

Trabalho de Estatística Não Paramétrica

Carlos Henrique Santos, Ítalo Ferreira, Luís Israel, Moira Ellen e Rafael Hilário

07/06/18

```
library('animation')
library('quantreg')
library('Rfit')
```

Criando a função bootstrap

```
# Função que realiza o bootstrap e teste de hipóteses
# H0: b1 == 0 vs H1: b1 != 0, dado dois grupos de Y,
# o tipo da regressão e o número de loops

bootstrap <- function(a, b, tipo, nloops) {

  # Cria um vetor com todos os Y e outro com variáveis dummy para X

  y <- c(a, b)
  x <- c(rep(0, length(a)), rep(1, length(b)))

  # Decide qual tipo de regressão aplicar

  if (tipo == 'lad') {regressao <- rq(y ~ x)}
  else {regressao <- rfit(y ~ x)}

  # Faz a primeira regressão e salva b0, b1 e os resíduos originais

  b0 <- regressao$coefficients[1]
  b1_original <- regressao$coefficients[2]
  residuos <- regressao$residuals

  # Determina a fórmula da regressão a ser repetida no bootstrap, ou seja,
# (Y' + resíduos reamostrados) em função da variável regressora x

  formula <- (b0 + sample(residuos, length(residuos), replace = TRUE)) ~ x

  # Replica a regressão adequada com a fórmula nloops vezes e salva um vetor
# com todos os b1

  if (tipo == 'lad') {b1 <- replicate(nloops, rq(formula)$coefficients[2])}
  else {b1 <- replicate(nloops, rfit(formula)$coefficients[2])}

  # Encontra os quantis 2,5% e 97,5% do vetor de b1s

  ic95 <- quantile(b1, c(0.025, 0.975))

  # Calcula o p-valor
```

```

if (b1_original == 0) {p_valor <- 1}

else {
  maior_que_b1 <- b1[b1 >= abs(b1_original)]
  menor_que_b1 <- b1[b1 <= (-1)*abs(b1_original)]
  p_valor <- (length(maior_que_b1) + length(menor_que_b1))/length(b1)
}

# Plota o histograma dos b1s com os respectivos percentis e o b1 original,
# além do p-valor e uma curva de densidade normal

hist(b1, freq = F, col = 'gray', border = 'Lavender', xlim = c(-6, 6), ylim = c(0, 0.5),
     main = 'Histograma de B1', xlab = 'B1', ylab = 'Frequência relativa')
abline(v = c(ic95, b1_original), lty = c(2, 2, 1), col = c("red","red","black"))
curve(dnorm(x, mean = mean(b1), sd = sqrt(var(b1))),
      col = 'darkblue', lwd = 2, add = TRUE, yaxt = "n")
legend('topright', paste('p-valor =', p_valor))

# Conclui o teste e printa o resultado

if (b1_original < ic95[2] & b1_original > ic95[1]) {
  cat(paste('0 p-valor', p_valor, 'é maior que 0.05, portanto',
            'não rejeito H0 a nível de significância de 5%.'))
}
else {
  cat(paste('0 p-valor', p_valor, 'é menor que 0.05, portanto',
            'rejeito H0 a nível de significância de 5%.'))
}
}

```

Executando a função

```

# Grupos A e B

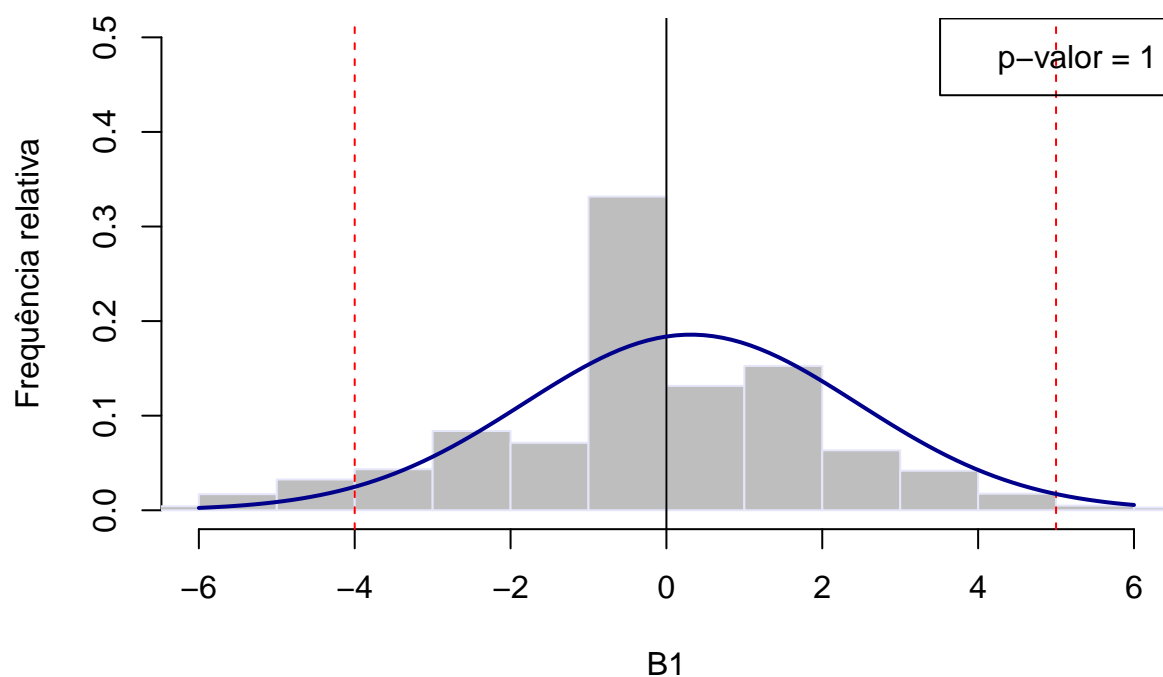
l1 <- c(14,16,21,14,18,24,16,14,21,23,13,18)
l2 <- c(18,14,19,11,23,21,10,13,19,24,15,20)

# Executa as funções

bootstrap(l1, l2, 'lad', 10000)

```

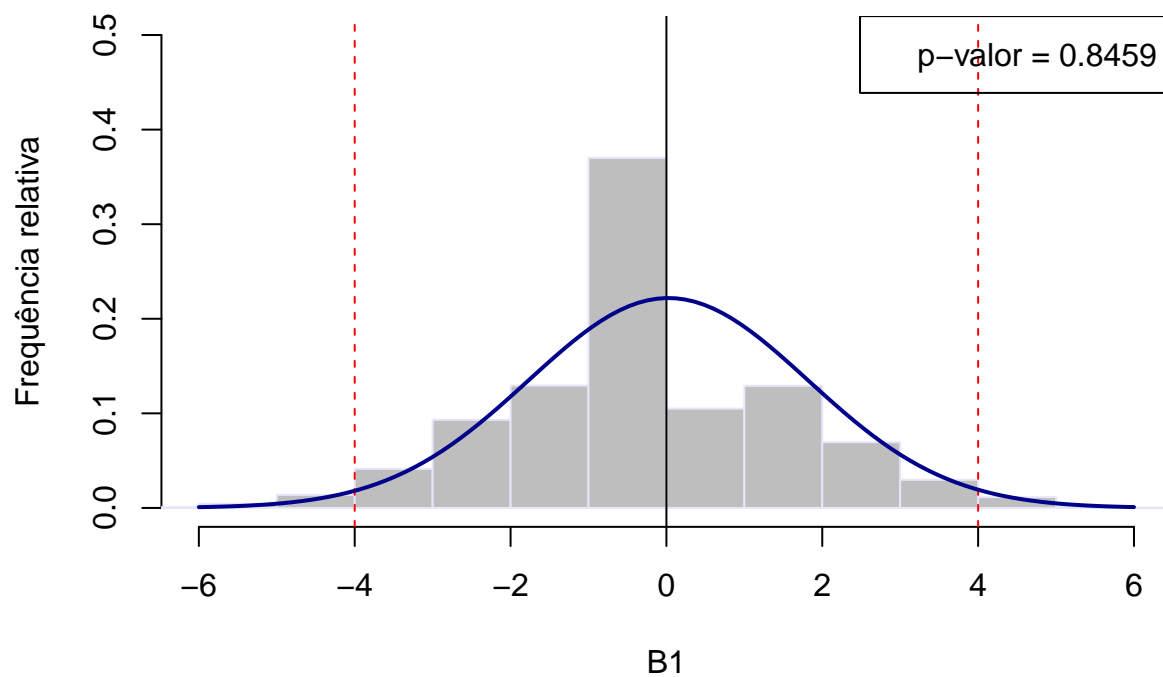
Histograma de B1



O p-valor 1 é maior que 0.05, portanto não rejeito H_0 a nível de significância de 5%.

```
bootstrap(l1, l2, 'rnp', 10000)
```

Histograma de B1



O p-valor 0.8459 é maior que 0.05, portanto não rejeito H_0 a nível de significância de 5%.

GIF

```
# Salva uma imagem GIF com a animação dos bootstraps:

saveGIF({for (i in seq(1, 100)) {bootstrap(l1, l2, 'lad', i)}},
        interval = 1, ani.width = 1000, ani.height = 1000)
saveGIF({for (i in seq(1, 100)) {bootstrap(l1, l2, 'rnp', i)}},
        interval = 1, ani.width = 1000, ani.height = 1000)
```