Bootstrap
(correção de viés e intervalos de confiança)

2026

Concordância

Apresentamos mais resultados sobre o coeficiente de correlação de concordância, complementando o exemplo do item "7. Erro padrão *bootstrap*.".

Concordância

Apresentamos mais resultados sobre o coeficiente de correlação de concordância, complementando o exemplo do item "7. Erro padrão *bootstrap*.".
Aplicando o método *bootstrap*, obtemos uma estimativa do coeficiente de correlação de concordância (ρ_c) com correção de viés e também intervalos de confiança para ρ_c.

Solução

```
## Bootstrap iterado
rca <- rc(x, y)  # Estimativa
set.seed(515)
B <- 5000  # Número de amostras (nível 1)
B2 <- 500  # Número de amostras (nível 2)
te <- c()  # tb**
```

Solução

```
i0 <- proc.time()[3]  # início
# Nível 1
rce <- c()
for (b in 1:B) {
   # Índices da amostra bootstrap
   inds <- sample(1:n, n, replace = TRUE)
   xs <- x[inds]  # x*
   ys <- y[inds]  # y*
   # Estimativa (*)
   rce[b] <- rc(xs, ys)
   # Nível 2
   rcee <- c()
   for (j in 1:B2) {
      # Estimativa (**)
      indss <- sample(inds, n, replace = TRUE)
      xss <- xs[indss]  # x**
      yss <- ys[indss]  # y**
      rcee[j] <- rc(xss, yss)
   }
   te[b] <- (rce[b] - rca) / sd(rcee)  # t**
}
cat("\n Tempo de processamento (min):", (proc.time()[3] - i0) / 60, "\n")
```

Solução

Com a função `proc.time` obtemos o tempo de processamento (em s) da parte iterada, que é menos de 1 min (em um computador com processador Apple M4 Pro, memória RAM de 24 GB e sistema operacional macOS 26.3.1).

```
Tempo de processamento (min): 0.7021833
```

Solução

```
# Distribuição de t**
dte <- density(te)
hist(te, freq = FALSE, main = "", xlab = expression(t^"**"),
     ylab = "Densidade", col = "lightgreen", ylim = c(0, max(dte$y)))
lines(dte, col = "blue", lty = 2, lwd = 2)
box()
```

Solução

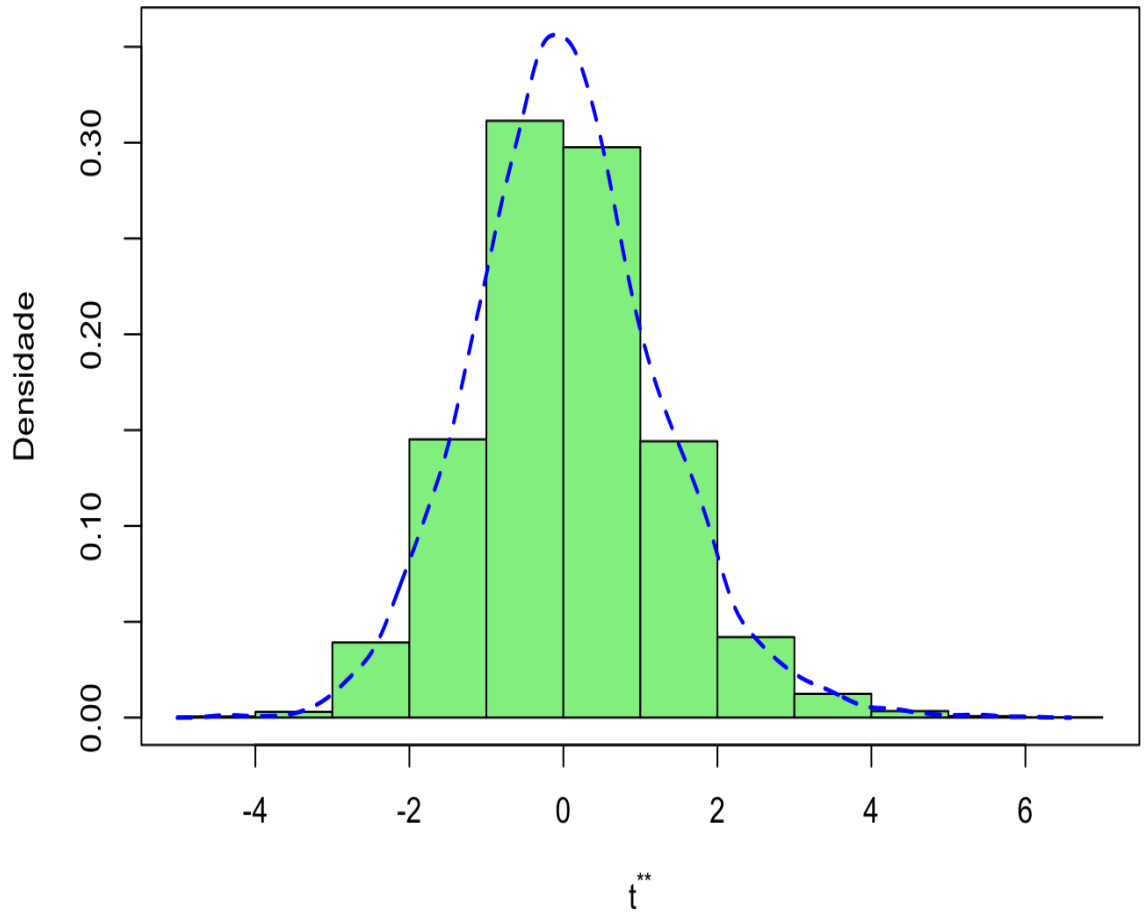

Figura 1: Distribuição de t^{**}.

Solução

```
## Resultados
epe <- sd(rce)  # Erro padrão bootstrap
rcae <- 2 * rca - mean(rce)  # Correção de viés
cat("\n B:", B, "\n e.p. bootstrap:", epe,
    "\n Estimativa:", rca,
    "\n Estimativa bootstrap:", mean(rce),
    "\n Estimativa com correção bootstrap:", rcae)
```

Solução

```
## Resultados
epe <- sd(rce)  # Erro padrão bootstrap
rcae <- 2 * rca - mean(rce)  # Correção de viés
cat("\n B:", B, "\n e.p. bootstrap:", epe,
    "\n Estimativa:", rca,
    "\n Estimativa bootstrap:", mean(rce),
    "\n Estimativa com correção bootstrap:", rcae)
```

```
 B: 5000
 e.p. bootstrap: 0.1249638
 Estimativa: 0.308384
 Estimativa bootstrap: 0.3039768
 Estimativa com correção bootstrap: 0.3127912
```

Solução

```
# Intervalos de confiança
conf <- 0.95
problu <- c(1 - conf, 1 + conf) / 2
cat("\n Intervalo percentil de", 100 * conf, "%: [",
    quantile(rce, probs = problu), "]")
```

Solução

```
# Intervalos de confiança
conf <- 0.95
problu <- c(1 - conf, 1 + conf) / 2
cat("\n Intervalo percentil de", 100 * conf, "%: [",
    quantile(rce, probs = problu), "]")
```

```
 Intervalo percentil de 95 %: [ 0.05052794 0.5364185 ]
```

Solução

```
# Intervalos de confiança
conf <- 0.95
problu <- c(1 - conf, 1 + conf) / 2
cat("\n Intervalo percentil de", 100 * conf, "%: [",
    quantile(rce, probs = problu), "]")
```

```
 Intervalo percentil de 95 %: [ 0.05052794 0.5364185 ]
```

```
q12 <- quantile(te, probs = problu)
ictb <- rca - q12[2:1] * epe
cat("\n B2:", B2, "\n Intervalo t bootstrap de", 100 * conf,
    "%: [", ictb, "]")
```

Solução

```
# Intervalos de confiança
conf <- 0.95
problu <- c(1 - conf, 1 + conf) / 2
cat("\n Intervalo percentil de", 100 * conf, "%: [",
    quantile(rce, probs = problu), "]")
```

```
 Intervalo percentil de 95 %: [ 0.05052794 0.5364185 ]
```

```
q12 <- quantile(te, probs = problu)
ictb <- rca - q12[2:1] * epe
cat("\n B2:", B2, "\n Intervalo t bootstrap de", 100 * conf,
    "%: [", ictb, "]")
```

```
 B2: 500
 Intervalo t bootstrap iterado de 95 %: [ -0.02806748 0.5868965 ]
```

Solução

O intervalo *t bootstrap* iterado não é simétrico em relação à estimativa r_c, pois a distribuição de t^{**} não é simétrica em relação a 0.

Solução

O intervalo *t bootstrap* iterado não é simétrico em relação à estimativa r_c, pois a distribuição de t^{**} não é simétrica em relação a 0.

```
print(q12)
```

```
     2.5%     97.5%
-2.228745  2.692391
```

Solução

```
# Distribuição e intervalo t bootstrap iterado
par(mai = c(1.2, 1.2, 0.1, 0.3))
hist(rce, freq = FALSE, main = "", xlab = expression(r[c]^"*"),
     ylab = "Densidade", cex.axis = 1.5, cex.lab = 1.5)
lines(density(rce), col = "blue", lwd = 2)
arrows(ictb[1], 0, ictb[2], 0, col = "red", code = 3, angle = 90,
       lwd = 2, length = 0.1)
points(rcae, 0, pch = 16, col = "green", cex = 1.5)
box()
```

Solução

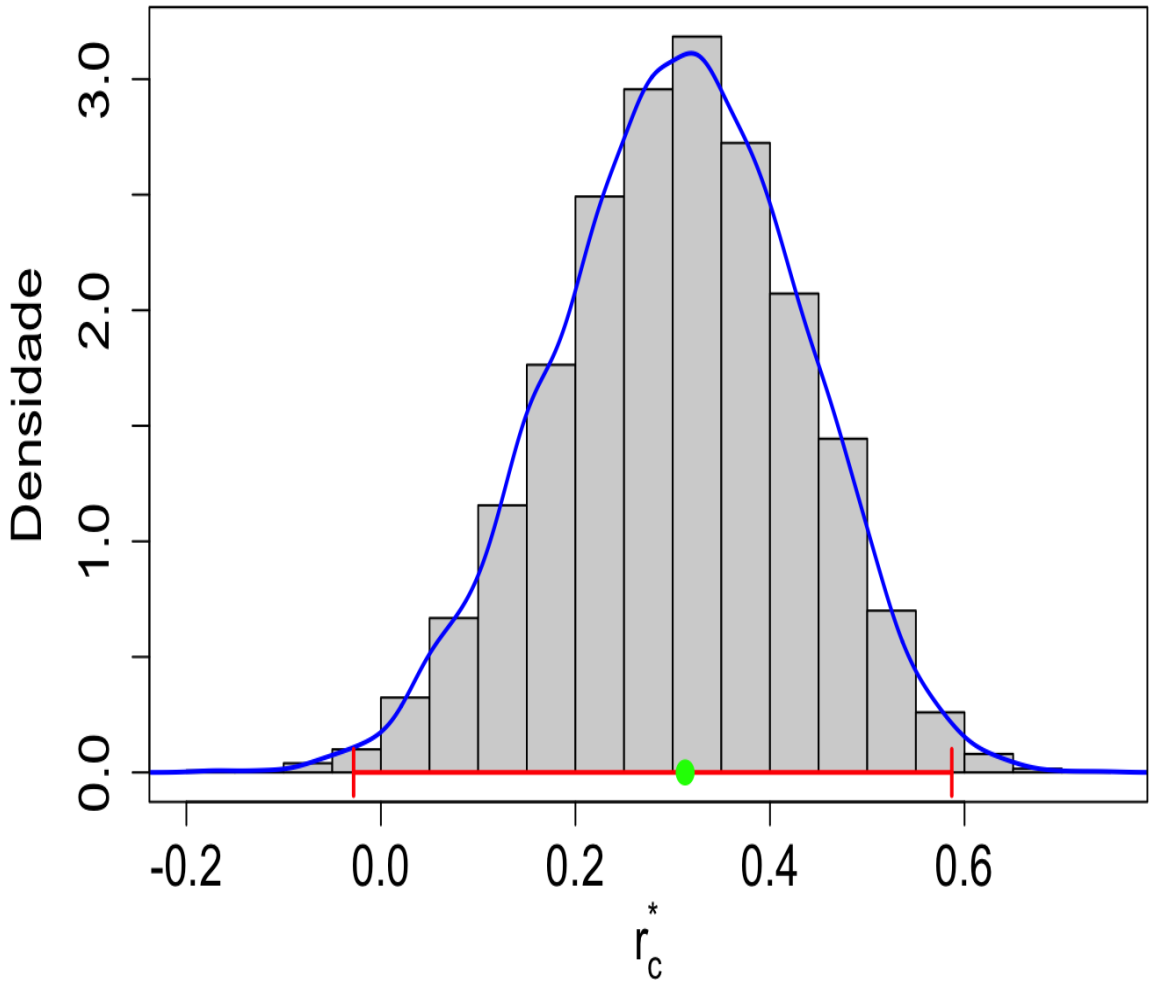

Figura 2: Histograma, estimativa da função densidade de r_c^*, estimativa corrigida e intervalo de confiança *t bootstrap* iterado de 95% para ρ_c.

Nota

1. Apresente o intervalo de confiança *t bootstrap* de 95% para ρ_c.