PUC-RIO
INF1036 - Probabilidade Computacional

Material 8 – Gerando Variáveis Aleatórias

Professora - Ana Carolina Letichevsky*

2022.1

*Material Adaptado de Professor Hélio Lopes

Introdução

Neste material iremos apresentar métodos e procedimentos computacionais dedicados à geração de variáveis aleatórias de algumas das diversas distribuições teóricas de probabilidade já estudadas.

Todos os métodos baseiam-se na prévia geração de um número aleatório U, uniformemente distribuído sobre o intervalo $(0, 1)$.

Importante observar que o método a ser empregado na geração das variáveis aleatórias, depende do tipo de distribuição e da eficiência que se está buscando no processo.

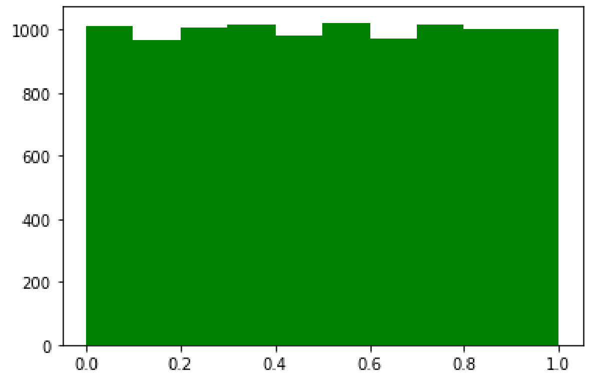
Variável Aleatória Discreta

Seja X uma variável aleatória discreta tal que:

$$P(X = x_j) = p_j; \qquad j = 0,1\ldots; \qquad \sum_j p_j = 1$$

Seja U uniformemente distribuído em $(0,1)$, temos então:

$$X = \begin{cases} x_0, & U < p_0 \\ x_1, & p_0 \le U < p_0 + p_1 \\ & \ldots \\ x_j, & \displaystyle\sum_{i=1}^{j-1} p_j \le U < \sum_{i=1}^{j} p_i \\ & \ldots \end{cases}$$



Desde que para $0 < a < b < 1$, $P(a \le U < b) = b - a$, temos que:

$$P(X = x_j) = P\left(\sum_{i=1}^{j-1} p_j \le U < \sum_{i=1}^{j} p_i\right) = p_j \Rightarrow X \ possui\ a\ distribuição\ desejada.$$

Variável Aleatória Discreta

Seja U uniformemente distribuído entre $(0, 1)$, temos então:

$$X = \begin{cases} x_0, & U < p_0 \\ x_1, & p_0 \leq U < p_0 + p_1 \\ & \cdots \\ x_j, & \sum_{i=1}^{j-1} p_j \leq U < \sum_{i=1}^{j} p_i \\ & \cdots \end{cases}$$

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

Se $x_0 < x_1 < \cdots < x_j < \cdots$ e F denotar a função de distribuição acumulada de X, então:

$$F(x_k) = P(X \leq x_k) = \sum_{i=1}^{k} p_j$$

e X será igual a x_j se

$$F(x_{j-1}) \leq U < F(x_j)$$

Em outras palavras, após gerar o número randômico U é feita a determinação do valor de X encontrando o intervalo $[F(x_{j-1}), F(x_j)]$em que U está, ou de forma equivalente, encontrando a inversa de $F(U)$. Daí o nome do método “Método da Inversa” para gerar X.

Variável Aleatória Discreta

Exemplo 1) Seja X uma variável aleatória discreta, tal que:

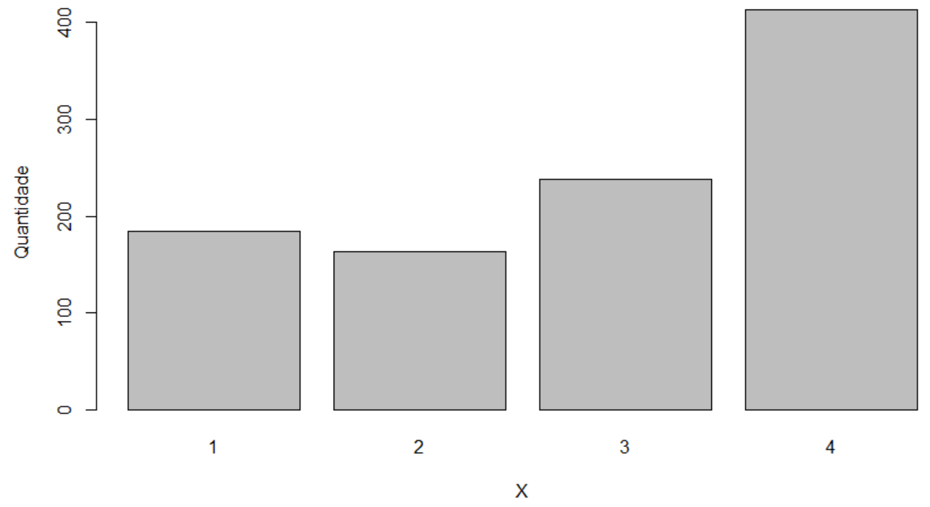
$$P(X = x_j) = p_j; \quad j = 1,2,3,4; \quad p_1 = 0,2 \quad p_2 = 0,15 \quad p_3 = 0,25 \quad p_4 = 0,4$$

Implemente uma função que possa gerar X.

```
gera.variavel <- function (nsamples, taxa) {
  X <- rep(0, nsamples) # nsamples posições com 0
  U <- runif(nsamples) # nsamples valores em [0.0, 1.0)
  for (i in 1:nsamples) {
    if (U[i] < 0.2)
      X[i] = 1
    else if (U[i] < 0.35) # 0.2 + 0.15
      X[i] = 2
    else if (U[i] < 0.6) # 0.2 + 0.15 + 0.25
      X[i] = 3
    else
      X[i] = 4
  }
  return (X)
}

nsamples = 50000

X <- gera.variavel(nsamples)
hist(X, breaks = 100, col = 'green')
```



Método da Inversa

Para gerar uma variável aleatória X cuja função distribuição acumulada é F, siga o seguinte algoritmo:

Passo 1: gere uma variável aleatória uniforme U no intervalo $(0, 1)$.

Passo 2: $X \leftarrow F^{-1}(U)$

Variável Aleatória Contínua

Além do “Método da Inversa”, existem outros possíveis métodos que podem ser utilizados para gerar variáveis aleatórias discretas como o “Método da Rejeição”.

Para variáveis aleatórias contínuas, iremos abordar, além do “Método da Inversa”, os métodos “Método da Rejeição” e o “Método Polar”.

Método da Inversa

Distribuição	Função de Densidade de Probabilidade	Inversa da Função de Distribuição Acumulada
Uniforme $\mathcal{U}(a, b)$ $a \leq x \leq b$	$\frac{1}{b-a}$	$a + (b-a)u$
Exponencial $\mathcal{E}(\lambda)$ $\lambda > 0; x \geq 0$	$\lambda e^{-\lambda x}$	$-\frac{1}{\lambda} ln(1-u)$
Beta $\mathcal{B}(\alpha, 1)$ $\alpha > 0;\ 0 \leq x \leq 1$	$\alpha x^{\alpha-1}$	$u^{\frac{1}{\alpha}}$
Beta $\mathcal{B}(1, \beta)$ $\beta > 0;\ 0 \leq x \leq 1$	$\beta(1-x)^{\beta-1}$	$1-(1-u)^{\frac{1}{\beta}}$
Logística $\mathcal{L}\ (\alpha, \beta)$ $\beta > 0;\ -\infty \leq x;\ \alpha < \infty$	$\frac{e^{-(x-\alpha)/\beta}}{\beta[1+e^{-(x-\alpha)/\beta}]^2}$	$\alpha + \beta ln[u/(1-u)]$
Weibull $\mathcal{W}\ (\alpha . \beta)$ $\alpha, \beta > 0;\ x \geq 0$	$\frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-(x/\beta)^{\alpha}}$	$\beta[-ln(1-u)]^{1/\alpha}$
Cauchy $\mathcal{C}\ (\alpha, \beta)$ $\beta > 0;\ -\infty \leq x;\ \alpha < \infty$	$\frac{1}{\pi}\frac{\beta}{(x-\alpha)^2+\beta^2}$	$\alpha + \beta \tan \pi\, [u-(1/2)]$

Método da Inversa

Exemplo 2) Implemente uma função que gere uma variável aleatória contínua X cuja função de distribuição acumulada é uma Exponencial de parâmetro $\lambda = 2$:

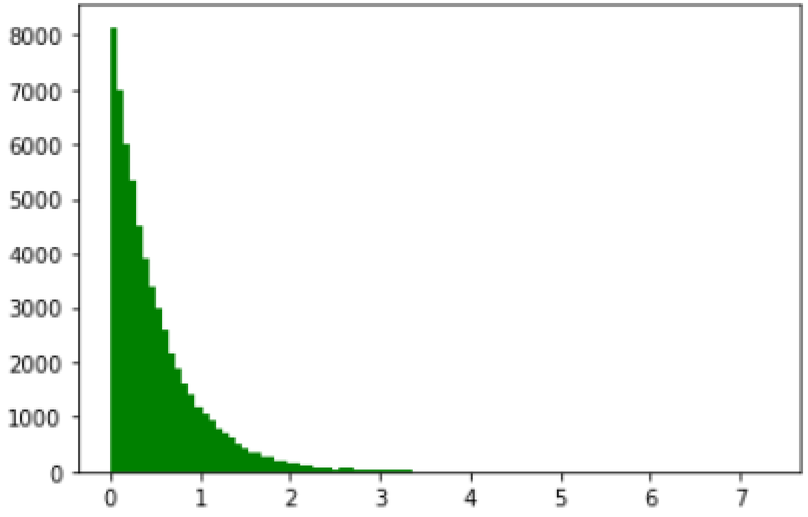
```
import numpy as np
import matplotlib.pyplot as plt
import math

def exponencial(nsamples, taxa):
    X = np.zeros(nsamples) # nsamples posições com 0
    U = np.random.sample(nsamples) # nsamples valores em [0.0, 1.0)
    for i in range(nsamples):
        X[i] = - math.log(1.0 - U[i])/taxa # Inversa da exponencial
    return (X)

taxa = 2.0
nsamples = 50000

X = exponencial(nsamples, taxa)

plt.hist(X, bins=100, facecolor='green')
plt.show()
```



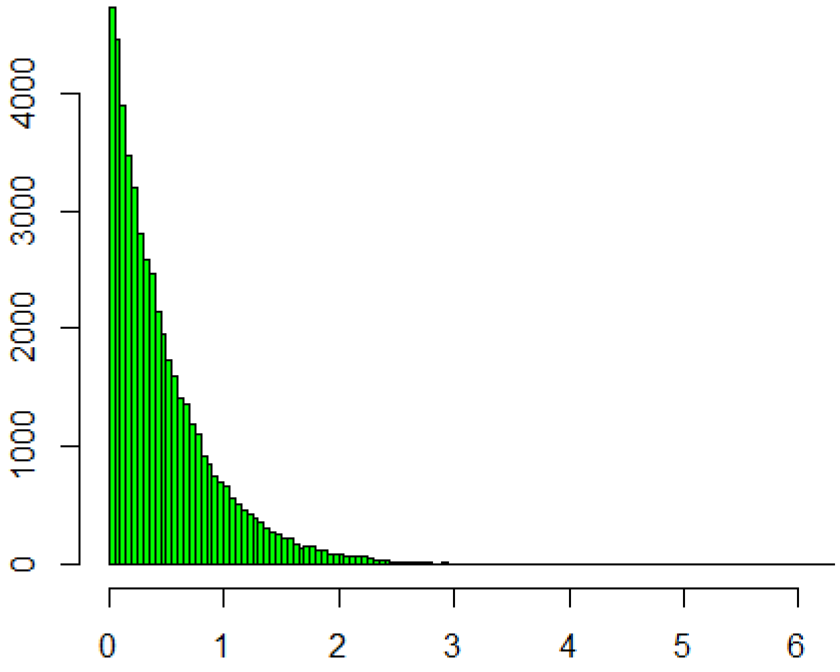
Método da Inversa

Continuação Exemplo 2) Implemente uma função que gere uma variável aleatória contínua X cuja função de distribuição acumulada é uma Exponencial de parâmetro $\lambda = 2$:

```
exponencial <- function (nsamples, taxa) {
  X <- rep(0, nsamples) # nsamples posições com 0
  U <- runif(nsamples) # nsamples valores em [0.0, 1.0)
  for (i in 1:nsamples) {
    X[i] = - log(1.0 - U[i])/taxa # Inversa da exponencial
  }
  return (X)
}

taxa = 2.0
nsamples = 50000

X <- exponencial(nsamples, taxa)
hist(X, breaks = 100, col = 'green')
```



Método da Inversa

Exercício 1) Implemente uma função que gere uma variável aleatória contínua X cuja função de distribuição acumulada é uma Weibull de parâmetros $\alpha = 2$ e $\beta = \frac{1}{2}$:

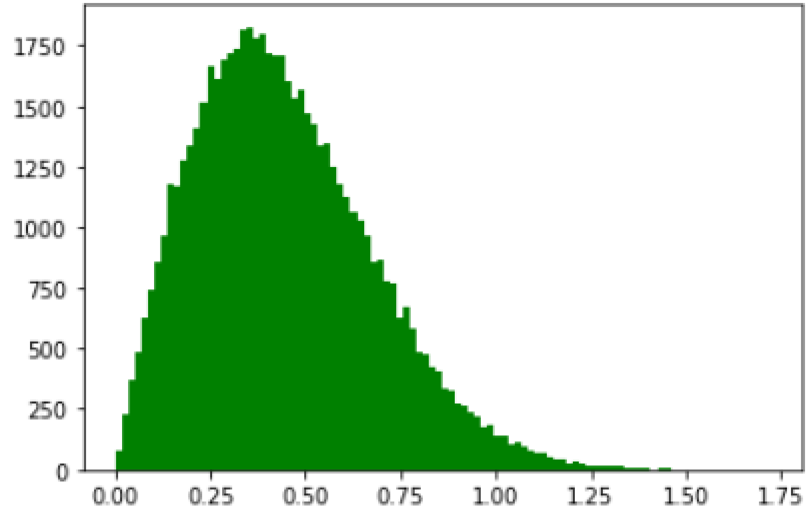
```
import numpy as np
import matplotlib.pyplot as plt
import math

def weibull(nsamples, a, b):
    X = np.zeros(nsamples)
    U = np.random.sample(nsamples)
    for i in range(nsamples):
        X[i] = b * math.pow(-math.log(1 - U[i]), 1.0 / a)
    return (X)

a = 2.0
b = 0.5
nsamples = 50000

X = weibull(nsamples, a, b)

plt.hist(X, bins=100, facecolor='green')
plt.show()
```



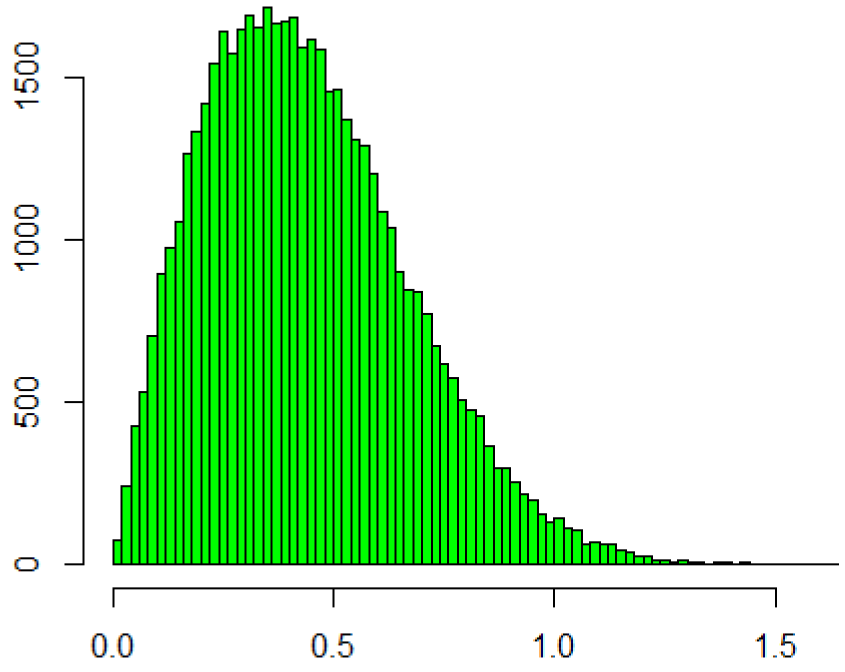
Método da Inversa

Continuação Exercício 1) Implemente uma função que gere uma variável aleatória contínua X cuja função de distribuição acumulada é uma Weibull de parâmetros $\alpha = 2$ e $\beta = \frac{1}{2}$:

```
weibull <- function (nsamples, a, b) {
  X <- rep(0, nsamples)
  U <- runif(nsamples)
  for (i in 1:nsamples) {
    X[i] = b * ((-log(1 - U[i]))^(1.0 / a))
  }
  return (X)
}

a = 2.0
b = 0.5
nsamples = 50000

X = weibull(nsamples, a, b)
hist(X, breaks = 100, col = 'green')
```



Método da Inversa

Exercício 2) Altere os parâmetros da distribuição acumulada Weibull do exercício anterior para $\alpha = 1$ e $\beta = \frac{1}{2}$:

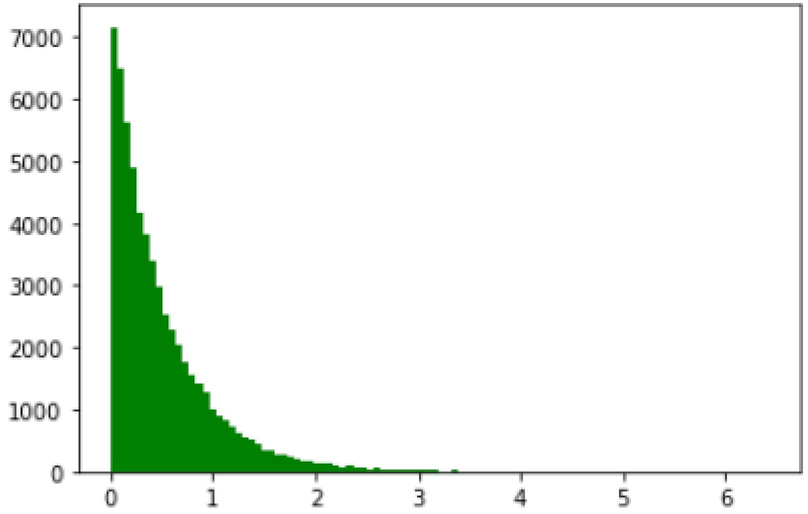
```
import numpy as np
import matplotlib.pyplot as plt
import math

def weibull(nsamples, a, b):
    X <- np.zeros(nsamples)
    U <- np.random.sample(nsamples)
    for i in range(nsamples):
        X[i] <- b * math.pow(-math.log(1 - U[i]), 1.0 / a)
    return (X)

a <- 1
b <- 0.5
nsamples = 50000

X <- weibull(nsamples, a, b)

plt.hist(X, bins=100, facecolor='green')
plt.show()
```



Método da Inversa

Continuação Exercício 2) Altere os parâmetros da distribuição acumulada Weibull do exercício anterior para $\alpha = 1$ e $\beta = \frac{1}{2}$:

```
weibull <- function (nsamples, a, b) {
  X <- rep(0, nsamples)
  U <- runif(nsamples)
  for (i in 1:nsamples) {
    X[i] = b * ((-log(1 - U[i]))^(1.0 / a))
  }
  return (X)
}

a = 1
b = 0.5
nsamples = 50000

X = weibull(nsamples, a, b)
hist(X, breaks = 100, col = 'green')
```