



**COMPARAÇÃO DE PROCEDIMENTOS  
ESTATÍSTICOS DE ALGUNS SOFTWARES  
USANDO SIMULAÇÃO DE DADOS**

**JOSÉ ERMELINO ALVES DAMASCENO**

**2003**

56979  
048658

**JOSÉ ERMELINO ALVES DAMASCENO**

**COMPARAÇÃO DE PROCEDIMENTOS ESTATÍSTICOS DE ALGUNS  
SOFTWARES USANDO SIMULAÇÃO DE DADOS**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-graduação em Agronomia, área de concentração em Estatística e Experimentação Agropecuária, para obtenção do título de “Mestre”.

**Orientador**  
**Prof. Ruben Delly Veiga**

**LAVRAS**  
**MINAS GERAIS – BRASIL**  
**JUNHO - 2003**

**Ficha Catalográfica Preparada pela Divisão de Processos Técnicos da  
Biblioteca Central da UFLA**

**Damasceno, José Ermelino Alves**

**Comparação de procedimentos estatísticos de alguns softwares  
usando simulação de dados / José Ermelino Alves Damasceno. – Lavras  
: UFLA, 2003.**

**64p. : il.**

**Orientador: Ruben Delly Veiga.  
Dissertação (Mestrado) – UFLA.  
Bibliografia.**

**1. Softwares. 2. Simulação. 3. Estatística. 4. Procedimentos.  
I. Universidade Federal de Lavras. II. Título.**

**CDD –005.1  
–519.5**

**JOSÉ ERMELINO ALVES DAMASCENO**

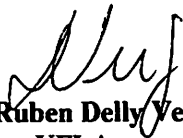
**COMPARAÇÃO DE PROCEDIMENTOS ESTATÍSTICOS DE ALGUNS  
SOFTWARES USANDO SIMULAÇÃO DE DADOS**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-graduação em Agronomia, área de concentração em Estatística e Experimentação Agropecuária, para obtenção do título de “Mestre”.

**APROVADA em 30 de junho de 2003**

Prof. Dr. Daniel Furtado Ferreira - UFLA

Prof. Dr. Paulo César Lima - UFLA

  
**Prof. Ruben Delly Veiga**  
UFLA  
(Orientador)

**LAVRAS  
MINAS GERAIS – BRASIL**

À memória de meu pai, Pedro.

À minha mãe, Arcena.

Às minhas irmãs, sobrinhas e cunhados.

**DEDICO**

## **AGRADECIMENTOS**

A Deus, por me dar certeza de sua existência e mostrar que sempre existe perspectiva.

À Universidade Federal de Lavras (UFLA), em especial ao Programa de Pós-graduação em Agronomia/Estatística e Experimentação Agropecuária, pela oportunidade de realização do curso.

Ao professor Ruben Delly Veiga pela orientação.

Ao professor Joel Augusto Muniz pelo incentivo.

Ao amigo Marcelo Ângelo Cirillo por estar sempre pronto a ajudar nos momentos de dificuldades.

Aos colegas de turma pelo companheirismo e amizade.

Ao inesquecível amigo João Marcos e a sua família que me acolheu e me ajudou a conquistar paz espiritual para levar adiante o curso.

Aos amigos Alberto, Torrinha, Bruno e Pig.

A todos os parentes e amigos que confiaram em mim e me apoiaram, em especial a minha mãe, irmãs e cunhados, meu reconhecimento.

# SUMÁRIO

<b>RESUMO</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>1 INTRODUÇÃO</b> .....	<b>1</b>
<b>2 REFERENCIAL TEÓRICO</b> .....	<b>3</b>
2.1 Erros em computação numérica .....	3
2.2 Simulação de dados .....	5
2.2.1 Jogos operacionais .....	6
2.2.2 Análise de Monte Carlo .....	7
2.3 Geração de variáveis estocásticas para simulação .....	7
2.3.1 Método da transformação inversa.....	8
2.3.2 Método da rejeição .....	10
2.3.3 Método da composição .....	11
2.3.4 Geração de uma variável aleatória com distribuição uniforme .....	12
2.3.5 Geração de uma variável aleatória com distribuição normal .....	15
2.3.6 Geração de uma variável aleatória com distribuição binomial.....	18
2.4 Geradores de variáveis aleatórias uniformes.....	19
2.4.1 Método de congruência para a geração de números aleatórios .....	20
2.4.2 Considerações computacionais.....	24
2.4.2.1 Problemas com o tamanho do registro dos números.....	24
2.4.2.2 Problemas com a precisão .....	25
2.5 Testes de geradores.....	25
2.5.1 Teste da frequência .....	26
2.5.2 Teste serial .....	27
2.5.3 Teste de Shapiro Wilk.....	29
2.6 Geradores fornecidos por alguns softwares .....	32
2.7 Estimação dos parâmetros de uma regressão linear.....	33
2.8 Confiabilidade de um software estatístico .....	36
<b>3 METODOLOGIA</b> .....	<b>38</b>
3.1 Geração de dados segundo distribuições de probabilidades conhecidas.....	38
3.2 Ajuste de uma Regressão linear simples – Simulação do método.....	39
3.3 Propriedades de estatísticas descritivas .....	41
<b>4 RESULTADOS E DISCUSSÃO</b> .....	<b>43</b>
4.1 Geração de distribuição estatística .....	43

4.2 Estimação dos coeficientes de uma regressão linear simples.....	45
4.3 Verificação de propriedades de estatísticas descritivas .....	47
<b>5 CONCLUSÕES .....</b>	<b>52</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>53</b>
<b>ANEXOS.....</b>	<b>55</b>



## RESUMO

DAMASCENO, José Ermelino Alves. **Comparação de procedimentos estatísticos de alguns softwares usando simulação de dados.** 2003. 64p. Dissertação (Mestrado em Estatística e Experimentação Agropecuária) – Universidade Federal de Lavras, Lavras, MG.<sup>1</sup>

Os procedimentos de análise de dados do Excel 2000, do SAS® 8.12 e do Matlab 6.1 foram testados e comparados utilizando os recursos de simulação de dados. Os três programas foram avaliados em três categorias de análise: geração de amostras aleatórias a partir de distribuições estatísticas conhecidas, estimação dos coeficientes de uma regressão linear simples e no cálculo e verificação de propriedades de algumas estatísticas. Para avaliar os geradores de amostras a partir de distribuições conhecidas, realizou-se uma simulação de dados composta de 1.000 experimentos considerando os diferentes tamanhos de amostras (30, 50, 75, 100 observações), os programas foram testados e comparados em termos do número de amostras geradas que não seguem a distribuição teórica e considerou-se o nível nominal de 5% de probabilidade. Em relação à estimação dos coeficientes de uma regressão linear simples, simularam-se 1.000 experimentos, cada qual com 30 observações, para cada experimento ajustaram-se modelos de regressão e estatísticas descritivas dos valores estimados foram obtidas e comparadas em termos de precisão e variabilidade. Para avaliar a precisão em relação ao cálculo de estatísticas descritivas e verificar algumas propriedades, gerou-se uma amostra aleatória de uma Poisson e constantes foram adicionadas e multiplicadas pelas observações em cada amostra. Problemas foram detectados no Excel 2000 em relação às propriedades das estatísticas descritivas, pois foi violada uma das propriedades da variância no que diz respeito à soma de constantes aos elementos de uma amostra, retornando valores totalmente discrepantes dos reais valores. Em relação à precisão das estimativas, os três programas apresentaram deficiências provocadas pelo arredondamento de fórmulas. Nas outras categorias de análise, não foram observados problemas que comprometessem o desempenho dos programas.

---

<sup>1</sup> Orientador: Ruben Delly Veiga – UFLA.

## ABSTRACT

DAMASCENO, José Ermelino Alves. **Comparison of statistical procedures of some softwares using simulation of data.** 2003. 64p. Dissertation (Master in Estatistics end Agricultural Experimentation) – Federal University of Lavras, Lavras, MG.<sup>1</sup>

The procedures of analysis of data of Excel 2000, of SAS® 8,12 and of Matlab 6.1 were tested and compared using the resources of simulation of data. The three programs were appraised in three analysis categories: generation of random samples starting from known statistical distributions; estimate of the coefficients of a simple linear regression and in the calculation and verification of properties of some statistics. To evaluate the generators of samples starting from known distributions, it took place a simulation of data composed of 1000 experiments considering the different sizes of samples (30, 50, 75, 100 observations), the programs were tested and compared in terms of the number of samples generated that don't follow the theoretical distribution, it was considered the nominal level of 5% of probability. In relation to the estimate of the coefficients of a simple linear regression, it was simulated 1000 experiments, each one with 30 observations, for each experiment was adjusted regression models and descriptive statistics of the dear values were obtained and compared in terms of precision and variability. To evaluate the precision in relation to the calculation of descriptive statistics and verification of some properties, a random sample of a Poisson was generated and constants were added and multiplied by the observations in each sample. Problems were detected in Excel 2000 in relation to the properties of the statistics, because it violated one of the properties of the variance in what concerns to the sum of constants to the elements of a sample, returning values totally conflicting of the real values, in relation to the precision of the estimates the three programs presented deficiencies provoked by the rounding of formulas. In the analysis categories, it was not observed problems to commit the acting of the programs.

---

<sup>1</sup> Adviser: Ruben Delly Veiga - UFLA

# 1 INTRODUÇÃO

É grande o número de softwares à disposição do usuário interessado em conduzir algum tipo de análise de dados. A utilização de um ou outro programa é de total responsabilidade do pesquisador. Considerando que a maioria desses softwares é protegido por patentes ou direitos autorais, a documentação da maioria deles não se encontra disponível para que seja consultada pelo usuário. Assim, informações relevantes sobre algoritmos, funções e precisão não podem ser utilizadas. Portanto, a escolha de um outro software recai na credibilidade da equipe responsável pelo seu desenvolvimento.

Se uma determinada análise é processada em mais de um software, e resultados diferentes são obtidos, o pesquisador não terá idéia de qual resultado é o correto. Portanto, os resultados de nenhum deles podem ser considerados confiáveis.

Alguns programas de análise estatística de dados mostram-se eficientes em algumas áreas, mas apresentam problemas em outras, o que muitas vezes se deve ao algoritmo usado para representar determinadas funções. Outros problemas podem ser acarretados devido ao fato de que os computadores representam os números em bases diferentes da decimal. Pequenas diferenças são notadas no método de conversão de decimal para a base utilizada pelo computador e vice-versa.

A precisão de um software é um conjunto de ferramentas para avaliar a sua performance numérica. Um importante aspecto da precisão é o fato de ela ser baseada na análise dos resultados computados, obtidos pela implementação computacional de um algoritmo numérico aplicado a um problema específico. Quando uma análise processada em mais de um software apresenta diferentes

resultados, torna-se necessário uma avaliação estatística das diferenças no sentido de mensurar o seu efeito sobre os resultados da análise.

A simulação de dados é uma poderosa ferramenta para avaliar e comparar a performance de softwares estatísticos, pois permite a realização de um modelo da situação real e nele levar a cabo experiências.

O presente trabalho teve por objetivo avaliar e comparar três softwares estatísticos na geração de distribuições estatísticas, estimação dos coeficientes de uma regressão linear simples, cálculo e verificação de propriedades da média e da variância, empregando a simulação de dados.

## 2 REFERENCIAL TEÓRICO

### 2.1 Erros em computação numérica

Segundo Ruggiero & Lopes (1996), um computador representa números na forma binária. Segundo o autor, a representação binária do decimal 0,1 é 0,0001100110011, em que o sublinhado indica uma repetição infinita da seqüência. Computadores, tendo precisão finita, não podem representar um número infinito de dígitos. Convertendo o binário 0,0001100110011 para forma decimal produz-se 0,99999964, o qual é preciso para sete casas decimais. Isto é o que o computador reconhece quando o número 0,1 é digitado. O número 100.000 é convertido de decimal para binário de maneira exata sem nenhum erro, mas somando  $0,1 + 100.000 = 100.000,1$ , convertendo para binário e depois para decimal temos 100.000,09375 o qual é preciso somente para duas decimais. Isto é ao, digitar  $100.000,1 - 100.000$  o computador lerá 0,09375.

Computadores geralmente cometem dois tipos de erros quando se envolvem com computação numérica: erros truncados e erros devido à representação binária de números com precisão finita.

Erro de arredondamento ocorre devido ao hardware; primariamente devido à precisão finita, isto é, um computador tem somente alguns bits com os quais representa vários números. Por exemplo, o IEEE-754 padrão para computação aritmética o qual é implementado no hardware de todo PC, tem precisão simples sobre seis ou sete dígitos, enquanto a precisão dupla tem 15 ou 16 dígitos de precisão. Ambos,  $a = 1.000\ 000$  e  $b = 0,000001$ , podem ser representados na precisão simples, mas sua soma,  $c = a + b$ , não pode. Em precisão simples o resultado seria  $c = 1.000\ 000$ , com os últimos dígitos significativos perdidos pelo erro de arredondamento. Em precisão dupla, a soma pode ser corretamente representada.

Outras conseqüências da precisão finita são erros que ocorrem ao se arredondarem duas fórmulas, pois enquanto algebricamente semelhantes, podem não ser numericamente equivalentes.

Exemplo:

$$\sum_{n=1}^{10000} n^{-2} \quad \text{e} \quad \sum_{n=1}^{10000} (10001-n)^{-2}$$

A primeira soma os termos na ordem crescente e a segunda, na ordem decrescente.

Erros truncados são conseqüências do software, e podem ser considerados um erro de aproximação. Algoritmos iterativos para problemas não lineares são sujeitos a erros de truncação porque o algoritmo somente produz a resposta correta para um número infinito de iterações, uma vez que na prática o número de iterações é finito. Como exemplo, considere calcular o seno de  $x$ ,  $\text{sen}(x)$ .

$$\text{sen}(x) = x - x^3 + \frac{x^5}{3!} - \frac{x^7}{7!} + \dots \quad (2.1)$$

Claramente, o computador não pode levar a soma até o infinito, e tem que parar a soma após algum número finito de termos, digamos  $k$ . Assumindo precisão finita, a diferença entre os valores exatos do  $\text{sen}(x)$  e valores encontrados pela soma de  $k$  termos é um erro truncado.

## 2.2 Simulação de dados

Os primeiros indícios de simulação de dados surgiram com a utilização do método de Monte Carlo, por Von Neuman, em 1940, com blindagem de reatores nucleares (Morgan, 1995).

Segundo Naylor (1971), a simulação é aplicada na construção de modelos de formas extremamente diversas, desde as esculturas e pinturas da Renascença até os modelos em escala, de aviões supersônicos e modelos analíticos de processos mentais, dessa forma, a simulação tornou-se algo quase que específico para cientistas e teóricos.

Conforme Naylor (1971), define-se simulação da seguinte maneira:

“x simula y” é verdade se e somente se:

- a) x e y forem sistemas formais;
- b) y for considerado como sendo o sistema real;
- c) x for considerado como sendo uma aproximação do sistema real;
- d) as regras de validade em x não estiverem isentas de erro.

Nos dias de hoje a simulação de dados é utilizada em diversas áreas de aplicação, basicamente sob duas linhas de atuação: problemas matemáticos completamente determinísticos, cuja solução é difícil; ou problemas que envolvem o processo estocástico de Monte Carlo, cuja técnica de simulação tem base probabilística ou estocástica.

O que torna a simulação de dados um recurso bastante utilizado é o fato de que esta técnica permite criar um modelo de uma situação real e nele levar a cabo experiências.

Para o uso da simulação em computadores digitais, torna-se necessário definir duas importantes variedades de simulação: jogos operacionais e análise de Monte Carlo.

### **2.2.1 Jogos operacionais**

Segundo Naylor (1971), a expressão “jogos operacionais” refere-se às simulações caracterizadas por alguma forma de conflito de interesses entre jogadores ou pessoas com poder de decisão dentro da estrutura do sistema simulado. Os jogadores ou pessoas que decidem agem dentro do sistema simulado e o experimentador, observando-os, deve ser capaz de testar as hipóteses relativas ao comportamento individual ou ao sistema de decisão como um todo.

As duas formas mais largamente usadas de jogos operacionais são os jogos militares e os jogos de gerência comercial. O jogo militar é essencialmente um dispositivo de treinamento para chefes militares que os capacita a testar os efeitos de estratégias diversas sob condições de guerra simulada. Jogos de negócios são também um tipo de ferramenta educacional para treinamento de chefes ou para executivos de negócios.

Um jogo de negócios é uma situação planejada que encaixa os jogadores em um sistema de negócios simulado em que eles devem tomar de tempos em tempos decisões de chefia. Suas escolhas geralmente afetam as condições do sistema em que a decisão subsequente deve ser tomada. Desta maneira, a interação entre decisão e o sistema é determinada por um processo de apuração que não sofre a influência dos argumentos dos jogadores.



### 2.2.2 Análise de Monte Carlo

Segundo o autor, a análise de Monte Carlo é uma técnica de simulação para problemas que têm base probabilística ou estocástica.

Dois diferentes tipos de problemas dão margem ao uso desta técnica:

**Primeiro:** os problemas que envolvem alguma forma de processo estocástico. A demanda de consumidores, o tempo de produção e o investimento total de uma organização são exemplos de variáveis econômicas que podem ser consideradas de natureza estocástica. Os métodos de Monte Carlo foram desenvolvidos não só para simular a maioria das bem conhecidas distribuições de probabilidades como para o caso das distribuições empíricas.

**Segundo:** certos problemas matemáticos completamente determinísticos não podem ser facilmente resolvidos por métodos estritamente determinísticos. Entretanto é possível obter um processo estocástico cujos momentos, funções de densidade ou funções de distribuição cumulativas satisfaçam as relações funcionais ou requisitos de solução de problemas determinísticos. Soluções de equações diferenciais de ordem superior e problemas de integrais múltiplas podem ser obtidas às vezes mais rapidamente pelo uso deste método de análise numérica que por outro método.

### 2.3 Geração de variáveis estocásticas para simulação

Naylor (1971) define a função  $F(x)$  chamada “função de distribuição acumulada” de  $x$ , que denota a probabilidade de que uma variável  $X$  possa assumir o valor de  $x$  ou menor. Se a variável aleatória for discreta, então  $x$  toma valores específicos. Se  $F(x)$  é contínua em todo o domínio de  $x$ , pode-se diferenciar esta função e definir:

$$f(x) = \frac{dF(x)}{dx}. \quad (2.2)$$

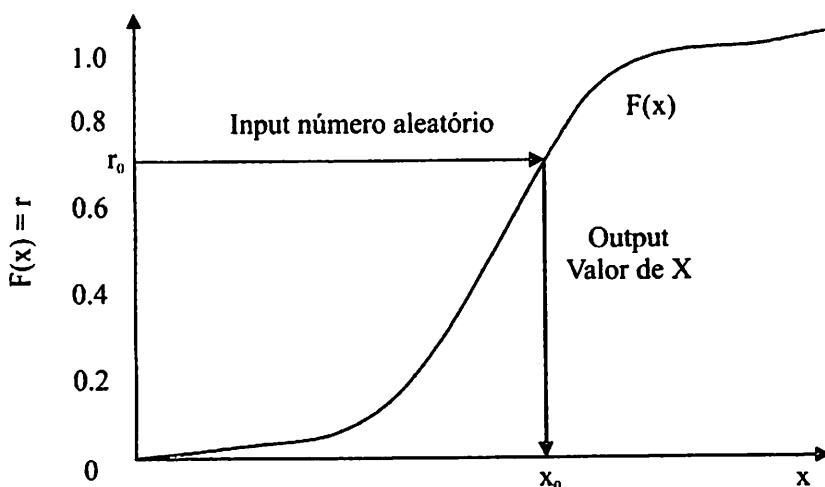
A derivada  $f(x)$  é chamada “função densidade de probabilidade”. A função  $F(x)$  pode ser matematicamente expressa como:

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t) dt, \quad (2.3)$$

em que  $F(x)$  é definida em todo o intervalo  $0 \leq F(x) \leq 1$ , e  $f(t)$  representa o valor da função densidade de probabilidade da variável aleatória  $X$  quando  $X = t$ .

### 2.3.1 Método da transformação inversa

Segundo Naylor (1971), para gerar valores aleatórios  $x_i$  de alguma população estatística particular, cuja função de densidade é dada por  $f(x)$ , deve-se primeiro obter a função de distribuição cumulativa  $F(x)$ .



**FIGURA 3-1.** Função de distribuição cumulativa.

Segundo o autor, uma vez que  $F(x)$  é definida em todo o intervalo 0 a 1, pode-se gerar números aleatórios uniformemente distribuídos e fazer  $F(x) = r$ . O valor de  $x$  é determinado univocamente por  $r = F(x)$ . Segue-se, por isso, que para qualquer valor particular de  $r$ , por exemplo,  $r_0$ , que tiver sido gerado, é possível achar o valor de  $x$ , nesse caso  $x_0$ , correspondente a  $r_0$ , pela função inversa de  $F$ , se a mesma for conhecida. Isto é:

$$x_0 = F^{-1}(r_0), \quad (2.4)$$

$F^{-1}(r)$  é a transformação inversa (ou mapeamento) de  $r$  no intervalo do domínio de  $x$ .

De modo geral, para gerar números aleatórios uniformes e fazer a correspondência para uma dada  $F(x)$ ,

$$r = F(x) = \int_{-\infty}^x f(t)dt ; \quad (2.5)$$

então:

$$P(X \leq x) = F(x) = P[r \leq F(x)] = P[F^{-1}(r) \leq x]; \quad (2.6)$$

conseqüentemente,  $F^{-1}(r)$  é uma variável que tem  $f(x)$  como sua função densidade de probabilidade.

Isto é equivalente a resolver a equação 2.5 para  $x$ , em termos de  $r$ .

### 2.3.2 Método da rejeição

Conforme Naylor (1971), se  $f(x)$  é limitada e  $x$  tem um domínio finito, ou seja,  $a \leq x \leq b$ , a técnica de rejeição pode ser usada para a geração de valores aleatórios. A aplicação desta técnica requer as seguintes etapas:

1. Normalização do domínio de  $f$  por um fator de escala "c" tal que:

$$c.f(x) \leq 1 \quad a \leq x \leq b .$$

2. Definir  $x$  como uma função linear de  $r$ ,

$$x = a + (b - a)r. \quad (2.7)$$

3. Gerar pares de números aleatórios  $(r_1, r_2)$ .
4. Sempre que encontrar um par de números aleatórios que satisfaça a relação:

$$r_2 \leq c.f[a + (b - a)r_1],$$

então aceitar o par e usar  $x = a + (b - a)r_1$  como valor aleatório gerado.

A teoria é baseada na consideração de a probabilidade de  $r$  ser menor ou igual a  $c.f(x)$  é dada por:

$$P[r \leq c.f(x)] = c.f(x). \quad (2.8)$$

Conseqüentemente, se  $x$  é escolhido ao acaso dentro do domínio  $(a, b)$ , de acordo com a equação 2.4 e então rejeitado se  $r > c.f(x)$ , a função de densidade de probabilidade dos  $x$  aceitos será exatamente  $f(x)$ .

### 2.3.3 Método da composição

Este método pode ser aplicado quando a função de distribuição da v.a.  $X$  é uma combinação de  $n$  funções de distribuição, ou seja:

$$F(x) = p_1 F_1(x) + p_2 F_2(x) + \dots + p_n F_n(x) = \sum_{j=1}^n p_j F_j(x); \quad (2.9)$$

$$\sum_{j=1}^n p_j = 1.$$

X pode ser vista como a variável gerada a partir de  $F_j$  com probabilidade  $p_j$ .

Pode-se aplicar o seguinte algoritmo:

1. Gera-se um inteiro positivo aleatório, tal que

$$P[J = j] = p_j \text{ para } j = 1, 2, \dots$$

2. Gera-se X a partir de  $F_j(x)$

$$P[X \leq x] = \sum_j P[X \leq x | J = j]P[J = j] = \sum_j F_j(x)p_j = F(x).$$

### 2.3.4 Geração de uma variável aleatória com distribuição uniforme

De acordo com Naylor (1971), o principal valor da distribuição uniforme em relação às técnicas de simulação repousa na simplicidade e no fato de que ela pode ser usada para simular variáveis aleatórias de quase todas as espécies de distribuições de probabilidades.

Matematicamente, a função de densidade uniforme é definida por:

$$f(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & \text{para outros valores} \end{cases} \quad (2.10)$$

A função de distribuição acumulada  $F(x)$  para uma variável aleatória uniformemente distribuída  $X$  é

$$F(x) = \int_a^x \frac{1}{b-a} dt = \frac{x-a}{b-a} \quad 0 \leq F(x) \leq 1. \quad (2.11)$$

O valor esperado e a variância de uma variável aleatória uniformemente distribuída são dados por:

$$E(X) = \int_a^b \frac{1}{b-a} x dx = \frac{b+a}{2} \quad (2.12)$$

e

$$V(X) = \int_a^b \frac{(x-E(X))^2}{b-a} dx = \frac{(b-a)^2}{12}. \quad (2.13)$$

Os valores dos parâmetros são obtidos através da resolução do sistema constituído pelas equações 2.12 e 2.13, em relação à “a” e “b”, uma vez que  $E(X)$  e  $V(X)$  são, por hipótese, conhecidos.

Este processo, semelhante ao método dos momentos, fornece as seguintes expressões:

$$a = E(X) - \sqrt{3V(X)} \quad (2.14)$$

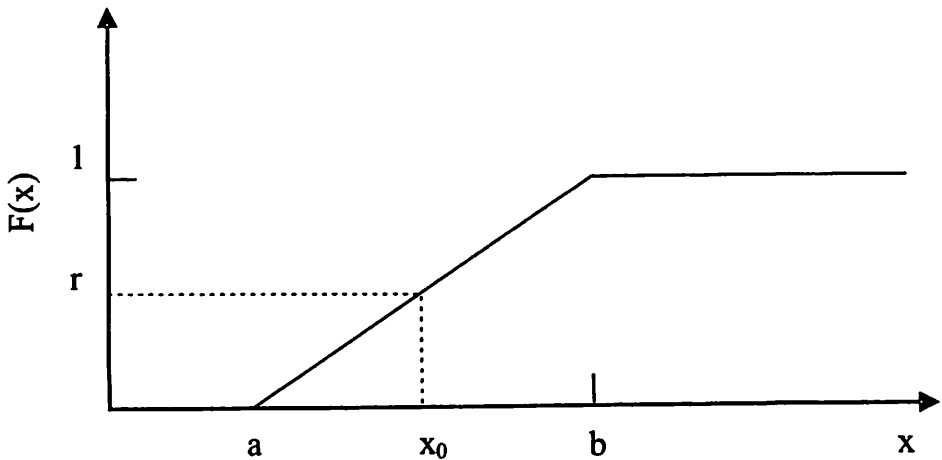
e

$$b = 2E(X) - a. \quad (2.15)$$

Para simulação de uma distribuição uniforme em um dado domínio (a, b), deve-se primeiro obter a transformação inversa para a equação 2.11, de acordo com a equação 2.2.

$$x = a + (b - a)r \quad 0 \leq r \leq 1. \quad (2.16)$$

Gera-se então um conjunto de números aleatórios correspondentes ao domínio das probabilidades acumuladas, isto é, valores aleatórios uniformes definidos no domínio de 0 a 1. Cada número aleatório  $r$  determina univocamente um valor uniformemente distribuído  $x$  (Figura 3-2).



**FIGURA 3-2.** Geração de números aleatórios através do uso das probabilidades cumulativas.



### 2.3.5 Geração de uma variável aleatória com distribuição normal

Se uma variável aleatória  $X$  tem uma função de densidade  $f(x)$  dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty < x < \infty, \quad (2.17)$$

em que  $\sigma$  é positivo, diz-se então que possui uma distribuição normal, com parâmetros  $\sigma$  e  $\mu$ .

O valor esperado e a variância da distribuição normal são dados por:

$$E(X) = \mu \quad (2.18)$$

e

$$V(X) = \sigma^2. \quad (2.19)$$

Para simular uma distribuição normal com um dado valor esperado  $\mu$  e um certo desvio padrão  $\sigma$ , a seguinte interpretação do teorema central do limite pode ser dada:

Se  $r_1, r_2, \dots, r_N$  são variáveis aleatórias independentes, cada uma com a mesma distribuição de probabilidades, com  $E(r_i) = \theta$  e  $V(r_i) = \sigma^2$ , então:

$$\lim_{N \rightarrow \infty} P \left[ a < \frac{\sum_{i=1}^N r_i - N\theta}{\sqrt{N}\sigma} < b \right] = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}z^2} dz, \quad (2.20)$$

em que:

$$E\left(\sum_{i=1}^N r_i\right) = N\theta, \quad (2.21)$$

$$V\left(\sum_{i=1}^N r_i\right) = N\sigma^2, \quad (2.22)$$

e

$$z = \frac{\sum_{i=1}^N r_i - N\theta}{\sigma\sqrt{N}}. \quad (2.23)$$

O processo para a simulação de valores normais em um computador inclui a soma de  $K$  valores aleatórios uniformemente distribuídos  $r_1, r_2, \dots, r_K$ , em que  $r_i$  é definido no intervalo  $0 \leq r_i \leq 1$ . Aplicando então a notação matemática do teorema central do limite e com o conhecimento prévio da distribuição uniforme, tem-se:

$$\theta = \frac{a+b}{2} = \frac{0+1}{2} = \frac{1}{2}, \quad (2.24)$$

$$\sigma = \frac{b-a}{\sqrt{12}} = \frac{1}{\sqrt{12}}, \quad (2.25)$$

e

$$z = \frac{\sum_{i=1}^K r_i - \frac{K}{2}}{\sqrt{\frac{K}{12}}}. \quad (2.26)$$

Como  $z$  é um valor normal padrão, obtém-se:

$$\frac{x - \mu}{\sigma} = \frac{\sum_{i=1}^K r_i - \frac{K}{2}}{\sqrt{\frac{K}{12}}}; \quad (2.27)$$

resolvendo em relação a  $x$ , tem-se:

$$x = \sigma \left( \frac{12}{K} \right)^{\frac{1}{2}} \left( \sum_{i=1}^K r_i - \frac{K}{2} \right) + \mu. \quad (2.28)$$

Para gerar um único valor de  $x$  simplesmente somam-se  $K$  números aleatórios definidos no intervalo 0 a 1. Substituindo o valor desse somatório na equação (2.14), bem como os valores de  $\mu$  e de  $\sigma$  para a distribuição desejada, determina-se um valor particular de  $x$ . Este processo pode ser repetido tantas vezes quanto forem os valores de distribuição normal requeridos.

### 2.3.6 Geração de uma variável aleatória com distribuição binomial

A distribuição binomial dá a probabilidade de que um evento favorável ocorra  $x$  vezes em  $n$  ensaios, em que a probabilidade de sucesso é  $p$ . A função de probabilidade para a distribuição binomial pode ser expressa como:

$$f(x) = \binom{n}{x} p^x q^{n-x}, \quad (2.29)$$

em que  $x$  é um valor inteiro definido no intervalo finito  $1, 2, 3, \dots, n$  e  $q = (1 - p)$ . O valor esperado e a variância da variável binomial  $X$  são:

$$E(X) = np \quad (2.30)$$

e

$$V(X) = npq. \quad (2.31)$$

Conhecidos  $E(X)$  e  $V(X)$ , tem-se:

$$p = \frac{(E(X) - V(X))}{E(X)} \quad (2.32)$$

e

$$n = \frac{E(X)^2}{(E(X) - V(X))}. \quad (2.33)$$

Os valores binomiais podem ser gerados de muitas maneiras diferentes, porém o método mais simples, para  $n$  não muito grande, é baseado na reprodução dos ensaios de Bernoulli, através de técnica de rejeição.

O processo começa com valores conhecidos de  $p$  e  $n$  e consiste na geração de  $n$  números aleatórios, após fixado  $x_0 = 0$ . Para cada número aleatório  $r_i$ , ( $1, 2, \dots, n$ ) faz-se um teste e a variável  $x_i$  é calculada da maneira que se segue:

$$x_i = x_{i-1} + 1 \quad \text{se } r_i \leq p \quad (2.34)$$

$$x_i = x_{i-1} \quad \text{se } r_i > p. \quad (2.35)$$

Após terem sido gerados  $n$  números aleatórios, o valor de  $x_n$  é igual ao valor binomial  $x$ . Este processo pode então ser repetido tantas vezes quantos forem os valores binomiais requeridos, ou seja,  $n$  vezes.

## 2.4 Geradores de variáveis aleatórias uniformes

A performance de uma simulação está fortemente correlacionada com o gerador de uniformes usado.

Uma seqüência de variáveis aleatórias  $(X_1, \dots, X_N)$  é dita “uma amostra de tamanho  $N$ ” da distribuição uniforme  $(0, 1)$  se:

- (a) para cada  $k \geq 2$  e cada  $1 \leq i_1 < \dots < i_k \leq N$  vale  $P(X_{i_1} \leq x_1, \dots, X_{i_k} \leq x_k) = P(X_{i_k} \leq x_k) = x_1 \cdot x_2 \cdot \dots \cdot x_k$  (2.36) quaisquer que sejam  $x_1, \dots, x_k$  em  $(0, 1)$ ;

(b) para cada  $i = 1, \dots, N$  e cada  $x \in R$  vale

$$P(X_i \leq x) = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{se } 0 \leq x \leq 1 \\ 1 & \text{se } x > 1. \end{cases} \quad (2.37)$$

Um bom gerador de uma seqüência  $x_1, \dots, x_N$  que seja uma realização das variáveis aleatórias  $X_1, \dots, X_N$ , além de passar em testes verificando (a) e (b), deve satisfazer as seguintes propriedades:

1. repetibilidade: dados os mesmos parâmetros, o gerador produza a mesma seqüência sempre que assim se desejar;
2. velocidade computacional: a velocidade está ligada à precisão desejada nos resultados finais da simulação na qual é usado o gerador. Quanto mais rápido seja o gerador, mais resultados serão obtidos no mesmo tempo de uso de computador. Em geral, para obter maior velocidade computacional será conveniente programar o algoritmo de geração usando uma linguagem de “baixo nível”.

#### 2.4.1 Método de congruência para a geração de números aleatórios

Os métodos de congruência para geração de números aleatórios são perfeitamente determinísticos porque os processos aritméticos envolvidos nos cálculos determinam univocamente cada termo na seqüência de números. De fato, existem fórmulas que permitem calcular antecipadamente o exato valor do  $i$ -ésimo termo de uma seqüência de números aleatórios  $(n_0, n_1, n_2, \dots, n_j, \dots)$  antes que a seqüência seja realmente gerada (Naylor, 1971).





Dos inteiros da seqüência  $n_i$  pode-se obter números racionais no intervalo unitário  $(0, 1)$  formando a seqüência  $\{r_i\} = \left\{ \frac{n_i}{m} \right\}$ .

Três métodos básicos de congruência foram desenvolvidos para a geração de números aleatórios, usando diferentes versões da fórmula dada pela equação (2.38). O objetivo de cada método é gerar seqüências com um período máximo, num mínimo de tempo. Estes métodos são chamados método da congruência aditiva, método da congruência multiplicativa e método da congruência mista (Naylor, 1971).

O método da congruência aditiva envolve  $k$  valores iniciais em que “ $k$ ” é um inteiro positivo, e calcula uma seqüência de números pela relação de congruência:

$$n_{i+1} = n_i + n_{i-k} \pmod{m} \quad (2.40)$$

Se  $k = 1$ , a equação (2.40) gera a conhecida seqüência de Fibonacci, que tem um comportamento semelhante às seqüências obtidas pelo método da congruência multiplicativa. As propriedades estatísticas da seqüência tendem a melhorar quando  $k$  cresce.

O método da congruência multiplicativa calcula uma seqüência  $\{n_i\}$  de inteiros não negativos, todos menores que “ $m$ ”, por meio da relação de congruência:

$$n_{i+1} = an_i \pmod{m}. \quad (2.41)$$



Este método é um caso especial da equação (2.38) em que  $c = 0$ . O método multiplicativo tem-se revelado estatisticamente bom. Isto é, os teste de frequência e seqüenciais, bem como outros testes de aleatoriedade, quando aplicados às seqüências geradas desta maneira, indicam que os números aleatórios são uniformemente distribuídos e não correlacionados.

Além disso, é possível impor condições tanto aos multiplicadores “a” como aos valores iniciais  $n_0$  de modo a assegurar um período máximo para as seqüências geradas por este método. O método multiplicativo oferece ainda vantagens relativas em termos de velocidade computacional.

Os números obtidos através da relação de congruência na sua forma original (equação 2.38), com “a” e “c” maiores que zero, são ditos serem gerados pelo método da congruência mista.

O método misto tem demonstrado algumas vantagens em relação ao multiplicativo no que diz respeito à maior velocidade computacional e à ausência de periodicidade nos últimos dígitos. Sua principal vantagem está no período amplo. Embora seu comportamento estatístico seja geralmente bom, em uns poucos casos ele é completamente inaceitável.

McLarem & Marsaglia (1965) sugeriram um método combinado no qual um método de congruência mista calcula os índices que determinam qual dos “p” números aleatórios previamente armazenados deve ser o seguinte na seqüência. Os “p” números  $n_1, n_2, \dots, n_i, \dots, n_p$  são gerados pelo método de congruência multiplicativa, de tal maneira que o i-ésimo número é substituído por um novo valor  $n_i$  se “i” é o número índice gerado pelo método misto. McClarem & Marsaglia (1965) usaram  $p = 128$  nos testes do método. O método combinado passou por todos os testes estatísticos que foram aplicados.

## 2.4.2 Considerações computacionais

O computador que está sendo usado deve ser capaz de computar corretamente o algoritmo de geração escolhido. O computador a ser utilizado na simulação de dados deve ter configuração suficiente para processar corretamente o algoritmo proposto. Essa afirmação parece óbvia, mas não é, se considerar que existem alguns problemas, tais como os que seguem:

### 2.4.2.1 Problemas com o tamanho do registro dos números

Considere o gerador:

$$n_{i+1} = 16807n_i \bmod (2^{31} - 1). \quad (2.42)$$

Dado que a seqüência assim construída está contida no conjunto  $\{1, \dots, 2^{31} - 2\}$  parece que se pode codificar o algoritmo de geração correspondente usando apenas inteiros de 32 bits. Porém, se  $n_i = 2^{31} - 2$  tem-se que  $16807n_i$  não cabe em um registro desse tamanho (32 bits); assim, quando se tenta fazer o produto ocorre um erro de overflow. Se o compilador em uso não estiver desenhado para avisar de tais erros, obterá, inadvertidamente, uma seqüência que não corresponderia às geradas verdadeiramente pelo gerador em questão. Assim, nada se poderia dizer sobre as propriedades da seqüência obtida.

Uma versão correta do algoritmo deveria se basear sobre operações aritméticas reais em precisão dupla.

O mesmo tipo de problema pode acontecer quando se tenta a implementação de outros geradores, mesmo não sendo eles do tipo congruencial.

#### **2.4.2.2 Problemas com a precisão**

Mesmo não ocorrendo erros de overflow, os compiladores apresentam sérias deficiências no tocante à precisão de diversas operações aritméticas.

Às vezes, embora muito infreqüentemente, é possível ter acesso à documentação sobre as limitações das aritméticas implementadas, que deveriam formar parte importante do conjunto dos manuais fornecidos pelo vendedor do equipamento computacional em uso.

Alguns autores têm documentado a necessidade de um conjunto padrão de testes para pacotes estatísticos (Wilkinson, 1985; Wampler, 1980; McCullough, 1998). Estes dizem ter incluído uma solicitação para vendedores de software com o intuito de relatar a precisão deles usando estes testes. Uma importante característica desta bateria de testes é que eles revelarão informações sobre a precisão de algoritmos do software desenvolvido.

#### **2.5 Testes de geradores**

É importante para todo pesquisador envolvido em trabalhos de simulação, conhecer a performance de cada gerador em uso e saber em que medida essa performance pode afetar os resultados finais do trabalho (Bustos & Orgambide, 1992).

Segundo Naylor (1971), as propriedades estatísticas dos números pseudo-aleatórios gerados pelos mais diversos métodos devem coincidir com as propriedades dos números gerados por qualquer dispositivo que selecione números no intervalo unitário, independentemente uns dos outros, e com todos eles igualmente prováveis.

Os números pseudo-aleatórios produzidos em computadores não são aleatórios na acepção do termo, uma vez que não são completamente determinados a partir dos dados iniciais, e tem precisão limitada. Porém, uma vez que os números pseudo-aleatórios podem passar pelo conjunto de testes estatísticos produzido pelo dispositivo aleatório, poderão ser tratados como verdadeiros números aleatórios, ainda que na realidade não o sejam.

Os testes estatísticos seguintes estão entre os mais importantes citados na literatura para a avaliação da aleatoriedade dos números gerados.

### 2.5.1 Teste da frequência

Segundo Naylor (1971), o teste da frequência é utilizado para verificar a uniformidade de uma seqüência de  $M$  conjuntos consecutivos com  $N$  números pseudo-aleatórios. Para cada conjunto de  $N$  números pseudo-aleatórios  $r_1, r_2, \dots, r_N$  divide-se o intervalo unitário  $(0, 1)$  em  $x$  subintervalos iguais. A quantidade esperada de números aleatórios em cada subintervalo é  $\frac{N}{x}$ . A seguir, seja  $f_j$ , em que  $j = 1, 2, \dots, x$  a quantidade real de números pseudo-aleatórios  $r_i$  ( $i = 1, 2, \dots, N$ ) encontrado no subintervalo  $\frac{(j-1)}{x} \leq r_i < \frac{j}{x}$ . A estatística

$$\chi_1^2 = \left(\frac{x}{N}\right) \sum_{j=1}^x \left(f_j - \frac{N}{x}\right)^2 \quad (2.43)$$

tem aproximadamente distribuição de qui-quadrado, com  $x - 1$  graus de liberdade para uma seqüência de verdadeiros números aleatórios.

Ela é então calculada para todos os  $M$  conjuntos consecutivos de  $N$  números pseudo-aleatórios. Seja então  $F_j$  a quantidade de valores  $M$  de  $\chi^2$  que estejam entre o  $(j - 1)$ -ésimo e a  $j$ -ésimo subintervalo de uma distribuição qui-quadrado com  $x - 1$  graus de liberdade ( $j = 1, 2, \dots, u$ ). Computa-se então a estatística:

$$\chi_F^2 = \left( \frac{u}{M} \right) \sum_{j=1}^u \left( F_j - \frac{M}{u} \right)^2. \quad (2.44)$$

A hipótese de que os números pseudo-aleatórios na seqüência de  $M$  conjuntos sejam verdadeiros números aleatórios será rejeitada se  $\chi_F^2$ , com  $u - 1$  graus de liberdade exceder o valor crítico estabelecido pelo nível de significância desejado. Um conjunto adequado de valores para este teste consiste em:  $x = u = 10$ ,  $M = 100$  e  $N = 1000$ . Os valores esperados  $\frac{N}{x}$  e  $\frac{M}{u}$  devem ser sempre maiores que 5.

### 2.5.2 Teste serial

Os testes de série são usados para a verificação do grau de aleatoriedade entre os números sucessivos de uma seqüência. Um teste de série é normalmente aplicado a pares de números, em que os números pseudo-aleatórios são considerados como coordenadas de um ponto em um quadrado unitário dividido em  $x^2$  células.

A idéia pode ser estendida a trincas de números pseudo-aleatórios, representando pontos em um cubo unitário. O teste da série também é baseado no teste de qui-quadrado e consiste nas seguintes etapas:

Começa por gerar uma seqüência de  $M$  conjuntos sucessivos de  $N$  números pseudo-aleatórios. Computa-se a estatística  $\chi_1^2$  para cada um dos  $M$  conjuntos, de acordo com a equação (2.43).

Então, para cada conjunto de  $N$  números pseudo aleatórios faz-se  $f_{jk}$  representar a quantidade de números pseudo-aleatórios  $r_i$  ( $i = 1, 2, \dots, N - 1$ ), que satisfaz  $\frac{(j-1)}{x} \leq r_i < \frac{j}{x}$  e  $\frac{(k-1)}{x} \leq r_{i+1} < \frac{k}{x}$ , em que  $j, k = 1, 2, \dots, x$ .

A seguir calcula-se a estatística:

$$\chi_2^2 = \frac{x^2}{N-1} \sum_{j=1}^x \sum_{k=1}^x \left( f_{jk} - \frac{N-1}{x^2} \right)^2 \quad (2.45)$$

para cada conjunto de  $N$  números pseudo-aleatórios.

Entretanto, Good (1957) mostrou que  $\chi_2^2 - \chi_1^2$  apresenta uma distribuição aproximadamente qui-quadrado, com  $x^2 - x$  graus de liberdade, para uma verdadeira seqüência aleatória.

A seguir calcula-se  $\chi_2^2 - \chi_1^2$  par a cada conjunto  $M$  de  $N$  números pseudo-aleatórios e chama-se de  $s$  o número dos  $M$  valores resultantes de  $\chi_2^2 - \chi_1^2$  que estejam entre  $(j - 1)$ -ésimo e  $j$ -ésimo subintervalo ( $j = 1, 2, \dots$ ,

$u$ ) de uma distribuição qui-quadrado com  $x^2 - x$  graus de liberdade. Finalmente calcula-se:

$$\chi_s^2 = \left( \frac{u}{M} \right) \sum_{j=1}^u \left( s_j - \frac{M}{u} \right)^2, \quad (2.46)$$

que tem  $u - 1$  graus de liberdade.

A aleatoriedade de uma seqüência de números pseudo-aleatórios é aceitável em certo nível de significância se os valores de  $\chi_F^2$  e  $\chi_s^2$  não forem inconsistentes com a hipótese de que foram extraídos, aleatoriamente, de uma distribuição qui-quadrado com graus de liberdade apropriados. Testes semelhantes podem ser desenvolvidos para trincas aleatórias.

### 2.5.3 Teste de Shapiro Wilk

O teste de normalidade proposto por Shapiro & Wilk (1965) basicamente consiste em obter uma estatística referente à razão entre o quadrado de uma combinação linear apropriada dos valores esperados das estatísticas de ordem da distribuição normal padrão e o quadrado dos desvios em relação à média. A estatística deste teste, representada por  $W$ , apresenta-se flexível à adaptação para hipóteses compostas, pois seus coeficientes lineares ( $a_i$ ) são facilmente computáveis.

A suposição exigida para aplicação deste teste é de que a amostra  $X_1, X_2, X_3, \dots, X_n$  seja aleatória, isto é, independente e identicamente distribuída e, ainda, associada a alguma função de distribuição desconhecida. Com esta suposição

sendo satisfeita, a obtenção da estatística W do teste de Shapiro-Wilk é dada conforme descrição que segue.

Inicialmente, associa-se um vetor de variáveis aleatórias  $\underline{y}$ , de modo que cada elemento será correspondente às amostras ordenadas do menor ao maior valor, ou seja,  $X^{(1)} \leq X^{(2)} \dots \leq X^{(n)}$ , em que  $X^{(i)}$  denota a i-ésima estatística de ordem. Associando essa amostra ordenada ao vetor  $\underline{y}$ , sua representação é dada por  $y_1 \leq y_2 \leq \dots \leq y_n$ . Esta associação é explicada facilmente pela igualdade  $y_i = \mu + \sigma X_i$ , em que cada  $X_i$  ( $i=1, 2, \dots, n$ ) tem distribuição normal padrão com média zero e variância 1.

Em seguida, calculam-se as quantidades:

$$D^2 = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (X_{(i)} - \bar{X})^2 \quad (2.47)$$

e

$$b = \sum_{i=1}^k a_{n-i+1} (y_{n-i+1} - y_i) \quad (2.48)$$

Definidas estas quantidades, é necessário obter os valores dos coeficientes  $\{a_{n-i+1}\}$ . Estes valores foram tabulados pelos próprios autores em função do tamanho amostral. Posteriormente define-se o valor de k, o qual será utilizado na equação 2.48 de modo que para n par usa-se a expressão (2.49), e para n ímpar, a expressão (2.50), respectivamente.

$$k = \frac{n}{2} \quad (2.49)$$



e

$$k = \frac{n-1}{2} \quad (2.50)$$

Definido o valor de  $k$ , a equação (2.48) pode ser desenvolvida da seguinte forma:

$$b = a_n (y_n - y_1) + \dots + a_{k+2} (y_{k+2} - y_k) \quad (2.51)$$

Sendo assim, a estatística  $W$  do teste é definida por:

$$W = \frac{b^2}{D^2} = \frac{\left( \sum_{i=1}^n a_i y_i \right)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.52)$$

Rejeita-se  $H_0$  caso o valor da estatística  $W$  apresentar valor inferior ao quantil apropriado a este teste. Dada a amostra, calculado o valor de  $W$ , o próximo passo é determinar a distribuição nula de  $W$  ou os valores críticos para determinados valores nominais de significância com a finalidade de realizar o teste da hipótese nula de normalidade dos dados. Shapiro & Wilk (1965) realizaram tabulações de valores críticos de  $W$  utilizando simulação Monte Carlo, alegando dificuldades de lidar analiticamente com a distribuição conjunta de  $W$  devido à presença de correlações entre valores ordenados. As aproximações de Royston (1982 e 1995) são usadas para o cálculo de p-valores,

relacionados às significâncias observadas de  $W$ . É importante ressaltar algumas propriedades relacionadas pelos próprios autores sobre a estatística  $W$ , sobre a qual afirmam que  $W$  mantém-se invariante quanto à escala; no caso de as amostras serem provenientes de uma distribuição normal, fica evidente que a estatística  $W$  depende somente do tamanho da amostra, apresentando-se independente da média e variância; o máximo da estatística  $W$  é 1 e o mínimo é dado por  $na_1^2/(n-1)$ .

Como desvantagem, os autores apontam a aplicação desta estatística em grandes amostras ( $n > 50$ ), sendo que, neste caso, torna-se complexo obter ou aproximar os valores referentes aos coeficientes  $a_{is}$  presentes no numerador. Royston (1982) mostra que a aproximação dos coeficientes dada por Shapiro & Wilk (1965) é bastante insatisfatória para  $n > 50$ . Assim, para  $n > 50$  Royston (1982) sugere uma aproximação por meio de suavizações polinomiais, ou seja, aproximações numéricas do coeficiente  $a_1$ , o que requer o uso de coeficientes específicos de polinômios para realizar a transformação.

Com o  $p$ -valor computado a decisão de rejeição ou não de  $H_0$  pode ser tomada, comparando o valor obtido com o valor nominal da significância  $\alpha$  adotado. O teste de normalidade, conhecido como teste de Shapiro Wilk, é considerado um excelente teste, com poder superior ao de seus concorrentes na maioria das situações e controle das taxas de erro tipo I, sempre em consonância com o valor nominal adotado, mesmo para pequenos valores de  $n$ .

## 2.6 Geradores fornecidos por alguns softwares

Às vezes pode não ser conveniente (ou impossível) escolher um gerador. Por exemplo, certos pacotes orientados à análise de dados, têm um gerador incorporado, e não dão opções de uso de geradores alternativos. Todavia, todo

usuário de um pacote deveria estar ciente do tipo de gerador nele implementado e conhecer as limitações do mesmo.

Quando o pesquisador trabalha com programas escritos por ele mesmo, ou usa pacotes de sub-rotinas como IMSL, NAG em linguagens computacionais como C, BASIC ou PASCAL, existe a possibilidade de escolher entre o gerador incorporado como parte da linguagem (em FORTRAN tal coisa não existe) ou a implementação de um gerador conhecido, ou ainda testar o gerador que será usado (Bustos & Orgambide, 1992).

Bustos & Orgambide (1992) fornecem uma síntese sobre vários pacotes e sistemas computacionais (software e hardware) em uso.

**SAS:** Usa os seguintes geradores

1. multiplicativo congruencial com  $a = 397204094$  e  $m = 2^{31} - 1$ .
2. um embaralhado do gerador congruencial multiplicativo com  $a = 16807$  e  $m = 2^{31} - 1$ .

**SPSS:** Oferece o gerador congruencial multiplicativo com  $a = 16807$  e  $m = 2^{31} - 1$ .

**IMSL:** Oferece duas sub-rotinas que implementam os geradores do SAS e também uma sub-rotina que implementa um embaralhado destas.

## 2.7 Estimação dos parâmetros de uma regressão linear

Segundo Draper & Smith (1998), pode-se classificar os modelos de regressão, em relação aos seus parâmetros, em lineares, linearizáveis e não-

lineares. Neste trabalho, interessam-nos os modelos lineares, com enfoque ao modelo de regressão linear simples.

Um modelo de regressão linear, conforme Draper & Smith (1998) e Hoffmann & Vieira (1998), pode ser expresso da forma que se segue:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \varepsilon_i, \quad (2.53)$$

em que:

$y_i$  :  $i$ -ésimo valor da variável resposta,  $i = 1, 2, \dots, N$  observações;

$x_{ki}$  :  $i$ -ésimo valor da  $k$ -ésima variável explicativa,  $k = 1, 2, \dots, K$  variáveis;

$\beta_k$  : parâmetros do modelo;

$\varepsilon_i$  : erros aleatórios.

Empregando a notação matricial, o modelo tem a seguinte forma:

$$y = X\beta + \varepsilon \quad (2.54)$$

em que:

$y$  : vetor de observações, de dimensões  $N \times 1$ , sendo  $N$  o número de observações;

$X$  : matriz das variáveis explicativas, de dimensões  $N \times (K + 1)$ , sendo  $K$  o número de variáveis explicativas;

$\beta$  : vetor de parâmetros, de dimensões  $(K + 1) \times 1$ , sendo  $(K + 1)$  o número de parâmetros;

$\varepsilon$  : vetor de erros aleatórios, de dimensões  $N \times 1$ .

Para a estimação do vetor de parâmetros  $\beta$ , comumente são empregados o método dos quadrados mínimos e método da máxima verossimilhança, que conduzem aos mesmos estimadores.

De acordo com as pressuposições que os erros podem assumir, existem variações no método de estimação dos quadrados mínimos para o modelo de regressão linear, relativa às diversas formas que a matriz de variâncias e covariâncias pode assumir. Estas variações são conhecidas como métodos dos quadrados mínimos ordinários, ponderado e generalizado.

Conforme Hoffmann & Vieira (1998), no ajuste de um modelo pelo método dos quadrados mínimos ordinários, pressupõe-se que a média dos erros é nula  $E(\varepsilon_i) = 0$ ; a variância do erro  $\varepsilon_i$ ,  $i = 1, 2, \dots, n$  é constante e igual a  $\sigma^2$ ; o erro de uma observação é não correlacionado com o erro de outra observação, isto é,  $E(\varepsilon_i \varepsilon_j) = 0$ , para  $i \neq j$ ; e os erros são variáveis aleatórias normalmente distribuídas.

Com base no método dos quadrados mínimos ordinários, estima-se um vetor  $\beta$ , considerando-se como condição que a soma de quadrados dos erros seja mínima. Como mostrado por Hoffmann & Vieira (1998), a forma quadrática  $Z$ , que representa a soma de quadrados dos erros, é dada por:

$$Z = \varepsilon' \varepsilon = (y - \beta X)' (y - \beta X). \quad (2.55)$$

Derivando parcialmente em relação a  $\beta$ , obtém-se o seguinte sistema de equações normais, conforme Graybill (1976):

$$X'X\hat{\beta} = X'y. \quad (2.56)$$

Como a matriz é de posto coluna completo, então  $X'X$  é uma matriz positiva definida e, assim,  $X'X$  é não singular. Portanto, existe a matriz inversa  $(X'X)^{-1}$  e o estimador para  $\beta$ , de acordo com Draper & Smith (1998) e Hoffmann & Vieira (1998), é:

$$\hat{\beta} = (X'X)^{-1}X'y \quad (2.57)$$

Esta solução única corresponde ao estimador linear não-tendencioso e de variância mínima para  $\beta$ .

## 2.8 Confiabilidade de um software estatístico

A confiabilidade de um software é um conjunto de ferramentas para avaliar a sua performance em relação ao tipo de análise que realiza. A confiabilidade é baseada na análise dos resultados computados, quando confrontados valores de referência.

McCullough & Wilson (1999) avaliaram a confiança dos procedimentos estatísticos do Excel 97 em três áreas: estimação (linear e não-linear), geração de números aleatórios e distribuições estatísticas e concluiu: “A performance do

Excel é inadequada. Pessoas desejando conduzir análise estatística de dados são advertidas a não usar o Excel”. Esta performance foi avaliada via Statistical Reference Data Sets (StRD), o qual recentemente foi liberado pela American “National Institute of Standards and Technology” (NIST).

McCullough (1999), usando a mesma metodologia avaliou a precisão de três pacotes estatísticos populares, SAS 6.12, SPSS e S-Plus 4.0, com atenção para detalhes importantes. Diferenças foram identificadas em todos os programas na geração de números aleatórios.

McCullough & Wilson (2002) encontraram as mesmas deficiências observadas no Excel 97 nas versões do Excel 2000 e do Excel 2002 (também chamado de Excel XP). Segundo os autores, os problemas na geração de amostras da distribuição normal padrão e da normal inversa têm piorado.

### 3 METODOLOGIA

A metodologia apresentada neste trabalho foi aplicada por meio de um estudo de simulação de dados, com a geração de distribuições comportadas nas suas propriedades. O objetivo foi avaliar e comparar três softwares estatísticos a partir de simulação e alguns métodos para avaliar a confiabilidade e precisão de relatórios em diferentes áreas de aplicação. Para conduzir este trabalho foram utilizados o SAS<sup>®</sup> 8.12, o Matlab 6.1 e a Ferramenta de Análise de Dados do Excel 2000.

Para avaliação da confiabilidade, os relatórios (output) dos programas foram confrontados e os parâmetros de interesse, comparados, tendo o SAS<sup>®</sup> 8.12 como referência. Para a realização deste estudo, foram considerados três procedimentos: geração de distribuições estatísticas, estimação dos coeficientes de uma regressão linear simples e cálculo e verificação de propriedades de algumas estatísticas, conforme descrito a seguir.

#### **3.1 Geração de dados segundo distribuições de probabilidades conhecidas**

Para comparação dos geradores de números aleatórios, realizou-se uma simulação de dados composta de 1.000 experimentos, para diferentes tamanhos de amostras. Foram geradas, nos três programas, amostras aleatórias da distribuição normal padronizada, da distribuição binomial e da distribuição Poisson. As amostras foram submetidas ao teste de Shapiro Wilk (equação 2.52) no caso da distribuição normal padrão, e nos outros dois casos foi utilizado o teste qui-quadrado (equação 2.44).

Os testes citados foram aplicados utilizando os recursos de análise do SAS<sup>®</sup> 8.12, foi considerado o fato de que a ferramenta de análise de dados do



Excel 2000 não possibilita a realização de tais testes, e procurou-se evitar que possíveis problemas com os algoritmos de implementação dos testes em um ou outro pacote comprometessem a eficiência dos geradores. Portanto, o fato de se ter escolhido o SAS® 8.12 para a realização dos testes foi o de proporcionar a todos os geradores a mesma ferramenta de análise.

### 3.2 Ajuste de uma Regressão linear simples – Simulação do método

Dada a seguinte relação linear

$$y_{hi} = \beta_0 + \beta_1 x_{hi} + \varepsilon_{hi}, \text{ sendo } \begin{cases} i = 1, 2, \dots, n \\ h = 1, 2, \dots, H \end{cases}$$

em que:

$y_{hi}$  : i-ésima observação da variável resposta do h-ésimo modelo;

$x_{hi}$  : i-ésimo valor da variável regressora do h-ésimo modelo;

$\beta_0$  e  $\beta_1$  : coeficientes do modelo;

$\varepsilon_{hi}$  : erro aleatório, associado à i-ésima observação do h-ésimo modelo, sendo supostos independentes e normalmente distribuídos, com média zero e variância comum, isto é,

$$\varepsilon_{hi} \sim NID(0, \sigma^2).$$

Realizou-se uma simulação de dados composta de 1000 experimentos (H), cada qual com 30 observações (n).

Estabeleceu-se inicialmente a relação  $y_{hi} = 5 + 2x_{hi}$ , sendo  $i = 1, 2, \dots, 30$  o número de observações e  $h = 1, 2, \dots, 1000$  o número de experimentos. Para cada experimento, foram obtidos valores da variável dependente  $y_{hi}$ , sendo que os valores da variável independente  $x_{hi}$ , foram obtidos em um intervalo fechado de 1 a 10, aleatoriamente gerados pela função RANUNI do SAS® 8.12.

Para a geração dos resíduos de cada modelo, foi necessário estimar a variância dos mesmos. Fixando-se o coeficiente de determinação  $R^2$  em 90%, e conhecida a relação  $R^2 = \frac{\sigma_{\text{modelo}}^2}{\sigma_{\text{modelo}}^2 + \sigma_{\text{erro}}^2}$ , em que  $\sigma_{\text{modelo}}^2$  corresponde à variância dos valores das variáveis dependentes, estimou-se a variância dos resíduos  $\sigma_{\text{erro}}^2$ .

Estimada a variância dos resíduos  $\sigma_{\text{erro}}^2$  geraram-se, no SAS® 8.12, os resíduos aleatórios de cada modelo, estes supostamente independentes e normalmente distribuídos, com média zero e variância comum, isto é,  $\varepsilon_i \sim NID(0, \sigma_{\text{erro}}^2)$ .

Para a estimação dos valores das variáveis dependentes  $y_{hi}$ , utilizou-se a relação:

$$\hat{y}_{hi} = 5 + 2x_{hi} + \varepsilon_{hi}.$$

Para os H conjuntos de observações  $(x_{hi}, \hat{y}_{hi})$ , ajustaram-se modelos de regressão e foram estimados  $\hat{\beta}_0$  e  $\hat{\beta}_1$  nos três programas. Estatísticas descritivas dos valores estimados  $\hat{\beta}_0$  e  $\hat{\beta}_1$  foram obtidas.

### 3.3 Propriedades de estatísticas descritivas

Dada uma amostra  $X_i = \{x_1, x_2, \dots, x_n\}$ , em relação à média e a variância, as seguintes propriedades podem ser verificadas:

- 1ª) Somando-se a todas observações uma constante k, a nova média fica acrescida de k e a variância não se altera.
- 2ª) Multiplicando-se todas as observações por uma constante k, a média fica multiplicada por k e a variância, por  $k^2$ .

Inicialmente, gerou-se uma amostra aleatória  $X_i = \{x_1, x_2, \dots, x_n\}$ , composta de 30 observações, a partir da distribuição de Poisson com  $\lambda = 5$ , por meio da função RANPOI do SAS® 8.12.

Para a primeira propriedade, considerou-se o modelo:

$$y_i = x_i + k, \text{ sendo } \begin{cases} i = 1, 2, \dots, 30 \\ k = 0, 10, 10^2, \dots, 10^{10} \end{cases}$$

em que:

$y_i$ : i-ésima observação da variável resposta;

$x_i$ : i-ésima observação gerada da distribuição de Poisson.

Para a segunda propriedade o modelo foi o que segue:

$$y_i = x_i \cdot k, \text{ sendo } \begin{cases} i = 1, 2, \dots, 30 \\ k = 1, 10, 10^2, \dots, 10^{10} \end{cases}$$

Objetivou-se neste estudo, verificar se algum dos programas apresentaria problemas com operações aritméticas simples envolvendo números grandes.

## 4 RESULTADOS E DISCUSSÃO

### 4.1 Geração de distribuição estatística


Para os casos de geração de números aleatórios, os softwares foram comparados em relação à proporção de amostras geradas que não seguiam a distribuição teórica. Considerou-se o nível nominal de 5% de probabilidade.

Para a distribuição normal, considerou-se a média  $\mu = 0$  e variância  $\sigma^2 = 1$  para diferentes tamanhos de amostras. Os resultados obtidos nos três programas, referentes à proporção ( $\hat{\rho}$ ) de amostras não-normais geradas, são os apresentados na Tabela 1.

**TABELA 1** – Proporção ( $\hat{\rho}$ ) de amostras não-normais geradas nos três programas.

Tamanho da Amostra	Valores de $\hat{\rho}$		
	Excel	Matlab	SAS
2	0,054	0,000	0,052
5	0,054	0,041	0,052
10	0,054	0,051	0,052
15	0,054	0,045	0,052
20	0,054	0,047	0,052
30	0,054	0,039	0,052

Conforme ilustra a Tabela 1, os relatórios dos três programas são semelhantes na geração da distribuição normal padrão, pequenas diferenças



foram observadas, mas, a proporção de amostras geradas que não seguem uma normal está de acordo com o nível de significância estabelecido para a aplicação do teste.

Na geração das amostras de uma Poisson considerou-se diferentes tamanhos de amostras para um dado valor de  $\lambda$ . A proporção ( $\hat{\rho}$ ) de amostras que não seguem uma Poisson de acordo os valores do parâmetro escolhido está apresentada na Tabela 2.

**TABELA 2** – Proporção ( $\hat{\rho}$ ) de amostras que não seguem um distribuição de Poisson geradas nos três programas.

Tamanho da Amostra	Valores de $\hat{\rho}$		
	Excel	Matlab	SAS
30	0,015	0,014	0,026
50	0,027	0,021	0,018
75	0,035	0,032	0,016
100	0,043	0,028	0,026

Pelo que se observou, todos os programas retornaram valores respeitando o nível de significância adotado pelo teste que foi de 5%, as pequenas diferenças entre os resultados devem ser consideradas de natureza meramente aleatórias.

Nos casos de geração de uma Binomial, amostras de diferentes tamanhos foram obtidas, estabeleceu-se  $p = 0,05$ . Foi considerada, como no caso das outras distribuições, a proporção ( $\hat{\rho}$ ) das amostras geradas que não seguem uma Binomial, como pode ser observado pela inspeção da Tabela 3.

**TABELA 3 – Proporção ( $\hat{\rho}$ ) de amostras que não seguem um distribuição Binomial geradas nos três programas.**

Tamanho da Amostra	Valores de $\hat{\rho}$		
	Excel	Matlab	SAS
30	0,020	0,016	0,026
50	0,025	0,027	0,033
75	0,025	0,020	0,025
100	0,017	0,022	0,020

Comportamento semelhante ao da geração de uma Poisson se observa na geração de binomiais, os programas se comportam de maneira semelhante para os diversos tamanhos amostrais.

#### 4.2 Estimação dos coeficientes de uma regressão linear simples

Em relação à estimação dos coeficientes de uma regressão linear, os valores médios da distribuição dos estimadores foram confrontados com os valores inicialmente fixados em termos de precisão. Em relação aos pacotes que apresentaram os mesmos valores médios, foram considerados os de variância mínima.

Os resultados referentes aos valores médios dos estimadores apresentados para os 1.000 experimentos simulados nos três programas encontram-se na Tabela 4.

**TABELA 4 – Média da distribuição dos estimadores  $\hat{\beta}_0$  e  $\hat{\beta}_1$  gerados pelos três programas.**

PROGRAMAS	ESTIMADORES	
	$\hat{\beta}_0$	$\hat{\beta}_1$
SAS 8.12	5,02125124	1,9966307
MATLAB	5,0212	1,9966
EXCEL	5,021251235	1,996630705

Observou-se, nos programas, que a distribuição dos estimadores retornaram a mesma média tanto para  $\hat{\beta}_0$  quanto para  $\hat{\beta}_1$ , cujos valores estão muito próximos dos valores inicialmente fixados, que eram  $\hat{\beta}_0 = 5$  e  $\hat{\beta}_1 = 2$ . Nota-se ainda que as pequenas diferenças podem ser corrigidas por meio de arredondamentos, não comprometendo, portanto, o resultado das análises.

Em relação à variância da distribuição dos estimadores, os programas retornaram também os mesmos valores (Tabela 5).

**TABELA 5 – Variância da distribuição dos estimadores  $\hat{\beta}_0$  e  $\hat{\beta}_1$  gerados pelos três programas.**

PROGRAMAS	ESTIMADORES	
	$\hat{\beta}_0$	$\hat{\beta}_1$
SAS 8.12	0,66607385	0,01562104
MATLAB	0,6661	0,0156
EXCEL	0,666073848	0,015621038



Portanto, não há evidências de que os três programas diferem entre si em relação à estimação dos coeficientes de uma regressão linear simples.

### 4.3 Verificação de propriedades de estatísticas descritivas

Os possíveis problemas com operações aritméticas foram observados em termos das propriedades das estatísticas geradas. A violação das propriedades consideradas foram constatadas por meio do viés em relação aos reais valores (erro absoluto).

Os resultados referentes à média das amostras (1ª propriedade do Item 3.3) obtidos nos três programas encontram-se sumariados na Tabela 6.

**TABELA 6 – Médias das amostras  $y_i = x_i + k$  para  $k = 0, 10, 10^2, \dots, 10^{10}$ , calculadas nos três programas.**

<b>MÉDIAS</b>		
<b>EXCEL</b>	<b>SAS</b>	<b>MATLAB</b>
5,533333	5,53333333	5,53333
15,53333	15,5333333	15,5333
105,5333	105,533333	105,5333
1005,533	1005,53333	1,0055e+003
10005,53	10005,5333	1,0006e+004
100005,5	100005,533	1,0001e+005
1000006	1000005,53	1,0000e+006
10000006	10000005,5	1,0000e+007
100000006	100000006	1,0000e+008
1000000006	1000000006	1,0000e+009
10000000006	1E10	1,0000e+010

Em relação ao cálculo da média das amostras os três programas apresentaram diferenças em relação ao verdadeiro valor a partir de um determinado valor de  $k$ .

No Excel, a partir de  $k = 10^6$  a média passa a ser de  $\bar{y} = 1,000006 \times 10^6$  ao invés de  $\bar{y} = 1,000005533333 \times 10^6$ . O erro absoluto observado é de aproximadamente 0,466667, erro este considerado como erro cometido devido ao arredondamento.

O SAS comete o mesmo erro a partir de  $k = 10^8$ . Entretanto, quando  $k = 10^{10}$ , o SAS comete um erro absoluto de aproximadamente 5,533333 ao expressar o resultado em potência de dez.

No Matlab, para  $k = 10^4$  o erro cometido é de 0,466667 e para  $k = 10^5$  o erro absoluto cometido é de aproximadamente 4,4667. A partir de  $k = 10^6$ , o erro absoluto cometido é de 5,533333.

Os problemas encontrados no SAS e no Matlab podem ser contornados se o usuário alterar a opção de formato dos números do output.

Para a variância das amostras, os resultados obtidos nos três programas encontram-se sumariados na Tabela 7.

**TABELA 7** – Variâncias das amostras  $y_i = x_i + k$  para  $k = 0, 10, 10^2, \dots, 10^{10}$ , calculadas nos três programas.

VARIÂNCIAS		
EXCEL 2000	SAS 8.12	MATLAB
6,395402	6,3954023	6,3954
6,395402	6,3954023	6,3954
6,395402	6,3954023	6,3954
6,395402	6,3954023	6,3954
6,395402	6,3954023	6,3954
6,395403	6,3954023	6,3954
6,395339	6,3954023	6,3954
6,396552	6,3954023	6,3954
4,4137931	6,3954023	6,3954
282,482759	6,3954023	6,3954
18078,89655	6,3954023	6,3954

Para esta situação notam-se sérios problemas com o Excel, pois a partir de  $k = 10^7$  violam-se totalmente a 1ª propriedade do Item 3.3 no que tange à variância das amostras, retornando valores sem nenhum padrão observável. É impossível, nesta situação detectar o tipo de erro cometido.

Em relação ao SAS e ao Matlab, nota-se que os valores da variância mantiveram-se constantes para todo os valores de  $k$ , as pequenas diferenças observadas devem-se a arredondamentos provocados por aritméticas de precisão.

A Tabela 8 ilustra a 2ª propriedade em relação à media da amostra quando se multiplicam os elementos por uma constante  $k$ .

**TABELA 8** – Médias das amostras  $y_i = x_i \cdot k$  para  $k = 1, 10, 10^2, \dots, 10^{10}$ , calculadas nos três programas.

<b>MÉDIAS</b>		
<b>EXCEL 2000</b>	<b>SAS 8.12</b>	<b>MATLAB</b>
5,533333	5,53333333	5,5333
55,33333	55,3333333	55,3333
553,3333	553,333333	553,3333
5533,333	5533,33333	5,5333e+003
55333,33	55333,3333	5,5333e+004
553333,3	553333,333	5,5333e+005
5533333	5533333,33	5,5333e+006
55333333	55333333,3	5,5333e+007
553333333	553333333	5,5333e+008
5533333333	5533333333	5,5333e+009
55333333333	5,53333E10	5,5333e+010

Pode-se observar que todos os programas apresentaram resultados semelhantes nesta categoria, indicando que em relação à multiplicação dos elementos da amostra por uma constante, a média ficou realmente multiplicada pela mesma constante.

Situação semelhante à anterior aconteceu com a variância, que ficou multiplicada por  $k^2$  quando os elementos da amostra foram multiplicados por  $k$  (Tabela 9).

**TABELA 9 – Variâncias das amostras  $y_i = x_i.k$  para  $k = 1, 10, 10^2, \dots, 10^{10}$ , calculada nos três programas.**

<b>VARIÂNCIAS</b>		
<b>EXCEL 2000</b>	<b>SAS 8.12</b>	<b>MATLAB</b>
6,395402	6,3954023	6,3954
639,5402	639,54023	639,5402
63954,02	63954,023	6,3954e+004
6395402	6395402,3	6,3954e+006
6,4E+08	639540230	6,3954e+008
6,4E+10	6,3954E10	6,3954e+010
6,4E+12	6,3954E12	6,3954e+012
6,4E+14	6,3954E14	6,3954e+014
6,395E+16	6,3954E16	6,3954e+016
6,3954E+18	6,3954E18	6,3954e+018
6,3954E+20	6,3954E20	6,3954e+020

Como se observa, os três programas retornaram os mesmos valores para a variância para os dados valores de  $k$ . Portanto, não se evidencia falha em nenhum deles.

## 5 CONCLUSÕES

Em relação à geração de amostras aleatórias a partir de distribuições conhecidas, não se observou nenhuma evidência que denuncie a presença de erros que inviabilizem a utilização de um outro dos softwares testados.

No que tange à estimação dos coeficientes de uma regressão linear simples, os três programas retornaram os mesmos valores, tanto para  $\hat{\beta}_0$  quanto para  $\hat{\beta}_1$ . Portanto, não há evidências de que os programas apresentem diferenças entre si nesta categoria de análise.

Ao se tratar de duas propriedades da média e da variância, em relação a operações elementares, tais como, multiplicação e soma de constantes, todos os programas testados apresentaram diferenças provocadas pela precisão na representação dos resultados do output. Problemas maiores foram detectados no Excel, que violou uma das propriedades da variância no que se refere à soma de constantes aos pontos amostrais, retornando valores totalmente discrepantes para a variância a partir de um dado valor da constante.

## REFERÊNCIAS BIBLIOGRÁFICAS

- BUSTOS, O. H.; ORGAMBIDE, A. C. F. Simulação Estocástica. Teoria e algoritmos. In: SIMPÓSIO NACIONAL DE PROBABILIDADE E ESTATÍSTICA, 10., 1992, Rio de Janeiro. *Anais...* Rio de Janeiro: UFRJ, 1992. 152 p
- DRAPER, N. R.; SMITH, H. *Applied regression analysis*. 2. ed. New York: Jhon Wiley, 1998. 709 p.
- GRAYBILL, F. A. *Theory and application of the linear model*. Belmont: Duxbury Press, 1976. 704 p.
- GOOD, I. J. On the serial test for random sequences. *Annals of Mathematical Statistics*, Hayward, v. 28, n. 1, p. 262-264, 1957.
- HOFFMANN, R.; VIEIRA, S. *Análise de regressão: uma introdução à econometria*. 3. ed. São Paulo: HUCITEC, 1998. 379 p.
- MACLAREN, D. M.; MARSAGLIA, G. Uniform random numbers generators. *Journal of the ACM*, London, v. 12, n. 1, p. 83-89, Jan. 1965.
- MCCULLOGH, B. D. Assessing the reliability of statistical software: Part I. *The American Statistician*, Washington, v. 52, n. 4, p. 358-366, Nov. 1998.
- MCCULLOGH, B. D. Assessing the reliability of statistical software: Part II. *The American Statistician*, Washington, v. 53, n. 2, p. 149-159, May 1999.
- MCCULLOGH, B. D.; WILSON, B. On the accuracy of statistical procedures in Microsoft Excel 97. *Computational Statistics & Data Analysis*, New York, v. 31, p. 27-37, 1999.
- MCCULLOGH, B. D.; WILSON, B. On the accuracy of statistical procedures in Microsoft Excel 2000 and Excel XP. *Computational Statistics & Data Analysis*, New York, v. 40, n. 4, p. 713-721, Oct. 2002.

MORGAN, B. J. T. **Elementos of simulation**. 6. ed. London: Chapman & Hall, 1995. 351 p.

NAYLOR, T. H.; BALINTFY, J. L.; BURDICH, D. S.; CHU, K. **Técnicas de simulação em computadores**. São Paulo: Vozes, 1971. 401 p.

ROYSTON, P. Algorithm AS 181: The W Teste for Normality. **Applied Statistics – Journal of the Royal Statistical Society Series C**, London, v. 31, p. 176-180, 1982.

ROYSTON, P. A Remark to Algorithm AS 181: The W Teste for Normality. **Applied Statistics – Journal of the Royal Statistical Society Series C**, London, v. 44, n. 4, p. 547-551, 1995.

RUGGIERO, G. A. M.; LOPES, R. L. V. **Cálculo Numérico, Aspectos Teóricos e Computacionais**. 2. ed. New York: Makron Books, 1996.

SAS® INSTITUTE. **SAS Procedures guide for computers**. 6. ed. Cary, NC, 1999. v. 3, 373 p.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance for normality (complete sample). **Biometrika**, London, v. 52, pt. 3/4, p. 591-611, 1965.

WAMPLER, R. H. Test procedures and test problems for least squares algorithms. **Journal of Econometrics**, Lausanne, v. 12, n. 1, p. 3-22, 1980.

WILKINSON, L. **Statistic quis**, Evanston, IL:SYSTAT, Inc. 1985. Disponível em: <<http://www.tspinl.com/Benchmarks>. Acesso em: 2003.



## ANEXOS

<b>ANEXOS</b>	<b>Pág.</b>
<b>Anexo 1 – Estrutura do Programa SAS para o teste de normalidade das amostras geradas no SAS .....</b>	<b>56</b>
<b>Anexo 2 – Estrutura do Programa SAS para o teste de normalidade das amostras geradas no Excel e no Matlab .....</b>	<b>57</b>
<b>Anexo 3 – Estrutura do Programa SAS para a geração de variáveis regressoras e ajuste de modelos de regressão .....</b>	<b>58</b>
<b>Anexo 4 – Estrutura do Programa SAS para teste de qui-quadrado – amostras geradas em outros programas .....</b>	<b>60</b>
<b>Anexo 5 – Estrutura do Programa SAS para teste de qui-quadrado – amostras geradas no SAS .....</b>	<b>63</b>

**Anexo 1 – Estrutura do Programa SAS para o teste de normalidade das amostras geradas no SAS**

**/\*Dissertação – Teste de normalidade – Amostras SAS\*/  
/\*José Ermelino A.Damasceno e Ruben Delly Veiga\*/**

```
data teste;  
do j=1 to 1000;  
do i=1 to 30;  
x=10+sqrt(1)*normal(100);  
output;  
end;  
end;  
run;  
proc print data=teste;  
proc univariate normal data=teste;  
output out=aula probn=pr;  
by j;  
var x;  
run; quit;  
proc print data=aula;  
run; quit;  
data aula; set aula;  
if pr<0.05 then ctb=1;  
else ctb=0;  
run; quit;  
proc means;  
var ctb;  
run; quit;
```

**Anexo 2 – Estrutura do Programa SAS para o teste de normalidade das amostras geradas no Excel e no Matlab**

**/\*Dissertação – normalidade – Amostras Excel e Matlab\*/  
/\*José Ermelino A.Damasceno e Ruben Delly Veiga\*/**

**data teste;**

do i=1 to 50;  
do j=1 to 10;  
input amostra @@; output;  
end;  
end;

**cards;**

;

**proc print data=teste;**  
**proc univariate normal data=teste nobs; output out=aula probn=pr;**  
by i;  
var amostra;  
**run; quit;**  
**proc print data=aula;**  
**run; quit;**  
**data aula; set aula;**  
if pr<0.05 then ctb=1;  
else ctb=0;  
**run; quit;**  
**proc means;**  
var ctb;  
**run; quit;**

### Anexo 3 – Estrutura do Programa SAS para a geração de variáveis regressoras e ajuste de modelos de regressão

```
/*Dissertação – Variáveis regressoras*/  
/*José Ermelino A.Damasceno e Ruben Delly Veiga*/
```

```
options ps = 60 nodate pageno = 1;  
data teste;  
  *file 'c:\temp\teste.txt';  
  do exp = 1 to 1000;  
    do i = 1 to 30;  
      x = ranuni(99)*10+1;  
      y = 5 + 2*x;  
      output;  
    end;  
  end;  
proc means data = teste noprint;  
  var y;  
  output out = varia var = var;  
  by exp;  
data junta;  
  merge teste varia;  
  by exp;  
data simula;  
  set junta;  
  error = sqrt(((1 - 0.9)/0.9)*var);  
  desvio = rannor(99)*error;  
  yest = y + desvio;  
proc reg data = simula noprint outest = est /*tableout*/;  
  model yest = x ;  
  by exp;
```

```
*proc print data = est;  
proc univariate data = est plot normal;  
  var intercept x;  
run;quit;
```

**Anexo 4 – Estrutura do Programa SAS para teste de qui-quadrado – amostras geradas em outros programas**

```
/*Dissertação – Teste qui-quadrado*/  
/*José Ermelino A.Damasceno e Ruben Delly Veiga*/
```

```
options ps = 65 ls = 80 nodate nocenter pageno = 1;  
data bin30 (keep = exp n p xbin);  
  infile 'c:\orienta\ermelino\bin30.prm';  
  do exp = 1 to 1000;  
    do i = 1 to 30;  
      input xbin005;  
      xbin = xbin005; p = 0.05; n = 30; output;  
    end;  
  end;  
data bin50 (keep = exp n p xbin);  
  infile 'c:\orienta\ermelino\bin50.prm';  
  do exp = 1 to 600;  
    do i = 1 to 50;  
      input xbin005;  
      xbin = xbin005; p = 0.05; n = 50; output;  
    end;  
  end;  
data bin75 (keep = exp n p xbin);  
  infile 'c:\orienta\ermelino\bin75.prm';  
  do exp = 1 to 400;  
    do i = 1 to 75;  
      input xbin005;  
      xbin = xbin005; p = 0.05; n = 75; output;  
    end;  
  end;
```

```

data bin100 (keep = exp n p xbin);
  infile 'c:\orienta\ermelino\bin100.prm';
  do exp = 1 to 300;
    do i = 1 to 100;
      input xbin005;
      xbin = xbin005; p = 0.05; n = 100; output;
      end;
    end;
data bin;
  set bin30 bin50 bin75 bin100;
proc sort data = bin;
  by exp n p;
proc freq;
  table xbin /nocum nopercnt noprint out=c ;
  by exp n p;
data teste2 (keep = exp n p xbin count cbinesp);
  set c;
  if xbin eq 0 then do;
    cbinesp = int(probbnml(p,n,xbin)*n)+1;
    end;
  else do;
    cbinesp = int((probbnml(p,n,xbin)-probbnml(p,n,xbin-1))*n)+1;
    end;
  output;
data testequi;
  set teste2;
  by exp n p;
  if first.p then do;
    qui = 0;
    classes = 0;
    end;
  qui + (count - cbinesp)**2/cbinesp;

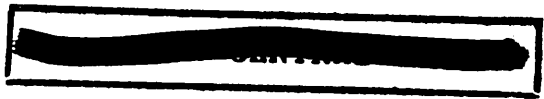
```

```

classes + 1;
if last,p then do;
  nvsig = 1-probchi(qui,classes-1);
output;
end;
proc format;
  value sig 0.00 -< 0.01 = "0.00 a 0.01"
            0.01 -< 0.025 = "0.01 a 0.025"
            0.025 -< 0.05 = "0.025 a 0.05"
            0.05 -< 0.10 = "0.05 a 0.10"
            0.10 -< 1.00 = "0.10 ou mais";
proc sort data = testequi;
  by n p;
proc freq data = testequi;
  title1 "Teste Qui Quadrado - Binomial - 07/07/2003 - Ermelino";
  title2 "Dados Gerados no Excel";
  table nvsig /*format = fmsig.*/;
  by n p;
format nvsig sig;
*proc print data = testequi (obs = 500);
run;run;

```





**Anexo 5 – Estrutura do Programa SAS para teste de qui-quadrado – amostras geradas no SAS**

*/\*Dissertação – Teste qui-quadrado\*/*

*/\*José Ermelino A.Damasceno e Ruben Delly Veiga\*/*

options ps = 65 ls = 80 nodate nocenter pageno = 1;

data teste;

do exp = 1 to 1000;

do n = 30,50,75,100;

do p = 0.05;

do i = 1 to n;

med = n\*p;

xpoi = ranpoi(1111,med);

output;

end; /\* do i \*/

end; /\* do p \*/

end; /\* do n \*/

end; /\* do exp\*/

proc freq;

table xpoi /nocum nopercnt noprint out=c ;

by exp n p;

data teste2 (keep = exp n p med xpoi count cpoiesp);

set c;

med = n\*p;

if xpoi eq 0 then do;

cpoiesp = int(poisson(med,xpoi)\*n)+1;

end;

else do;

cpoiesp = int((poisson(med,xpoi)-poisson(med,xpoi-1))\*n)+1;

end;

```

output;
data testequi;
set teste2;
by exp n p;
if first.p then do;
    qui = 0;
        classes = 0;
        end;
qui + (count - cpoiesp)**2/cpoiesp;
classes + 1;
if last.p then do;
    nivsig = 1-probchi(qui,classes-1);
        output;
        end;
proc format;
value sig  0.00 -< 0.01 = "0.00 a 0.01"
           0.01 -< 0.025 = "0.01 a 0.025"
           0.025 -< 0.05 = "0.025 a 0.05"
           0.05 -< 0.10 = "0.05 a 0.10"
           0.10 -< 1.00 = "0.10 ou mais";
proc sort data = testequi;
by n p;
proc freq data = testequi;
title1 'Teste Qui Quadrado - POISSON - 07/07/2003 - Ermelino';
title2 'Dados Gerados no Sistema SAS';
table nivsig /*format = fntsig.*/*;
by n p;
format nivsig sig.;
*proc print data = testequi (obs = 500);
run;

```