Lista 2 - Monte Carlo

AUTHORS
Andreia Queiroz Correia Dummar (222103912)
Fernando da Silva Costa (232102946)
Roberto Jorge Dummar FIlho (232103587)

1) Geração de Números Aleatórios

a. O método Inverse Transform - Exercício 3.3 do livro da Rizzo.

3.2 The Pareto(a, b) distribution has cdf

$$F(x) = 1 - (\frac{b}{x})^a, x \geq b \geq 0, a \geq 0$$

Derive the probability inverse transformation $F^{-1}(U)$ and use the inverse transform method to simulate a random sample from the Pareto(2, 2) distribution. Graph the density histogram of the sample with the Pareto(2, 2) density superimposed for comparison.

Resposta

Para cálculo da transformação inversa, temos que:

$$u = F(x) = 1 - (\frac{b}{x})^a \Rightarrow 1 - u = (\frac{b}{x})^a \Rightarrow (1-u)^{\frac{1}{a}} = \frac{b}{x} \Rightarrow x = \frac{b}{(1-u)^{\frac{1}{a}}} \Rightarrow F_x^{-1}(u) = b * (1-u)^{\frac{-1}{a}}$$

Como explicado no Exemplo 3.3 (p.51), U e $1 - U$ possuem a mesma distribuição $Uniform(0, 1)$, logo é mais simples utilizar $b * (u)^{\frac{-1}{a}}$.

Por se tratar de função de distribuição acumulada, é necessária derivá-la para encontrar a função densidade de probabilidade, logo:

$$F(x) = 1 - (\frac{b}{x})^a, x \geq b \geq 0, a \geq 0$$

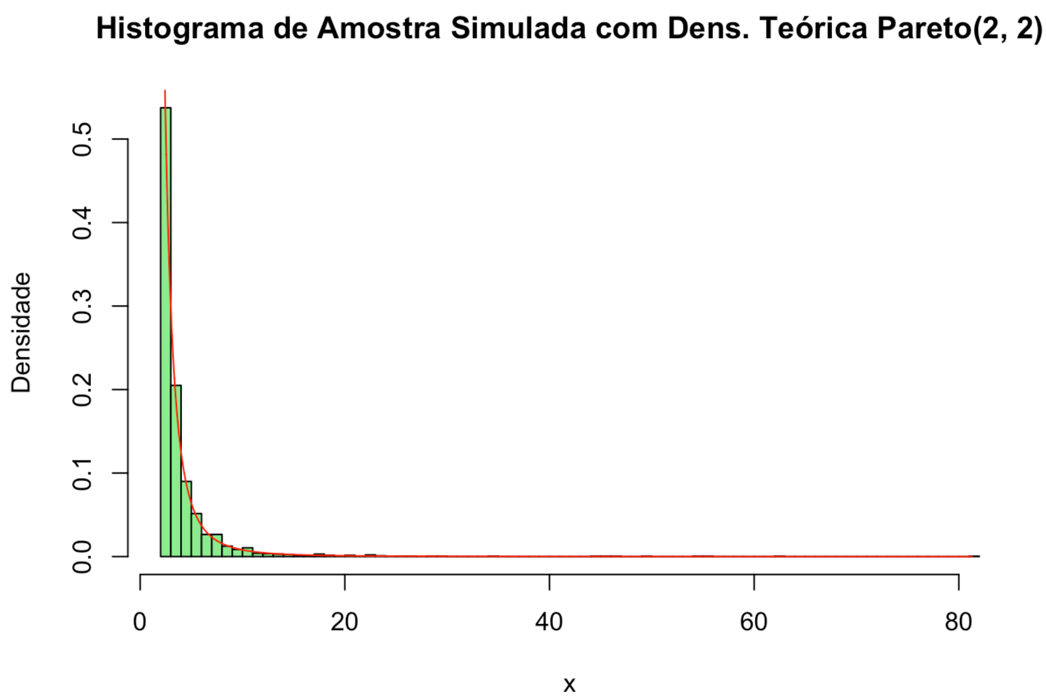
$$f(x) = F'(x) = 0 - (a) * (b^a * (-1) * x^{-a-1}) \Rightarrow a * b^a * x^{-(a+1)}, x \geq b$$

```
set.seed(1)
# Valores iniciais dados pelo problema
a <- 2
b <- 2
n <- 2000
u <- runif(n)
x <- b * (u)^(-1/a)
print(summary(x))
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 2.000   2.303   2.880   4.024   4.075  81.294
```

```
hist(x, breaks = 100, prob = TRUE, col = 'lightgreen'
     , main = "Histograma de Amostra Simulada com Dens. Teórica Pareto(2, 2)"
     , xlab = 'x', ylab = 'Densidade')
y <- sort(x)
```

```
        fy <- a * b^a * y^(- (a + 1))
        lines (y, fy, col = 'red', lwd = 1)
```



b. O método Acceptance-Rejection - Exercício 3.7 do livro da Rizzo. (Exemplo 3.7 faz isso para Beta(2,2). Para simplificar vocês podem resolver a questão somente para α e β maiores que 1.)

3.7 Write a function to generate a random sample of size n from the Beta(a, b) distribution by the acceptance-rejection method. Generate a random sample of size 1000 from the Beta(3,2) distribution. Graph the histogram of the sample with the theoretical Beta(3,2) density superimposed.

Resposta

Vamos calcular a função densidade de probabilidade (PDF) para a distribuição Beta com parâmetros $\alpha = 3$ e $\beta = 2$.

Definição Geral da PDF da Distribuição Beta

A função densidade de probabilidade (PDF) da distribuição Beta é dada por:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

onde $B(\alpha, \beta)$ é a função beta: $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

Para $\alpha = 3$ e $\beta = 2$, a PDF se torna:

$$f(x; 3, 2) = \frac{x^{3-1}(1-x)^{2-1}}{B(3, 2)} = \frac{x^2(1-x)}{B(3, 2)}$$

Cálculo da Função Beta B(3, 2)

A função beta para $\alpha = 3$ e $\beta = 2$ é: $B(3,2) = \frac{\Gamma(3)\Gamma(2)}{\Gamma(3+2)}$

Usando as propriedades da função gama: $\Gamma(n) = (n-1)!$, então, $\Gamma(3) = 2! = 2$, $\Gamma(2) = 1! = 1$, $\Gamma(5) = 4! = 24$

Portanto, $B(3,2) = \frac{2!\cdot 1!}{4!} = \frac{2\cdot 1}{24} = \frac{2}{24} = \frac{1}{12}$

Substituindo $B(3,2) = \frac{1}{12}$ na expressão da PDF:

$$f(x;3,2) = \frac{x^2(1-x)}{\frac{1}{12}} = x^2(1-x)\cdot 12 = 12x^2(1-x)$$

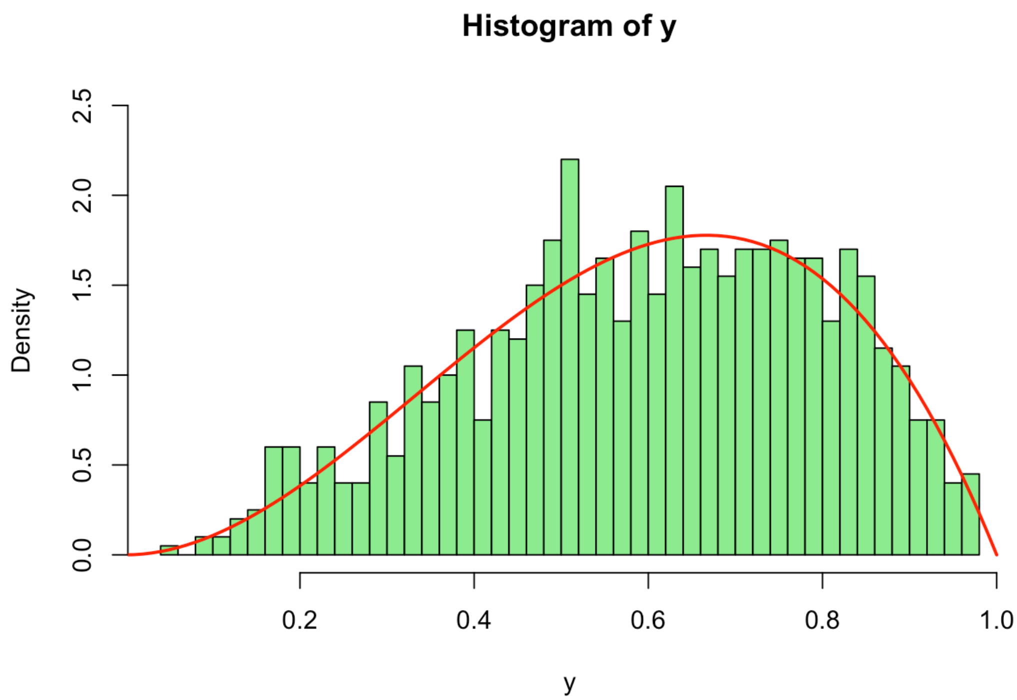
No exercício, para os valores de $\alpha = 3$ e $\beta = 2$, temos uma distribuição unimodal com um pico, assimétrica e levemente inclinada para a direita.

```
function_Beta <- function(n, a, b)
{
  k <- 0
  y <- numeric(n)
  count <- 0
  while (k < n)
  {
    u <- runif(1)
    x <- runif(1)
    if (x^(a - 1) * (1 - x)^(b - 1) > u)
    {
      k <- k + 1
      y[k] <- x
    }
    count <- count + 1
  }
  print(paste("Quantidade de iterações: ", count))
  return(y)
}

y <- function_Beta(1000, a = 3, b = 2)
```

```
[1] "Quantidade de iterações:  11880"
```

```
hist(y, breaks = 50, prob = TRUE, ylim = c(0, 2.5), col='lightgreen')
z <- seq (0, 1, .01)
f.z <- 12 * z^2 * (1-z)
lines (z, f.z, col = 'red', lwd=2)
```



2) Integração por Monte Carlo

a. Exercício 5.3 do livro da Rizzo. (Opcional: gera da exponencial truncada ao intervalo [0;0,5] e compara a variância.)

Compute a Monte Carlo estimate $\hat{\theta}$ of

$$\theta = \int_0^{0.5} e^{-x}\, dx$$

by sampling from Uniform(0, 0.5), and estimate the variance of $\hat{\theta}$*. Find another Monte Carlo estimator* $\hat{\theta}^*$ *by sampling from the exponential distribution. Which of the variances (of* $\hat{\theta}$ *and* $\hat{\theta}^*$*) is smaller, and why?*

Resposta

O valor exato da integral é dado por:

$$\theta = -e^{-0.5} - (-e^{0}) = 1 - e^{-0.5} = 1 - 0.606531 = 0.393469$$

O estimador de Monte Carlo é dado pela expressão:

$$\hat{\theta} = (b-a)\int_a^b g(x)\, dx = \frac{1}{2}\left(\frac{1}{m}\sum_{i=1}^{m} e^{-u}\right),$$

sendo que u é gerado a partir de uma distribuição $Uniforme(0, 0.5)$.

```r
n <- 20000
# Geração da distribuiçào uniforme entre os intervalor 0 e 0.5
u <- runif(n, 0, 0.5)
theta <- 0.5 * mean(exp(-u))
theta
```

```
[1] 0.393242
```

```r
# Para cada 20.000 amostras da distribuição uniforme no intervalo [0, 0.5], vamos
# calcular o estimador de Monte Carlo usando a função 0.5 * exp(-u), fazendo isso
# 2.000 vezes
estimator <- replicate(2000, expr = {
  u <- runif(n, 0, 0.5)
  theta <- 0.5 * mean(exp(-u))
  theta
})
print(paste("Média do estimador: ", mean(estimator)))
```

```
[1] "Média do estimador:  0.39347051705846"
```

```r
print(paste("Variância do estimador: ", var(estimator)))
```

```
[1] "Variância do estimador:  1.59552490493912e-07"
```

```r
print(paste("DP do estimador: ", sd(estimator)))
```

```
[1] "DP do estimador:  0.000399440221427327"
```

Para estimarmos a variância de $\hat{\theta}^*$, vamos utilizar a seguinte expressão:

$$\hat{\theta}^* = \frac{1}{m}\sum_{i=1}^{m} I(v < 0.5)$$

, sendo v gerado a partir de uma distribuição $Exponential(1)$.

```r
n <- 20000
v <- rexp(n, 1)
theta <- mean(v <= 0.5)
print(theta)
```

```
[1] 0.3918
```

```r
estimator_1 <- replicate(2000, expr = {
  v <- rexp(n, 1)
  theta <- mean(v <= 0.5)
  theta
})
print(paste("Média do estimador", mean(estimator_1)))
```

```
[1] "Média do estimador 0.393552925"
```

```
print(paste("Variância do estimador: ", var(estimator_1)))
```

```
[1] "Variância do estimador:  1.19860106997249e-05"
```

```
print(paste("Desvio-padrão do estimador: ", sd(estimator_1)))
```

```
[1] "Desvio-padrão do estimador:  0.00346208184474672"
```

```
print(paste("Razão entre os estimadores: ", var(estimator)/var(estimator_1)))
```

```
[1] "Razão entre os estimadores:  0.0133115591576749"
```

A variância do estimador $\hat{\theta}$ (baseado na amostragem uniforme) é maior do que a variância do estimador $\hat{\theta}^*$ (baseado na amostragem exponencial) porque a transformação usada na amostragem exponencial pode reduzir a variabilidade dos valores amostrados, uma vez que é melhor ajustada à forma da função de densidade e^{-x}.

Assim, o estimador $\hat{\theta}^*$ terá uma variância menor devido à melhor adequação da amostragem à distribuição exponencial.

b. Variáveis Antitéticos: Exercício 5.10 do livro da Rizzo. (Opcional: Trocar esse exercício para 5.9, mas saiba que a função não é monotônica [pode quebrar em pedaços monotônicos] e o limite superior é infinito [precisa fazer alguma transformação de variáveis x-> 1/x funcionaria])

Use Monte Carlo integration with antithetic variables to estimate

$$\int_0^1 \frac{e^{-x}}{1+x^2}\,dx$$

and find the approximate reduction in variance as a percentage of the variance without variance reduction.

Resposta

O uso de variáveis antiéticas ajudam a reduzir a variabilidade de um método e estimação.

Como regra geral, temos que:

$$Var(\frac{U_1+U_2}{2}) = \frac{1}{4} * (Var(U_1) + Var(U_2) + 2 * Cov(U_1, U_2))$$

sendo que a variância de $\frac{(U_1+U_2)}{2}$ é menor se U_1 e U_2 são negativamente correlacionadas do que quando as variáveis são independentes.

Neste caso, iremos gerar uma distribuição $Uniforme(0, 1)$ e aplicar o estimador de Monte Carlo para o valor de u.

Ainda, para calcularmos com o uso de variáveis antiéticas, iremos aplicar a expressão abaixo:

$$\hat{\theta} = \frac{1}{m}\left\{Y_1 + Y_1' + Y_2 + Y_2' + \cdots + Y_{m/2} + Y_{m/2}'\right\}$$

```
set.seed(10)
m <- 20000
monte_carlo <- replicate (2000, expr =
                  {
                    u <- runif (m)
```

```
                         mean(exp(-u)/(1 + u^2))
                        }
                     )
set.seed(10)
anti_ethic <- replicate (2000, expr =
                               { u <- runif (m/2)
                               x1 <- exp(-u)/(1 + u^2)
                               x2 <- exp(-(1 -u))/(1 + ((1 -u)^2))
                               mean(c (x1, x2))
                               }
                             )

print(c(mean(monte_carlo), var(monte_carlo)))
```

```
[1] 5.247396e-01 3.019848e-06
```

```
print(c(mean(anti_ethic), var(anti_ethic)))
```

```
[1] 5.247863e-01 1.180597e-07
```

```
approx_reduction = 100 * (var (monte_carlo) - var (anti_ethic)) / var (monte_carlo)
print(paste(round(approx_reduction,4), "%"))
```

```
[1] "96.0905 %"
```

A redução da variância entre o uso de Monte Carlo padrão e o uso de Monte Carlo com variáveis antiéticas foi de 96,09%.

c. Importance Sampling - Exercício 5.14 do livro da Rizzo.

Obtain a Monte Carlo estimate of

$$\int_1^\infty \frac{x^2}{\sqrt{2\pi}} e^{-x^2/2}\, dx$$

by importance sampling.

Resposta

Para a análise, foram utilizadas com as distribuições:

1. Distribuição x^2 com 1 grau de liberdade
2. Distribuição Γ (Gamma) com parâmetros $\alpha = \frac{3}{2} \quad \text{e} \quad \beta = 2$
3. Distribuição exponencial
4. Distribuição Cauchy (com localização 2 e escala 1)

```
set.seed(1)
n <- 20000
imp_sampl_1 <- replicate(2000, expr =
                   {
                     x <- sqrt(rchisq(n, 1)) + 1
                     f <- 2 * dnorm(x, 1)
                     g <- x^2 * exp(-x^2/2)/sqrt(2 * pi)
                     mean (g/f)
                   }
)
```

```r
set.seed(1)
imp_sampl_2 <- replicate(2000, expr =
                   {
                     x <- rgamma(n, 3/2, 2) + 1
                     f <- dgamma(x - 1, 3/2, 2)
                     g <- x^2 * exp(-x^2/2)/sqrt(2 * pi)
                     mean (g/f)
                   }
)
set.seed(1)
imp_sampl_3 <- replicate(2000, expr =
                   {
                     x <- rexp(n, 1)
                     f <- dexp(x, 1)
                     g <- x^2 * exp(-x^2/2)/sqrt(2 * pi)
                     mean (g/f)
                   }
)
set.seed(1)
imp_sampl_4 <- replicate(2000, expr =
                   {
                     x <- rcauchy(n, 2, 1)
                     f <- dcauchy(x, 2, 1)
                     g <- x^2 * exp(-x^2/2)/sqrt(2 * pi)
                     mean (g/f)
                   }
)


print(c(mean(imp_sampl_1), mean(imp_sampl_2), mean(imp_sampl_3), mean(imp_sampl_4)))
```

```
[1] 0.4006161 0.4006423 0.4999949 1.0002580
```

```r
print(c(var(imp_sampl_1), var(imp_sampl_2), var(imp_sampl_3), var(imp_sampl_4)))
```

```
[1] 9.727324e-08 6.082952e-07 1.465524e-05 2.061787e-04
```

```r
print(var(imp_sampl_1)/var(imp_sampl_2))
```

```
[1] 0.1599112
```

No caso em tela, A média das estimativas estão próximas, pois todas estão estimando a mesma integral.

A 1^a função importância produz um estimador mais eficiente, com maior precisão e menor variabilidade que as demais funções uma vez que possui a menor variância.

d. Stratified Importance Sampling - Exercício 5.15 do livro da Rizzo. (Não esquece de usar os quantís da função importância para definir os intervalos.)

Obtain the stratified importance sampling estimate in Example 5.13 and compare it with the result of Example 5.10.

Resposta

O exemplo 5.13 traz a seguinte integral:

$$\int_0^1 \frac{e^{-x}}{1+x^2}\,dx$$

com a função importância dada por:

$$f(x) = \frac{e^{-x}}{1-e^{-1}}, 0 < x < 1,$$

nos cinco subintervalos: $(j/5, (j+1)/5), j = 0, 1, \ldots, 4.$.

As variáveis no j^{th} subintervalo são geradas a partir da densidade:

$$\frac{5e^{-x}}{1-e^{-1}}, \quad \frac{j-1}{5} < x < \frac{j}{5}.$$

Para encontrar a transformação inversa usada no código, considere a função de importância dada por:

$$f(x) = \frac{e^{-x}}{1-e^{-1}} \quad \text{para } 0 < x < 1$$

Primeiro, encontramos a função de distribuição acumulada (CDF) da função de importância (f(x)):

$$F(x) = \int_0^x \frac{e^{-t}}{1-e^{-1}}\,dt$$

Integrando, obtemos:

$$F(x) = \left[\frac{-e^{-t}}{1-e^{-1}}\right]_0^x = \frac{1-e^{-x}}{1-e^{-1}}$$

Para obter a transformação inversa, resolvemos (F(x) = u) para (x):

$$u = \frac{1-e^{-x}}{1-e^{-1}} => u(1-e^{-1}) = 1-e^{-x} => e^{-x} = 1-u(1-e^{-1}) =>$$

$$=> ln(e^{-x}) = ln(1-u(1-e^{-1})) => x = -ln(1-u(1-e^{-1})$$

que é a transformação inversa da função de distribuição acumulada (CDF) da distribuição de importância

$$f(x) = \frac{e^{-x}}{1-e^{-1}}$$

```r
set.seed(1)
M <- 20000
k <- 5
m <- M/k
si_stratified <- numeric(k)
v_stratified <- numeric(k)
g <- function(x) exp(-x)/(1 + x^2)
f <- function(x) ((k * exp(-x))/(1 - exp(-1)))

for(j in 1:k) {
  u <- runif(m, (j - 1)/k, j/k)
  x <- -log(1 - (1 - exp(-1)) * u)
  gf <- g(x)/f(x)
```

```
  si_stratified[j] <- mean(gf)
  v_stratified[j] <- var(gf)
}

print(paste("A soma da média dos estratos para estimativa da integral é: ",sum(si_stra
```

```
[1] "A soma da média dos estratos para estimativa da integral é:  0.524763642375128"
```

```
print(paste("A média da variância dos estratos é: ", mean(v_stratified)))
```

```
[1] "A média da variância dos estratos é:  1.74807187879508e-05"
```

```
print(sqrt(mean(v_stratified)))
```

```
[1] 0.004180995
```

No caso do exemplo 5.10, se não fosse usada a estratificação, os valores seriam:

```
set.seed(1)
M <- 20000
si <- 0
v <- 0
g <- function(x) exp(-x)/(1 + x^2)
f <- function(x) (exp(-x)/(1 - exp(-1)))
u <- runif(M)
x <- -log(1 - (1 - exp(-1)) * u)
gf <- g(x)/f(x)

si_n_stratified <- mean(gf)
v_n_stratified <- var(gf)

print(si_n_stratified)
```

```
[1] 0.5245328
```

```
print(v_n_stratified)
```

```
[1] 0.009477452
```

```
print(sqrt(v_n_stratified))
```

```
[1] 0.0973522
```

```
print((v_n_stratified - mean(v_stratified))/v_n_stratified)
```

```
[1] 0.9981555
```

No exemplo calculado, a variância por amostragem estratificada foi reduzida para 99,82% do valor da variância por amostragem não estratificada.

3) Inferência com Monte Carlo

a. MSE (EQM): Exercício 6.1 do livro da Rizzo. (aproveite os códigos para a Normal com contaminação)

Estimate the MSE of the level k trimmed means for random samples of size 20 generated from a standard Cauchy distribution. (The target parameter θ is the center or median; the expected value does not exist.) Summarize the estimates of MSE in a table for $k = 1, 2, \ldots, 9$.

Resposta:

A distribuição de Cauchy é uma distribuição contínua com densidade de probabilidade dada por:

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]}$$

Para a distribuição de Cauchy padrão, temos $(x_0 = 0)$ e $(\gamma = 1)$:

$$f(x) = \frac{1}{\pi(1 + x^2)}$$

Trimmed mean (média podada) é uma média calculada excluindo os valores extremos de uma amostra. Para uma amostra ordenada $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$:

- O nível de podagem $ k $ indica que os k menores e os k maiores valores são excluídos.
- A média podada é então calculada sobre os $ n - 2k $ valores restantes.

O erro quadrático médio (MSE) é uma medida de precisão de um estimador. Para um estimador $\hat{\theta}$ do parâmetro θ:

$\text{MSE}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2]$

```
set.seed(1)
n <- 20
K <- n/2 - 1
m <- 1000
mse <- matrix(0, n/2, 2)
trimmed.mse <- function(n, m, k)
  {
    tmean <- numeric(m)
    for (i in 1:m)
    {
      x <- sort(rcauchy(n))
      tmean[i] <- sum(x[(k + 1):(n - k)])/(n - 2 * k)
    }
    mse.est <- mean(tmean^2)
    se.mse <- sqrt(mean((tmean - mean(tmean))^2))/sqrt(m)
    return(c(mse.est, se.mse))
}

for (k in 0:K)
{
  mse[k + 1, 1:2] <- trimmed.mse(n = n, m = m, k = k)
}
```

```r
mse <- as.data.frame(cbind(0:K, mse))
names(mse) <- c("k", "média podada", "erro padrão")
print(mse)
```

```
   k média podada erro padrão
1  0 1766.9212480  1.32920584
2  1    1.9158594  0.04374635
3  2    0.3245856  0.01795765
4  3    0.2503728  0.01579560
5  4    0.1752799  0.01322104
6  5    0.1476811  0.01213569
7  6    0.1447897  0.01203215
8  7    0.1174757  0.01083862
9  8    0.1284601  0.01133352
10 9    0.1400133  0.01179556
```

A tabela de resultados mostra como o MSE da média podada varia com diferentes níveis de podagem k. Conforme o nível de k aumenta, excluem-se os valores mais extremos, melhorando a precisão do estimador, reduzindo o erro quadrático médio. No caso, a podagem ajuda a reduzir o efeito dos outliers na distribuição de Cauchy, que afetam mais significativamente a média.

Entretanto, o MSE e o erro padrão atingem seus menores valores não no máximo valor de k, mas quando ele é igual a 7.

b. Poder de testes: Exercício 6.3 do livro da Rizzo. (aproveite os códigos do Exemplo 6.9)

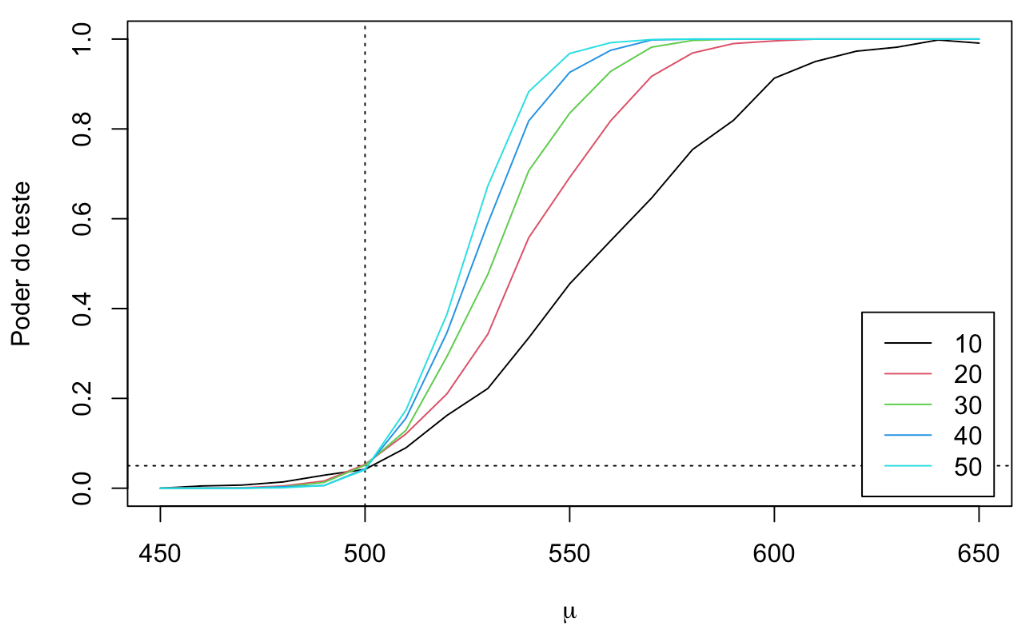
Plot the power curves for the t-test in Example 6.9 for sample sizes 10, 20, 30, 40, and 50, but omit the standard error bars. Plot the curves on the same graph, each in a different color or different line type, and include a legend. Comment on the relation between power and sample size.

<u>Resposta</u>

```r
N <- c(10, 20, 30, 40, 50)
m <- 1000
mu0 <- 500
sigma <- 100
mu <- c(seq(450, 650, 10)) #alternatives
M <- length(mu)
power <- matrix(0, M, 5)
for (j in 1:5)
{
  n <- N[j]
  for (i in 1:M)
  {
    mu1 <- mu[i]
    pvalues <- replicate(m, expr =
    {
      #simulate under alternative mu1
      x <- rnorm(n, mean = mu1, sd = sigma)
      ttest <- t.test(x, alternative = "greater", mu = mu0)
      ttest$p.value
    })
    power[i, j] <- mean(pvalues <= .05)
  }
}
```

```
plot(mu, power[, 1], type = "l", ylim = range(power), xlab = bquote(mu), ylab = "Poder
abline(v = mu0, lty = 3)
abline(h = 0.05, lty = 3)
for (j in 2:5)
  {
    lines(mu, power[, j], col = j)
  }

legend("bottomright", inset = 0.02, legend = N, col = 1:5, lty = 1)
```



Quando o tamanho da amostra aumenta, a precisão das estimativas de parâmetros também aumenta. A média amostral se aproxima da verdadeira média populacional μ_0 , diminuindo a variabilidade da média amostral. Isso resulta em um maior poder estatístico, permitindo detectar diferenças menores entre a média amostral e μ_0. Portanto, quanto mais se aumentam as amostras, mais a curva do poder estatístico se aproxima de μ_0, indicando uma maior capacidade de detectar desvios da hipótese nula.

c. Níveis de confiança: Exercício 6.4 ou 6.5 do livro da Rizzo

6.4 Suppose that $X_1, \ldots, X_n$ are a random sample from a lognormal distribution with unknown parameters. Construct a 95% confidence interval for the parameter μ. Use a Monte Carlo method to obtain an empirical estimate of the confidence level.

Resposta

Iremos gerar uma amostra de tamanho 50 a partir de uma distribuição lognormal padrão (com média logarítmica $\mu = 0$ e desvio padrão logarítmico $\sigma = 1$).

O intervalo de confiança é calculado utilizando-se a fórmula:

$$\left(\bar{y} - 1.96 \cdot \frac{s_y}{\sqrt{30}}, \bar{y} + 1.96 \cdot \frac{s_y}{\sqrt{30}}\right)$$

```
set.seed(1)
n <- 50
confid_interv <- replicate(20000, expr =
  {
    x <- rlnorm(n, 0, 1)
    y <- log(x)
    ybar <- mean(y)
    se <- sd(y)/sqrt(n)
    ybar + se * qnorm(c(0.025, 0.975))
  })
L_confid_interv <- confid_interv[1, ]
U_confid_interv <- confid_interv[2, ]

print(paste("Quantidade de ICs que contém o valor da média logarítmica: ", sum(L_confi
```

```
[1] "Quantidade de ICs que contém o valor da média logarítmica:  18904"
```

```
print(paste("Proporção de ICs que contêm o verdadeiro valor de mi: ", 100 * mean(L_con
```

```
[1] "Proporção de ICs que contêm o verdadeiro valor de mi:  94.52 %"
```

Ao usarmos a simulação Monte Carlo para validar a construção de intervalos de confiança (IC) para a média logarítmica μ de uma distribuição lognormal, do total de 20.000 IC, temos 18.904 IC que contém o valor da média logarítmica ($\mu = 0$ para a distribuição lognormal padrão).

Ainda, temos que a proporção de IC que contêm o valor 0, pela simulação, foi de 94,52%, um valor muito próximo de 95%.