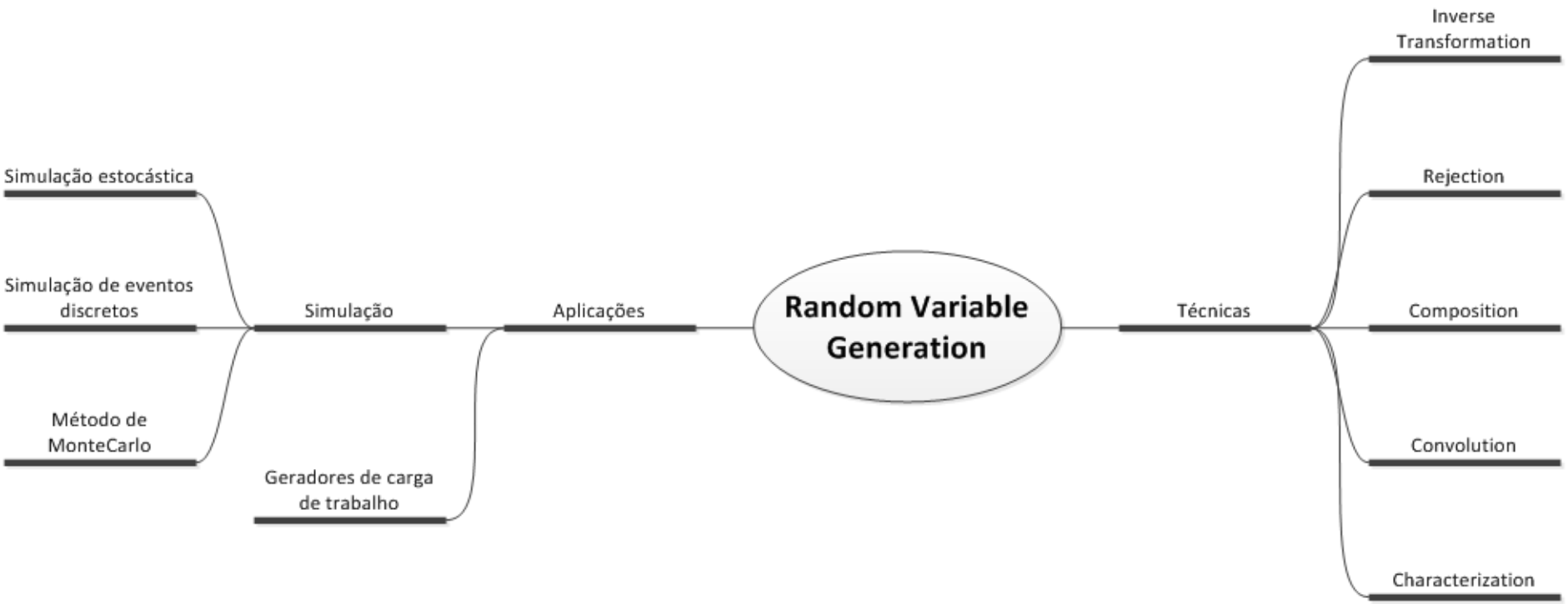
Geração de variáveis aleatórias

Danilo Oliveira, Matheus Torquato

Centro de Informática
Universidade Federal de Pernambuco

5 de setembro de 2012

Resumo


Random Variable Generation
Aplicações
Simulação
Simulação estocástica
Simulação de eventos discretos
Método de MonteCarlo
Geradores de carga de trabalho
Técnicas
Inverse Transformation
Rejection
Composition
Convolution
Characterization

Sumario

Introdução

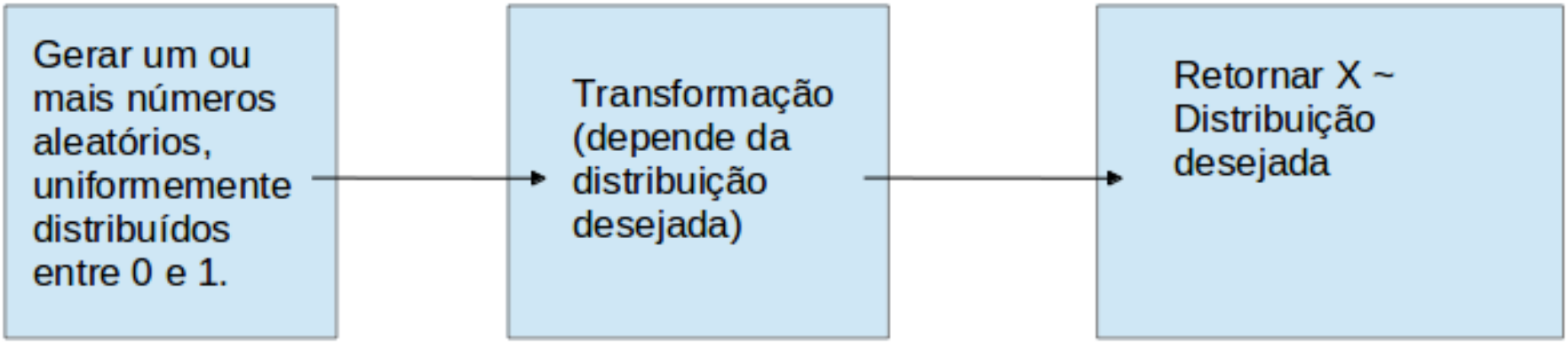
- Estatística trabalha com incertezas;
- É necessário estimar:
  - Número de pessoas procurando emprego;
  - Índice Bovespa diário;
  - Carga submetida ao sistema em determinados horários.

Introdução

- Obedecendo-se a distribuição de probabilidade é possível obter, por intermédio de transformações, valores para as variáveis aleatórias.
- Particularmente útil para simulações em sistemas.

Introdução

- Independentemente da distribuição, o processo de geração é sempre dividido em três partes:



Sumario

Geração de números aleatórios

- O primeiro passo para gerar uma variável aleatória é gerar números aleatórios uniformemente distribuídos entre 0 e 1
- A maioria das linguagens de programação possuem uma função rand() que retorna um número aleatório entre 0 e 1
- A sequência gerada é caracterizada por um número denominado *semente*, que deve ser definido à priori

Geração de números aleatórios

- Gerar valores uniformes entre 0 e 1.
  - Submete-se uma semente a um transformação ciclíca.
- Exemplo:

$$x_n = (5x_{n-1} + 1) \ mod \ 16$$

Geração de números aleatórios

$$x_n = (5x_{n-1} + 1) \bmod 16$$

32 primeiros números

$x_0 = 5$

10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5
10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.

Dividir por 16

0.6250, 0.1875, 0.0000, 0.0625, 0.3750, 0.9375, 0.7500, 0.8125, 0.1250, 0.6875, 0.5000, 0.5625,
0.8750, 0.4375, 0.2500, 0.3125, 0.6250, 0.1875, 0.0000, 0.0625, 0.3750, 0.9375, 0.7500, 0.8125,
0.1250, 0.6875, 0.5000, 0.5625, 0.8750, 0.4375, 0.2500, 0.3125.

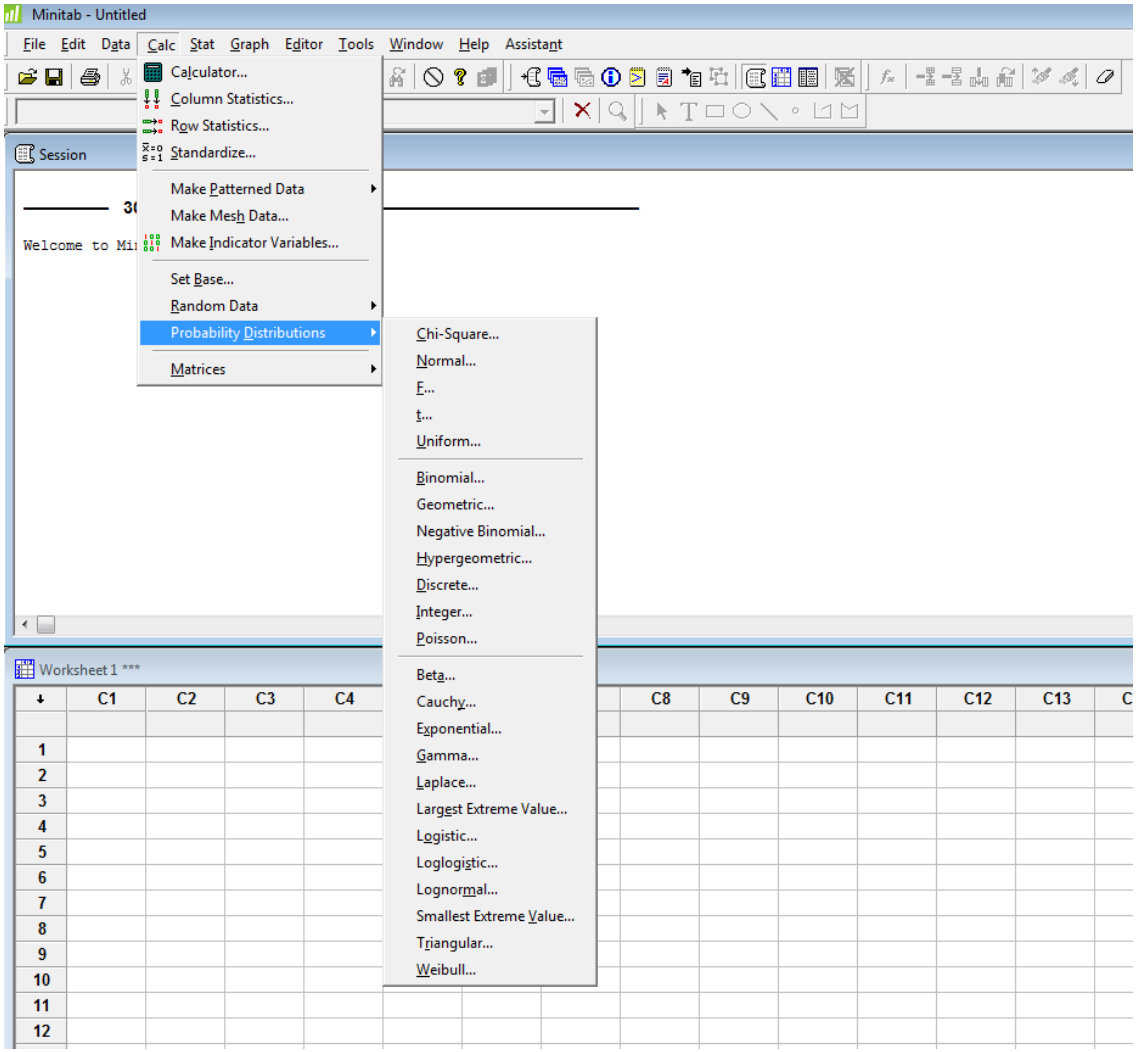
Números pseudo-aleatórios

Sumario

Utilizando softwares de estatística

- Não reinvente a roda, geradores de números aleatórios são fornecidos pelos softwares de estatística mais utilizados

Minitab


Minitab - Untitled
File Edit Data Calc Stat Graph Editor Tools Window Help Assistant
Calculator...
Column Statistics...
Row Statistics...
Standardize...
Make Patterned Data
Make Mesh Data...
Make Indicator Variables...
Set Base...
Random Data
Probability Distributions
Matrices
Chi-Square...
Normal...
F...
t...
Uniform...
Binomial...
Geometric...
Negative Binomial...
Hypergeometric...
Discrete...
Integer...
Poisson...
Beta...
Cauchy...
Exponential...
Gamma...
Laplace...
Largest Extreme Value...
Logistic...
Loglogistic...
Lognormal...
Smallest Extreme Value...
Triangular...
Weibull...
Session
Welcome to Mi
Worksheet 1 ***
C1 C2 C3 C4 C8 C9 C10 C11 C12 C13

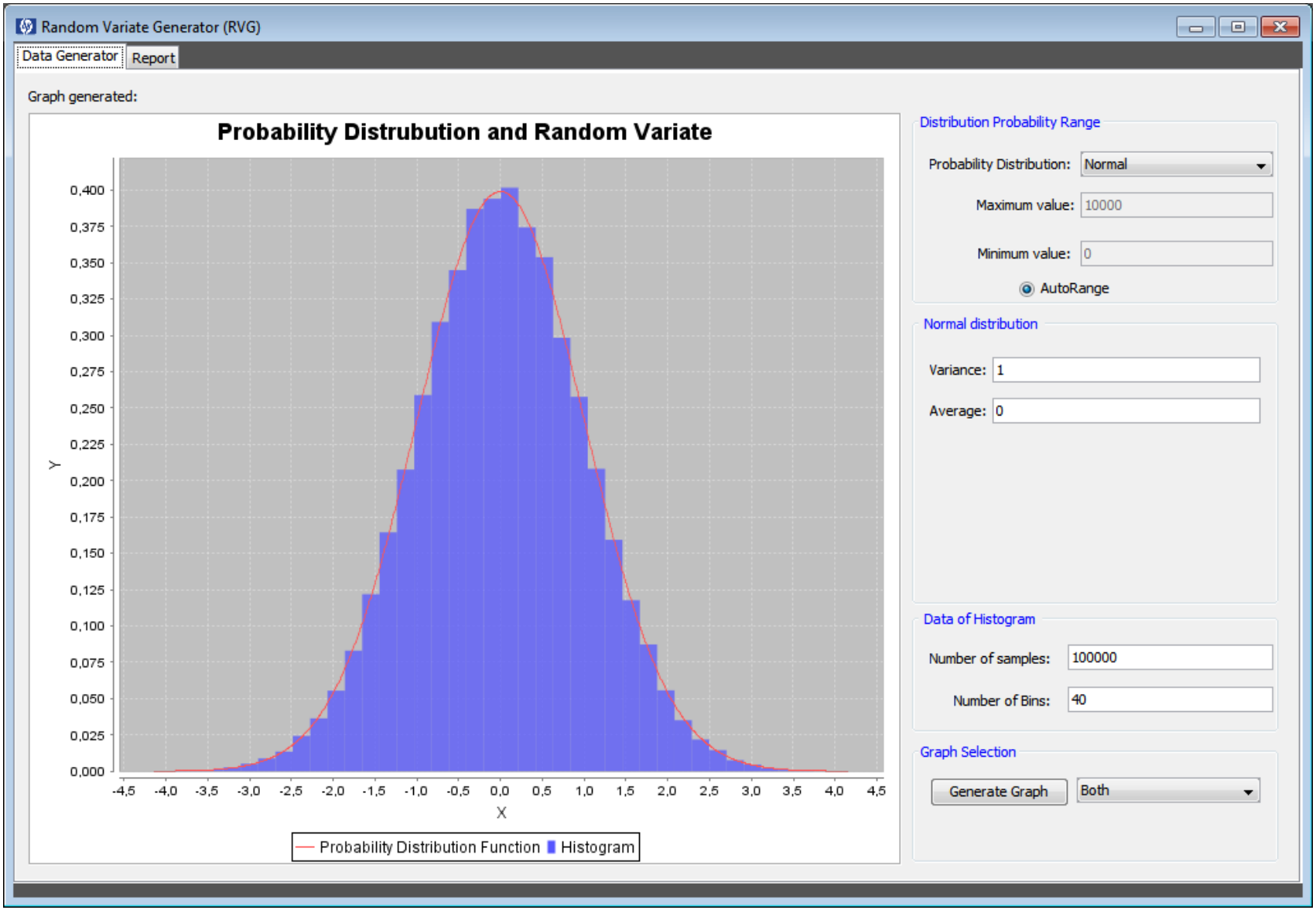
Alguns exemplos:

- **Uniforme**: runif(n, min=0, max=1)
- **Normal**: rnorm(n, mean=0, sd=1)
- **Exponencial**: rexp(n, rate=1)

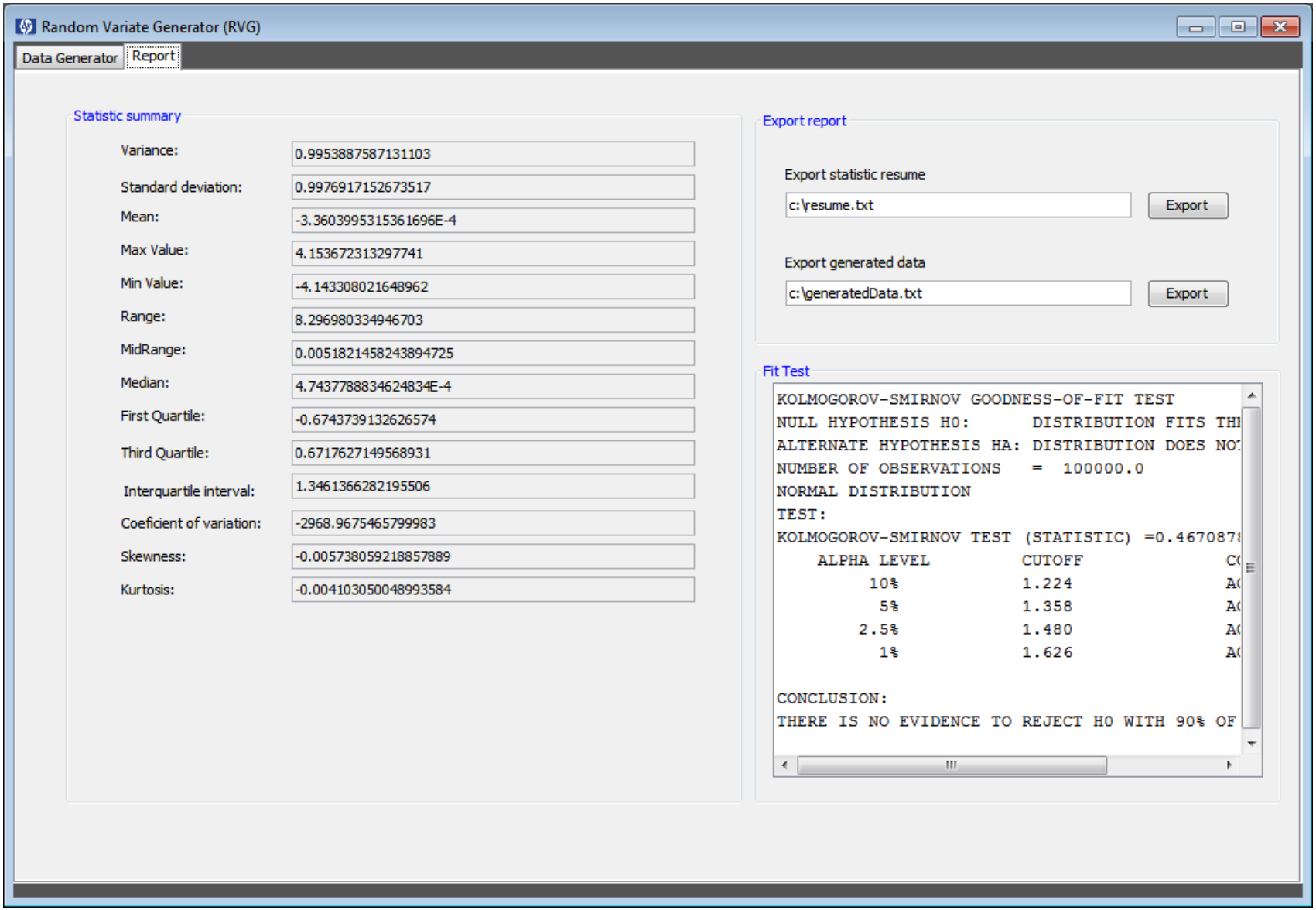
```
Console ~/
> rnorm(100, 10, 1)
  [1]  9.744720  9.495892  9.409301  9.450441 10.476732 11.568990  9.687061  9.794956 10.557614
 [10] 10.191142  9.954670 10.671792 11.107222 11.270932  9.973668 10.015046  8.012956  9.488550
 [19] 10.693954 10.881325  9.148016  9.679518 10.803631 10.097004  7.998736 10.115915  9.958896
 [28] 10.058196  9.413176 10.393964  9.508182 11.302813  8.825665 10.496673 10.018716  9.312772
 [37] 10.785604 10.346448  9.619381 10.237415  8.123532 11.419877  9.933029  8.117339  9.681594
 [46]  8.820324  9.147330 10.158316 10.151577  8.686367  6.923516  9.963983  9.814120 10.039231
 [55]  9.003283  8.501021  8.795137 11.990907  7.039475  9.321258  9.170965  9.577936 10.434104
 [64]  9.956789  8.800380 10.906716  9.694387 11.243990  9.467445 10.032502  9.232671  7.274910
 [73] 10.194464  8.217640 10.754139 11.002639 11.066049 10.224467  9.638626  8.780690 10.415295
 [82]  9.628247  9.276653  8.621708  9.032095 10.781312 10.406845 10.740497 10.532482 10.532831
 [91] 10.120953 10.544423  9.636733  8.934489  9.911907 10.685453  9.858285 10.436472 11.414764
[100]  9.821092
>
```

- Lista de distribuições disponíveis:
  - `http://stat.ethz.ch/R-manual/R-devel/library/stats/html/Uniform.html`
  - `http://en.wikibooks.org/wiki/R_Programming/Probability_Distributions`
  - Distribuições Phase-Type - `http://www.louisaslett.com/PhaseType/`

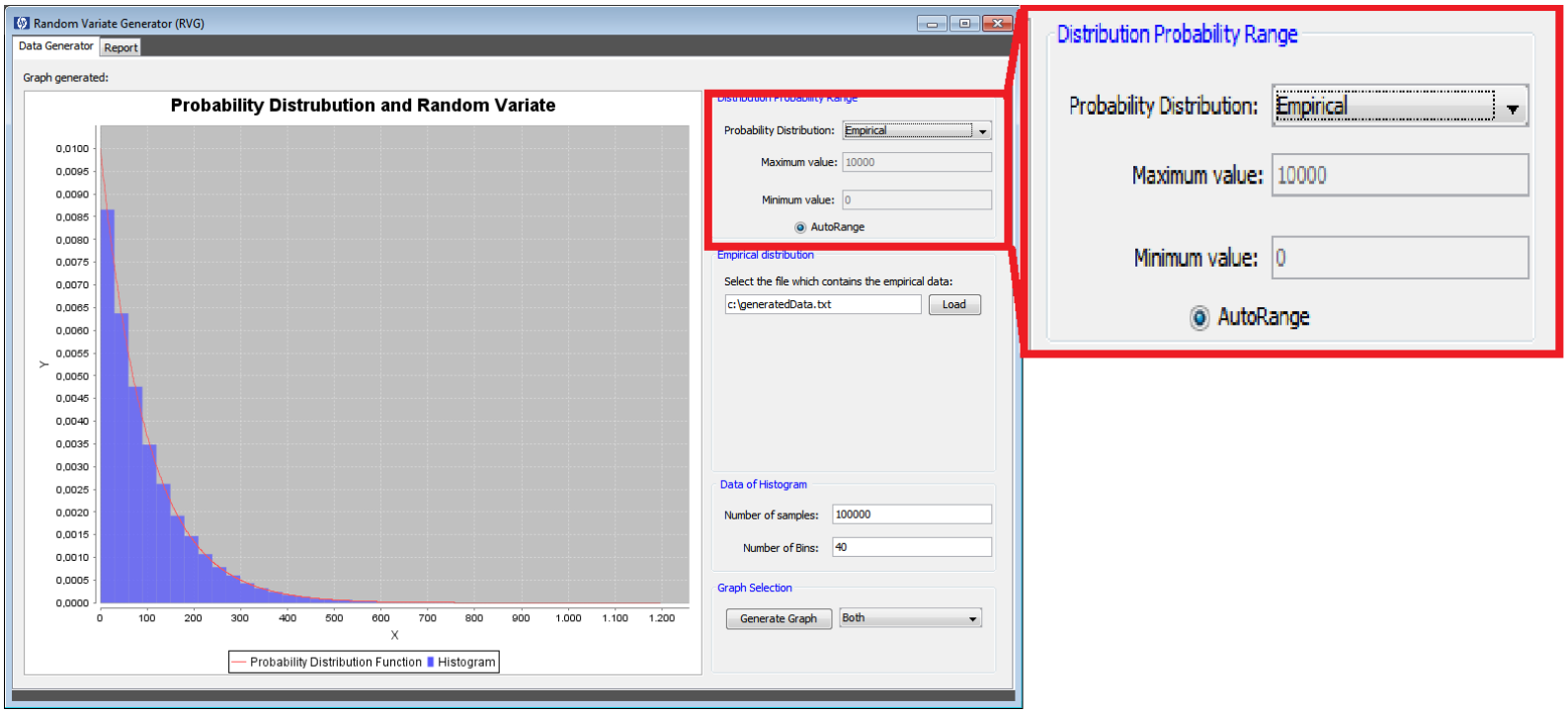
Mercury


Random Variate Generator (RVG)
Data Generator
Report
Graph generated:
Probability Distrubution and Random Variate
Y
X
Probability Distribution Function
Histogram
Distribution Probability Range
Probability Distribution:
Normal
Maximum value:
10000
Minimum value:
0
AutoRange
Normal distribution
Variance:
1
Average:
0
Data of Histogram
Number of samples:
100000
Number of Bins:
40
Graph Selection
Generate Graph
Both

Mercury


Random Variate Generator (RVG)
Data Generator
Report
Statistic summary
Variance:
0.9953887587131103
Standard deviation:
0.9976917152673517
Mean:
-3.6039955315361696E-4
Max Value:
4.153672313297741
Min Value:
-4.143308021648962
Range:
8.296980334946703
MidRange:
0.0051821458243894725
Median:
4.7437788834624834E-4
First Quartile:
-0.6743739132626574
Third Quartile:
0.6717627149568931
Interquartile interval:
1.3461366282195506
Coeficient of variation:
-2968.9675465799983
Skewness:
-0.005738059218857889
Kurtosis:
-0.004103050048993584
Export report
Export statistic resume
c:\resume.txt
Export
Export generated data
c:\generatedData.txt
Export
Fit Test
KOLMOGOROV-SMIRNOV GOODNESS-OF-FIT TEST
NULL HYPOTHESIS H0: DISTRIBUTION FITS THI
ALTERNATE HYPOTHESIS HA: DISTRIBUTION DOES NO
NUMBER OF OBSERVATIONS = 100000.0
NORMAL DISTRIBUTION
TEST:
KOLMOGOROV-SMIRNOV TEST (STATISTIC) =0.467087
ALPHA LEVEL CUTOFF
10% 1.224
5% 1.358
2.5% 1.480
1% 1.626
CONCLUSION:
THERE IS NO EVIDENCE TO REJECT H0 WITH 90% OF

Mercury


Random Variate Generator (RVG)
Data Generator
Report
Graph generated:
Probability Distrubution and Random Variate
Probability Distribution Function
Histogram
Distribution Probability Range
Probability Distribution:
Empirical
Maximum value:
10000
Minimum value:
0
AutoRange
Empirical distribution
Select the file which contains the empirical data:
c:\generatedData.txt
Load
Data of Histogram
Number of samples:
100000
Number of Bins:
40
Graph Selection
Generate Graph
Both

Utilizando softwares de estatística

- Ter pacotes de software à disposição é bom, mas...
  - E se quiséssemos gerar uma distribuição que não estivesse implementada no R ou Minitab?
  - E se quiséssemos gerar uma distribuição em uma linguagem de programação como Java ou C?
  - E se estivéssemos utilizando um pacote de simulação (NS-2, Omnet++, TimeNet, etc.), que não implementa a distribuição que a gente quer?

Utilizando softwares de estatística

- É preciso conhecer técnicas de geração de variáveis aleatórias
- Conhecer todas as principais técnicas é importante, tendo em vista que não existe uma técnica que consiga gerar todas as distribuições de forma eficiente
- As técnicas que serão apresentadas aqui são:
  - Transformação inversa
  - Aceitação-Rejeição
  - Composição
  - Convolução
  - Caracterização

Sumario

Método da transformação inversa

- Seja X uma variável aleatória com função de distribuição acumulada F. Definimos F^{-1} como a função

$$F^{-1}(y) = inf\{x : F(x) \geq y\}, 0 \leq y \leq 1.$$

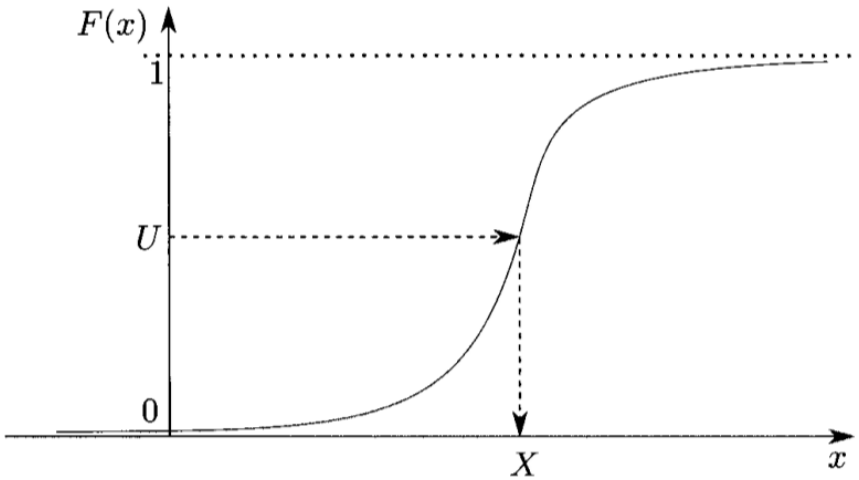
- Seja $U \sim U(0,1)$. A cdf da transformada inversa $F^{-1}(U)$ é dada por

$$P(F^{-1}(U) \leq x) = P(U \leq F(X)) = F(x)$$

- Portanto, para gerar a variável X, dado uma variável aleatória $U \sim U(0,1)$ e sua cdf inversa F^{-1}, aplicamos o seguinte algoritmo:
  1. Generate $U \sim U(0,1)$
  2. Return $X = F^{-1}(U)$

Método da transformação inversa

O método pode ser visualizado através do seguinte gráfico:

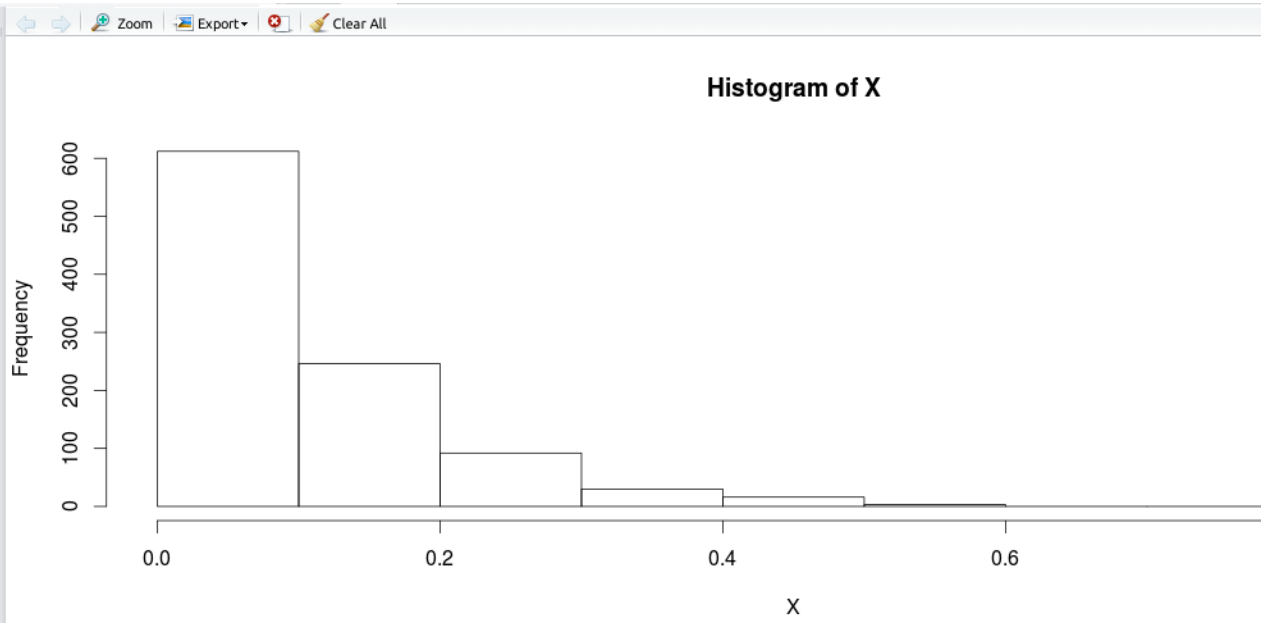


Método da transformação inversa

- Exemplo: gerando uma variável aleatória exponencial
  - $F(x) = 1 - e^{\lambda x}$, para $x \geq 0$ e $F(x) = 0$, caso contrário
  - $F^{-1}(x) = \frac{-1}{\lambda} \ln U$

Usando o R:

```
Console ~/
> U = runif( 1000 )
> X = (-1/10) * log(U)
> hist(X)
>
```



Método da transformação inversa

- Este método exige que a função de distribuição acumulada (cdf) F exista numa forma que a função inversa correspondente F^{-1} possa ser encontrada analiticamente ou algoritmicamente
- Algumas funções cujo método é aplicável são: exponencial, uniforme, Weibull, logistic, Cauchy
- Entretanto, para muitas outras distribuições de probabilidade, é ou impossível ou difícil encontrar a transformada inversa, ou seja, resolver

$$F(x) = \int_{-\inf}^{x} f(t)dt = u$$

com respeito a X.

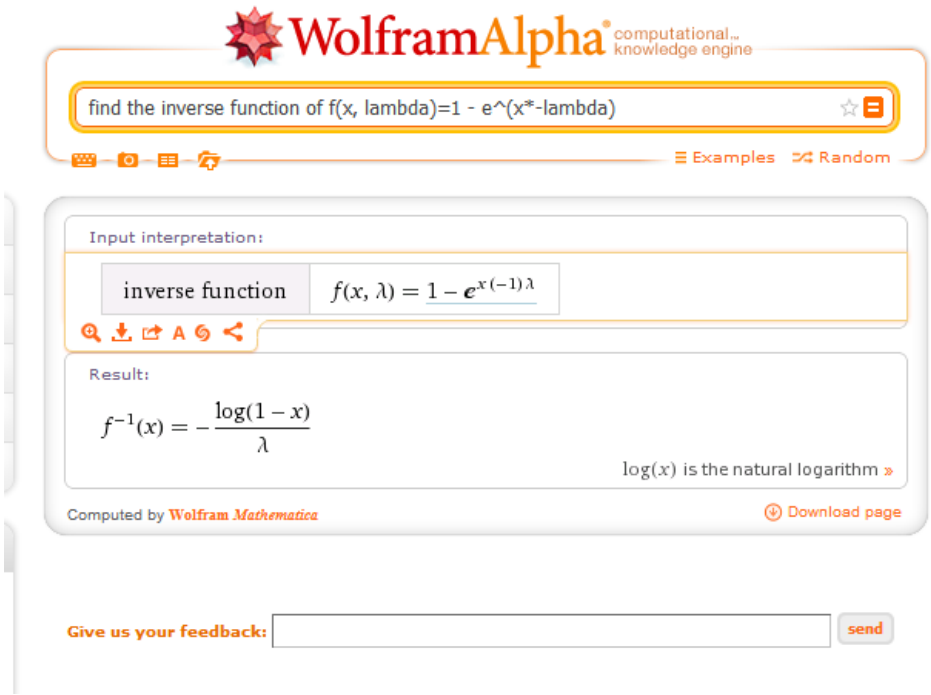
Método da transformação inversa

TABLE 28.1 Applications of the Inverse-Transform Technique

Distribution	CDF $F(x)$	Inverse
Exponential	$1 - e^{-x/a}$	$-a \ln(u)$
Extreme value	$1 - e^{-e^{(x-a)/b}}$	$a + b \ln \ln u$
Geometric	$1 - (1-p)^x$	$\left\lceil \frac{\ln(u)}{\ln(1-p)} \right\rceil$
Logistic	$1 - \frac{1}{1 + e^{(x-\mu)/b}}$	$\mu - b \ln\left(\frac{1}{u} - 1\right)$
Pareto	$1 - x^{-a}$	$1/u^{1/a}$
Weibull	$1 - e^{(x/a)^b}$	$a(\ln u)^{1/b}$

Método da transformação inversa

Podemos calcular a função inversa de uma determinada função com a ajuda do WolframAlpha (ou Mathematica):



Método da transformação inversa

- Outro exemplo: a distribuição Weibull
- Propriedades:
  - pdf - $f(x) = \frac{k}{\lambda}(\frac{x}{\lambda})^{k-1}e^{-(x/\lambda)^{\boldsymbol{k}}}$, para $x \geq 0$
  - cdf - $P(X \geq x) = 1 - e^{-(x/\lambda)^{\boldsymbol{k}}}$
  - função da taxa de falha - $h(t) = \frac{\gamma}{\theta}(\frac{t}{\theta})^{\gamma-1}$
  - cdf inversa - $F^{-1}(x) = \frac{(-ln(x))^{\frac{\mathbf{1}}{\alpha}}}{\lambda}$
- O parâmetro k é denominado *shape* (forma) e o λ é denominado *scale*

Método da transformação inversa

- Se uma quantidade x é um "Tempo até a falha" de algum dispositivo ou componente de um sistema, a distribuição Weibull representa uma distribuição para a qual a taxa de falha é proporcional a uma potência do tempo.
- A variação da taxa de falha é determinada pelo parâmetro k, da forma descrita a seguir:
  - $K < 1$ indica que a taxa de falha diminui com o passar do tempo
  - $k = 1$ indica que a taxa de falha é constante com o passar do tempo, o que sugere que as falhas são causadas por agentes externos ao sistema
  - $k > 1$ indica que a taxa de falha aumenta com o passar do tempo, devido a algum processo de envelhecimento do componente

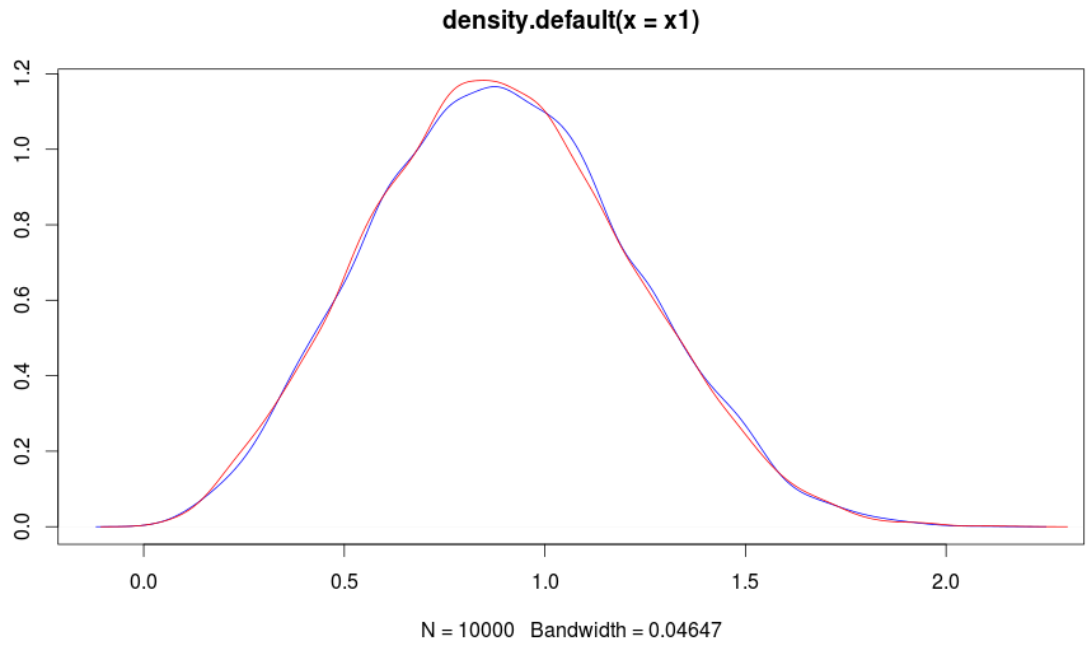
Método da transformação inversa

Implementação em R:

```
weibullgenerator = function(n, shape, scale){
  U = runif(n);

  return( (scale) * ((-log(U)) ^ (1/shape)) );
}

x1 = weibullgenerator(1e4, 3, 1);

d1 = density(x1);

plot(d1, col="blue");

x2 = rweibull(n=1e4, shape=3, scale=1);

d2 = density(x2);

lines(d2, col="red");
```

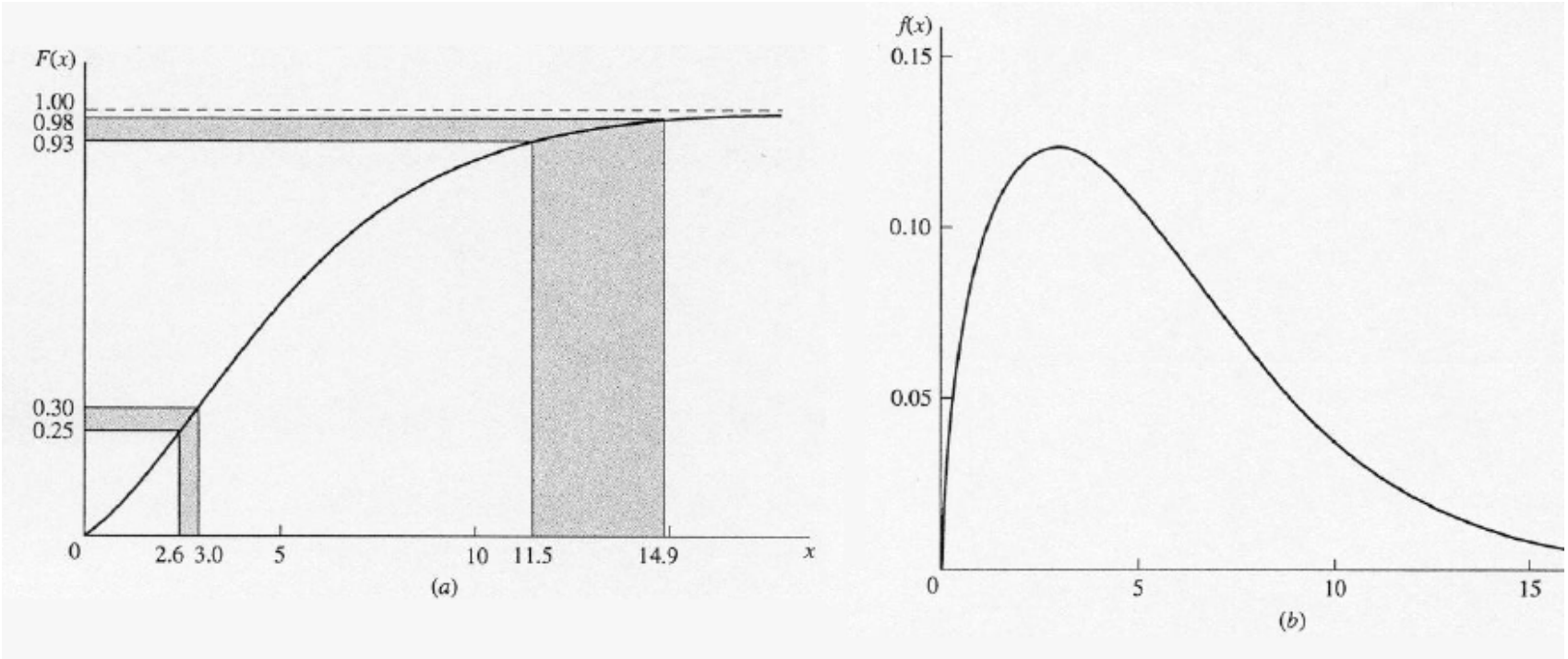
Método da transformação inversa

Implementação em R:



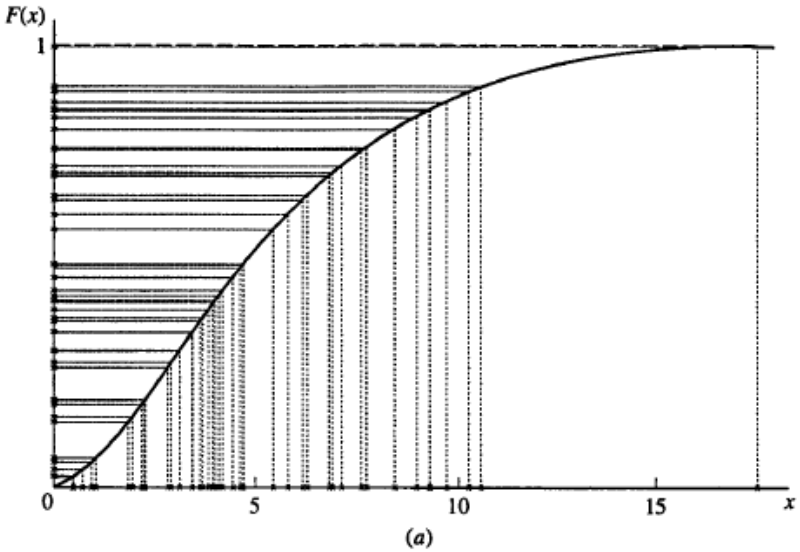
Método da transformação inversa

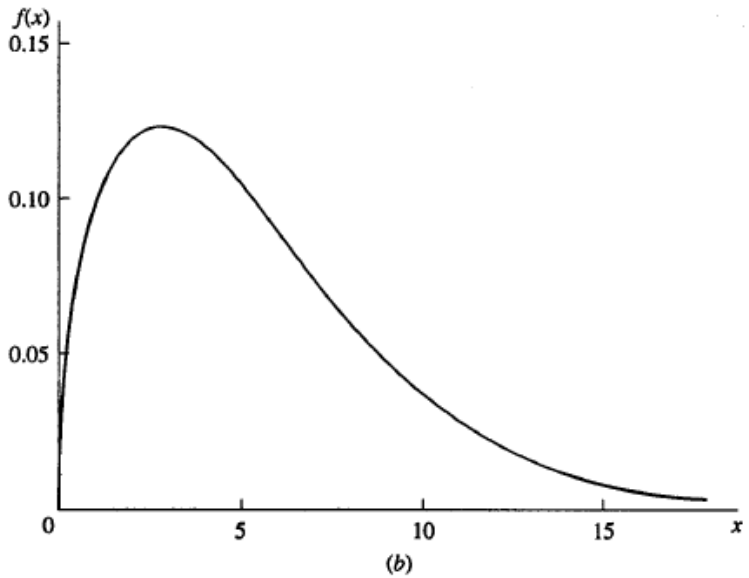
Entendendo o método da transformação inversa:



Método da transformação inversa

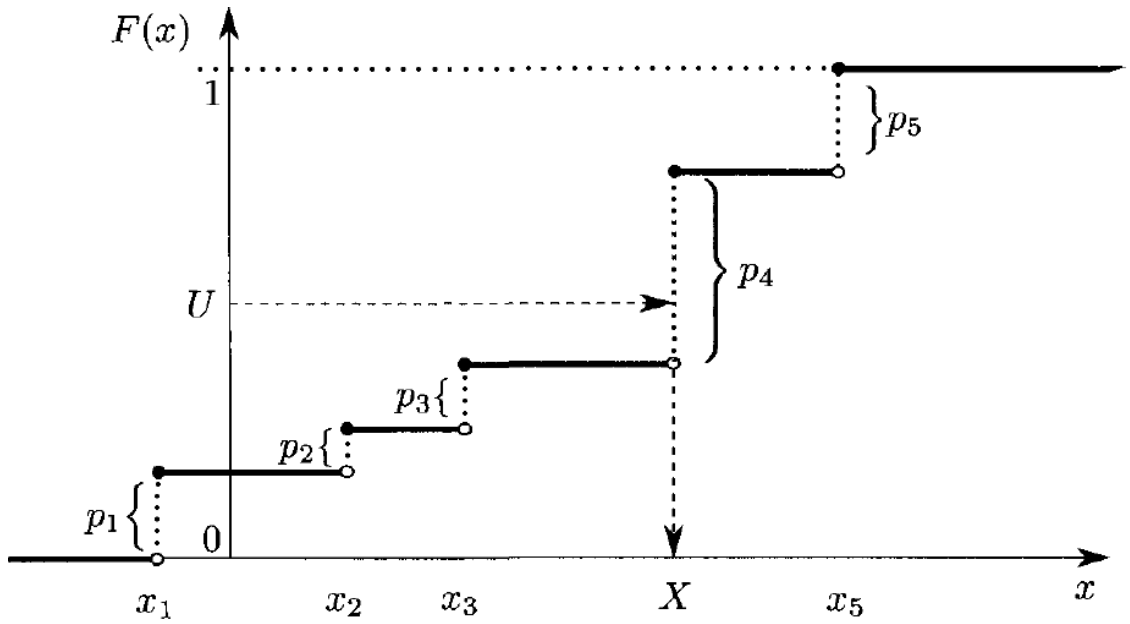
O algoritmo em ação:





Método da transformação inversa

- O método também pode ser usado para gerar variáveis aleatórias discretas:
  - Gere $U \sim U(0,1)$
  - Encontre o menor inteiro positivo k tal que $F(x_l) \geq U$, e retorne $X = x_k$



Método da transformação inversa

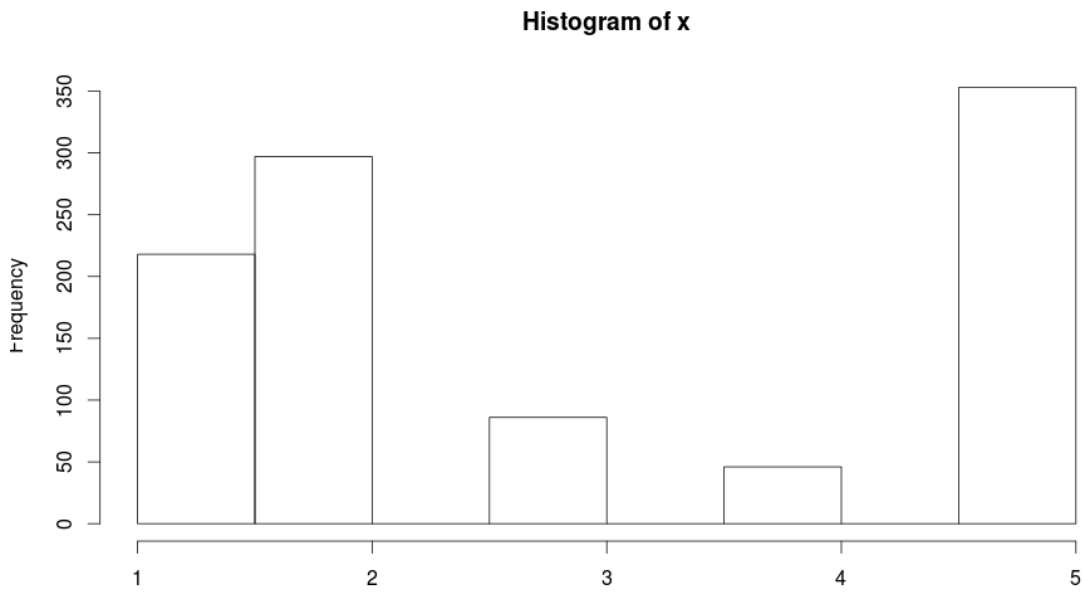
Exemplo:
Considere a seguinte distribuição discreta:

x	1	2	3	4	5
f(x)	0.2	0.3	0.1	0.05	0.35

A seguir, o código que implementa o método em R:

```
p = c(0.2, 0.3, 0.1, 0.05, 0.35)
F = cumsum( c(0.2, 0.3, 0.1, 0.05, 0.35) )

U = runif(1000)

x = rep(1:1000)

for(i in 1:1000){

  index = 1;

  while(U[i] > F[index]){
    index = index+1;

  }

  x[i] = index;
}

hist(x)
```

Método da transformação inversa



Sumario

Método da composição

- A técnica é utilizada quando a função de distribuição acumulada pode ser expressa como soma ponderada de outras.
- Comumente utilizada quando a transformação inversa é difícil ou lenta.

Método da composição

- Suponhamos que é possível encontrar uma função de distribuição acumulada(cdf) F(x) que satisfaça:
  - $F(x) = p_1.F_1(x) + p_2.F_2(x) + ...$
  - $p_1, p_2, ...$ são os pesos de cada cdf.
  - $F_1(x), F_2(x), ...$ são as cdfs componentes de F(x)

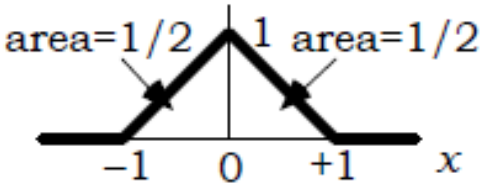
Método da composição

- Gerar um número aleatório que
  - $P(I = i) = p_i$
- Utiliza-se a i-ésima função de densidade de probabilidade para gerar o valor para a variável aleatória.

Método da composição

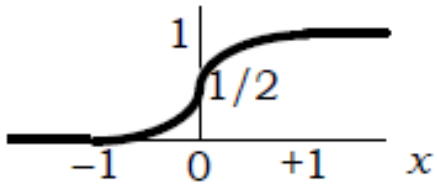
Symmetric triangular distribution on [–1, +1]:

$$\text{Density: } f(x) = \begin{cases} x+1 & \text{if } -1 \le x \le 0 \\ -x+1 & \text{if } 0 \le x \le +1 \\ 0 & \text{otherwise} \end{cases}$$



CDF:

$$F(x) = \begin{cases} 0 & \text{if } x < -1 \\ x^2/2 + x + 1/2 & \text{if } -1 \le x \le 0 \\ -x^2/2 + x + 1/2 & \text{if } 0 < x \le +1 \\ 1 & \text{if } x > +1 \end{cases}$$



Inverse-transform:

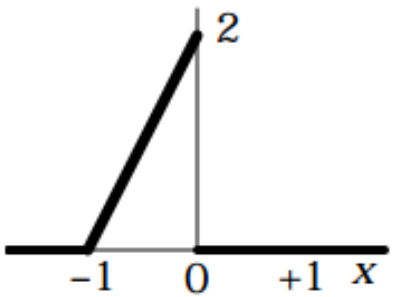
$$U = F(X) = \begin{cases} X^2/2 + X + 1/2 & \text{if } U < 1/2 \\ -X^2/2 + X + 1/2 & \text{if } U \ge 1/2 \end{cases}; \text{ solve for}$$

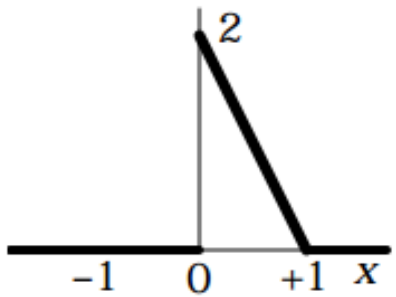
$$X = \begin{cases} \sqrt{2U} - 1 & \text{if } U < 1/2 \\ 1 - \sqrt{2(1-U)} & \text{if } U \ge 1/2 \end{cases}$$

Método da composição

Composition: Define *indicator function* for the set A as $I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$

$$\begin{aligned} f(x) &= (x+1)I_{[-1,0]}(x) + (-x+1)I_{[0,+1]}(x) \\ &= \underbrace{0.5}_{p_1}\underbrace{\{2(x+1)I_{[-1,0]}(x)\}}_{f_1(x)} + \underbrace{0.5}_{p_2}\underbrace{\{2(-x+1)I_{[0,+1]}(x)\}}_{f_2(x)} \end{aligned}$$





$F_1(x) = x^2 + 2x + 1$

$F_1^{-1}(U) = \sqrt{U} - 1$

$F_2(x) = -x^2 + 2x$

$F_2^{-1}(U) = 1 - \sqrt{1-U}$

Composition algorithm:

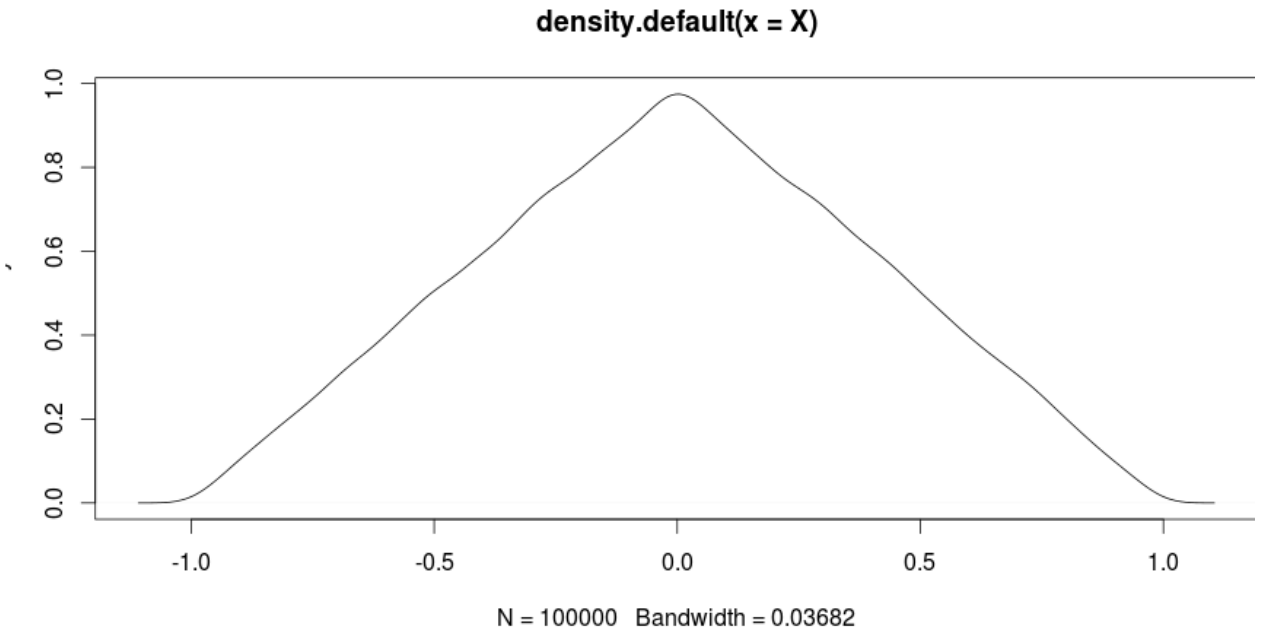
1. Generate $U_1, U_2 \sim \mathrm{U}(0,1)$ independently
2. If $U_1 < 1/2$, return $X = \sqrt{U_2} - 1$

   Otherwise, return $X = 1 - \sqrt{1-U_2}$

(Can eliminate $\sqrt{}$'s)

Método da composição

```r
N = 10000

U_1 = runif(N)
U_2 = runif(N)

X = rep(1:N)

for(i in 1:N){
  if(U_1[i] < 0.5){
    X[i] = sqrt(U_2[i]) - 1
  }else{
    X[i] = 1 - sqrt(1-U_2[i])
  }
}

d = density(X)
plot(d)
```



Sumario

Método da convolução

- Ideia semelhante à composição.
- No caso, é utilizado quando o valor de variável aleatória pode ser obtido através da soma de outras variaveis aleatórias.
- $x = y_1 + y_2 + ... + y_n$

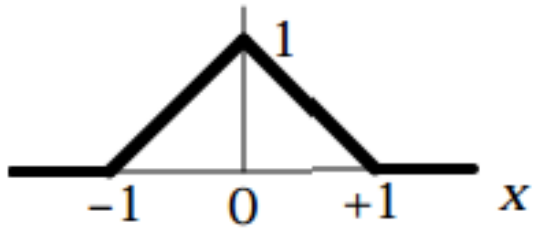
Método da convolução

- Algoritmo
  - Gerar n variáveis aleatórias (y1, y2, y3, . . .)
  - Retornar a soma destes valores.

Método da convolução

Symmetric triangular distribution on [–1, +1] (again):

$$\text{Density: } f(x) = \begin{cases} x+1 & \text{if } -1 \le x \le 0 \\ -x+1 & \text{if } 0 \le x \le +1 \\ 0 & \text{otherwise} \end{cases}$$



By simple conditional probability: If $U_1, U_2 \sim$ IID U(0,1), then $U_1 + U_2 \sim$ symmetric triangular on [0, 2], so just shift left by 1:

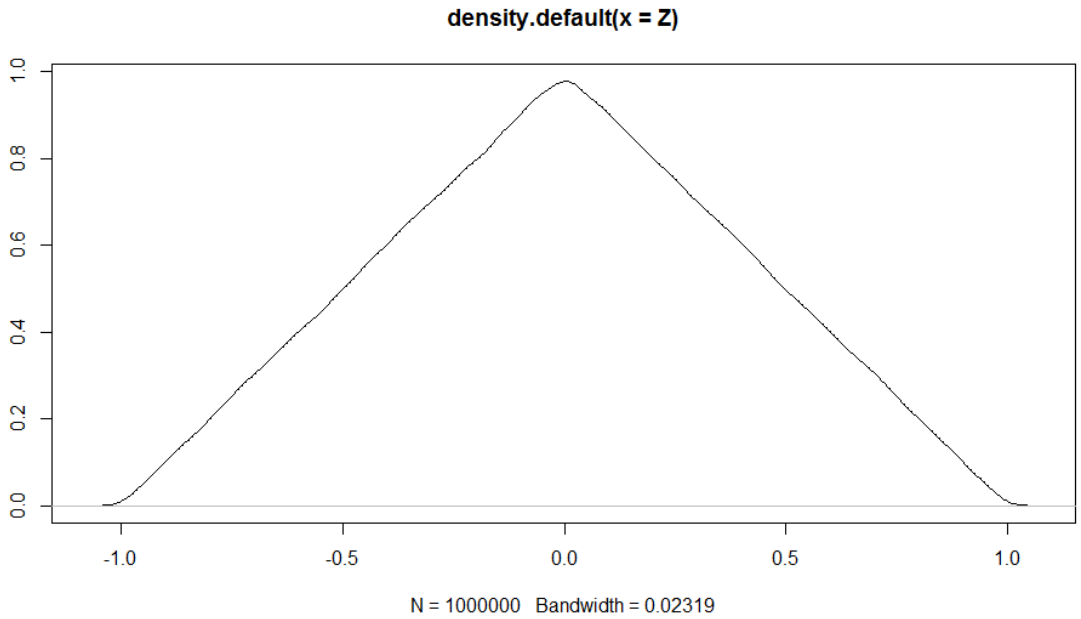
$$\begin{aligned} X &= U_1 + U_2 - 1 \\ &= \underbrace{(U_1 - 0.5)}_{Y_1} + \underbrace{(U_2 - 0.5)}_{Y_2} \end{aligned}$$

2 *U*s, 2 adds; no compares, multiplies, or $\sqrt{}$ s—clearly beats inverse transform, composition

Método da convolução

Exemplo anterior em R:

- X = (runif(1000000) - 0.5)
- Y = (runif(1000000) - 0.5)
- Z = X + Y
- d = density(Z)
- plot(d)



Método da convolução

Distribuições que podem ser geradas através desse método:

- Erlang-k = $\sum_{i=1}^{k} Exp(\lambda)$
- Binomial(n,p) = $\sum_{i=1}^{n} Bernoulli(p)$
  - Gere $n \times U(0,1)$ e retorne o número dos qe são menores que p
- Chi-square: $\chi^2(v) = \sum_{i=1}^{v} N(0,1)^2$
- $\Gamma(a, b_1) + \Gamma(a, b_2) = \Gamma(a, b_1 + b_2)$

Sumario

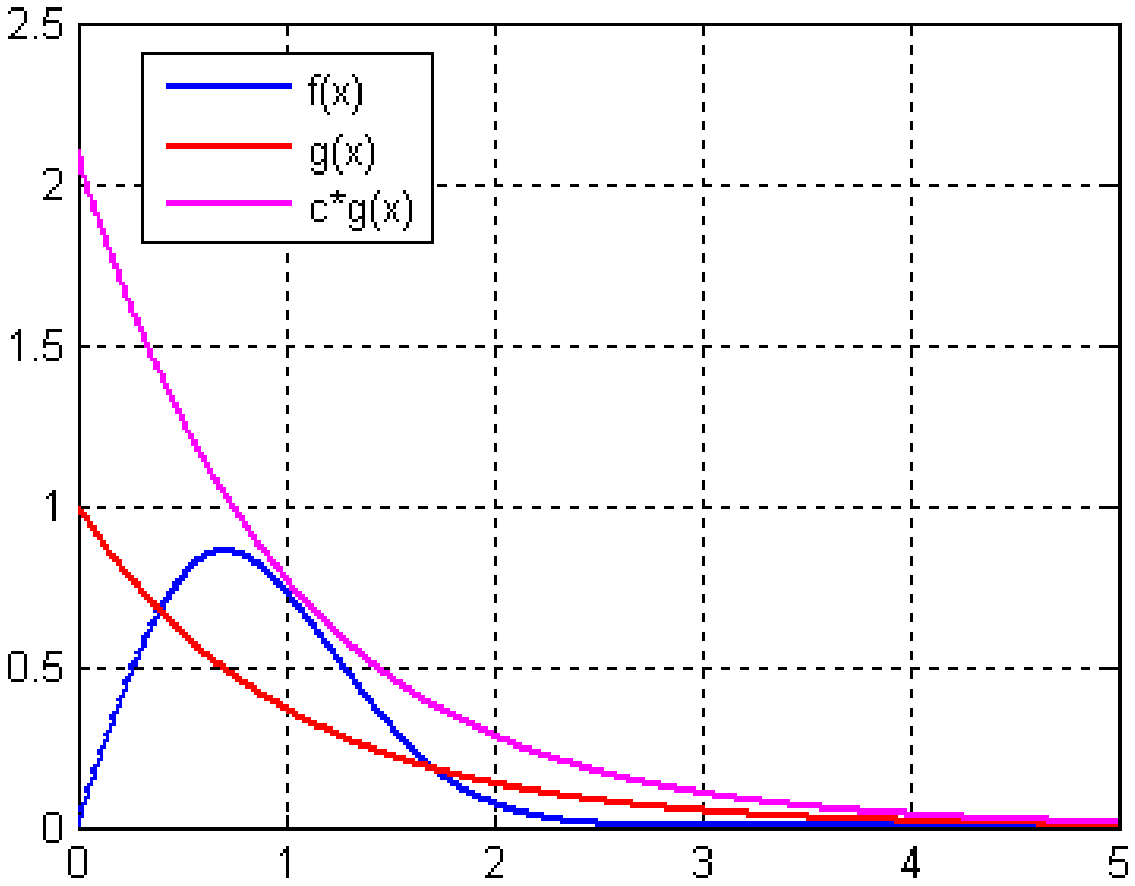
Método da aceitação-rejeição

- Este método é um dos mais úteis para realizar amostragem de distribuições genéricas. Ele consiste em encontrar uma função $g(x)$ tal que exista $cg(x)$ que domina a função de densidade $f(x)$, isto é, $cg(x) \geq f(x)$ para todos os valores de x
- O algoritmo é descrito a seguir:
  1. Escolha uma função de densidade g
  2. Escolha uma constante c tal que $f(x)/g(x) \leq c$ para todo x
  3. Gere um número aleatório uniforme u
  4. Gere um número aleatório v a partir de g
  5. Se $c.u \geq f(v)/g(v)$, aceite e retorne v
  6. Caso contrário, rejeite v e volte ao passo 3

Em outras palavras, gere $X \sim g$ e aceite isso com probabilidade $f(X)/(Cg(X))$.

Método da aceitação-rejeição

Descrição visual do método:



Método da aceitação-rejeição

Implementação do método em R:

```
accrejrnd = function(f, g, grnd, c, n) {
    X = rep(0, n);

    for (i in 0:n){
        accept = FALSE;

        while(! accept ){
            u = runif(1);
            v = grnd();

            if( c*u <= f(v)/g(v) ){
                X[i] = v;
                accept = TRUE;
            }
        }
    }

    return (X);
}
```

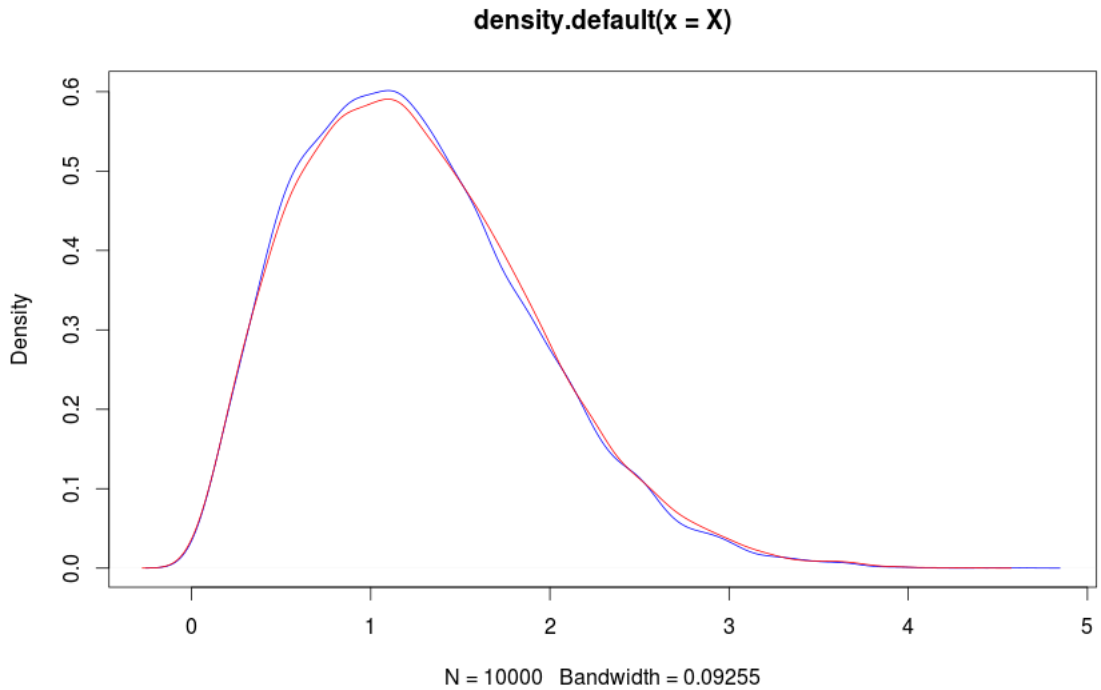
Método da aceitação-rejeição

Implementação do método em R (cont.):

```
#definindo as funções usadas pelo método
f = function(x) x * exp(-(x^2)/2)
g = function(x) exp(-x)
grnd = function() rexp(1)

X = accrejrnd(f, g, grnd, 2.2, 1e4)

#plotando a curva de densidade
d = density(X)
plot(d, col="blue")

#gerando a mesma variável aleatória
#com a função do pacote VGAM
library("VGAM")
Y = rrayleigh(1e4, 1)

d2 = density(Y)
lines(d2, col="red")
```

Método da aceitação-rejeição

Resultados do script anterior:



Método da aceitação-rejeição

- Variável aleatória gerada através de g deve ser simples, caso contrário o método será ineficiente
- A eficiência do método depende de como a função $g(x)$ "envelopa" a função $f(x)$. Quanto maior a área entre as funções, maior a quantidade de valores rejeitados

Sumario

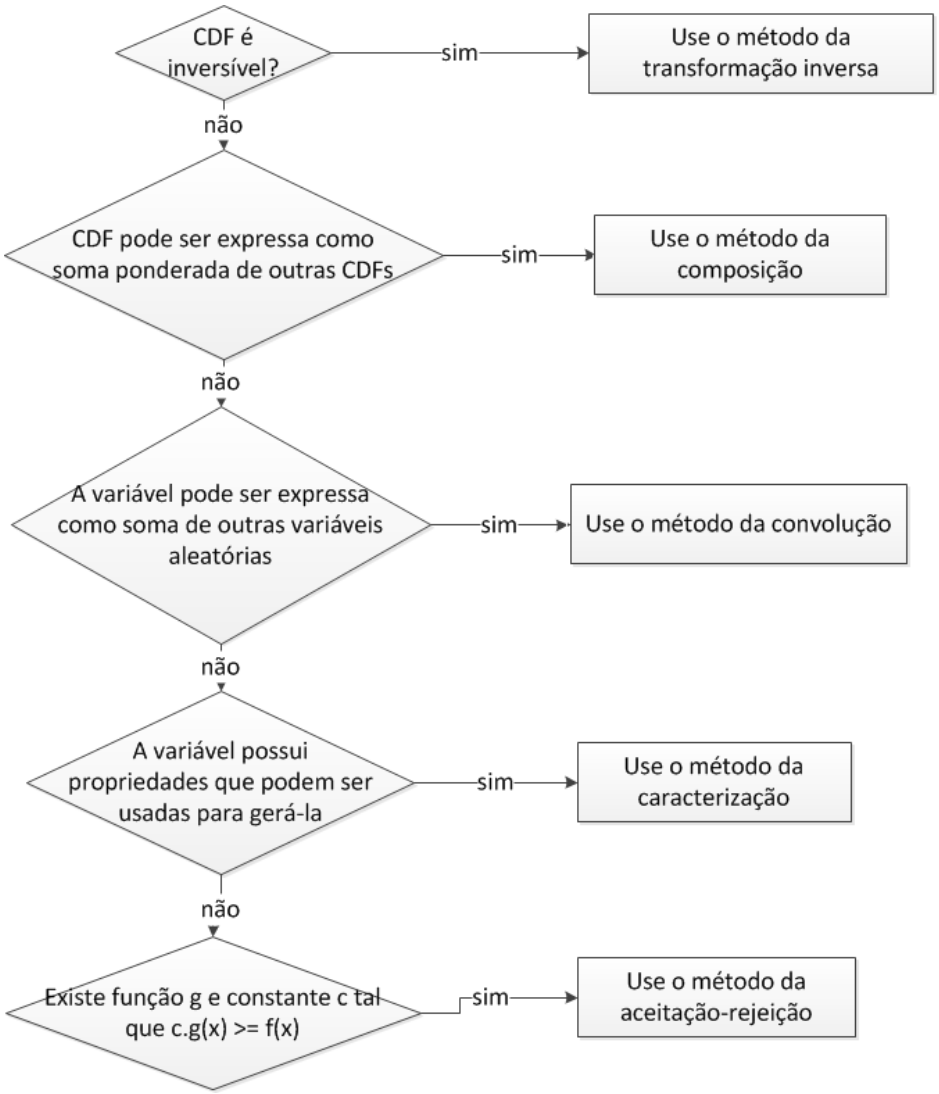
Método da caracterização

- Características especiais de algumas distribuições permitem que suas variáveis sejam geradas usando algoritmos especialmente adaptados para ela
- Exemplos de variáveis aleatórias que podem ser geradas com esta técnica:
  - Se o tempo entre chegadas de um processo é exponencialmente distribuído com média $1/\lambda$, o número de chegadas sobre um período T tem uma distribuição de Poisson com parâmetro λT. Portanto, uma variável de Poisson pode ser obtida continuamente gerando variáveis exponenciais até que sua soma exceda o período T, e retornando a quantidade de variáveis geradas
  - O a-ésimo menor número em uma sequência de $a + b + 1$ variáveis uniformes $U(0, 1)$, possui uma distribuição beta(a,b)
  - A razão entre duas variáveis normal padrão é uma variável Cauchy(0, 1)
  - Uma distribuição qui-quadrada com graus de liberdade par $\chi^2(v)$ é a mesma que uma variável gamma $\gamma(2, v/2)$

Método da caracterização

- Exemplos de variáveis aleatórias que podem ser geradas com esta técnica:
  - Se x_1 e x_2 são duas variáveis gamma $\gamma(a, b)$ e $\gamma(a, c)$, respectivamente, a razão $x_1/(x_1 + x_2)$ tem uma distribuição beta(b,c)
  - Se X tem uma variável normal padrão, $e^{\mu+\sigma x}$ é uma variável lognormal(μ, σ)

Sumário das técnicas



Sumario

Método de Monte Carlo

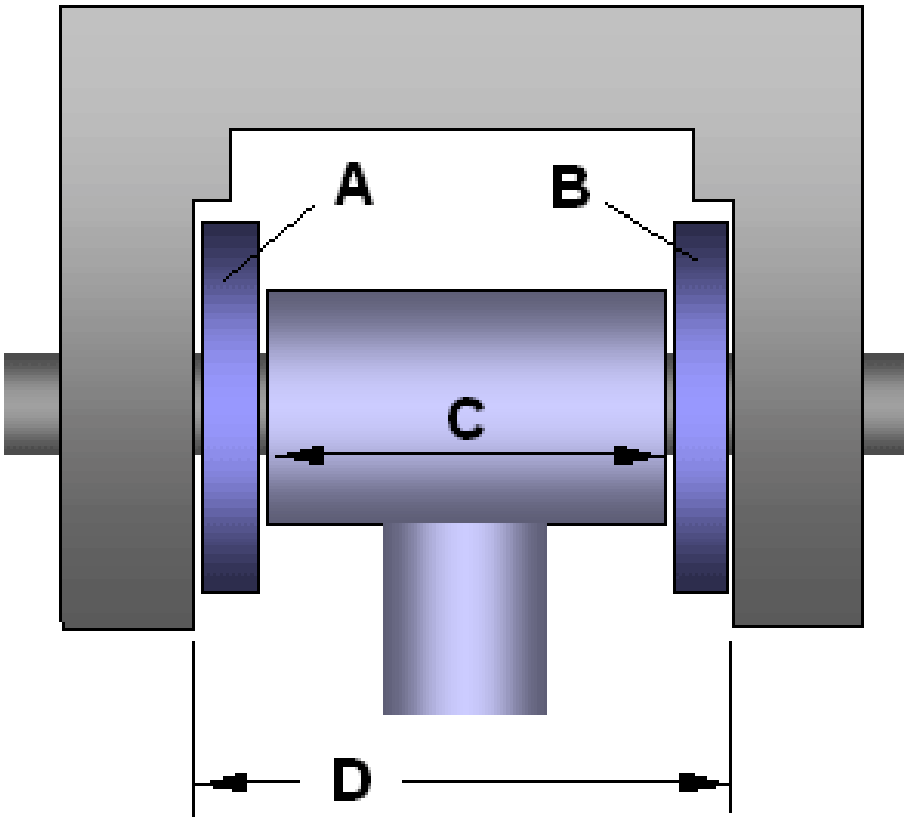
- Simulações estáticas ou que não variam o resultados conforme uma base temporal são chamadas simulações de Monte Carlo. Esta técnica é útil para modelar fenômenos probabilísticos que não variam com o avanço do tempo.
- A técnica se baseia na lei dos grandes números para aproximar valores esperados.
- Monte Carlo é uma técnica utilizada para analizar *propagação de incertezas*, cujo objetivo é determinar como variação aleatória, falta de conhecimento e erros afetam a sensitividade, desempenho ou confiabilidade do sistema sendo modelado

Método de Monte Carlo

O método de Monte Carlo pode ser implementado através do seguinte algoritmo:

1. Crie um modelo paramétrico $y = f(x_1, x_2, ...x_q)$
2. Gere um conjunto de parâmetros aleatórios $x_1, x_2, ..., x_q$
3. Avalie o modelo e armazene o resultado como y_i
4. Repita os passos 2 e 3 para i = 1 até n (n consideravelmente grande)
5. Analise os resultados utilizando resumos estatísticos, histogramas, intervalos de confiança, etc.

Exemplo: Modelo de construção de dobradiças



Método de Monte Carlo

- Queremos, através de um modelo de simulação, detectar a proporção de dobradiças defeituosas. Uma dobradiça é defeituosa quando o valor da folga: $D - (A + B + C)$ for menor que zero.
- Podemos simular a montagem de uma dobradiça selecionando valores para A, B, C e D aleatoriamente a partir de uma distribuição de probabilidade. Suponha que os tamanhos A, B, C e D sigam uma distribuição normal com os seguintes parâmetros:

Variavel	Média	Desvio padrão
A	2	0.05
B	2	0.05
C	30	0.5
D	34.5	0.5

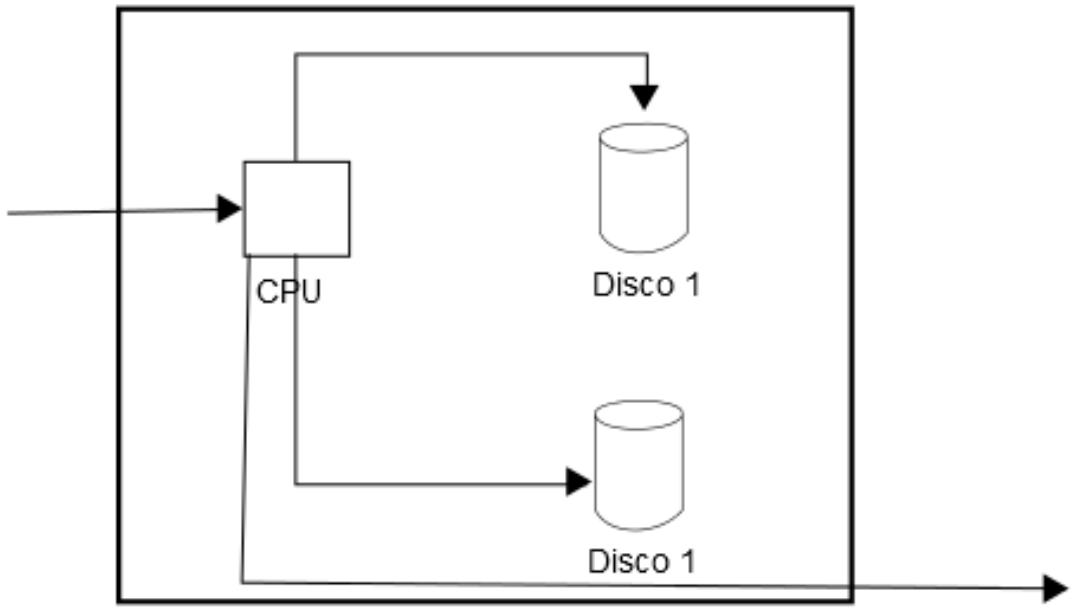
Método de Monte Carlo

```
n = 100000

A = rnorm(n, 2, 0.05)
B = rnorm(n, 2, 0.05)
C = rnorm(n, 30, 0.5)
D = rnorm(n, 34.5, 0.5)

folga = D - (A+B+C)

#Valor esperado da folga
print(mean(folga))

#Proporção de peças defeituosas
num_defeituosos = sum(folga < 0)
p = num_defeituosos/n
print(p)
```

Sumario

Atividade prática

Considere um servidor de arquivos composto por uma CPU e dois discos, representado pelo modelo da figura abaixo:



Atividade prática

Resumo de Análise Operacional:

- T - tempo de observação
- K - número de recursos
- B_i - tempo total ocupado do recurso i durante o tempo de observação T
- A_i - número de requisições feitas para o recurso i durante o tempo T
- C_i - número de requisições completadas pelo recurso i durante o tempo T
- A_0, C_0 - indica número de requisições feitas e completadas pelo sistema

Atividade prática

Resumo de Análise Operacional:

- S_i: tempo médio de serviço do recurso i: $S_i = B_i / C_i$
- U_i: utilização do recurso i: $U_i = B_i / T$
- X_i: vazão do recurso i: $X_i = C_i / T$
- X_0: vazão do sistema: $X_0 = C_0 / T$

Atividade prática

Leis operacionais:

- Lei da utilização: $U_i = X_i \times S_i = \lambda_i \times S_i$
- Lei do fluxo forçado: $X_i = V_i \times X_0$
- Lei da demanda de serviço: $D_i = V_i = S_i = U_i / X_0$
- Lei de Little: $N = X \times R$
- Lei do tempo de resposta interativo $= R = \frac{M}{X_0} - Z$

Atividade prática

Características do servidor de banco de dados:

- $S_1 = 0.02,\ S_2 = 0.32,\ S_3 = 0.24$
- $V_1 = 2,\ V_2 = 4,\ V_3 = 7$
- $\mathrm{T} = 1\mathrm{h}$
- $\lambda_0 = X_0 = 0.5$ req/sec

Atividade prática

Suponha que as condições do workload não mudem com o tempo, mas não sejam determinísticos. Considere que a partir de logs coletados, através de testes de aderência, conseguimos aproximar os parâmetros anteriores pelas seguintes distribuições:

- $S_1 \sim Norm(0.02, 0.01)$
- $S_2 \sim Norm(0.32, 0.05)$
- $S_3 \sim Norm(0.24, 0.05)$
- $V_2 \sim Geom(0.4)$
- $V_3 \sim Geom(0.18)$

Calcule o valor esperado da utilização de cada recurso, faça um resumo estatístico e gere intervalos de confiança para $\alpha = 0.05$