Simulação de variáveis aleatórias: Parte 1

Prof. Caio Azevedo

- Simular: reproduzir, controlando características de interesse, fenômenos reais. Exemplo: avaliar o efeito de um Tsunami numa parede de concreto.
- Simulação variáveis aleatórias: gerar valores para representar o comportamento de fenômenos aleatórios.
- Aplicações: comparar estimadores, comparar modelos, estimar a distribuição de uma variável aleatória.

Lembrete

- Sempre conferir a “qualidade” dos valore simulados.
- Formas simples:
  - Calcular momentos amostrais.
  - Comparar histogramas (curvas de nível) observados com os teóricos.

Método da transformada inversa (v.a. contínua)

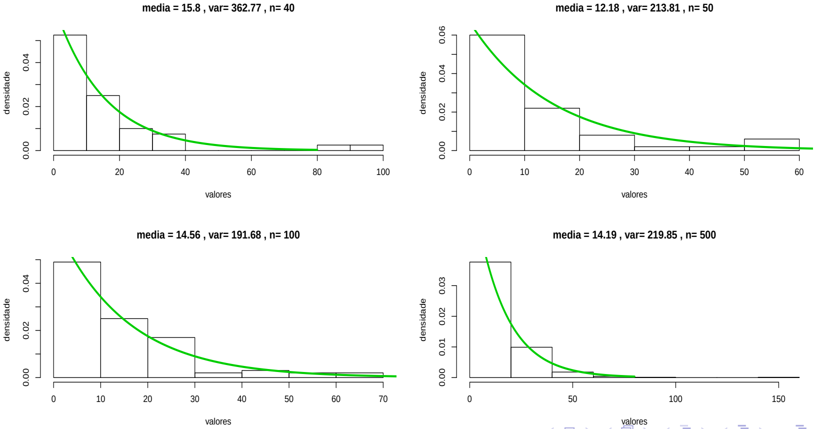
- Seja $X \sim F_X(.;\theta)$, $F_X(.;\theta)$ é uma função de distribuição acumulada (fda) contínua.
- Então $Y = F_X(X;\theta) \sim U(0,1)$ (exercício). OBS: resultado também vale se usar $S_X(x;\theta) = 1 - F_X(x;\theta)$ (função de sobrevivência).
- Algoritmo para simular uma amostra aleatória (aa, conjunto de variáveis independentes e identicamente distribuídas) de $X \sim F_X(.;\theta)$, faça (também pode ser usada a $S_X(\ \theta)$):
  1. Simule $u_i \sim U(0,1), i = 1, ..., n$.
  2. Calcule $x_i = F_X^{-1}(u_i), i = 1, ..., n$.

Exponencial

- Seja $X \sim \exp(\theta), \theta > 0, \mathcal{E}(X) = \theta$.
- Temos que $F_X(x) = (1 - e^{-x/\theta})\mathbb{1}_{(0,\infty)}(x)$. Portanto, $x = -\theta \ln(1 - u)$.
- Programa no R:

```
n<-50
theta <-2
u <- runif(n)
x <- -theta*log(1-u)
```

Valores simulados



Normal truncada

- Seja $X \sim N_{[a,b]}(\mu, \sigma^2), \mu \in \mathcal{R}, \sigma^2 > 0, a < b$.
- Temos que $F_X(X) = \frac{F_Y(x) - F_Y(a)}{F_Y(b) - F_Y(a)}$, $Y \sim N(\mu, \sigma^2)$ (exercício).
- Assim $x = F_Y^{-1}\left[u\left(F_Y(b) - F_Y(a)\right) + F_Y(a)\right]$.
- Temos que

$$\mathcal{E}(X) = \mu + \sigma \frac{\phi\left(\frac{a-\mu}{\sigma}\right) - \phi\left(\frac{b-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \phi\left(\frac{a-\mu}{\sigma}\right)}$$

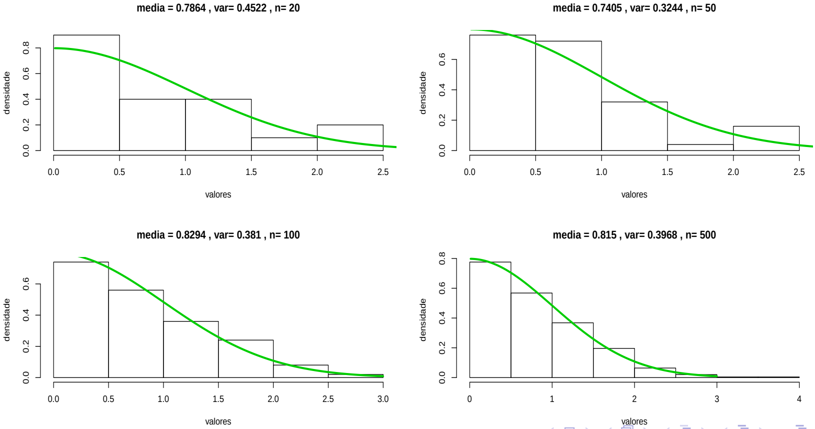
$$\mathcal{V}(X) = \sigma^2 \left[1 + \frac{\frac{a-\mu}{\sigma}\phi\left(\frac{a-\mu}{\sigma}\right) - \frac{b-\mu}{\sigma}\phi\left(\frac{b-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) + \phi\left(\frac{a-\mu}{\sigma}\right)} - \left(\frac{\phi\left(\frac{a-\mu}{\sigma}\right) - \phi\left(\frac{b-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) + \phi\left(\frac{a-\mu}{\sigma}\right)}\right)^2\right]$$

Cont.

- Programa no R:

```
v.u <- (runif(n))
aux <- cbind(u*(pnorm(b,mu,sqrt(sigma2))
-pnorm(a,mu,sqrt(sigma2))) + pnorm(a,mu,sqrt(sigma2)))
v.x <- qnorm(aux,mu,sqrt(sigma2))
```

Valores simulados



Normal multivariada

- $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{\mu} \in \mathcal{R}^p, \boldsymbol{\Sigma} > 0$.
- Sabemos que $\mathbf{X} = \boldsymbol{\Psi}\mathbf{Z} + \boldsymbol{\mu}$, $\boldsymbol{\Psi} = \text{Cholesky}(\boldsymbol{\Sigma})$,
  $\mathbf{Z} = (Z_1, ..., Z_n), Z_i \overset{i.i.d.}{\sim} N(0,1), i = 1, ..., p$.
- $\mathbf{x} = \boldsymbol{\Psi} F_Z^{-1}(\mathbf{u}) + \boldsymbol{\mu}, \mathbf{u} = (u_1, ..., u_p)$.

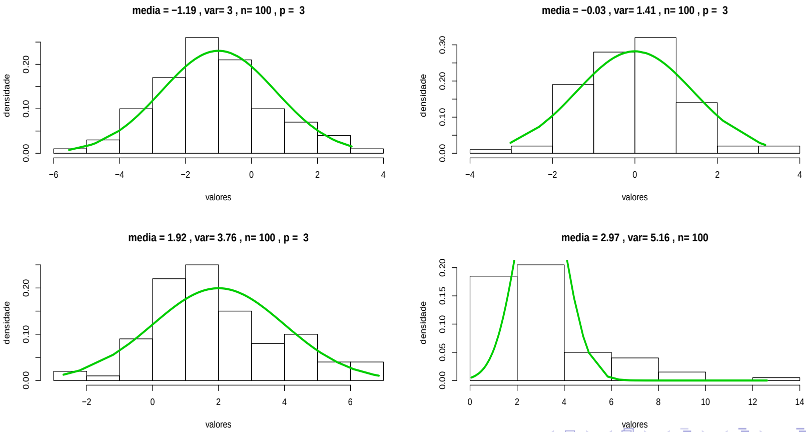
```
m.u <- cbind(runif(n))
m.z <- qnorm(m.u)
for (i in 2:p)
{m.u <- cbind(runif(n)) m.z <- cbind(m.z,qnorm(m.u)) }
m.x <- t(chol(m.sigma)%*%t(m.z) + matrix(v.mu,p,n))
```

Comentários

- Neste caso, tem-se que $X_i \sim N_1(\mu_i, \sigma_i^2)$, em que $\mathcal{E}(X_i) = \mu_i$ e $\mathcal{V}(X_i) = \sigma_i^2$.
- Além disso, a distância de Mahalanobis tem distribuição qui-quadrado com p graus de liberdade, ou seja:

$$(\mathbf{X} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu}) \sim \chi_p^2$$

Valores simulados



Método da transformada inversa (v.a. discreta)

- Seja X uma v.a discreta, $P_X(X = x_i) = p_i \mathbb{1}_{\{0,1,2,\ldots\}}(i), \sum_i p_i = 1$.
- Algoritmo para simular uma amostra aleatória (aa, conjunto de variáveis independentes e identicamente distribuídas) de $X \sim P_X(X = i)$, faça:

Cont.

1. Simule $u_i \sim U(0,1), i = 1, ..., n$.
2. Faça

$$x_i = \begin{cases} x_0, & \text{se } u < p_0 \\ x_1, & \text{se } p_1 \leq u < p_0 + p_1 \\ \vdots & \\ x_j, & \text{se } \sum_{i=0}^{j-1} p_i \leq u < \sum_{i=0}^{j} p_i \\ \vdots & \end{cases}$$

- Dado que $0 < a < b < 1, P(a \leq U < b) = b - a$, temos que

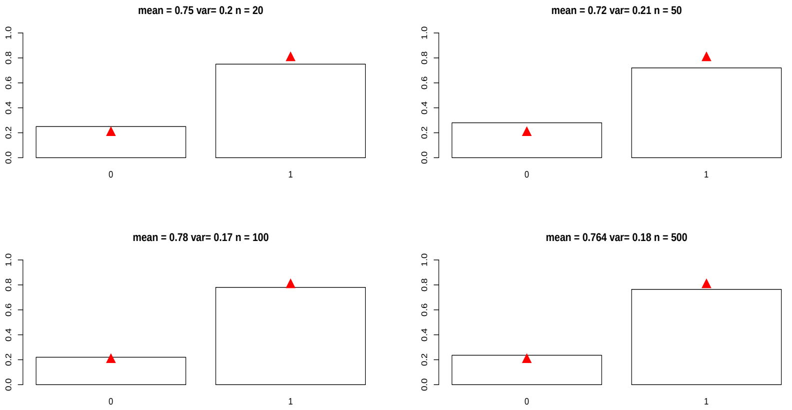
$$P(X = x_j) = P(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i) = p_j$$

Bernoulli

- Seja $X \sim \text{Bernoulli}(\theta), \theta \in (0, 1)$.
- Algoritmo: gere $u_i, U \sim U(0, 1), i = 1, 2..., n$ e faça

$$x_i = \begin{cases} 0, & \text{se } u > \theta \\ 1, & \text{se } u \leq \theta \end{cases}$$

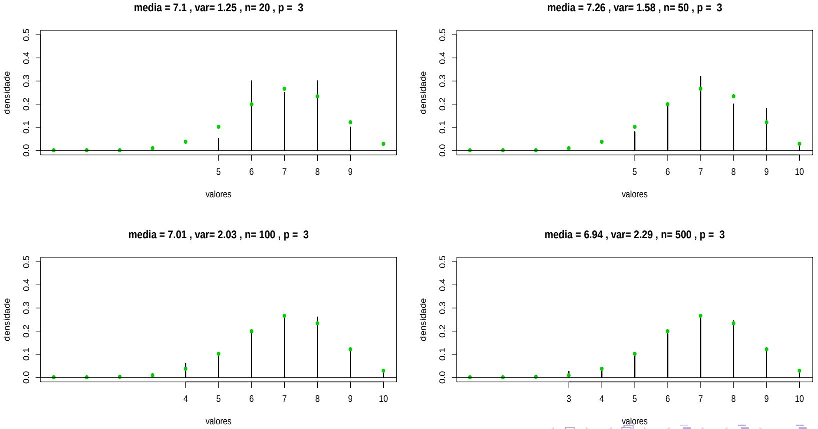
Valores simulados



Binomial

- Seja $X \sim \text{Binomial}(m, \theta), \theta \in (0, 1)$, m conhecido.
- Duas formas:
  - Método da transformada inversa. Repetir n vezes: Gerar u, de $U(0, 1), i = 1, .., n$ e atribuir $x = x_j$ se $\sum_{i=1}^{j-1} p_j \leq u < \sum_{i=1}^{j} p_j$.
  - Representação estocástica : $X = \sum_{i=1}^{m} Y_i$, $Y_i \overset{i.i.d.}{\sim}$ Bernoulli(θ). Repetir n vezes: gerar $y_1, ..., y_m$ (usando o método da transformada inversa e fazer $x = \sum_{i=1}^{m} y_i$.

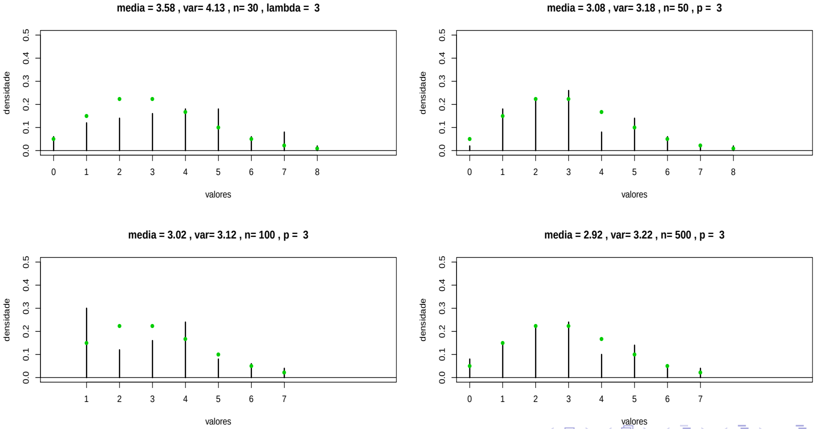
Valores simulados



Poisson

- Seja $X \sim \text{Poisson}(\lambda), \lambda > 0$. $p_i = P(X = i) = \frac{e^{-\lambda}\lambda^i}{i!}\mathbb{1}_{\{0,1,2,\ldots\}}(i)$
- Pode-se provar que $p_{i+1} = \frac{\lambda}{i+1}p_i$
- Algoritmo:
  1. Simule u de $U(0,1)$.
  2. Faça i = 0, $p = e^{-\lambda}, F = p$.
  3. Se u ¡ F, atribua $X = i$ e pare.
  4. $p = \frac{\lambda p}{i+1}, F = F + p.i = i + 1$
  5. Vá para o passo 3.

Valores simulados



Método da rejeição (v.a. contínua)

- Seja $X \sim f_X(.;\theta)$, $f_X(.;\theta)$ é uma função densidade de probabilidade com suporte $X(\Omega)$.
- Queremos simular de f_X. Para isso podemos encontrar um "envelope" constante $(c \geq 1)$ e uma densidade envelope g_Y com mesmo suporte de X, tal que

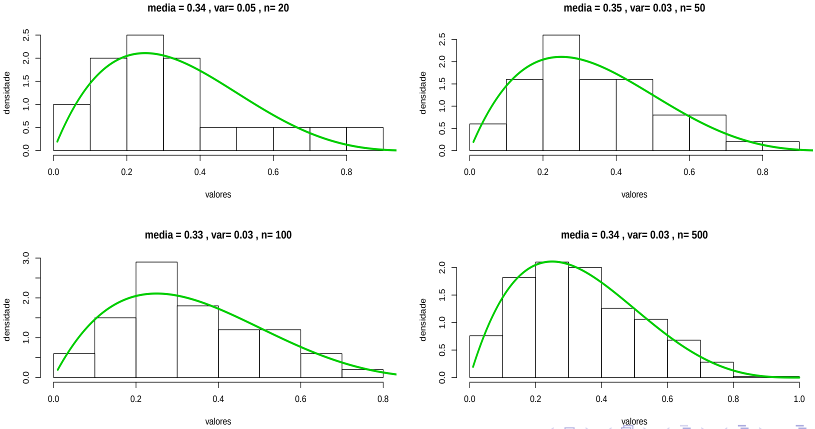
$$\frac{f_X(x)}{g_Y(x)} \leq c, \forall x \in X(\Omega)$$

- O algotimo da rejeição pode ser descrito como
  1. Simule $u_i \sim U(0,1), i = 1, ..., n$ e y de g_y de modo independente.
  2. Se $u \leq \frac{f_X(y)}{cg_Y(y)}$ faça x = y, caso contrário, volte ao Passo 1.

Beta a = 2 e b =4

- X uma va tal que $f_X(x) = 20x(1-x)^3$, Beta(2,4).
- Seja $g_Y(y) = \mathbb{1}(y)_{(0,1)}$.
- O máximo de $\frac{f(x)}{g(x)}$ é obtido para $x = 1/4$ e assim $\frac{f(1/4)}{g(1/4)} = \frac{135}{64}$.
- Portanto, $\frac{f(x)}{cg(x)} = \frac{256}{27}x(1-x)^3$
- Algoritmo
  1. Gere, de modo independentes, u_1 e u_2 de $U(0,1)$.
  2. Se $u_2 \leq \frac{256}{27}u_1(1-u_1)^3$ faça $x = u_1$, caso contrário, volte ao passo 1.

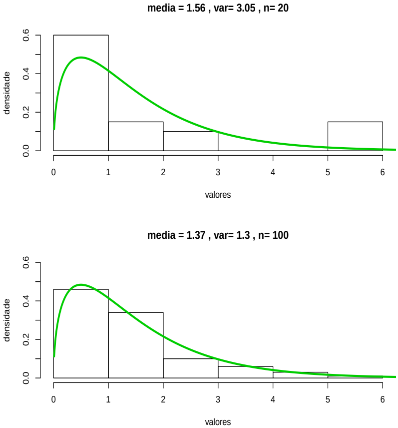
Valores simulados



Gama r $= 3/2$ e λ =1

- X uma va tal que $f_X(x) = Kx^{1/2}e^{-x}\mathbb{1}(x)_{(0,\infty)}, K = 2/sqrt\pi$, gama(3/2,4).
- Seja $g_Y(y) = \frac{2}{3}e^{-2x/3}\mathbb{1}(y)_{(0,\infty)}$.
- O máximo de $\frac{f(x)}{g(x)}$ é obtido para $x = 3/2$ e assim $\frac{f(3/2)}{g(3/2)} = \frac{3^{3/2}}{(2\pi e)^{1/2}}$.
- Portanto, $\frac{f(x)}{cg(x)} = (2e/3)^{1/2}x^{1/2}e^{-x/3}$
- Algoritmo
  1. Gere, u_1 de $U(0,1)$ e faça $y = -\frac{3}{2}\ln u_1$.
  2. Gere u_2 de $U(0,1)$.
  3. Se $u_2 \leq (2ey/3)^{1/2}e^{-y/3}$ faça $x = y$, caso contrário, volte ao passo 1.

Valores simulados



Método da rejeição (v.a. discreta)

- Seja $X \sim f_X(.;\theta)$, $f_X(.;\theta)$ é uma **função de probabilidade** com suporte $X(\Omega)$.
- Queremos simular de f_X. Para isso podemos encontrar um "envelope" constante $(c \geq 1)$ e uma densidade envelope g_Y com mesmo suporte de X, tal que

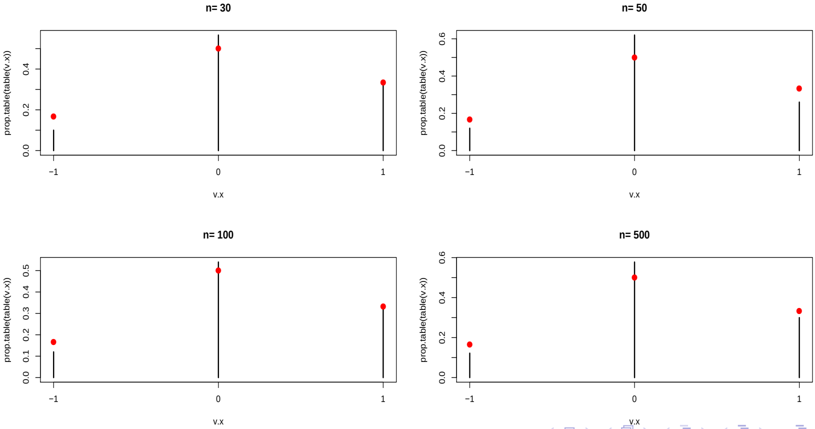
$$\frac{f_X(x)}{g_Y(x)} \leq c, \forall x \in X(\Omega), f(x) \neq 0, g(x) \neq 0$$

- O algotimo da rejeição pode ser descrito como
  1. Simule $u_i \sim U(0,1), i = 1, ..., n$ e y de g_y de modo independente.
  2. Se $u \leq \frac{f_X(y)}{cg_Y(y)}$ faça x = y, caso contrário, volte ao Passo 1.

Variável aleatórioa discreta

- Seja X, $P(X=-1)=1/6, P(X=0)=3/6, P(X=1)=2/6$.
- Seja $g_Y(y)=\frac{1}{3}\mathbb{1}(y)_{\{-1,0,1\}}$
- O máximo de $\frac{f_X(x)}{g_Y(x)}$ ocorre para x $=0$ e, nesse caso, $\frac{f_X(x)}{g_Y(x)}=9/6$
- Algoritmo:
  1. Simule u_1 de $U(0,1)$ e u_2 de $U_{\{-1,0,1\}}$
  2. Se $u_1 \leq f(u_2)/((9/6)g(u_2))$ faça $x=u_2$, caso contrário, volte ao passo 1.

Valores simulados

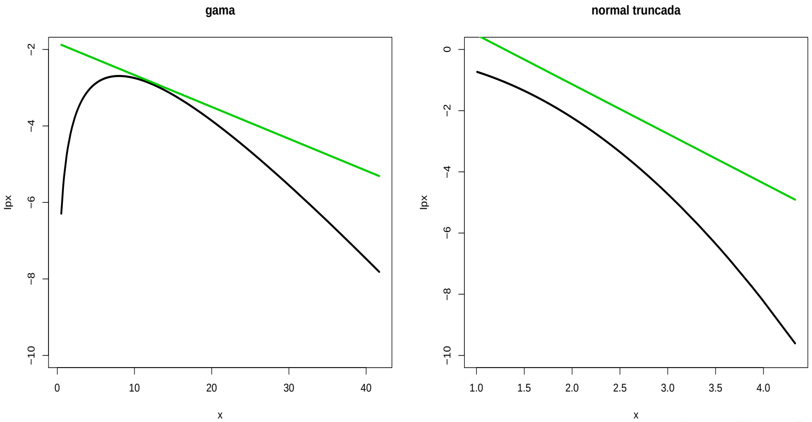


Eficiência do método da rejeição

- A eficiência do método da rejeição é determinada pela probabilidade de aceitação $\frac{1}{\theta}$.
- Quanto menor o valor de c maior será a eficiência do método.
- Em geral, queremos encontrar c,

$$c_\theta = \max_{x \in X(\Omega)} \frac{f(x)}{g_\theta(x)}$$

Distribuição log côncava



Exemplo: normal truncada $f(x) = N_{[1,\infty]}(0,1)$

- Neste caso, a distribuição é log-côncava.
- Uma densidade candidata é $g(x) = be^{-b(x-a)}\mathbb{1}_{(a,\infty)}(x)$
- Pode-se provar que

$$c_b = \max_{x>a} \frac{f(x)}{be^{-b(x-a)}} = \begin{cases} (bd)^{-1}e^{(0{,}5b^2-ba)}, & \text{se } b > a, \\ (bd)^{-1}e^{-0{,}5b^2}, & \text{se } b \leq a \end{cases}$$

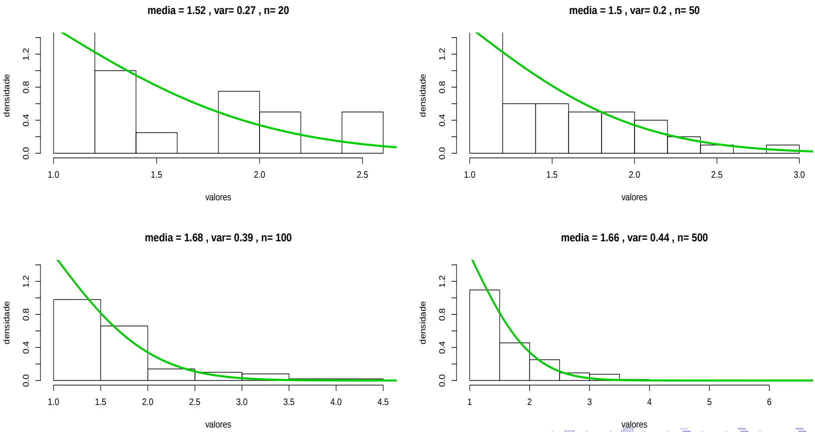
em que $d = \sqrt{2\pi}(1-\Phi(a))$. O primeiro limite é minimizado em $b^* = (a+\sqrt{a^2+4})/2$ e o segundo em $\widehat{b} = a$. A escolha de b é b^*.

Cont.

- Algoritmo

1. Gere u_1, u_2 de $U(0,1)$ independentemente e faça $y = -\frac{1}{b^*}\ln(u_1) + a$
2. Se $u_2 \leq e^{-y^2+b^*y-(b^*)^2/2}$ faça x = y, caso contrário, volte ao passo 1.

Valores simulados



Método da amostragem (reamostragem) por importância

- Seja $X \sim f_X(.;\theta)$, $f_X(.;\theta)$ é uma fdp com suporte $X(\Omega)$.
- Considere uma fdp de amostragem por importância, com mesmo suporte de $f_X(.;\theta)$.
  1. Simule $x_1, ..., x_m$ $iid \sim g(x), m >>> n$ (n tamanho da amostra).
  2. Defina $r(x_j) = \frac{f(x_j)}{g(x_j)}, j = 1, 2, ..., m$
  3. Selecione, **sem reposição** e via reamostragem, da distribuição discreta uma amostra de tamanho n.

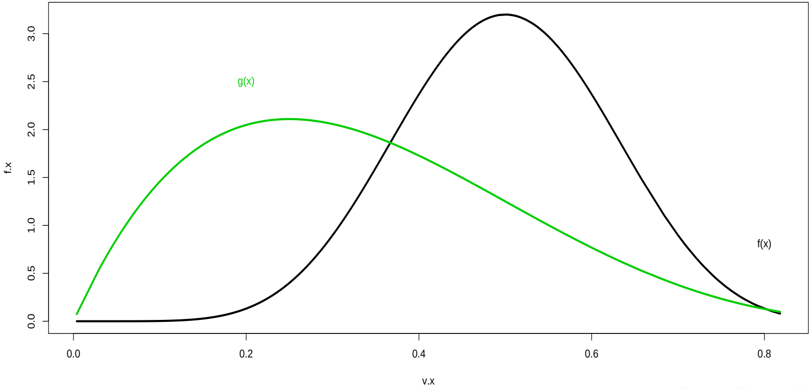
Exemplo: Distribuição definida no intervalo (0,1)

- Seja $X \sim f_X(.; r)$,

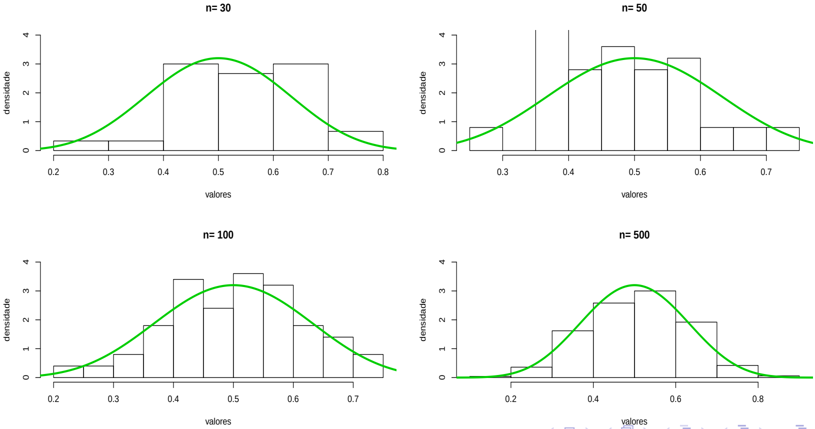
$$f(x) = \frac{\pi sin^r(\pi x)}{beta(0.5, (r+1)/2)} \mathbb{1}_{(0,1)}(x)$$

- Para r $= 6$, considere $g(x) \sim$ beta$(2, 4)$
- Algoritmo
  1. Como definido anteriormente

Comportamento: f(x) x g(x)



Valores simulados



Representação Estocástica

- Dois vetores aleatórios $\mathbf{X}$ e $\mathbf{Y}$ tal que $\mathbf{X} = g(\mathbf{Y})$.
- Simulação de uma distribuição beta(a, b).
- Simular $x_1 \sim \text{gama}(a, 1)$ e $x_2 \sim \text{gama}(b, 1)$.
- Faça $y = x_1/(x_1 + x_2)$.
- Repetir o processo acima n vezes.

Programa

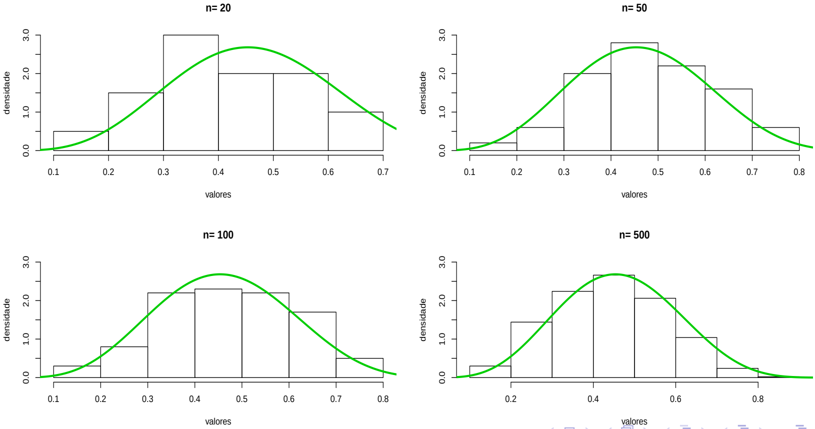
- Programa

```
v.x1 <- r1*(runif(n,0,0.5) -
cbind(apply(log(replicate(round(a),runif(n,0,1))),1,sum)))

v.x2 <- r2*(runif(n,0,0.5) -
cbind(apply(log(replicate(round(b),runif(n,0,1))),1,sum)))

v.y <- cbind(v.x1/(v.x1 + v.x2))
```

Valores simulados



Beta binomial

- Seja $X|Y \sim$ binomial(m, y) e $Y \sim$ beta(a, b).
- Então $X \sim$ beta-binomial, $f_X(x) = \int_0^1 f(x|y)f(y)dy$.
- Algoritmo
  1. Simular n vezes y de beta(a, b).
  2. Dado o vetor y simular $x|y$ de binomial(m, y).
  3. O vetor x terá a distribuição desejada.

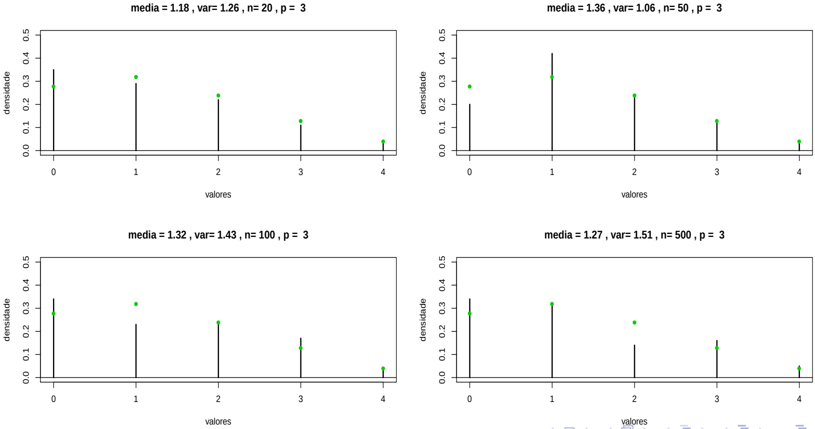
Programa

- Programa

```
v.x1 <- cbind(rbeta(n,a,b))

v.x2 <- cbind(rbinom(n,m,v.x1))
```

Valores simulados



Outros métodos

- Rejeição adaptativa.
- Amostragem condicional.
- Métodos específicos (em geral levam em consideração representações estocásticas e pelo menos um dos algoritmos anteriores).