7, 8.4, 11.1, 11.7, 12.2, 12.5, 10.0,
  9.8, 10.3, 10.6, 8.6, 10.7, 10.6, 9.5, 7.6, 9.6, 9.0, 12.3, 8.6,
  11.1, 10.0, 10.5, 12.0, 11.2, 10.0, 11.2, 11.7, 9.3, 7.9, 10.8,
  11.1, 10.0, 9.8, 9.3, 7.7, 10.1, 10.9, 11.5, 10.6, 9.7, 10.0, 13.4)

muc <- mean(dados) # EMV
sigma2c <- (n - 1) * var(dados) / n
cat("\n n = ", length(dados), "\n EMV de mu e sigma2 = ",
  c(muc, sigma2c))

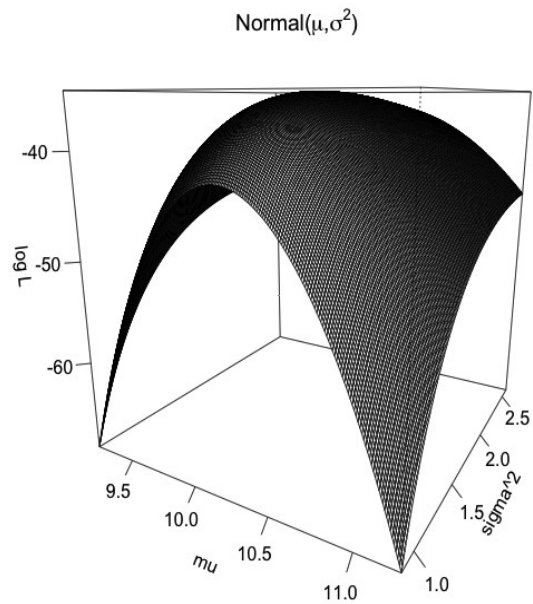
n = 45
EMV de mu e sigma2 = 10.23111 1.734143

# Contornos
mux <- seq(0.9 * muc, 1.1 * muc, length = 200)
sigma2y <- seq(0.5 * sigma2c, 1.5 * sigma2c, length = 200)
fxy <- function(x, y) logL(x, y, amostra = dados) # Função auxiliar
z <- outer(mux, sigma2y, fxy)
contour(mux, sigma2y, z, main =
  expression(paste('Normal(', mu, ',', sigma^2, ')')),
  xlab = expression(mu), ylab = expression(sigma^2), nlevels = 20)
points(muc, sigma2c, pch = 20)
text(muc, sigma2c, round(logL(muc, sigma2c, dados), 1), adj = 1.2)

```



```
# Tridimensional
persp(mux, sigma2y, z, ticktype = "detailed",
      main = expression(paste('Normal(', mu, ', ', ', sigma^2, ')')),
      theta = 30, xlab = "mu", ylab = "sigma^2", zlab = "log L")
```



```
fitdistr(dados, "normal")
```

```
      mean      sd
10.2311111  1.3168687
(0.1963072) (0.1388102)
```

Nota 3. Em R, os parâmetros da distribuição normal são a média ($\text{mean} = \mu$) e o desvio padrão ($\text{sd} = \sigma$), conforme pode ser visto com o comando `?dnorm`.