

Lista 1

Técnicas Computacionais em Estatística

Tailine J. S. Nonato

April 9, 2025

Lista 1 - Geração de NPA's (Números Pseudo-Aleatórios)

```
# preparação do ambiente  
set.seed(345)
```

Exercício 1

A distribuição Laplace padrão tem densidade $f(x) = \frac{1}{2}e^{-|x|}$, $x \in \mathbb{R}$. Use o método da transformada inversa para gerar uma amostra aleatória de tamanho 1000 dessa distribuição. Plote um histograma.

Resolução

A função de distribuição acumulada (cdf) da distribuição Laplace padrão é dada por:

$$F(x) = \begin{cases} \frac{1}{2}e^x, & x < 0 \\ 1 - \frac{1}{2}e^{-x}, & x \geq 0 \end{cases}$$

Para $u \leq \frac{1}{2}$:

$$u = \frac{1}{2}e^x \Rightarrow x = \log(2u)$$

Para $u > \frac{1}{2}$:

$$u = 1 - \frac{1}{2}e^{-x} \Rightarrow x = -\log(2(1-u))$$

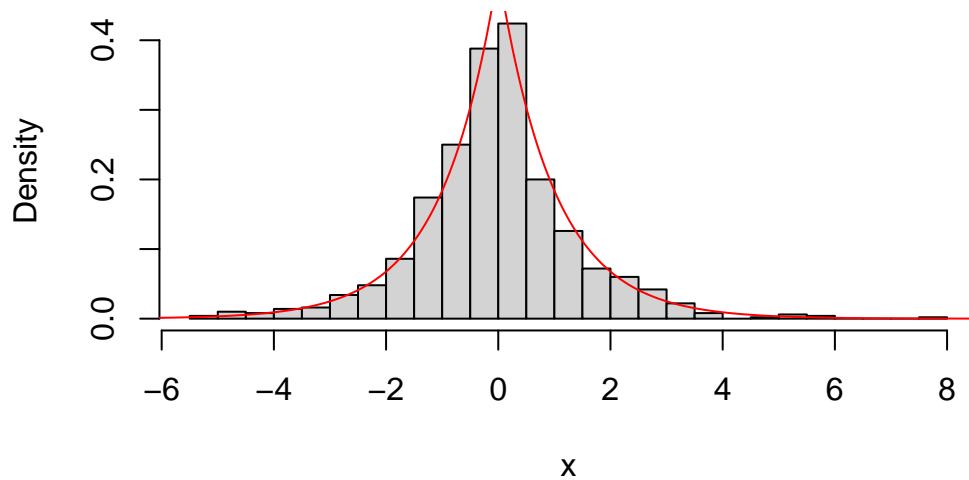
Logo, a inversa da cdf é dada por:

$$F^{-1}(u) = \begin{cases} \log(2u), & u \leq \frac{1}{2} \\ -\log(2(1-u)), & u > \frac{1}{2} \end{cases}$$

Assim,

```
n <- 1000
u <- runif(n)
x <- ifelse(u <= 0.5, log(2*u), -log(2*(1 - u)))

t <- seq(-10, 10, 0.01)
hist(x, probability = TRUE, main = "", breaks = 30)
lines(t, 0.5 * exp(-abs(t)), col = "red")
```



Exercício 2

Dada a densidade $f(x \mid \theta)$ e a densidade a priori $\pi(\theta)$, se observamos $x = x_1, \dots, x_n$, a distribuição a posteriori de θ é dada por:

$$\pi(\theta \mid x) = \pi(\theta \mid x_1, \dots, x_n) \propto \prod_i f(x_i \mid \theta) \pi(\theta),$$

onde $\prod_i f(x_i \mid \theta) = L(\theta \mid x_1, \dots, x_n)$ é a função de verossimilhança. Para estimar uma média normal, uma priori robusta é a Cauchy. Para $X_i \sim N(\theta, 1)$, $\theta \sim \text{Ca}(0, 1)$, a distribuição a posteriori é:

$$\pi(\theta \mid x) \propto \frac{1}{\pi} \frac{1}{1 + \theta^2} \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n e^{-(x_i - \theta)^2/2}.$$

Seja $\theta_0 = 3$, $n = 10$, e gere $X_1, \dots, X_n \sim N(\theta_0, 1)$. Use o algoritmo da Aceitação-Rejeição com uma candidata $\text{Ca}(0, 1)$ para gerar uma amostra da distribuição a posteriori. Avalie quão bem o valor θ_0 é recuperado. Estenda o código de maneira que $n = 10, 25, 50, 100$. Assuma que $M = L(\hat{\theta} \mid x_1, \dots, x_n)$, ou seja, M é a função de verossimilhança avaliada no estimador de máxima verossimilhança.

Resolução

$$f: \frac{1}{\pi} \frac{1}{1 + \theta^2} \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n e^{-(x_i - \theta)^2/2}, \theta \sim \text{Ca}(0, 1)$$

$$g: \text{Ca}(0, 1)$$

Verificação de condições:

[X] f e g têm suportes compatíveis.

[X] Há uma constante M tal que $f(x) \leq M g(x)$ para todo x .

```
theta_0 <- 3
n <- c(10, 25, 50, 100)
Nsim <- 2500

set.seed(123)

posteriori <- function(n, theta0, Nsim) {
  # step 1: amostra de x
  x <- rnorm(n, theta0, 1)

  # step 2: amostra de y e u (candidatos e uniforme)
  y <- rcauchy(Nsim)
  g_y <- dcauchy(y)
```

```

# f(y): priori * verossimilhança
priori_y <- dcauchy(y)
lik_y <- sapply(y, function(t) prod(dnorm(x, mean = t, sd = 1)))
f_y <- priori_y * lik_y

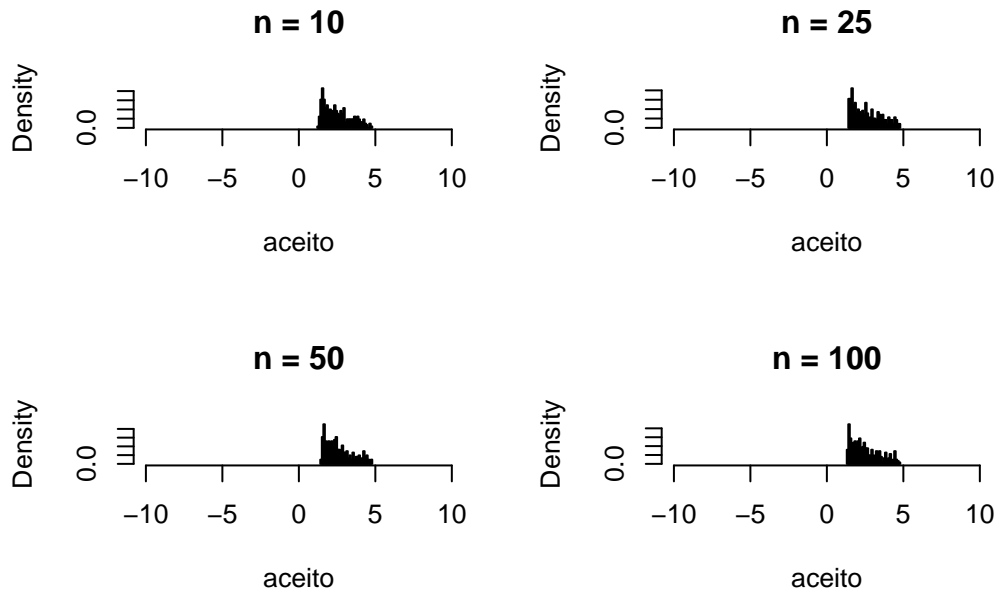
ratio <- f_y / g_y
# o menor valor que garante que M >= f_y/g_y
M <- max(ratio)
u <- runif(Nsim, 0, M)

# step 3: aceitação-rejeição
aceito <- y[u <= (ratio / M)]

# visualização
hist(aceito, probability = TRUE, main = paste("n =", n),
     xlim = c(-10, 10), breaks = 30)
}

# estendendo para n = 10, 25, 50, 100
par(mfrow = c(2,2))
sapply(c(10, 25, 50, 100), posteriori, theta0 = theta_0, Nsim = Nsim)

```



	[,1]	[,2]	[,3]	[,4]
breaks	numeric,37	numeric,35	numeric,35	numeric,36
counts	integer,36	integer,34	integer,34	integer,35
density	numeric,36	numeric,34	numeric,34	numeric,35
mids	numeric,36	numeric,34	numeric,34	numeric,35
xname	"aceito"	"aceito"	"aceito"	"aceito"
equidist	TRUE	TRUE	TRUE	TRUE

Explicação do porquê usar $M = \max(\text{ratio})$:

Lembrando o objetivo do algoritmo

A ideia é gerar uma amostra de uma distribuição complicada $f(\theta)$, usando uma distribuição mais simples $g(\theta)$ como candidata.

Para isso funcionar, a função $f(\theta)$ precisa ser menor ou igual a $M \cdot g(\theta)$ em todos os pontos:

$$f(\theta) \leq M \cdot g(\theta), \quad \text{para todo } \theta$$

Por que o máximo da razão $\frac{f(\theta)}{g(\theta)}$?

Porque essa razão mede o quanto f é maior do que g em cada ponto. Então:

$$\frac{f(\theta)}{g(\theta)} \leq M \iff f(\theta) \leq M \cdot g(\theta)$$

Logo, o menor valor de M que garante isso é justamente o maior valor da razão:

$$M = \sup_{\theta} \left(\frac{f(\theta)}{g(\theta)} \right)$$

Usar o máximo observado (ou uma boa aproximação dele) torna o algoritmo mais eficiente: aceita mais amostras e rejeita menos.

Ilustração intuitiva:

Imagine que $f(\theta)$ é uma montanha e $g(\theta)$ é uma colina.

Você precisa inflar g com o fator M até que ela cubra completamente f .

- Muito pequeno \rightarrow rejeita demais ou nem cobre $f \rightarrow$ errado
- Muito grande \rightarrow cobre tudo, mas com desperdício \rightarrow ineficiente
- Máximo exato da razão \rightarrow cobre só o necessário \rightarrow ideal

Exercício 3

Gere 200 observações aleatórias de uma distribuição normal multivariada de dimensão 3 com vetor de médias $\mu = (0, 1, 2)^\top$ e matriz de covariância:

$$\Sigma = \begin{bmatrix} 1.0 & -0.5 & 0.5 \\ -0.5 & 1.0 & -0.5 \\ 0.5 & -0.5 & 1.0 \end{bmatrix}.$$

Use o método de decomposição de Cholesky.

Resolução

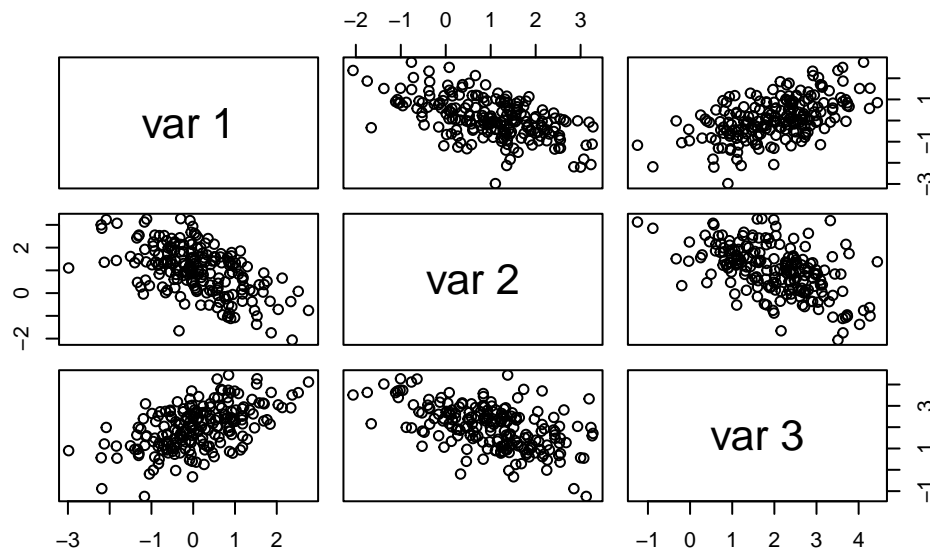
```
rmvn.Choleski <- function(n, mu, Sigma) {
  d <- length(mu)
  Q <- chol(Sigma)
  Z <- matrix(rnorm(n*d), nrow=n, ncol=d)
  X <- Z %*% Q + matrix(mu, n, d, byrow=TRUE)
  X}
```

```

Sigma <- matrix(c(1.0, -0.5, 0.5,
                 -0.5, 1.0, -0.5,
                 0.5, -0.5, 1.0), nrow = 3, byrow = TRUE)

mu <- c(0, 1, 2)
n <- 200
X <- rmvn.Choleski(n, mu, Sigma)
pairs(X)

```



Exercício 4

Considere o artigo “Bivariate Birnbaum–Saunders distribution and associated inference” (Kundu et al., 2010), disponível em PDF, onde os autores apresentam uma formulação para a distribuição bivariada de Birnbaum–Saunders (BVBS). A geração de dados desta distribuição é descrita na equação (8) do artigo. Utilize a parametrização apresentada no artigo para simular 1.000 observações de um vetor aleatório bivariado (T_1, T_2) com distribuição $BVBS(\alpha_1 = 0.5, \alpha_2 = 0.8, \beta_1 = 1.0, \beta_2 = 2.0, \rho = 0.7)$. Apresente um gráfico de dispersão dos dados gerados.

Resolução

```
alpha1 <- 0.5
alpha2 <- 0.8
beta1 <- 1
beta2 <- 2
rho <- 0.7
```

- Step 1: Generate independent U_1 and U_2 from $N(0, 1)$.

```
U1 <- rnorm(1000, 0, 1)
U2 <- rnorm(1000, 0, 1)
```

- Step 2: Compute

$$Z_1 = \frac{\sqrt{1+\rho} + \sqrt{1-\rho}}{2} * U_1 + \frac{\sqrt{1+\rho} - \sqrt{1-\rho}}{2} * U_2$$

$$Z_2 = \frac{\sqrt{1+\rho} - \sqrt{1-\rho}}{2} * U_1 + \frac{\sqrt{1+\rho} + \sqrt{1-\rho}}{2} * U_2$$

```
Z1 <- (sqrt(1 + rho) + sqrt(1 - rho)) / 2 * U1 +
      (sqrt(1 + rho) - sqrt(1 - rho)) / 2 * U2
Z2 <- (sqrt(1 + rho) - sqrt(1 - rho)) / 2 * U1 +
      (sqrt(1 + rho) + sqrt(1 - rho)) / 2 * U2
```

- Step 3: Obtain

$$T_i = \beta_i \left[\frac{1}{2} \alpha_i Z_i + \sqrt{\left(\frac{1}{2} \alpha_i Z_i \right)^2 + 1} \right]^2, \quad i = 1, 2$$

```
T1 <- beta1 * (0.5 * alpha1 * Z1 + sqrt((0.5 * alpha1 * Z1)^2 + 1))^2
T2 <- beta2 * (0.5 * alpha2 * Z2 + sqrt((0.5 * alpha2 * Z2)^2 + 1))^2
```

- Gráfico de dispersão dos dados gerados:


```
plot(T1, T2, xlab = "T1", ylab = "T2", main = "Gráfico de dispersão dos dados gerados")
```

