

SME0806 - Estatística Computacional - Trabalho 2

Matheus Victal Cerqueira

17/06/2021

Alunos:

- Aline Fernanda da Conceição, 9437275
- Diego J Talarico Ferreira, 3166561
- Matheus Victal Cerqueira, 10276661
- Murilo Henrique Soave, 10688813
- Nelson Calsolari Neto, 10277022

Docente: Professor Dr. Mário de Castro

Introdução

O presente documento se trata de uma solução para os exercícios propostos no Trabalho 2 da disciplina SME0806 - Estatística Computacional, oferecida pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo no primeiro semestre de 2021. As temáticas abordadas são métodos computacionais de reamostragem para a obtenção de estimativas pontuais e intervalares.

Desenvolvimento e Metodologia

Exercício 1

No presente exercício, quer-se analisar o coeficiente de Gini para o Produto Interno Bruto (PIB) *per capita* dos municípios do estado de São Paulo. Para tal análise, foi considerado que os valores observados para tais municípios correspondiam à uma amostra de observações e em tal amostra performou-se um processo de reamostragem baseado em *bootstrap* para obter-se estimativas pontuais e intervalares para o coeficiente de desigualdade. O sistema de referência utilizado foi o de uma pesquisa de PIB municipal realizada pelo Instituto Brasileiro de Geografia e Estatística - IBGE; Fundação Seade em 2018. A função para obter o coeficiente de Gini foi implementada em R e encontra-se abaixo.

```
# Coeficiente de Gini
Gini <- function(v){ # função recebe vetor de interesse v
  mu <- mean(v)
  n <- length(v)
  m_sub <- outer(v,v, FUN = '-') # obtém uma matriz com todas as subtrações
#possíveis entre os termos de v (xi-xj); i,j = {1,2,...,n}
  mod_m_sub <- abs(m_sub) # obtém a matriz com o módulo das subtrações
  somatorio <- sum(mod_m_sub) # obtém o somatório dos elementos de mod_m_sub
  gini <- somatorio/(2*n^2*mu) # cálculo do coeficiente de Gini
  return(gini)
}
```

Bootstrap iterado

```
data <- read.csv("dadosIBGE_f.csv",
                header = TRUE,
                encoding = "UTF-8") # dados dos municípios de SP, IBGE

data <- data.frame(data)

PIBpc <- data$PIBpc # valores para todos os municípios

## Bootstrap iterado

set.seed(2112)
n <- length(PIBpc) # número de observações (municípios: 645)
G <- Gini(PIBpc) # estimativa pontual do coef de Gini
B1 <- 100 # Número de amostras bootstrap no primeiro nível
B2 <- 10 # Número de amostras bootstrap no segundo nível
t <- c() # vetor vazio para receber os valores de t* (bootstrap)

system.time({ # Obter tempo de compilação do segmento
  # Nível 1 (N1)
  ge1 <- c() #vetor para valores do Gini estimado para cada amostra bootstrap(N1)

  for(b1 in 1:B1){
    # Amostra bootstrap (nível 1)
    ab1 <- sample(PIBpc, n, replace = TRUE)
    # Estimativa bootstrap (*)
    ge1[b1] <- Gini(ab1)

    # Nível 2 (N2)
    ge2 <- c() #recebe valores do Gini estimado para cada amostra bootstrap(N2)

    for(b2 in 1:B2){
      # Amostra bootstrap (nível 2)
      ab2 <- sample(ab1, n, replace = TRUE)
      # Estimativa bootstrap (**)
      ge2[b2] <- Gini(ab2)
    }

    t[b1] <- (ge1[b1] - G)/sd(ge2)
  }
})

##      user  system elapsed
##    13.02    5.51    19.39
```

Obtenção de estimativas pontuais e intervalares

```
# Estimativas pontuais
epb <- sd(ge1) # erro padrão bootstrap
```

```
gb_cvies <- 2 * G - mean(ge1) # estimativa bootstrap com correção de viés
gb_vies <- mean(ge1)
```

```
## Erro padrão bootstrap: 0.01898816
```

```
## Estimativa direta: 0.3262675
```

```
## Estimativa Bootstrap (sem correção de viés): 0.3208676
```

```
## Estimativa Bootstrap (com correção de viés): 0.3316673
```

```
# Estimativas intervalares
conf <- .95 # nível de confiança de 95%
coefq <- c(1-conf,1+conf)/2 # coeficientes de probabilidade dos quantis
cat("Intervalo percentil: IC[G, ", 100*conf, "%] = [",
    quantile(ge1, probs = coefq, type = 6), "]" )
```

```
## Intervalo percentil: IC[G, 95 %] = [ 0.2863556 0.3603914 ]
```

```
qs12 <- quantile(t, probs = coefq, type = 6) # quantis da distribuição de t*
ictb <- G - qs12[2:1]*epb # intervalo de confiança usando q12
cat("Intervalo t* (t bootstrap): IC[G, ", 100*conf, "%] = [", ictb, "]" )
```

```
## Intervalo t* (t bootstrap): IC[G, 95 %] = [ 0.2915509 0.3975486 ]
```

Exercício 2

```
cor(data[,c(2,3,4,5,6)])
```

```
##               Agropecuaria  Industria    Adm_pub  T_ex_adm_pub      Total
## Agropecuaria  1.0000000000  0.01031106  0.006243629 -0.005827323 -0.0006927772
## Industria     0.0103110586  1.00000000  0.921608723  0.899280372  0.9220296264
## Adm_pub       0.0062436293  0.92160872  1.000000000  0.992062090  0.9951054886
## T_ex_adm_pub -0.0058273227  0.89928037  0.992062090  1.000000000  0.9984369962
## Total        -0.0006927772  0.92202963  0.995105489  0.998436996  1.0000000000
```

```
x <- data$T_ex_adm_pub
y <- data$Total

set.seed(2112)
n <- length(data$Municipios) # número de observações (municípios: 645)
pearson <- cor(x,y)
B1 <- 1000 # Número de amostras bootstrap no primeiro nível
B2 <- 100  # Número de amostras bootstrap no segundo nível
t <- c() # vetor vazio para receber os valores de t* (bootstrap)

system.time({ # Obter tempo de compilação do segmento
  # Nível 1 (N1)
```

```

pe1 <- c() #vetor para valores da cor estimada para cada amostra bootstrap(N1)

for(b1 in 1:B1){
  # Amostra bootstrap dos indices (nível 1)
  indic1 <- sample(n, n, replace = TRUE)
  # Estimativa bootstrap (*)
  pe1[b1] <- cor(x[indic1],y[indic1])

  # Nível 2 (N2)
  pe2 <- c() #recebe valores da cor estimada para cada amostra bootstrap(N2)

  for(b2 in 1:B2){
    # Amostra bootstrap dos indices (nível 2)
    indic2 <- sample(indic1, n, replace = TRUE)
    # Estimativa bootstrap (**)
    pe2[b2] <- cor(x[indic2],y[indic2])
  }

  t[b1] <- (pe1[b1] - pearson)/sd(pe2)
}
})

```

```

##      user  system elapsed
##  30.92    0.10   32.57

```

```

# Estimativas pontuais
epb <- sd(pe1) # erro padrão bootstrap
pb_vies <- mean(pe1)

```

```
## Erro padrão bootstrap: 0.009360919
```

```
## Estimativa direta: 0.998437
```

```
## Estimativa Bootstrap (sem correção de viés): 0.9923042
```

```

# Estimativas intervalares
conf <- .95 # nível de confiança de 95%
coefq <- c(1-conf,1+conf)/2 # coeficientes de probabilidade dos quantis
cat("Intervalo percentil: IC[pearson,",100*conf,"%] = [",
    quantile(pe1, probs = coefq, type = 6),"]" )

```

```
## Intervalo percentil: IC[pearson, 95 %] = [ 0.971612 0.9996202 ]
```

```

qs12 <- quantile(t, probs = coefq, type = 6) # quantis da distribuição de t*
ictb <- pearson - qs12[2:1]*epb # intervalo de confiança usando qs12
cat("Intervalo t* (t bootstrap): IC[pearson,",100*conf,"%] = [",ictb,"]" )

```

```
## Intervalo t* (t bootstrap): IC[pearson, 95 %] = [ 0.993954 1.044837 ]
```