

Estatística Computacional

2020

1 Distribuição exponencial exponencializada

A distribuição exponencial exponencializada¹ tem função distribuição acumulada

$$F(x; \alpha, \lambda) = (1 - e^{-\lambda x})^\alpha, \text{ para } x > 0 \quad \text{e} \quad F(x; \alpha, \lambda) = 0, \text{ para } x \leq 0,$$

em que $\alpha > 0$ e $\lambda > 0$. Esta distribuição é utilizada como uma alternativa às distribuições gama e Weibull na modelagem de dados positivos.

1. A distribuição exponencial exponencializada generaliza a distribuição exponencial. Justifique.
2. Apresente um gerador de amostras aleatórias da distribuição exponencial exponencializada.
3. Os dados abaixo representam milhões de ciclos até a falha de rolamentos que compõem uma máquina.

17,88 28,92 33,00 41,52 42,12 45,60 48,80 51,84 51,96 54,12 55,56 67,80
68,64 68,64 68,88 84,12 93,12 98,64 105,12 105,84 127,92 128,04 173,40

Ajuste as distribuições exponencial, gama, Weibull e exponencial exponencializada a estes dados e apresente as estimativas dos parâmetros. Qual distribuição leva a um melhor ajuste?

4. Com relação ao item 3 e para a distribuição exponencial exponencializada, apresente estimativas dos erros padrão das estimativas dos parâmetros com base em resultados assintóticos e em métodos de reamostragem.

¹Gupta & Kundu (2001, Exponentiated exponential family: An alternative to gamma and Weibull distributions, *Biometrical Journal* 43, 117–130).

A Código em linguagem R

```
library(MASS)

# Função densidade - exponencial exponencializada
deexp <- function(x, shape, rate = 1) {
  elam <- exp(-rate * x)
  dens <- shape * rate * (1 - elam)^(shape - 1) * elam
  return(dens)
}

# Função distribuição acumulada - exponencial exponencializada
peexp <- function(x, shape, rate = 1) {
  cdf <- (1 - exp(-rate * x))^shape
  return(cdf)
}

# Amostras aleatórias - exponencial exponencializada
reexp <- function(n, shape, rate = 1) {
  raux <- runif(n)
  return(-log(1 - raux^(1 / shape)) / rate)
}

x <- scan()
17.88 28.92 33.00 41.52 42.12 45.60 48.80 51.84 51.96 54.12 55.56 67.80
68.64 68.64 68.88 84.12 93.12 98.64 105.12 105.84 127.92 128.04 173.40

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

## Ajustes
# Exponencial
(lam1 <- 1 / mean(x))

fitdistr(x, "exponential")

(mgam <- fitdistr(x, "gamma"))

(mwei <- fitdistr(x, "weibull"))
```

```

(meexp <- fitdistr(x, deexp, start = list(shape = mgam$estimate[1],
    rate = lam1)))

# Gráficos
hist(reexp(200, 0.5, 1))

mycolors <- rainbow(4)

plot(ecdf(x), main = "", xlab = "Número de ciclos (em milhões)",
    pch = 20, ylab = "Funções distribuição")
curve(pexp(x, rate = lam1), add = TRUE, col = mycolors[1])
curve(pgamma(x, shape = mgam$estimate[1], rate = mgam$estimate[2]),
    add = TRUE, col = mycolors[2])
curve(pweibull(x, shape = mwei$estimate[1], scale = mwei$estimate[2]),
    add = TRUE, col = mycolors[3])
curve(peexp(x, shape = meexp$estimate[1], rate = meexp$estimate[2]),
    add = TRUE, col = mycolors[4])
legend("bottomright", c("Exponencial", "Gama", "Weibull",
    "Exp. exponenc."), lty = 1, col = mycolors, bty = "n")

hist(x, freq = FALSE, main = "", xlab = "Número de ciclos (em milhões)",
    pch = 20, ylab = "Funções densidade")
curve(dexp(x, rate = lam1), add = TRUE, col = mycolors[1])
curve(dgamma(x, shape = mgam$estimate[1], rate = mgam$estimate[2]),
    add = TRUE, col = mycolors[2])
curve(dweibull(x, shape = mwei$estimate[1], scale = mwei$estimate[2]),
    add = TRUE, col = mycolors[3])
curve(deexp(x, shape = meexp$estimate[1], rate = meexp$estimate[2]),
    add = TRUE, col = mycolors[4], from = 0)
legend("topright", c("Exponencial", "Gama", "Weibull",
    "Exp. exponenc."), lty = 1, col = mycolors, bty = "n")
box()

# Bootstrap
B <- 1000
tetacs <- matrix(NA, B, 2)
for (b in 1:B) {
    xs <- sample(x, n, replace = TRUE)

```

```
meexps <- fitdistr(xs, deexp, start = list(shape = meexp$estimate[1],
  rate = meexp$estimate[2]), method = "Nelder-Mead")
tetacs[b, ] <- meexps$estimate
}
```

```
summary(tetacs)
```

```
cat("\n Erros padrão bootstrap:", apply(tetacs, 2, sd))
```