

Estatística Computacional

Técnicas de integração



Prof. Paulo Cerqueira Jr
Faculdade de Estatística - FAEST
Programa de Pós-graduação em Matemática e Estatística - PPGME

<https://github.com/paulocerqueirajr> 

Introdução

Introdução

- A necessidade de resolver uma integral numericamente aparece com bastante frequência.
- Principalmente quando:
 1. ajustamos modelos de regressão com efeitos aleatórios;
 2. precisamos resolver o denominador de uma distribuição **a posteriori**;
- Diversos métodos de integração numérica podem ser encontrados em textos clássicos de cálculo numérico.
- O método do retângulo, dos trapézios, do ponto central e suas diversas variações, são métodos simples de serem implementados.

Introdução

- Dentre os diversos métodos possíveis vamos descrever:
 - O método de trapezoidal de Simpson,
 - Quadratura Gaussiana usando os polinômios de Hermite, próprios para a integração na reta real.
- Métodos baseados em simulação:
 - Integração Monte Carlo;

Método trapezoidal

Método trapezoidal

- Consiste no uso de uma função linear para aproximar o integrando ao longo do intervalo de integração.
- O uso do polinômio de Newton entre os pontos $x = a$ e $x = b$ resulta em:

$$f(x) \approx f(a) + (x - a) \left[\frac{f(b) - f(a)}{b - a} \right].$$

Método trapezoidal

Com a integração analítica, obtém-se:

$$I(f) \approx \int_a^b f(x) + (x - a) \left[\frac{f(b) - f(a)}{b - a} \right] dx.$$

$$I(f) = f(a) + (x - a) \frac{1}{2} [f(b) - f(a)](b - a).$$

Simplificando o resultado, obtém-se uma fórmula aproximada popularmente conhecida como regra ou método trapezoidal.

$$I(f) \approx \left[\frac{f(b) + f(a)}{2} \right] (b - a).$$

Método trapezoidal - Exemplo

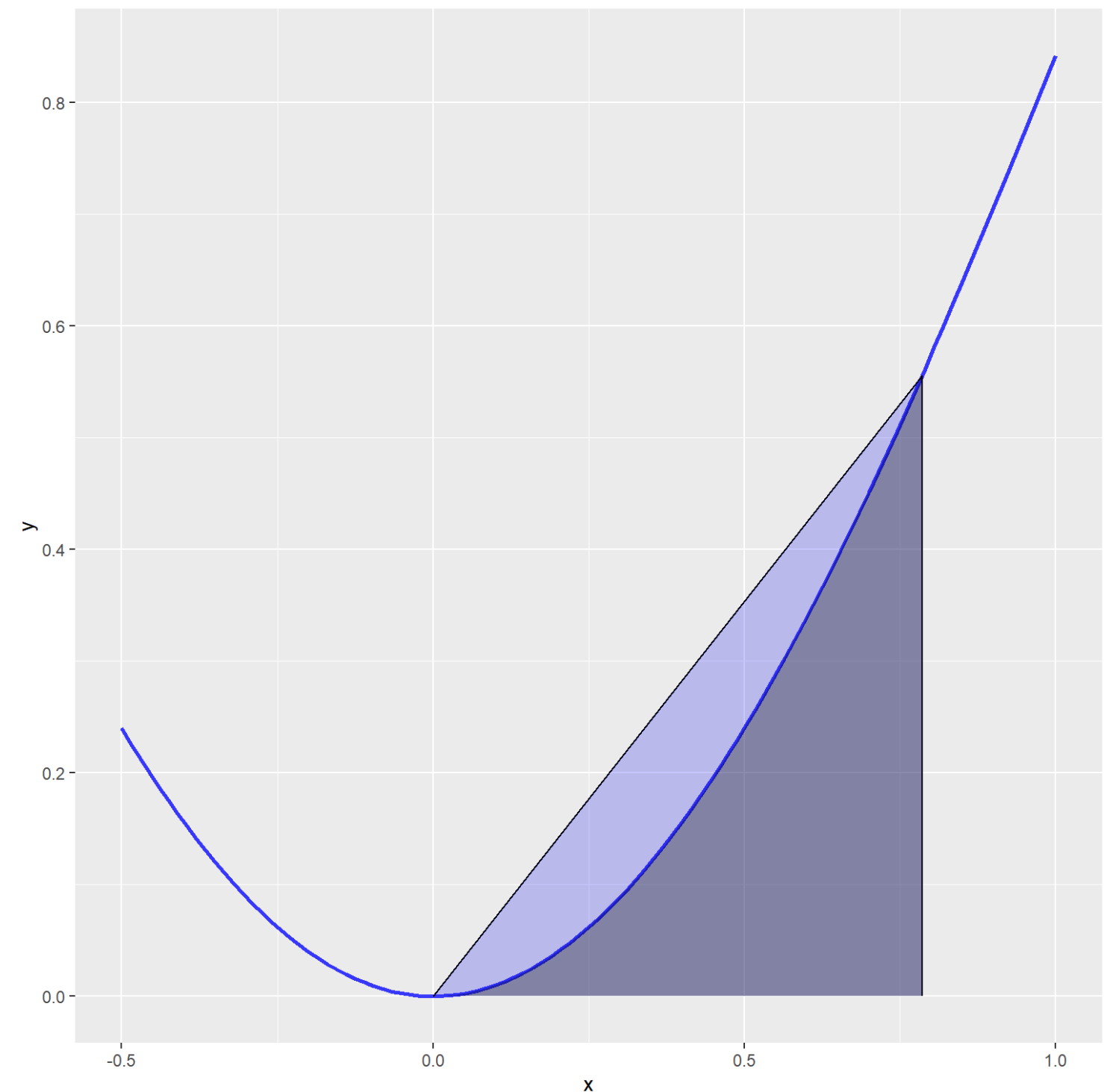
Se temos a seguinte função:

$$f(x) = x\sin(x).$$

E queremos resolver a seguinte integral:

$$I(f) = \int_0^{\pi/4} x\sin(x)dx.$$

Graficamente



Método trapezoidal - Exemplo

- No **R** temos a seguinte função:

```
1 trapezio <- function(integrando, a, b, ...){  
2   Int <- ((integrando(a, ...) + integrando(b, ...))/2) * (b-a)  
3   return(Int)  
4 }
```

- No pacote **pracma** podemos usar a função **trapzfun(f, a, b, ...)**.
- Dessa forma, a solução para a integral:

```
$value  
[1] 0.1517464
```

```
$iter  
[1] 11
```

```
$rel.err  
[1] 4.641731e-08
```

Método de Simpson

Método de Simpson

- Neste método, um polinômio de segunda ordem é usado para aproximar o integrando.
- Os coeficientes de um polinômio quadrático podem ser determinadas a partir de três pontos.
- Para uma integral ao longo do domínio $[a, b]$, são usados os dois pontos finais $x_1 = a, x_3 = b$, e o ponto central, $x_2 = (a + b)/2$.
- O polinômio pode ser escrito na forma:

$$p(x) = \alpha + \beta(x - x_1) + \lambda(x - x_2),$$

onde α , β e λ são constantes desconhecidas avaliadas a partir da condição que diz que o polinômio deve passar por todos os pontos, $p(x_1) = f(x_1)$, $p(x_2) = f(x_2)$ e $p(x_3) = f(x_3)$.

Método de Simpson

$$\alpha = f(x_1), \quad \beta = [f(x_2) - f(x_1)] / (x_2 - x_1)$$

e

$$\lambda = \frac{f(x_3) - 2f(x_2) + f(x_1)}{2(h)^2}$$

onde $h = (b - a) / 2$.

- Substituindo as constantes de volta e integrando $p(x)$ ao longo do intervalo $[a, b]$, obtém-se

$$I = \int_{x_1}^{x_3} f(x) dx \approx \int_{x_1}^{x_3} p(x) dx = \frac{h}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

Método de Simpson - Exemplo

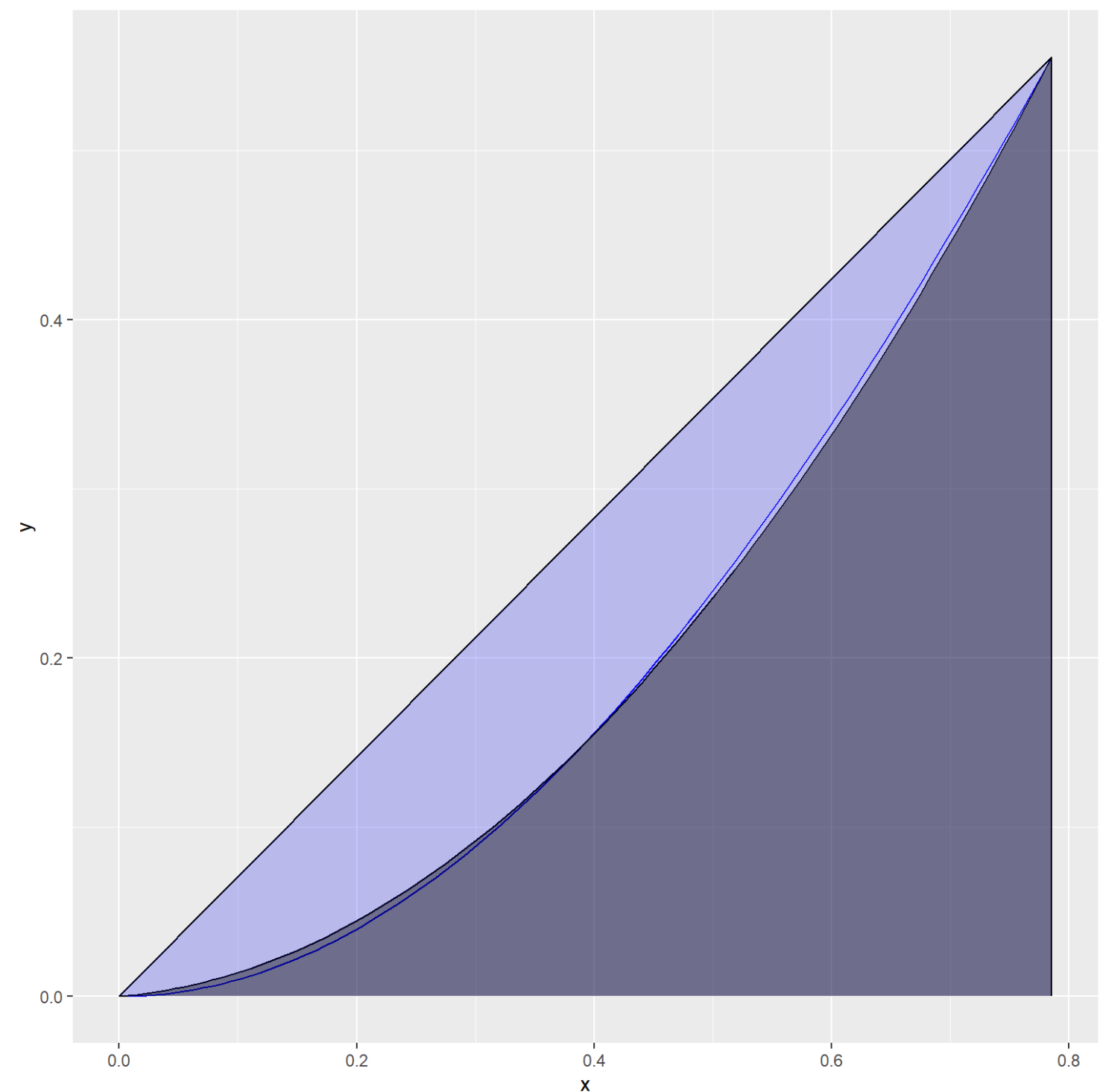
Se temos a seguinte função:

$$f(x) = x\sin(x).$$

E queremos resolver a seguinte integral:

$$I(f) = \int_0^{\pi/4} x\sin(x)dx.$$

Graficamente



Método de Simpson - Exemplo

- No R temos a seguinte função:

```
1 simpsons.rule <- function(f, a, b) {  
2   if (is.function(f) == FALSE) {  
3     stop('f deve ser uma funcao com argumento')  
4   }  
5   h <- (b - a) / 2  
6   x0 <- a  
7   x1 <- a + h  
8   x2 <- b  
9   s <- (h / 3) * (f(x0) + 4 * f(x1) + f(x2))  
10  return(s)  
11 }
```

- Dessa forma, a solução para a integral:

```
1 simpsons.rule(f, 0, pi/4) # Valor da integral
```

```
[1] 0.1513826
```

Quadratura de Gauss-Hermite

Quadratura de Gauss-Hermite

- Nos dois métodos de integração apresentados até agora, a integral de $f(x)$ ao longo do intervalo $[a, b]$ foi avaliada representando $f(x)$ como um polinômio de fácil integração.
- A integral é avaliada como uma soma ponderada dos valores de $f(x)$ nos diferentes pontos.
- A localização dos pontos comuns é predeterminada em um dos métodos de integração.
- Até agora os dois métodos consideram pontos igualmente espaçados.
- Na quadratura de Gauss, a integral também é avaliada usando uma soma ponderada dos valores de $f(x)$ em pontos distintos ao longo do intervalo $[a, b]$ (chamados pontos de Gauss).
- Estes pontos, contudo, não são igualmente espaçados e não incluem os pontos finais.

Quadratura de Gauss-Hermite

- O método de Gauss-Hermite é uma extensão do método de Quadratura Gaussiana para resolver integrais da forma:

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx.$$

- A integral é aproximada por uma soma ponderada, da função avaliada nos pontos de Gauss e pesos de integração:

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=1}^n \omega_i f(x_i),$$

onde n é o número de pontos usadas para a aproximação.

Quadratura de Gauss-Hermite

- Os x_i são as raízes do polinômio de Hermite $H_n(x)$ ($i = 1 < 2, \dots, n$) e os pesos w_i associados são dados por

$$w_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(x_i)]^2}$$

- Para a aproximação de integrais via o método de Gauss-Hermite precisamos dos pesos de integração w_i e dos pontos de Gauss x_i
- A função `gauss.quad()` do pacote `statmod` calcula os pesos e os pontos de Gauss-Hermite.

Quadratura de Gauss-Hermite

```
1 gauss.hermite <- function(integrando, n.pontos, ...){  
2   pontos <- gauss.quad(n.pontos, kind="hermite")  
3   integral <- sum(pontos$weights*integrando(pontos$nodes,...) / exp(-pontos$nodes^2))  
4   return(integral)  
5 }
```

- Esta função tem apenas dois argumentos, o primeiro é a função a ser integrada e o segundo o número de pontos a ser utilizado na aproximação.
- A segunda linha da função faz apenas uma soma ponderada da função avaliada nos pontos de Gauss.

Quadratura de Gauss-Hermite

- O método de Gauss-Hermite apresenta duas grandes limitações.
- A primeira está relacionada a escolha dos pontos de Gauss, que são escolhidos baseados em e^{-x^2} , independente da função $f(x)$ no integrando.
- Dependendo do suporte de $f(x)$, os pontos selecionados podem ou não estar dentro da área de interesse.
- Uma ideia natural é reescalonar os pontos de modo a colocá-los na área de maior densidade da função $f(x)$ o que gera o método chamada de Quadratura Adaptativa de Gauss-Hermite.

Quadratura de Gauss-Hermite - Exemplo

Se temos a seguinte função:

$$f(x) = x\sin(x).$$

E queremos resolver a seguinte integral:

$$I(f) = \int_0^{\pi/4} x\sin(x)dx.$$

```
1 gauss.hermite(f, n.pontos=50)
```

```
[1] 19.28854
```

Quadratura de Gauss-Hermite

- O segundo problema do método de Gauss-Hermite está relacionado com a dimensão da integral a ser resolvida.
- Quando a função é unidimensional, basta espalhar os pontos sobre a reta real e avaliar a função neste pontos.
- Para funções em duas ou mais dimensões precisamos do produto cartesiano dos pontos de integração para espalhar na função multidimensional, ou seja, o número de pontos cresce exponencialmente de acordo com a dimensão da função a ser integrada.

Quadratura de Gauss-Hermite

```
1 gauss.hermite.multi <- function(integrando,n.dim,n.pontos, ...){
2   normaliza <- function(x){exp(-t(as.numeric(x))%*%as.numeric(x))}
3   pontos <- gauss.quad(n.pontos,kind="hermite")
4   nodes <- matrix(rep(pontos$nodes,n.dim),ncol=n.dim)
5   pesos <- matrix(rep(pontos$weights,n.dim),ncol=n.dim)
6   lista.nodes <- lista.pesos <- list()
7   for(i in 1:ncol(nodes)){
8     lista.nodes[[i]] <- nodes[,i]
9     lista.pesos[[i]] <- pesos[,i]
10  }
11  nodes = as.matrix(do.call(expand.grid,lista.nodes))
12  pesos = do.call(expand.grid,lista.pesos)
13  pesos.grid = apply(pesos,1,prod)
14  norma = apply(nodes,1,normaliza)
15  integral <- sum(pesos.grid*(integrando(nodes)/norma))
```

Integração Monte Carlo

Integração Monte Carlo

Seja X uma variável aleatória com função densidade f . Considere o seguinte

Problema 1: Calcular

$$I = E(h(x)) = \int h(x)f(x)dx,$$

para alguma função $h: \mathbb{R} \rightarrow \mathbb{R}$ tal que $E(|h(X)|) < \infty$.

Integração Monte Carlo

- Em muitas situações é difícil fazer um cálculo exato da quantidade I .
- Uma alternativa usual consiste em estimar I através de um método computacional conhecido como integração Monte Carlo.
- A estratégia do método consiste em usar um gerador de números pseudo-aleatórios para obter uma amostra (X_1, \dots, X_n) i.i.d. da distribuição de X e estimar I pela média amostral

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^n h(X_i)$$

Integração Monte Carlo

- Não é difícil ver que \hat{I}_n possui as seguintes propriedades:
 1. $E(\hat{I}_n) = I$, para qualquer n fixo;
 2. $Var(\hat{I}_n) = \sigma^2 / n$, onde $Var(h(X)) = \sigma^2$;
 3. $\hat{I}_n \rightarrow I$;
 4. $\sqrt{n}(\hat{I}_n - I) / \sigma \xrightarrow{D} N(0, 1)$.
- Estas propriedades informam que \hat{I}_n é um estimador não viciado e consistente para I .
- Além disso, a última propriedade pode ser usada na construção de uma estimativa por intervalo para I .

Exemplo 1

- Estime

$$I = \int_0^1 2x dx.$$

- Observe que $I = E(2X)$, onde $X \sim U(0, 1)$, portanto, estima-se I por

$$\hat{I} = \frac{1}{n} \sum_{i=1}^n 2X_i,$$

onde $X_i \sim U(0, 1)$.

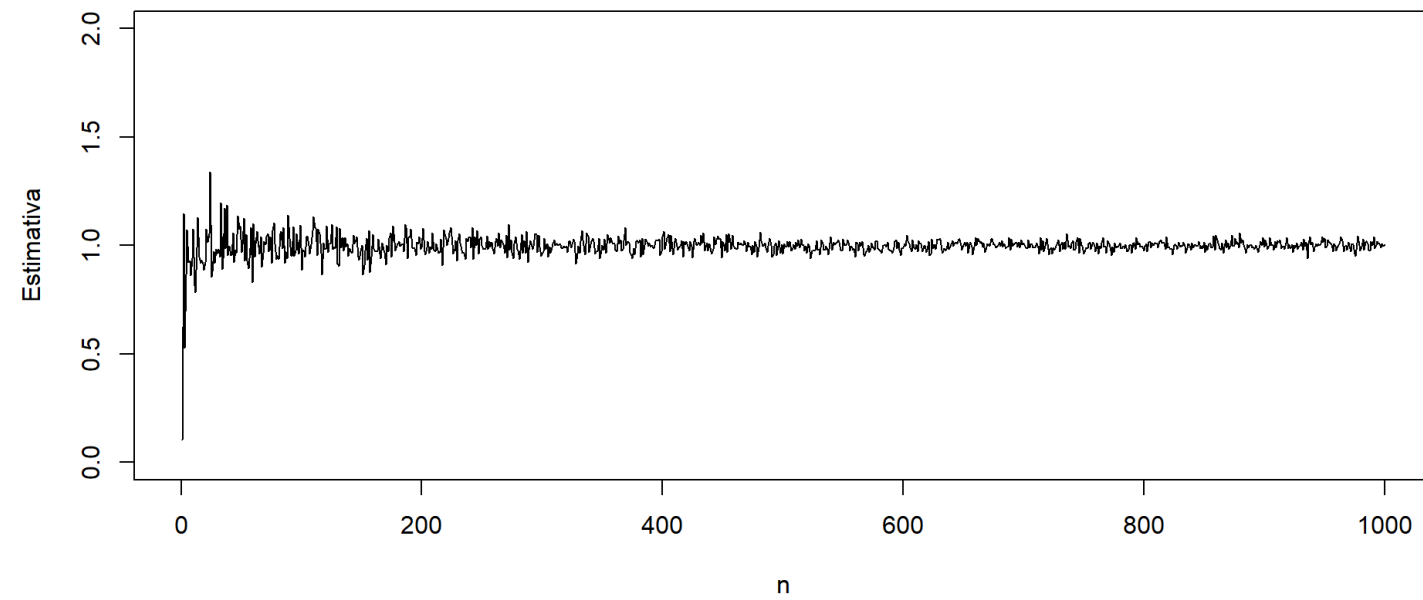
Exemplo 1

```
1 estimativa <- function(n) {  
2   u <- runif(n,0,1)  
3   h <- 2*u  
4   y <- mean(h)  
5   return(y)  
6 }  
7  
8 n.sizes <- 1:1000  
9 i.hat <- rep(0, length(n.sizes))  
10 for(k in 1:1000) {  
11   i.hat[k] <- estimativa(n=k)  
12 }  
13  
14 # O valor exato de I é 1  
15 erro <- (abs(i.hat - 1)) * 100
```

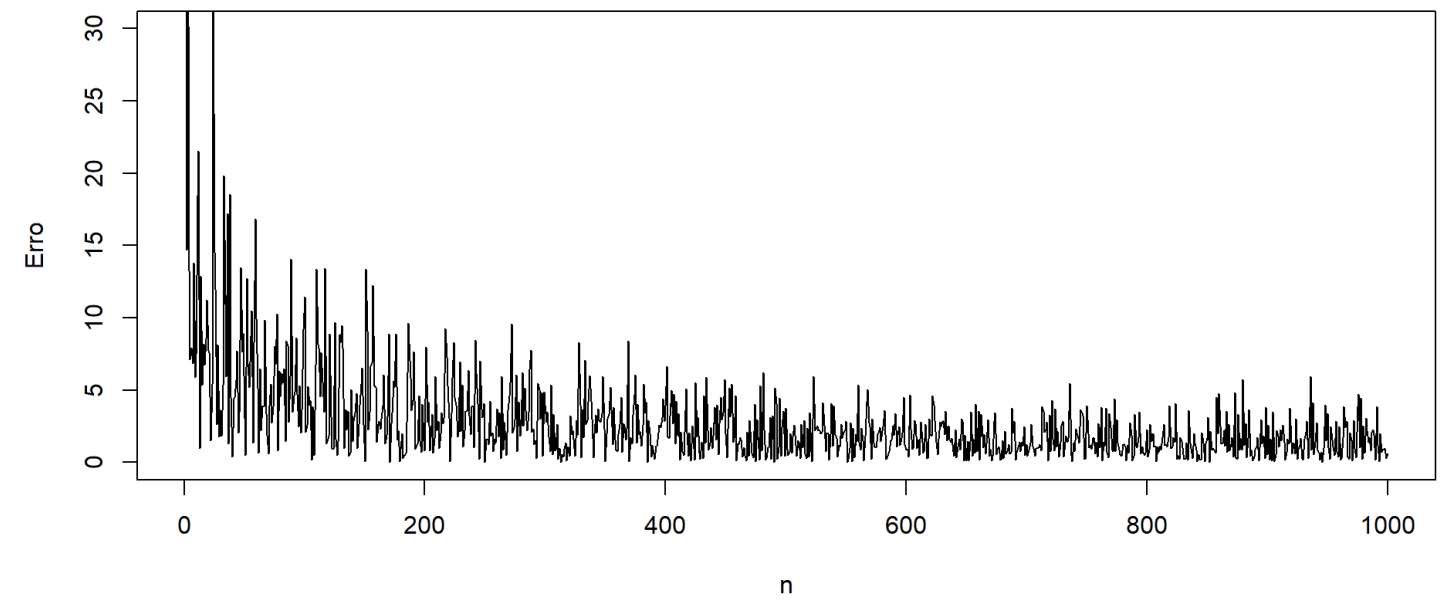
```
[1] 89.542541280 14.700418734 47.133549862 13.243799342 7.096297909  
[6] 7.902858228 6.870054806 13.782369572 5.876298269 7.506485782  
[11] 18.431612104 21.525793480 0.959062580 12.877131270 5.355658767  
[16] 8.202859835 6.742498542 8.551346878 11.225664150 7.756229795  
[21] 7.597426079 1.502962330 5.405895679 33.771649374 14.515947826  
[26] 11.965901551 2.634796366 8.127060787 1.759775811 3.643538263  
[31] 1.836423864 5.373637596 19.802172067 10.949293612 5.917013925  
[36] 17.191956523 1.310234401 18.517247310 4.506400162 0.375335501  
[41] 4.387496360 4.522529242 5.646861040 7.692077216 2.048418110  
[46] 3.484635922 13.484795457 7.624413473 8.921209548 3.031408574  
[51] 0.498722070 12.737345115 6.836233055 5.211202340 6.957872201  
[56] 10.463058583 1.352006899 8.281400557 16.833476351 9.918702382
```

Exemplo 1

```
1 plot(n.sizes,i.hat,type="l",xlab="n",  
2      ylab="Estimativa",ylim=c(0,2))
```



```
1 plot(n.sizes,erro,type="l",xlab="n",  
2      ylab="Erro",ylim=c(0,30))
```



Exemplo 2

- Estime

$$I = \int_a^b g(x) dx.$$

- Faça $y = (x - a)/(b - a)$ e $dy = dx/(b - a)$. Assim I pode ser escrita como

$$I = (b - a) \int_0^1 g((b - a)y + a) dy$$

.

- Portanto, estima-se I por

$$\hat{I} = \frac{b - a}{n} \sum_{i=1}^n g((b - a)X_i + a),$$

onde $X_i \sim U(0, 1)$.

Exemplo 3

- Estime

$$I = \int_0^{\infty} g(x) dx.$$

- Faça $y = \frac{1}{x-1}$ e $dy = \frac{-dx}{(x-1)^2} = -y^2 dx$. Assim I pode ser escrita como

$$I = \int_0^1 g\left(\frac{1-y}{y}\right) \frac{1}{y^2} dy.$$

- Portanto, estima-se I por

$$\hat{I} = \frac{1}{n} \sum_{i=1}^n g\left(\frac{1-X_i}{X_i}\right) \frac{1}{X_i^2},$$

onde $X_i \sim U(0, 1)$.