

Simulação

Ricardo Ehlers
ehlers@icmc.usp.br

Departamento de Matemática Aplicada e Estatística
Universidade de São Paulo

Cap. 11 Sheldon Ross, Introduction to Probability Models

- 1 Introdução
- 2 Simulando de Distribuições Discretas
- 3 Simulando de Distribuições Contínuas
- 4 Métodos de Aceitação/Rejeição
- 5 Gerando Vetores Aleatórios
- 6 Redução de Variância
 - Variáveis Antitéticas
 - Amostragem por Importância

Seja $\mathbf{X} = (X_1, \dots, X_n)$ um vetor aleatório com função de densidade $f(x_1, \dots, x_n)$ e deseja-se calcular,

$$E[g(\mathbf{X})] = \int \int \cdots \int g(x_1, \dots, x_n) f(x_1, \dots, x_n) dx_1 dx_2 \dots dx_n.$$

- Em muitas situações não é possível resolver esta integral analiticamente. Uma possibilidade é aproximar $E[g(\mathbf{X})]$ por simulação.
- Podemos tentar aproximar $E[g(\mathbf{X})]$ por simulação.

Para aproximar $E[g(\mathbf{X})]$,

- simule valores,

$$\mathbf{X}^{(1)} = (x_1^{(1)}, \dots, x_n^{(1)})$$

$$\mathbf{X}^{(2)} = (x_1^{(2)}, \dots, x_n^{(2)})$$

$$\vdots$$

$$\mathbf{X}^{(r)} = (x_1^{(r)}, \dots, x_n^{(r)})$$

- calcule $Y^{(i)} = g(\mathbf{X}^{(i)}), i = 1, \dots, r.$

Se $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots$ forem simulados de forma independente,

- Pela lei forte dos grandes números temos que,

$$\lim_{r \rightarrow \infty} \frac{1}{r} \sum_{i=1}^r Y^{(i)} = E[g(\mathbf{X})].$$

- Para r finito podemos usar então a média aritmética dos Y 's simulados como aproximação para $E[g(\mathbf{X})]$.

- Como simular os vetores $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(r)}$?
- Os elementos de \mathbf{X} podem ser variáveis aleatórias dependentes.
- A distribuição de \mathbf{X} pode ser conhecida apenas parcialmente,

$$f(\mathbf{x}) = Cf^*(\mathbf{x}),$$

sendo $f^*(\cdot)$ conhecida e a constante C desconhecida.

Note que,

$$C^{-1} = \int \int \cdots \int f^*(\mathbf{x}) dx_1 dx_2 \dots dx_n.$$

- Na prática são simulados no computador números pseudo-aleatórios ao invés de realmente aleatórios.

$$X_{n+1} = (aX_n + c) \text{ modulo } m, \quad n \geq 0.$$

- X_{n+1} é o resto da divisão $aX_n + c$ por m .
- Cada X_n assume valores $\{0, 1, \dots, m-1\}$
- X_n/m é um valor aproximado da distribuição $U(0, 1)$.

Assume-se que um gerador de números aleatórios da distribuição $U(0, 1)$ está disponível.

Exemplo. Seja U uma variável aleatória com distribuição uniforme continua no intervalo $(0,1)$. Então, para um inteiro positivo m

$$X = mU \sim U(0, m)$$

Portanto,

$$\begin{aligned} P(i-1 < X < i) &= P(X < i) - P(X < i-1) \\ &= \frac{i}{m} - \frac{i-1}{m} = \frac{1}{m}. \end{aligned}$$

A variável aleatória discreta $I = [X] + 1$ é tal que,

$$P(I = i) = P([X] = i-1) = P(i-1 < X < i) = \frac{1}{m}.$$

Então I tem distribuição Uniforme discreta nos inteiros $\{1, 2, \dots, m\}$.

Portanto, para gerar um valor da distribuição uniforme discreta em $\{1, 2, \dots, m\}$.

- 1 Gere um valor u da distribuição $U(0, 1)$.
- 2 Retorne o valor $1 + [mu]$

Para gerar uma amostra I_1, \dots, I_n desta distribuição repita os passos 1 e 2 n vezes de forma independente.

Gerando 50 valores da distribuição uniforme discreta em $\{1, 2, \dots, 20\}$.

```
> m = 20
```

```
> n = 50
```

```
> u = runif(n, 0, 1)
```

```
> x = m*u
```

```
> i = floor(x) + 1
```

```
> print(i)
```

```
[1] 16 15 14  5  4 16  7  3 18 14 17 13  9  9 13 17 10 12 16  7  4  
[26] 10 10 19 18  2 16 10 13 19  2 15  7 15  1 12 13 15  6  5  9 16
```

Simulando de Distribuições Discretas

Assume-se que um gerador de números aleatórios da distribuição $U(0, 1)$ está disponível.

Seja X uma variável aleatória discreta em $\{x_1, \dots, x_k\}$ com

$$P(X = x_i) = p_i \quad \text{e} \quad \sum_{i=1}^k p_i = 1.$$

Defina,

- $F_i = p_1 + \dots + p_i$, $i = 1, \dots, k$, sendo $F_0 = 0$,
- $I_i = (F_{i-1}, F_i]$.

Propriedade

Se $U \sim U(0, 1)$ então $P(U \in I_i) = F_i - F_{i-1} = P(X = x_i)$.

Portanto, gerando um valor u da distribuição $U(0, 1)$ e verificando a qual subintervalo I_i ele pertence temos um valor simulado de X .

- 1 Gere um valor u da distribuição $U(0, 1)$.
- 2 Para $i = 1, \dots, k$,
 - ▶ se $u \in I_i$ retorne $x = x_i$.
 - ▶ caso contrário faça $i = i + 1$ e repita.

Pense sobre a (in)eficiência computacional deste algoritmo.

Exemplo. Seja uma variável aleatória X assumindo valores em $\{1, 2, 3, 4, 5\}$ com probabilidades iguais a $1/5$. Neste caso,

$$F_0 = 0, F_1 = 1/5, F_2 = 2/5, F_3 = 3/5, F_4 = 4/5, F_5 = 1$$

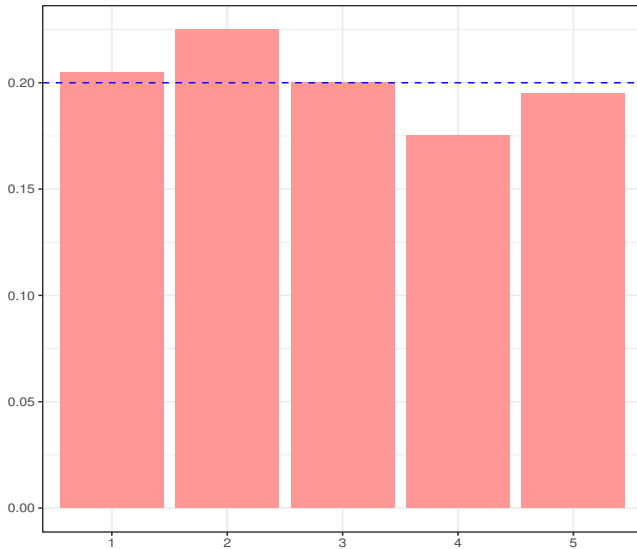
e os intervalos são,

$$\begin{aligned} I_1 &= (0, 1/5], I_2 = (1/5, 2/5], \\ I_3 &= (2/5, 3/5], I_4 = (3/5, 4/5], \\ I_5 &= (4/5, 1]. \end{aligned}$$

Função em R para executar o algoritmo anterior.

```
> sunif <- function(x,prob) {  
+   k = length(x)  
+   F = array(0,k+1)  
+   F[1] = 0  
+   for (i in 1:k) {  
+     F[i+1] = F[i] + prob[i]  
+   }  
+   u = runif(n=1, 0, 1)  
+   for (i in 1:k) {  
+     if (F[i] < u & u <= F[i+1]) {  
+       return(x[i])  
+       stop  
+     }  
+   }  
+ }
```

Frequências de 200 valores simulados.

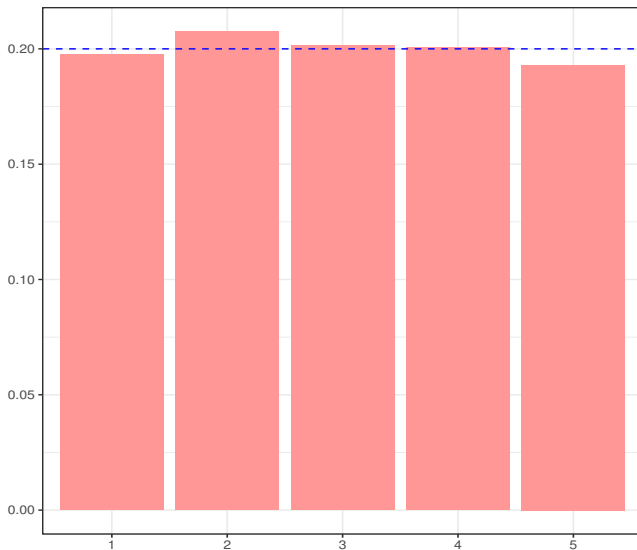


Comandos do R.

```
> x = 1:5
> p = rep(0.2,5)
> n = 200
> x.sim = array(0,n)
> set.seed(1234)
> for (j in 1:n) x.sim[j] = sunif(x=x, prob=p)

> tab= table(x.sim)
> sp = data.frame(tab/n)
> library(ggplot2)
> ggplot(data=sp,aes(x=x.sim,y=Freq))+
+   geom_bar(stat="identity", fill="#FF9999")+
+   theme_bw()+labs(x="",y="")+
+   geom_hline(yintercept=0.2,colour="blue",lty=2)
```


Frequencias de 2000 valores simulados.



Distribuição Binomial

Se X_1, \dots, X_n são variáveis aleatórias independentes tais que $X_i \sim \text{Bernoulli}(p)$ então

$$Y = \sum_{i=1}^n X_i \sim \text{Binomial}(n, p).$$

Deseja-se gerar valores para Y .

Para isso, primeiro precisamos gerar os n valores de X usando o algoritmo anterior.

Os intervalos são $I_1 = (0, p]$ e $I_2 = (p, 1]$ portanto,

- 1 Simule valores $u_1, \dots, u_n \sim U(0, 1)$.
- 2 Para $i = 1, \dots, n$, se $u_i \in (0, p]$ o valor simulado é $x_i = 1$. Caso contrário, $x_i = 0$.
- 3 Faça $y = \sum_{i=1}^n x_i$.

Função em R para executar o algoritmo anterior.

```
> sbinom <- function(n,p){  
+   u = runif(n,0,1)  
+   x = u <= p  
+   y = sum(x)  
+   return(y)  
+ }
```

Exemplo. Gerando $m = 100$ valores da distribuição binomial com parâmetros $n = 20$ e $p = 1/2$.

Valores simulados e suas frequências,

y	Freq
3	1
4	1
6	4
7	5
8	16
9	18
10	14
11	11
12	16
13	10
14	2
15	2

Comandos do R.

```
> m = 100  
> y = array(m)  
> set.seed(1234)  
> for (j in 1:m) y[j] = rbinom(n = 20, p = 0.5)  
  
> library(xtable)  
> mat= as.data.frame(table(y))  
> a= xtable(mat)  
> print(a,include.rownames=F)
```

Distribuição Geométrica

Seja uma variável aleatória $X \sim Geometrica(p)$ com função de probabilidade,

$$P(X = i) = p(1 - p)^{i-1}, \quad i = 1, 2, \dots$$

X é o número de ensaios de Bernoulli até ocorrer o primeiro sucesso.

Para obter os intervalos precisamos calcular,

$$F_{j-1} = \sum_{i=1}^{j-1} P(X = i) = 1 - P(X \geq j)$$

Note que,

$$\begin{aligned}P(X \geq j) &= p(1-p)^{j-1}[1 + (1-p) + (1-p)^2 + \dots] \\&= (1-p)^{j-1},\end{aligned}$$

e portanto,

$$F_{j-1} = \sum_{i=1}^{j-1} P(X = i) = 1 - P(X \geq j) = 1 - (1-p)^{j-1}.$$

Portanto podemos simular $X \sim \text{Geometrica}(p)$ como,

- 1 Gere um valor u da distribuição $U(0, 1)$.
- 2 Retorne o valor $X = j$ tal que,

$$1 - (1-p)^{j-1} < u < 1 - (1-p)^j$$

Equivalentemente, temos que $X = j$ se,

$$(1 - p)^j < 1 - u < (1 - p)^{j-1}$$

e como $1 - U$ e U tem a mesma distribuição $U(0, 1)$,

$$\begin{aligned} X &= \min\{j : (1 - p)^j < U\} \\ &= \min\left\{j : j > \frac{\log(u)}{\log(1 - p)}\right\} \\ &= 1 + \left\lceil \frac{\log(u)}{\log(1 - p)} \right\rceil. \end{aligned}$$

Simulando de Distribuições Contínuas

Método da Transformação Inversa

Seja X uma variável aleatória contínua com função de distribuição $F_X(x) = P(X \leq x)$. Então, $Y = F(X) \in (0, 1)$ tem função de distribuição dada por,

$$G_Y(y) = y$$

$$\begin{aligned} G_Y(y) &= P(Y \leq y) \\ &= P(F_X(X) \leq y) \\ &= P(X \leq F_X^{-1}(y)) \\ &= F_X(F_X^{-1}(y)) = y. \end{aligned}$$

Conclui-se então que $G_Y(y) = y$ e $Y \sim U(0, 1)$. Portanto, para gerar um valor de X podemos usar o seguinte algoritmo,

- 1 Gere um valor y da distribuição $U(0, 1)$.
- 2 Aplique a transformação $x = F_X^{-1}(y)$.

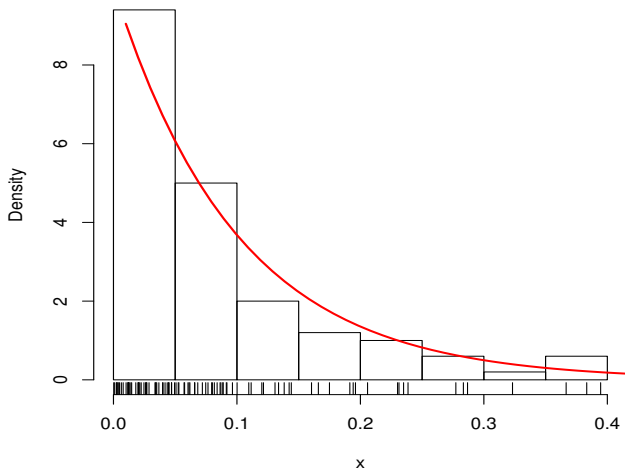
Exemplo. Simulando valores da distribuição Exponencial. Seja $X \sim \text{Exp}(\lambda)$. Então,

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0$$

$$F(x) = 1 - \exp(-\lambda x), \quad x > 0$$

$$F^{-1}(u) = -\frac{1}{\lambda} \log(1 - u), \quad 0 < u < 1.$$

Simulando $n = 100$ valores da distribuição exponencial com média 0.1.



Comandos do R.

```
> sexp <- function(n,lambda){  
+   u = runif(n)  
+   x = -(1/lambda)* log(1-u)  
+   return(x)  
+ }  
  
> n= 100  
> lambda=10  
> x = sexp(n=100,lambda=10)  
  
> hist(x, main="", prob=T)  
> rug(x)  
> curve(dexp(x,10),from=0.01,to=1,add=T,col=2,lwd=2)
```

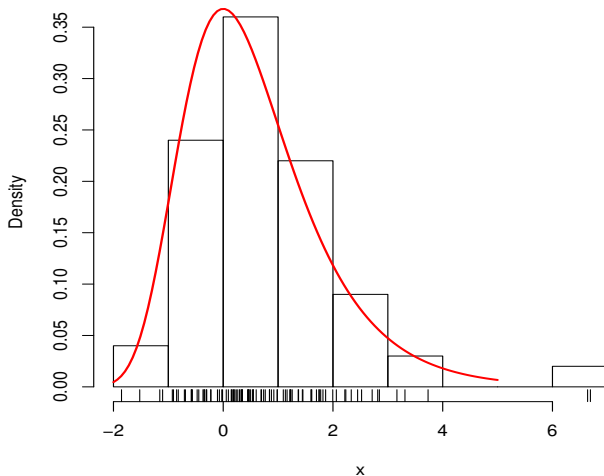
Exemplo. Simulando valores da distribuição de Gumbel. Seja X uma variável aleatória com distribuição Gumbel com locação μ e escala σ . Então,

$$f(x) = \frac{1}{\sigma} \exp \left\{ - \left(\frac{x - \mu}{\sigma} \right) - \exp \left(- \frac{x - \mu}{\sigma} \right) \right\}$$

$$F(x) = \exp \left\{ - \exp \left(- \frac{x - \mu}{\sigma} \right) \right\}$$

$$F^{-1}(u) = \mu - \sigma \log(-\log u).$$

100 valores simulados da distribuição Gumbel com $\mu = 0$ e $\sigma = 1$.



Comandos do R.

```
> sgumbel <- function(n,mu,sigma) {  
+   u = runif(n)  
+   x = mu-sigma*log(-log(u))  
+   return(x)  
+ }  
  
> dgumbel <- function(x,mu,sigma) {  
+   z = (x-mu)/sigma  
+   ldens = -z-exp(-z)-log(sigma)  
+   return(exp(ldens))  
+ }  
  
> x = sgumbel(n=100,mu=0,sigma=1)  
> hist(x, main="", prob=T)  
> rug(x)  
> curve(dgumbel(x,0,1),from=-2,to=5,add=T,col=2,lwd=2)
```

Exemplo. Seja X uma variável aleatória contínua com distribuição Logística com parâmetro de locação μ e escala σ .

Para simular valores de X podemos usar o método da inversão pois,

$$F(x) = \frac{1}{1 + e^{-(x-\mu)/\sigma}} = \frac{e^{(x-\mu)/\sigma}}{1 + e^{(x-\mu)/\sigma}}.$$

e podemos verificar que,

$$F^{-1}(u) = \mu + \sigma \log \left(\frac{u}{1-u} \right).$$

Exemplo. Seja X uma variável aleatória contínua com distribuição Cauchy com parâmetro de locação μ e escala σ .

Para simular valores de X podemos usar o método da inversão.
A função de densidade é dada por,

$$f(x) = \frac{1}{\pi\sigma} \frac{1}{1 + \left(\frac{x-\mu}{\sigma}\right)^2}, \quad x \in \mathbb{R}.$$

e sua função de distribuição acumulada e

$$F(x) = \frac{1}{2} + \frac{1}{\pi} \arctan \left(\frac{x - \mu}{\sigma} \right).$$

- O método de inversão só é realmente útil se a inversa da função de distribuição for fácil de ser obtida e calculada.
- Por exemplo, a função de distribuição inversa Φ^{-1} da distribuição normal padrão não pode ser obtida analiticamente e sua avaliação numérica é lenta.
- Existem métodos mais específicos e eficientes.

Transformações de Variáveis Contínuas

Suponha que foram gerados n valores de uma variável aleatória contínua X de forma independente.

Então x_1, \dots, x_n formam uma amostra da distribuição de X .

Obtém-se uma amostra y_1, \dots, y_n da distribuição de $Y = h(X)$ para qualquer transformação inversível $h(\cdot)$ como,

$$y_1 = h(x_1), y_2 = h(x_2), \dots, y_n = h(x_n).$$

Exemplo. Suponha que temos uma amostra de valores simulados x_1, \dots, x_n da distribuição $N(0, 1)$. Segue então que,

$$\mu + \sigma x_1, \mu + \sigma x_2, \dots, \mu + \sigma x_n$$

é uma amostra de valores simulados da distribuição $N(\mu, \sigma^2)$.

Exemplo. Uma variável aleatória Y tem distribuição log-normal com parâmetros μ e σ^2 se e somente se,

$$\log(Y) \sim N(\mu, \sigma^2).$$

Portanto, se tivermos uma amostra de valores simulados x_1, \dots, x_n da distribuição $N(\mu, \sigma^2)$ segue que,

$$\exp(x_1), \dots, \exp(x_n)$$

é uma amostra de valores simulados da distribuição log-normal com parâmetros μ e σ^2 .

Métodos de Aceitação/Rejeição

Suponha que,

- Sabemos gerar valores de uma variável aleatória Y com função de densidade $g(y)$.
- Porém queremos gerar valores de X com função de densidade $f(x)$.

O método da rejeição consiste em,

- 1 Gerar um valor *candidato* y da distribuição g .
- 2 Aceitar o valor gerado com probabilidade proporcional a $f(y)/g(y)$.

Seja c uma constante positiva tal que

$$\frac{f(y)}{g(y)} \leq c \quad \forall y.$$

Então, a probabilidade de aceitação é

$$P(\text{aceitar } y) = \frac{f(y)}{c g(y)}.$$

Podemos usar o seguinte algoritmo para gerar valores de X ,

- 1 Gere um valor y da distribuição g .
- 2 Gere $u \sim U(0, 1)$.
- 3 Se $u \leq \frac{f(y)}{c g(y)}$ faça $x = y$.
- 4 Caso contrário retorne ao passo 1.

A probabilidade global de aceitação é,

$$\begin{aligned}P\left(U \leq \frac{f(Y)}{cg(Y)}\right) &= \int P\left(U \leq \frac{f(Y)}{cg(Y)} \middle| Y = y\right) g(y) dy \\&= \int \frac{f(y)}{cg(y)} g(y) dy \\&= \frac{1}{c} \int f(y) dy \\&= \frac{1}{c}\end{aligned}$$

que é independente do valor simulado y .

Seja N o número de valores candidatos propostos até a aceitação.

- Então N tem distribuição Geométrica com parâmetro $p = 1/c$.
- Portanto,

$$P(N = k) = (1 - p)^{k-1}p = \left(1 - \frac{1}{c}\right)^{k-1} \frac{1}{c}, \quad k = 1, 2, \dots$$

$$E(N) = \frac{1}{p} = c.$$

Observações,

- A distribuição auxiliar g deve ser fácil de ser simulada.
- As densidades f e g devem ter o mesmo suporte, i.e. $g(x) > 0$ nos mesmos valores em que $f(x) > 0$.
- Pode ser necessário gerar muitos candidatos até aceitar um valor proposto.
- Quanto mais próximas forem f e g mais eficiente será o algoritmo.
- $cg(x)$ é chamado de *envelope* de $f(x)$.

Conclui-se então que o método estará correto para qualquer constante c tal que,

$$\frac{f(x)}{g(x)} \leq c, \quad \forall x.$$

- Na prática devemos procurar o menor valor possível de c para tentar obter eficiencia computacional.
- Matematicamente temos que,

$$c = \sup_x \left\{ \frac{f(x)}{g(x)} \right\}.$$

Exemplo. Deseja-se gerar um valor de X cuja função de densidade é,

$$f(x) = 20x(1 - x)^3, \quad x \in (0, 1).$$

Considere o método de rejeição com $g(x) = 1, x \in (0, 1)$.

Qual o menor valor da constante c tal que $\frac{f(y)}{g(y)} \leq c$?

Note que,

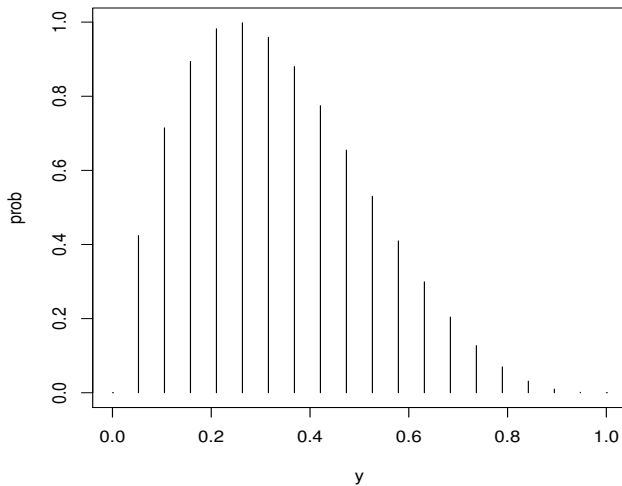
$$\frac{d}{dx} \left(\frac{f(x)}{g(x)} \right) = 0 \iff x = \frac{1}{4},$$

portanto $c = f(1/4) = \frac{135}{64}$ garante $f(y) \leq c, \forall y$

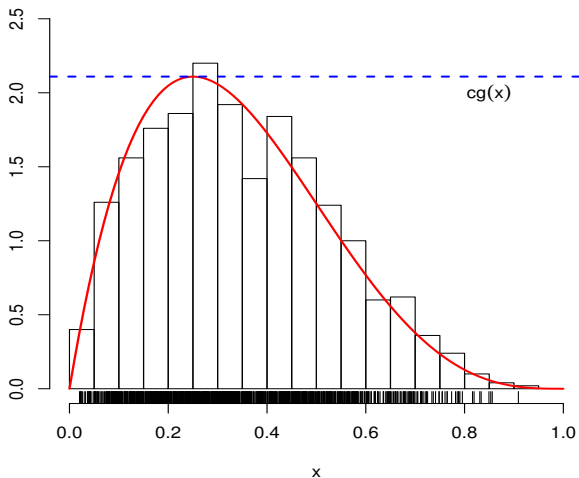
O algoritmo de rejeição neste exemplo fica,

- 1 Gere y da distribuição $U(0, 1)$ (esta é a distribuição auxiliar).
- 2 Gere u da distribuição $U(0, 1)$.
- 3 Se $u \leq \frac{20y(1-y)^3}{135/64}$ faça $x = y$, caso contrário retorne ao passo 1.

Probabilidades de aceitação em uma grade de valores $y \in \{0, 1\}$.



1000 valores simulados de X .



```

> sbeta <- function(n=100){
+   x= NULL
+   c= 135/64
+   for (i in 1:n){
+     accept = FALSE
+     while (!accept){
+       y = runif(1)
+       u = runif(1)
+       if (u < 20*y*(1-y)^3/c) {
+         x = c(x,y)
+         accept=TRUE
+       }
+     }
+   }
+   return(x)
+ }

```

Exemplo. Gerar valores de $X \sim \text{Beta}(a, b)$ usando a distribuição $U(0, 1)$ como auxiliar.

A função de densidade de X é,

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \quad a > 0, \quad b > 0, \quad x \in (0, 1),$$

a a densidade auxiliar é $g(x) = 1, x \in (0, 1)$.

Portanto o ponto x que maximiza $f(x)/g(x)$ é solução de

$$\frac{d \log(f(x))}{dx} = \frac{a-1}{x} - \frac{b-1}{1-x} = 0.$$

Resolvendo a equação obtemos,

$$x = \frac{a-1}{a+b-2},$$

e podemos usar,

$$\begin{aligned} c &= f((a-1)/(a+b-2)) \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \left(\frac{a-1}{a+b-2} \right)^{a-1} \left(\frac{b-1}{a+b-2} \right)^{b-1} \end{aligned}$$

no algoritmo de rejeição.

1 Gere y da distribuição $U(0, 1)$ (distribuição auxiliar).

2 Gere u da distribuição $U(0, 1)$.

3 Se

$$u \leq \frac{f(y)}{c}$$

faça $x = y$, caso contrário retorne ao passo 1.

No passo 3,

$$\frac{f(y)}{c} = \left(\frac{y}{x}\right)^{a-1} \left(\frac{1-y}{1-x}\right)^{b-1},$$

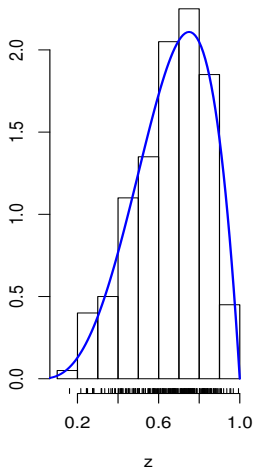
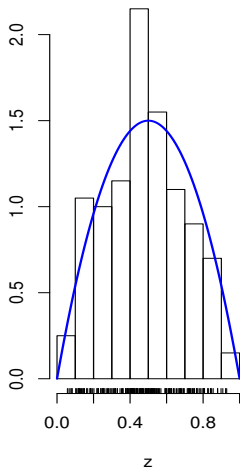
sendo $x = \frac{a-1}{a+b-2}$.

```

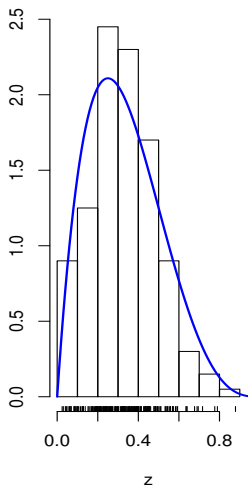
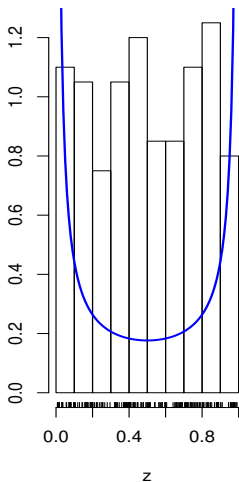
> sbeta <- function(n,a,b){
+   x= array(n)
+   c= dbeta((a-1)/(a+b-2), shape1=a,shape2=b)
+   for (i in 1:n){
+     accept = FALSE
+     while (!accept){
+       y = runif(1)
+       u = runif(1)
+       prob = dbeta(y, shape1 = a, shape2 = b)/c
+       if (u < prob) {
+         x[i] = y
+         accept=TRUE
+       }
+     }
+   }
+   x
+ }

```

200 valores simulados da $\text{Beta}(2,2)$ e $\text{Beta}(4,2)$.



200 valores simulados da $\text{Beta}(0.1,0.1)$ e $\text{Beta}(2,4)$.



O que deu errado com a distribuição Beta(0.1,0.1)?

- O valor $x = (a - 1)/(a + b - 2)$ utilizado na função sbeta é o ponto de máximo somente se $a > 1$ e $b > 1$.
- Quando $a < 1$ e $b < 1$ este é o ponto que minimiza $f(x)/g(x)$.

Exemplo. Simulando um valor da distribuição normal padrão usando a distribuição Cauchy como auxiliar.

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$
$$g(x) = \frac{1}{\pi(1+x^2)}$$

Para $c = \sqrt{2\pi} \exp(-1/2)$ tem-se que $\frac{f(x)}{g(x)} \leq c, \forall x$.

1 Gere $y \sim \text{Cauchy}$

2 Calcule,

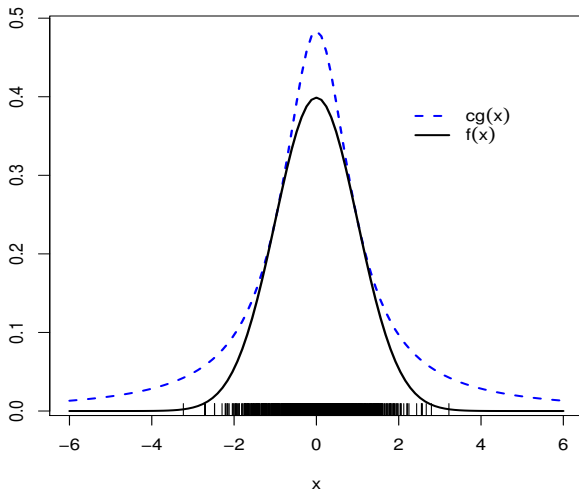
$$p = \frac{f(x)}{cg(x)} = \frac{\pi}{2}(1 + y^2) \exp[-(y^2 + 1)/2]$$

3 Gere $u \sim U(0, 1)$,

4 Se $u < p$ aceite y como um valor simulado de X ,

5 Caso contrário, retorne ao passo 1.

1000 valores simulados da $N(0,1)$.



```

> scauchy <- function(n=100) {
+   x= NULL
+   c= sqrt(2*pi)*exp(-0.5)
+   for (i in 1:n) {
+     accept= FALSE
+     while (!accept) {
+       y= rt(1,df=1)
+       prob= dnorm(y,0,1)/ (c*dt(y,df=1))
+       u= runif(1)
+       if (u < prob) {
+         x= c(x,y)
+         accept= TRUE
+       }
+     }
+   }
+   return(x)
+ }

```

Simulando da Distribuição Normal

Sejam X e Y independentes com distribuição normal padrão. A função de densidade conjunta então é,

$$p(x, y) = p(x)p(y) = \frac{1}{2\pi} \exp(-(x^2 + y^2)/2), \quad x \in \mathbb{R}, \quad y \in \mathbb{R}.$$

Reescrevendo em coordenadas polares,

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \theta &= \tan^{-1}(y/x) \end{aligned}$$

e a transformação inversa fica,

$$\begin{aligned} x &= r \cos(\theta) \\ y &= r \sin(\theta). \end{aligned}$$

A função de densidade conjunta de (R, Θ) então fica,

$$p(r, \theta) = p(x, y) \left| \frac{\partial(x, y)}{\partial(r, \theta)} \right|$$

sendo

$$J = \frac{\partial(x, y)}{\partial(r, \theta)} = \begin{pmatrix} \cos(\theta) & -r \sin(\theta) \\ \sin(\theta) & r \cos(\theta) \end{pmatrix}$$

Portanto,

$$p(r, \theta) = \frac{1}{2\pi} \exp(-r^2/2) r.$$

Além disso, R e Θ são independentes sendo,

$$p(\Theta) = \frac{1}{2\pi} \quad \text{e} \quad p(r) = r e^{-r^2/2}$$

e pode-se verificar que

$$\begin{aligned}\Theta &\sim U(0, 2\pi) \\ R^2 &\sim \text{Exp}(1/2)\end{aligned}$$

pois

$$p(r^2) = \frac{1}{2} \exp(-r^2/2).$$

Temos assim um algoritmo para simular 2 variáveis aleatórias independentes com distribuição normal padrão.

- 1 Gere r^2 da distribuição exponencial com média 2.
- 2 Gere θ da distribuição $U(0, 2\pi)$.
- 3 Faça $x = r \cos(\theta)$ e $y = r \sin(\theta)$.

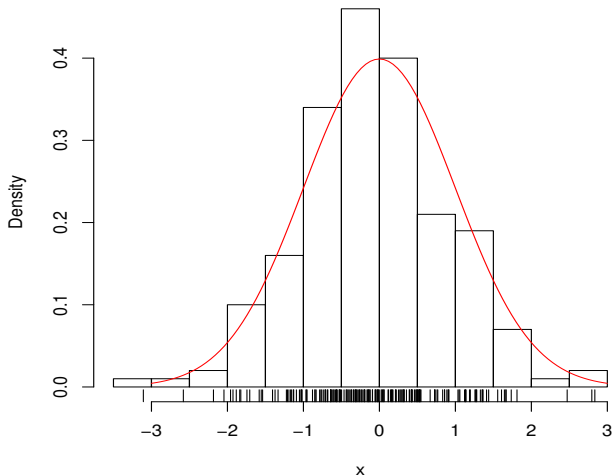
Este algoritmo é chamado método de *algoritmo de Box-Muller*.

Se quisermos simular um valor da distribuição $N(\mu, \sigma^2)$ basta usar este algoritmo e aplicar as transformações

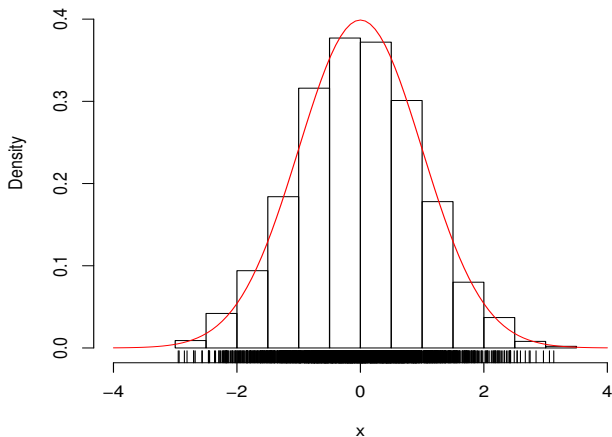
$$\mu + \sigma x \quad \text{e} \quad \mu + \sigma y.$$

```
> snorm <- function(n, mu=0, sd=1){  
+   r2 = sexp(n, lambda=0.5)  
+   r = sqrt(r2)  
+   theta = 2*pi*runif(n)  
+   x = r * cos(theta)  
+   y = r * sin(theta)  
+   return(sd*x + mu)  
+ }
```

Histograma de 200 valores simulados da distribuição normal padrão com a função de densidade superposta.



Histograma de 2000 valores simulados da distribuição normal padrão com a função de densidade superposta.



Simulando da Distribuição Qui-quadrado

Se $Z_1, \dots, Z_n \sim N(0, 1)$ são independentes então

$$X = Z_1^2 + \dots + Z_n^2 = \sum_{i=1}^n Z_i^2 \sim \chi_n^2.$$

e poderíamos usar o algoritmo anterior para gerar um valor de X .

Porém,

$$Z_i^2 \sim \chi_1^2 \text{ ou } \text{Gama}\left(\frac{1}{2}, \frac{1}{2}\right)$$

$$Z_j^2 \sim \chi_1^2 \text{ ou } \text{Gama}\left(\frac{1}{2}, \frac{1}{2}\right)$$

$$Z_i^2 + Z_j^2 \sim \text{Gama}\left(1, \frac{1}{2}\right) \text{ ou } \text{Exp}\left(\frac{1}{2}\right)$$

para quaisquer i, j .

Então, para um número par de graus de liberdade $n = 2k$,

$$X = (Z_1^2 + Z_2^2) + \cdots + (Z_{2k-1}^2 + Z_{2k}^2)$$

é uma soma de k variáveis aleatórias independentes cada uma com distribuição exponencial com média 2.

Portanto,

$$X \sim \text{Gama} \left(k, \frac{1}{2} \right) \text{ ou } X \sim \chi_{2k}^2.$$

Podemos usar então o seguinte algoritmo para gerar um valor de X ,

- 1 Gere valores u_1, \dots, u_k independentes da distribuição $U(0, 1)$.
- 2 Faça $x = -2 \sum_{i=1}^k \log u_i$.

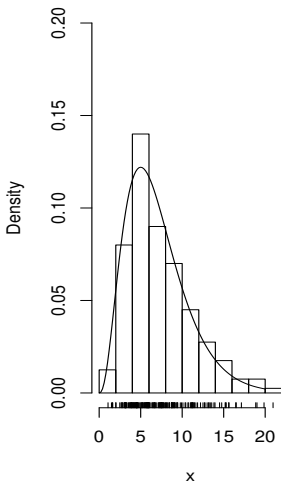
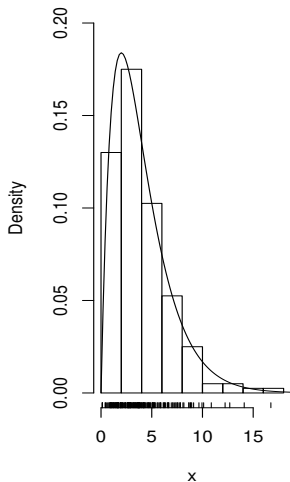
O valor obtido de x é um valor simulado da distribuição qui-quadrado com $2k$ graus de liberdade (embora tenhamos simulado somente k valores).

Para um número ímpar de graus de liberdade $n = 2k + 1$,

- 1 Gere valores u_1, \dots, u_k independentes da distribuição $U(0, 1)$.
- 2 Gere $z \sim N(0, 1)$
- 3 Faça $x = z^2 - 2 \sum_{i=1}^k \log u_i$.

```
> sqq <- function(n,df){  
+   x = rep(0,n)  
+   k = floor(df/2)  
+   if (floor(df/2) != ceiling(df/2))  
+     z = snorm(n) else z = rep(0,n)  
+   for (i in 1:n) {  
+     u = runif(k)  
+     x[i] = -2 * sum(log(u))  
+   }  
+   return(x + z^2)  
+ }
```

Histogramas de 200 valores simulados de distribuições qui-quadrado com 4 e 7 graus de liberdade.



Gerando Vetores Aleatórios

Seja $\mathbf{X} = (X_1, \dots, X_n)$ um vetor aleatório de dimensão n . Uma possível forma de escrever sua função de densidade conjunta é,

$$f(\mathbf{x}) = f(x_1)f(x_2|x_1)f(x_3|x_2, x_1) \dots f(x_n|x_{n-1}, \dots, x_1)$$

Então um algoritmo para simular um valor do vetor \mathbf{X} é,

- 1 Simule um valor x_1 da distribuição de X_1 ,
- 2 Simule um valor x_2 da distribuição condicional de $X_2|X_1 = x_1$
- 3 Simule um valor x_3 da distribuição condicional de $X_3|X_2 = x_2, X_1 = x_1$
- 4 ...
- 5 Simule um valor x_n da distribuição condicional de $X_n|X_{n-1} = x_{n-1}, \dots, X_1 = x_1$

Exemplo. Deseja-se simular valores do vetor aleatório (X, Y, Z) . Um possível algoritmo é,

- 1 Simule um valor x da distribuição de X ,
- 2 Simule um valor y da distribuição condicional de $Y|X = x$
- 3 Simule um valor z da distribuição condicional de $Z|Y = y, X = x$

- Assume-se que sabemos simular valores das distribuições condicionais.
- Os componentes não precisam ser escalares, podem ser vetores ou matrizes.

Exemplo. Simulando de misturas de distribuições. Seja a variável aleatória contínua X com função de densidade,

$$f(x) = \sum_{j=1}^k p_j f_j(x),$$

sendo

$$p_j > 0 \quad \text{e} \quad \sum_{j=1}^k p_j = 1.$$

Neste caso, deve-se simular valores do vetor (X, j) sendo j o indicador de componente da mistura.

Portanto, o seguinte algoritmo pode ser usado,

- 1 Gere j de uma distribuição discreta finita com probabilidades p_j ,
- 2 Gere x da distribuição cuja densidade é f_j .
- 3 Retorne x como um valor simulado da mistura.

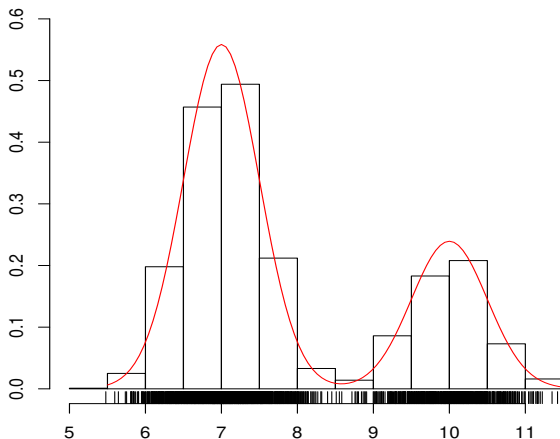
Exemplo. Deseja-se gerar valores de uma variável aleatória X cuja distribuição é uma mistura de distribuições normais.

$$X \sim p N(\mu, \tau^2) + (1 - p)N(\theta, \sigma^2)$$

com $p = 0.7$, $\mu = 7$, $\tau^2 = 0.5^2$, $\theta = 10$ e $\sigma^2 = 0.5^2$.

- 1 Gere $j \in \{0, 1\}$ da distribuição Bernoulli(0.7),
- 2 Gere x da distribuição $N(7, 0.5)$ se $j = 1$.
- 3 Caso contrário, gere x da distribuição $N(10, 0.5)$.
- 4 Retorne x como um valor simulado da mistura.

Histograma de 2000 valores simulados.



Exemplo. Deseja-se gerar valores de um variável aleatória contínua X tal que,

$$\begin{aligned} X|Y = y &\sim N(0, 1/y) \\ Y &\sim \text{Gama}(\nu/2, \nu/2). \end{aligned}$$

A função de densidade (marginal) de X seria obtida por integração,

$$f(x) = \int_0^{\infty} f(x|y)f(y)dy.$$

Podemos simular um valor de (X, Y) em 2 estágios e retornar somente o valor de X simulado,

- 1 Gere um valor y da distribuição $\text{Gama}(\nu/2, \nu/2)$,
- 2 Gere um valor x da distribuição condicional $N(0, 1/y)$,
- 3 Retorne x como um valor simulado da distribuição marginal de X .

Neste exemplo a variável X tem distribuição t -Student com $\nu > 0$ graus de liberdade. Sua função de densidade é,

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right) \sqrt{\pi\nu}} \left[1 + \frac{x^2}{\nu}\right]^{-(\nu+1)/2},$$

que pode ser obtida resolvendo-se a integral,

$$f(x) = \int_0^\infty f(x|y)f(y)dy.$$

Redução de Variância

Seja $\mathbf{X} = (X_1, \dots, X_n)$ um vetor aleatório e deseja-se calcular,

$$\theta = E[g(\mathbf{X})].$$

- Suponha que foram simulados k vetores $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)}$ de forma independente.

- Calcule

$$Y_i = g(\mathbf{X}^{(i)}), \quad i = 1, \dots, k.$$

- Defina,

$$\bar{Y} = \frac{1}{k} \sum_{i=1}^k Y_i,$$

sendo Y_1, \dots, Y_k independentes.

Então,

$$\begin{aligned}E(\bar{Y}) &= \theta \\ E(\bar{Y} - \theta)^2 &= \text{Var}(\bar{Y}).\end{aligned}$$

Ao usar \bar{Y} para estimar θ deseja-se reduzir $\text{Var}(\bar{Y})$.

Variáveis Antitéticas

O objetivo é reduzir a variância introduzindo correlação negativa entre pares de valores simulados.

Dados Y_1 e Y_2 tais que $Var(Y_1) = Var(Y_2) = \sigma^2$ e $E(Y_1) = E(Y_2) = \theta$ temos que,

$$\begin{aligned} Var\left(\frac{Y_1 + Y_2}{2}\right) &= \frac{1}{4} [Var(Y_1) + Var(Y_2) + 2Cov(Y_1, Y_2)] \\ &= \frac{\sigma^2}{2} + \frac{Cov(Y_1, Y_2)}{2} \end{aligned}$$

Portanto, pode ser uma vantagem se Y_1 e Y_2 forem negativamente correlacionados ao invés de independentes.

Suponha que X_1, \dots, X_n são independentes e,

$$X_i = F_i^{-1}(U_i), \quad i = 1, \dots, n$$

sendo F_i a função de distribuição de X_i e $U_i \sim U(0, 1)$.

- Se $U \sim U(0, 1)$ então $1 - U \sim U(0, 1)$ e $\text{Cov}(U, 1 - U) < 0$.
- Se g é uma função monótona, fazendo

$$\begin{aligned} Y_1 &= g(F_1^{-1}(U_1), \dots, F_n^{-1}(U_n)) \\ Y_2 &= g(F_1^{-1}(1 - U_1), \dots, F_n^{-1}(1 - U_n)) \end{aligned}$$

Y_1 e Y_2 serão identicamente distribuídos e negativamente correlacionados.

Amostragem por Importância

Seja $\mathbf{X} = (X_1, \dots, X_n)$ um vetor aleatório com função de densidade (ou massa de probabilidade) $f(\mathbf{x})$.

Deseja-se calcular,

$$\theta = E[h(\mathbf{X})] = \int h(\mathbf{x})f(\mathbf{x})d\mathbf{x}.$$

Muitas vezes não conhecemos completamente $f(x)$ e/ou não é possível simular seus valores diretamente.

Mas $E[h(\mathbf{X})]$ pode ser reescrita como,

$$\begin{aligned}\theta = E[h(\mathbf{X})] &= \int \frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x} \\ &= \int w(\mathbf{x})g(\mathbf{x})d\mathbf{x}.\end{aligned}$$

Ou seja,

$$E_f[h(\mathbf{X})] = E_g[w(\mathbf{X})]$$

Dada uma amostra aleatória $\mathbf{x}_1, \dots, \mathbf{x}_n$ da distribuição com densidade g um estimador de Monte Carlo da integral é,

$$\hat{I} = \frac{1}{n} \sum_{i=1}^n \frac{h(\mathbf{x}_i)f(\mathbf{x}_i)}{g(\mathbf{x}_i)}.$$

Quando $n \rightarrow \infty$,

$$\hat{I} \longrightarrow \int h(\mathbf{x})f(\mathbf{x})d\mathbf{x}.$$

- 1 Gere valores $\mathbf{x}_1, \dots, \mathbf{x}_n$ de $g(\cdot)$.
- 2 Calcule os pesos,

$$w(\mathbf{x}_i) = \frac{h(\mathbf{x}_i)f(\mathbf{x}_i)}{g(\mathbf{x}_i)}, i = 1, \dots, n.$$

- 3 Calcule

$$\hat{l} = \frac{1}{n} \sum_{i=1}^n w(\mathbf{x}_i).$$

Portanto, o algoritmo produz uma amostra ponderada que é usada para estimar $E_f[h(\mathbf{X})]$.

- $g(x)$, usualmente chamada de *função de importância*, deve ser de fácil amostragem.
- O procedimento é comumente chamado de *amostragem por importância*.
- No caso de $f(x)$ ser completamente conhecida o estimador usa todos os valores gerados com os mesmos pesos $1/n$.
- Na prática resultados melhores serão obtidos quando $g(x)$ for uma boa aproximação para $f(x)$.
- Se alguns poucos pesos forem muito maiores do que os demais estes irão dominar a estimativa. Para tentar evitar isto, $g(x)$ em geral deve ter caudas mais pesadas do que $f(x)$, ou seja $f(x) \leq g(x)$ nas caudas.

Exemplo. Deseja-se calcular $P(Z > 4)$ sendo $Z \sim N(0, 1)$.

Aplicando o método de Monte Carlo usual, gera-se $z_1, \dots, z_n \sim N(0, 1)$ e,

$$P(Z > 4) \approx \frac{1}{n} \sum_{i=1}^n I(z_i > 4).$$

- Como $P(Z > 4)$ é muito pequena teremos muitos $I(z_i > 4) = 0$ e $P(Z > 4) \approx 0$.
- É mais eficiente gerar z_i de uma distribuição $q(z)$ com probabilidades maiores nas caudas e aplicar a aproximação,

$$P(Z > 4) \approx \frac{1}{n} \sum_{i=1}^n I(z_i > 4) \frac{p(z_i)}{q(z_i)}.$$

Suponha agora que $p(x) = kf(x)$, sendo k uma constante desconhecida. Então,

$$p(x) = \frac{f(x)}{\int f(x)dx}$$

e deseja-se calcular $E[g(X)]$.

Mas,

$$\begin{aligned} E[g(X)] &= \frac{1}{\int f(x)dx} \int g(x) f(x)dx \\ &= \frac{1}{\int \frac{f(x)}{q(x)} q(x)dx} \int \frac{g(x) f(x)}{q(x)} q(x)dx \\ &= \frac{E_q[g(X)w(X)]}{E_q[w(X)]}. \end{aligned}$$

sendo,

$$w(x) = \frac{f(x)}{q(x)}.$$

- 1 Gere x_1, \dots, x_n da distribuição q .
- 2 Calcule, $w(x_i) = f(x_i)/q(x_i)$, $i = 1, \dots, n$.
- 3 Estime a esperança como,

$$\widehat{E[g(X)]} = \frac{1}{\sum_{i=1}^n w(x_i)} \sum_{i=1}^n g(x_i) w(x_i).$$

Pode-se mostrar que,

$$\widehat{E[g(X)]} \longrightarrow E[g(X)] \quad \text{q.c.}$$