Técnicas de Amostragem - Simulação

Gabriella de Oliveira Argenton 255677 Gabriela Namie Hidaka 204274 Pedro Montei

```r
set.seed(123)

#### 1) Gerar uma população "oculta" com heterogeneidade ####

N <- 5000  # tamanho da população

# "Conectividade" dos indivíduos (análoga ao b_j do modelo de Rasch)
b <- rnorm(N, mean = 0, sd = 1)

# Variável de interesse Y (ex: número de clientes, gasto, etc.)
# Aqui fazemos Y correlacionada com b, só para ficar mais realista
Y <- 10 + 2*b + rnorm(N, mean = 0, sd = 3)

# Verdadeira média populacional (para comparar depois)
mu_true <- mean(Y)
mu_true
```

```
[1] 9.986339
```

```r
#### 2) Definir probabilidades de inclusão heterogêneas ####

# Parâmetros da "função logística" (simplificação do Rasch)
alpha <- -2          # nível geral de inclusão
beta  <-  1          # quanto a conectividade b_j influencia a inclusão

# Probabilidade de inclusão de cada indivíduo
pi_vec <- 1 / (1 + exp(-(alpha + beta * b)))

summary(pi_vec)      # só pra ver a distribuição das probabilidades
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.005839 0.065673 0.118410 0.154968 0.207672 0.809381
```

```r
mean(pi_vec) * N     # tamanho médio de amostra esperado
```

```
[1] 774.8422
```

```r
#### 3) Desenhar UMA amostra da população ####

# Cada indivíduo entra na amostra com probabilidade pi_j (amostragem Bernoulli)
S <- rbinom(N, size = 1, prob = pi_vec)  # 1 = na amostra, 0 = fora
sum(S)                                   # tamanho da amostra obtida
```

```
[1] 773
```

```r
Y_s  <- Y[S == 1]
pi_s <- pi_vec[S == 1]

length(Y_s)
```

```
[1] 773
```

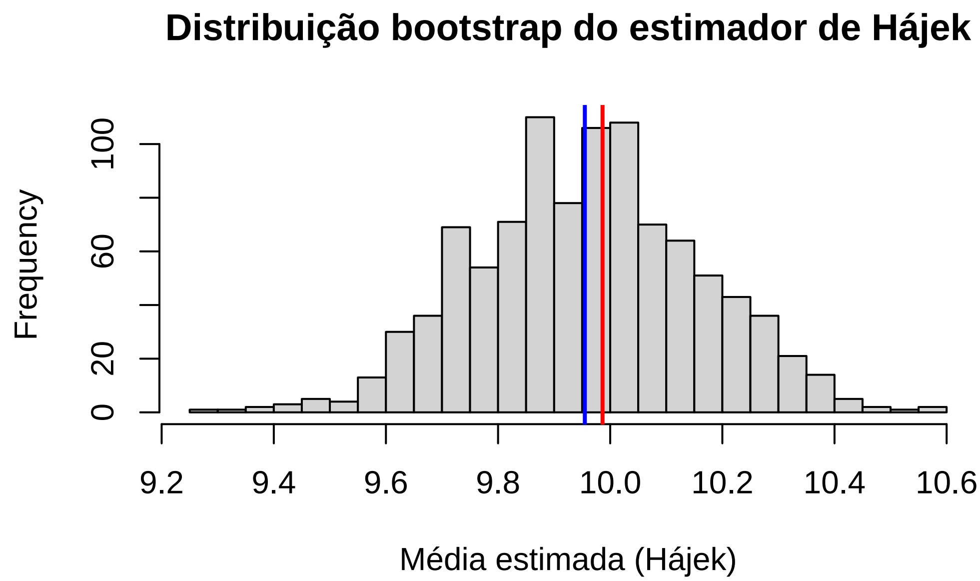
```r
#### 4) Três estimadores da média ####

# 1) Média ingênua (sem pesos)
mean_naive <- mean(Y_s)

# 2) Estimador tipo Horvitz-Thompson para o total
T_HT <- sum(Y_s / pi_s)
N_hat_HT <- sum(1 / pi_s)  # estimativa de N (só pra referência)
mu_HT <- T_HT / N          # se N é conhecido
# ou mu_HT <- T_HT / N_hat_HT se você quiser também ilustrar com N estimado

# 3) Estimador tipo Hájek para a média
mu_Hajek <- sum(Y_s / pi_s) / sum(1 / pi_s)

c(mu_true = mu_true,
  naive   = mean_naive,
  HT      = mu_HT,
  Hajek   = mu_Hajek)
```

```
  mu_true     naive        HT     Hajek 
 9.986339 11.454791  9.845307  9.954740 
```

```r
#### 5) Bootstrap para o estimador de Hájek ####

B <- 1000  # número de réplicas bootstrap
mu_Hajek_boot <- numeric(B)

for (b_rep in 1:B) {
  # reamostragem com reposição DENTRO da amostra obtida
  idx <- sample(seq_along(Y_s), size = length(Y_s), replace = TRUE)
  Y_b  <- Y_s[idx]
  pi_b <- pi_s[idx]

  mu_Hajek_boot[b_rep] <- sum(Y_b / pi_b) / sum(1 / pi_b)
}

# Erro-padrão bootstrap
se_boot <- sd(mu_Hajek_boot)

# Intervalo de confiança normal aproximado
alpha_ci <- 0.05
z <- qnorm(1 - alpha_ci/2)
IC_lower <- mu_Hajek - z * se_boot
IC_upper <- mu_Hajek + z * se_boot

c(mu_true = mu_true,
  mu_Hajek = mu_Hajek,
  IC_lower = IC_lower,
  IC_upper = IC_upper)
```

```
  mu_true  mu_Hajek  IC_lower  IC_upper
 9.986339  9.954740  9.561835 10.347644
```

```r
hist(mu_Hajek_boot,
     main = "Distribuição bootstrap do estimador de Hájek",
     xlab = "Média estimada (Hájek)",
     breaks = 30)
abline(v = mu_true, col = "red", lwd = 2)   # verdadeira média populacional
abline(v = mu_Hajek, col = "blue", lwd = 2) # estimativa observada
```


Distribuição bootstrap do estimador de Hájek
Frequency
0
20
60
100
9.2
9.4
9.6
9.8
10.0
10.2
10.4
10.6
Média estimada (Hájek)