SoftBoots: Um software para calcular estimativas *Bootstrap*

Renata Garcia Oliveira , **Antonyus Pyetro do A. Ferreira**
Centro de Informática, UFPE,
Caixa Postal 7851, Cidade Universitária, 50.732-970 Recife, PE, Brasil
E-mail: rgo@cin.ufpe.br, apaf@cin.ufpe.br,

Bruno Correia da Silva, **Marcília Andrade Campos**
Centro de Informática, UFPE,
Caixa Postal 7851, Cidade Universitária, 50.732-970 Recife, PE, Brasil
E-mail: bcs2@cin.ufpe.br, mac@cin.ufpe.br.

Resumo Este artigo trata das diretivas da construção de um software estimador de intervalos de confiança *bootstrap*, para os casos de distribuição de probabilidade conhecida e desconhecida. Adicionalmente, são apresentados exemplos de uso do software.

Palavras-chave *bootstrap*, intervalo de confiança, geração de variáveis aleatórias

1 Introdução

A técnica do *bootstrap* foi introduzida por *Bradley Efron* [3] em 1979. Apesar de ser relativamente recente, a técnica tem tido aceitação, tanto por ser computacionalmente aplicável quanto por apresentar-se como uma alternativa simples aos problemas complexos da teoria estatística tradicional.
A partir disso, foi produzido um software que fornece o intervalo de confiança de *bootstrap*, em que é informado uma pequena amostra e a distribuição de probabilidade. Nesse trabalho podem ser computadas as distribuições de Normal, Exponencial, Poisson, Binomial e Desconhecida.
O programa foi implementado em java, tornando-o portável a qualquer sistema operacional. Sua arquitetura permite que ele possa ser integrado a qualquer sistema facilmente, ou pode ter mais funcionalidades agregadas a ele. O *SoftBoots* se encontra em *http://www.cin.ufpe.br/rgo/SoftBoots*

2 Estimativa *Bootstrap*

Será mostrado como encontrar estimativas *bootstrap* para a média ($\mu = \mathrm{E}(X)$), variância ($\sigma^2 = \mathrm{V}(X)$) e desvio padrão (σ) populacionais; a seguir será visto como calcular intervalos de confiança também via *bootstrap*.

Cenário: X é uma variável aleatória populacional que segue uma lei de probabilidade $f(x, \underset{\sim}{\theta})$ onde $\underset{\sim}{\theta}$ é um vetor de parâmetros.

Problema: Geralmente o interesse é encontrar estimativas para μ,σ^2 e σ.

Solução: Considerando algum tipo de dificuldade teórica e disponibilidade de uso de computador usar *bootstrap*.

A metodologia, nesse caso, divide-se em duas considerando a lei de probabilidade populacional é ou nao conhecida.

2.1 Lei de Probabilidade Populacional Conhecida

Sem perda de generalidade, seja $\underset{\sim}{\theta} = \theta$, isto é, tem-se apenas um parâmetro populacional. Convém salientar que os parâmetros populacionais estão extremamente relacionados com a média e variância populacionais, portanto, obter estimativas para os parâmetros pode ser o mesmo que obter estimativas para a média ou variância. A Tabela 1 enfatiza esse fato.

Tabela 1: Parâmetros μ e σ^2 para algumas variáveis aleatórias

Variável Aleatória	Parâmetro(s)	E(X)	V(X)
Normal	μ,σ^2	μ	σ^2
Exponencial	λ	$1/\lambda$	$1/\lambda$
Poisson	λ	λ	λ
Binomial principal	n,p	np	np(1-p)

O Algoritmo 1 a seguir descreve como calcular as estimativas *bootstrap*.

Algoritmo 1. Cálculo das estimativas *bootstrap*

(i) Retirar uma amostra aleatória $(x_1, x_2, \ldots, x_n)$ da população.

(ii) Obter uma estimativa amostral, θ_a, para o parâmetro θ.

(iii) Usar $f(x, \theta_a)$.

(iv) Gerar N amostras cada uma de tamanho n; estas são as amostras *bootstrap*.

(v) Calcular estimativas *bootstrap* θ_{ib} para cada uma delas, como mostra a Tabela 2.

(vi) Calcular as estimativas *bootstrap* para a média, variância e desvio padrão populacionais, respectivamente, μ, σ^2 e σ.

$$\overline{\theta_b} = \frac{1}{N}\sum_{i=1}^{N}\theta_{ib} \ ,$$

$$s_{_b}^2 = \frac{1}{N}\sum_{i=1}^{N}(\theta_{ib} - \overline{\theta_b})^2 \ ,$$

$$s_{_b} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_{ib} - \overline{\theta_b})^2} \ .$$

Tabela 2: Amostras e estimativas *bootstrap*

Amostras($f(x,\theta_n)$)	Estimativas *bootstrap* ($\theta_{ib}, i = 1,\ldots,N$)
$(x_{_{11}}, x_{_{12}}, \ldots, x_{_{1n}})$	θ_{1b}
$(x_{_{21}}, x_{_{22}}, \ldots, x_{_{2n}})$	θ_{2b}
$\ldots$	$\ldots$
$(x_{_{N1}}, x_{_{N2}}, \ldots, x_{_{Nn}})$	θ_{Nb}

2.2 Lei de Probabilidade Populacional Desconhecida

Neste caso, a amostra $(x_{_1}, x_{_2}, \ldots, x_{_n})$ será considerada como sendo a população e as N amostras *bootstrap*:

$$(x_{_{11}}, x_{_{12}}, \ldots, x_{_{1n}})$$
$$\ldots$$
$$(x_{_{N1}}, x_{_{N2}}, \ldots, x_{_{Nn}})$$

conterão observações $(x_{ij}), i = 1,\ldots,N$ e $j = 1,\ldots,n$ obtidas através da amostragem, com reposição, da população. O algoritmo 1 apresentado anteriormente agora volta a ser seguido a partir de (v).

2.3 Cálculo do Intervalo de Confiança

O Algoritmo 2 apresentado a seguir descreve os procedimentos necessários para encontrar o intervalo de confiança (IC) bilateral para o parâmetro populacional de interesse no problema. O algoritmo se inicia a partir dos dados da Tabela 2.

Algoritmo 2. Cálculo do intervalo de confiança (IC)

(i) Calcular as diferenças

$$P_i = \theta_{ib} - \overline{\theta_b},\ i = 1,\ldots,N$$

(ii) Ordenar os P_i, obtendo $P_{(i)}$ tal que

$$P_{_{(1)}} \leqslant P_{_{(2)}} \leqslant \ldots \leqslant P_{_{(N)}} \ .$$

(iii) Obter os $P_{_{(\alpha/2)}}$ e $P_{_{(1-\alpha/2)}}$ através da fórmula $[(N-1)\alpha + 1]$, a qual dará a ordem dos percentis desejados.

(iv) Calcular os limites inferior (I) e superior (S) do intervalo de confiança

$$I = \theta_a - P_{_{(1-\alpha/2)}} \ ,$$
$$S = \theta_a - P_{_{(\alpha/2)}} \ .$$

(v) O IC para θ com $(1-\alpha)\%$ de confiança é

$$(\theta_a - P_{_{(1-\alpha/2)}} \ , \ \theta_a - P_{_{(\alpha/2)}}) \ .$$

2.3.1 Exemplo 1: Cálculo dos Intervalos de confiança *Bootstrap* quando a lei de probabilidade é conhecida

A variável aleatória é T, tempo de falha de um modo eletrônico e sua distribuição de probabilidade é exponencial com parâmetro desconhecido. Como visto na Tabela 1, $E(T) = 1/\lambda$ e $V(T) = 1/\lambda^2$. [6] Seguindo-se os passos do Algoritmo 1 tem-se o que segue:

(i) Amostra aleatória

$(11,96; 5,03; 67,40; 16,07; 31,50; 7,73; 11,10; 22,38)$

(ii) Obtendo estimativa para λ

$$\overline{x} = \frac{1}{8}\sum_{j=1}^{8} x_{_j} = 21,65 \ . \qquad (2.1)$$

logo,

$$\lambda = \frac{1}{\overline{x}} = 0,0462 \ . \qquad (2.2)$$

(iii) A densidade da variável T é

$$f_{_T}(t) = 0,0462e^{-0,0462t},\ t > 0 \ . \qquad (2.3)$$

(iv) Gerar $N = 200$ amostras *bootstrap* cada uma de dimensão $n = 8$ de uma exponencial com densidade em 2.3

(v) Calcular as estimativas *bootstrap* λ_{ib} para cada uma delas, segundo 2.1 e 2.2.

¨

Tabela 3: Amostras geradas usando os parâmetros da amostra inicial

Amostras $f(t; 0,0462)$
$(8,01; 28,85; 14,14; 49,12; 3,11; 32,19; 5,26; 14,17)$
$(32,27; 2,10; 40,17; 32,43; 6,94; 30,66; 18,99; 5,61)$
...
$(40,26; 39,26; 19,59; 43,53; 9,55; 7,07; 6,03; 8,94)$

Tabela 4: Estimativas *bootstrap* calculadas para cada amostra gerada

	Estimativas *bootstrap* (λ_{ib})
amostra 1	$\lambda_{1b} = 0,0485$
amostra 2	$\lambda_{2b} = 0,0470$
...	...
amostra 200	$\lambda_{200b} = 0,0459$

(vi) Calculando as estimativas *bootstrap* para a média, variância e desvio padrão populacionais.

média: $\overline{\lambda_b} = \frac{1}{200}\sum_{i=1}^{200}\lambda_{ib} = 0,0513$.

variância: $s_b^2 = \frac{1}{200}\sum_{i=1}^{200}(\lambda_{ib} - \overline{\lambda_b})^2 = 0,0004$.

desvio padrão: $s_b = +\sqrt{0,0004} = 0,020$.

O próximo passo é encontrar o intervalo de confiança (IC) para o parâmetro λ aplicando-se o Algoritmo 2.

(i) Calculando P_i, $i = 1, \ldots, 200$

$P_1 = \lambda_{1b} - \overline{\lambda_b} = 0,0485 - 0,0513 = -0,0088$

...

$P_{200} = \lambda_{200b} - \overline{\lambda_b} = 0,0459 - 0,0513 = -0,0054$

(ii) Ordenar os P_i, obtendo $P_{(i)}$.

(iii) Supondo um nível de confiança de 90%, isto é, $1 - \alpha = 90\%$, obter então $P_{(\alpha/2)}$ e $P_{(1-\alpha/2)}$ através da fórmula $\lfloor (N-1)\alpha + 1 \rfloor$:

$\lfloor (200-1) \times 0,05 + 1 \rfloor = \lfloor 10,95 \rfloor = 10$

$\lfloor (200-1) \times 0,95 + 1 \rfloor = \lfloor 199,05 \rfloor = 199$

Portanto, $P_{(10)} = -0,0228$ e $P_{(199)} = -0,03135$

(iv) $I = 0,0462 - 0,03135 = 0,0149$

$S = 0,0462 - (-0,0228) = 0,0690$

(v) O IC de 90% para λ é

$$(0,0149\ ;\ 0,0690)$$

2.3.2 Exemplo 2: Cálculo dos Intervalos de Confiança *Bootstrap* quando a lei de probabilidade é desconhecida

No exemplo anterior foi considerado que a lei de probabilidade era conhecida, se esta não for conhecida, a amostra dada é considerada como sendo a população, isto é, a população passa a ser

$$(11,96; 5,03; 67,40; 16,07; 31,50; 7,73; 11,10; 22,38)\ .$$

e amostras aleatórias de tamanho N são retiradas, com reposição, desta população. A partir disso o procedimento segue-se como mencionado no exemplo anterior.

3 Geração de Variáveis Aleatórias

3.1 Binomial

Para a geração de uma variável Binomial levou-se em conta a particularidade de que o número de sucessos x de uma seqüência de n tentativas de Bernoulli possui uma distribuição Binomial. Foi utilizado o método da Composição aplicado à distribuição binomial. O algoritmo foi retirado de Jain [4]:

Gere n números aleatórios em $U(0,1)$. A quantidade de números sorteados que são menores que p representa a Binomial(n, p)

3.2 Exponencial

A distribuição exponencial tem a particularidade de sua função de densidade acumulada FDA é inversível. Além disso, a FDA ($u = F(x)$) é uniformemente distribuída entre 0 e 1. Para o caso da exponencial: $F(x) = 1 - e^{\lambda x} = u$ ou $x = -\frac{1}{\lambda}\ln(1-u)$. Por simplificação chegamos a $x = -\frac{1}{\lambda}\ln(u)$. Assim o problema de gerar a variável aleatória exponencial se resumiu a gerar u aleatoriamente em $U(0,1)$, baseados no Método da Transformação Inversa.

3.3 Normal

Na geração de uma variável com distribuição Normal foi realizada pelo método da convolução. Essa escolha partiu da característica do método, aplicável para os casos em que a variável aleatória pode ser expressada como uma soma de variáveis

aleatórias que possam ser geradas. Assim somando-se as n variáveis geradas obtemos a variável desejada.
Se x é uma variável aleatória que é a soma de duas outras, então a fdp de x pode ser analiticamente obtida pela convolução das fdp das duas variáveis. [4]
Uma soma de um grande número de variáveis de qualquer distribuição tem uma distribuição normal. Nesse caso podemos usar uma distribuição facilmente gerada como uma variável $U(0,1)$.
Na geração de $N(\mu,\sigma)$ procedemos o cálculo de:

$$\mu + \sigma \frac{(\sum_{i=1}^{n} u_i) - n/2}{(n/12)^{1/2}}$$

Onde cada u_i foi escolhido aleatóriamente com reposição em $U(0,1)$ e, segundo Jain [4], para n comumente se usa o valor 12.

3.4 Poisson

Uma variável aleatória que se distribui como uma Poisson pode ser gerada pelo seguinte algoritmo extraído de Jain [4].

Inicie com $n = 0$, *gere* $u_n \sim U(0,1)$ *e compute o produto* $\prod_{i=0}^{n} u_i$. *Tão logo que o produtório se torne menor que* $e^{-\lambda}$, *retorne n tal que* $u_0 u_1 \ldots u_{n-1} > e^{-\lambda} \geq u_0 u_1 \ldots u_n$.

4 Implementação e funcionalidades

O *SoftBoots* foi implementado todo em linguagem java. Assim contém toda a portabilidade inerente à linguagem de programação, ou seja, pode ser executado em qualquer sistema operacional com suporte a java. Sua implementação utiliza o padrão de projeto *Singleton* [7]. A escolha por esse padrão de projeto foi tomada para obter maior performance de execução e garantir a consistência de dados. Pela filosofia do Singleton existe apenas uma instância da classe ao invés de criar vários objetos da mesma, conseguindo otimizar gerenciamento dos recursos.

Foram implementadas interfaces bem definidas tanto para realizar chamadas do programa, quanto para as chamadas das distribuições. Isso torna o programa facilmente reusável e estendível. Em outras palavras, ele pode ser agregado a outros softwares e mais distribuições podem ser adicionadas a ele.

A Figura 4 mostra como foi estruturada a implementação do programa. Também pode ser visto a utilização do *Singleton* na classe *Bootstrap*. O resultado da estimativa *Bootstrap* é armazenado em DadosDoParametro. Além dessas existem as classes de distribuição, as quais contêm a geração da variável, a *GUI* (*Graphic User Interface*) e a classe Util que contém os cálculos matemáticos.

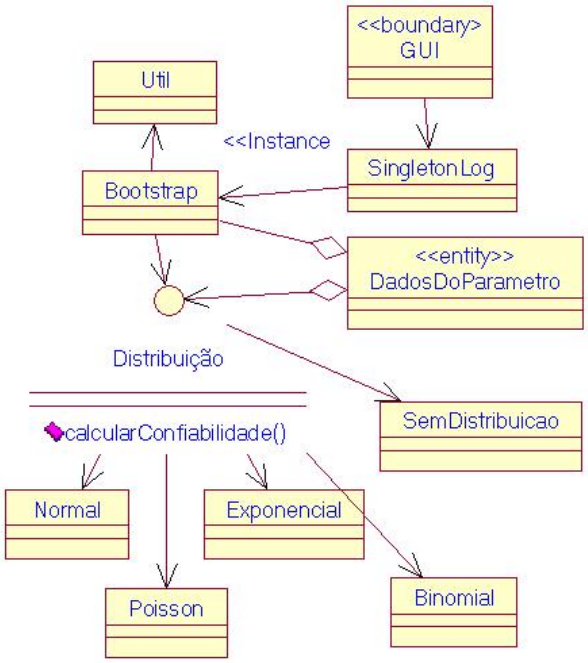

Figura 1: Diagrama da Estrutura de Implementação

O programa é capaz de realizar uma estimativa *bootstrap* de quatro distribuições conhecidas e para o caso da distribuição desconhecida. Atualmente estão implementadas as distribuições Normal, Binomial, Exponencial e *Poisson*, como pode ser visto na Figura 4. Ele retorna ao usuário informações sobre o parâmetro da distribuição, a média, a variância e o desvio padrão populacionais.

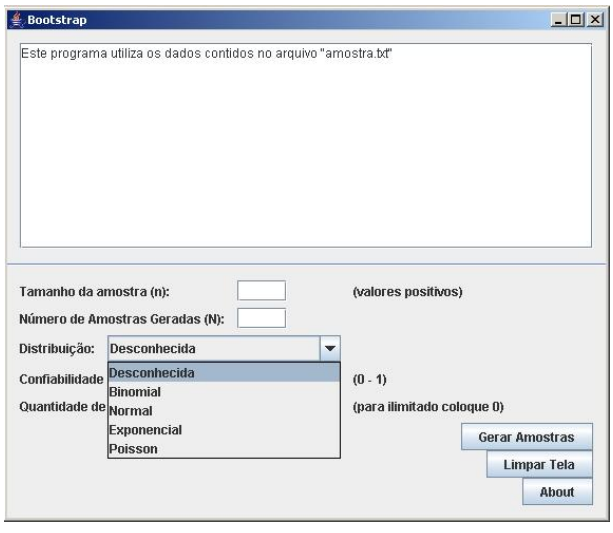

Figura 2: Tela de Distribuições

5 Exemplos de utilização do software

No primeiro passo o usuário deve inserir as amostras de entrada no arquivo chamado "amostra.txt", localizado no diretório raiz do programa. O arquivo de amostras deve ser formatado com uma entrada por linha e a indicação de decimal feita com ponto. Veja a seguir na Figura 5 .Em seguida o usuário abrira o programa. Onde terá a seguinte tela inicial. Veja Figura 5:

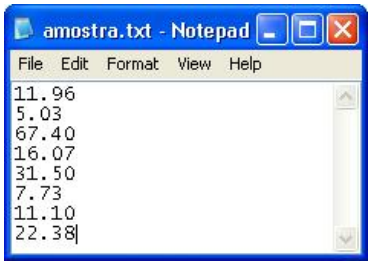

Figura 3: Tela da Amostra

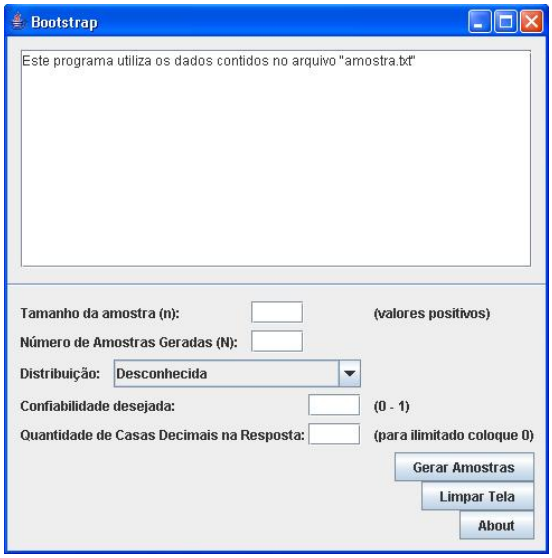

Figura 4: Tela inicial

Na tela inicial o usuário tem a opção de definir qual será o tamanho da amostra de entrada. Nesse caso temos uma amostra de 8 elementos e a distribuição escolhida é a exponencial. Deve-se também escolher a quantidade de amostras *bootstrap*, foi escolhido o valor 200, conforme o exemplo. O próximo passo é especificar o nível de confiança para o intervalo, o valor 0.90 foi inserido. Veja Figura 5.
O *SoftBoots* mostra o resultado na própria tela. Veja a Figura 5. A discrepância entre os valores é dado pela aproximação dos algoritmos de geração

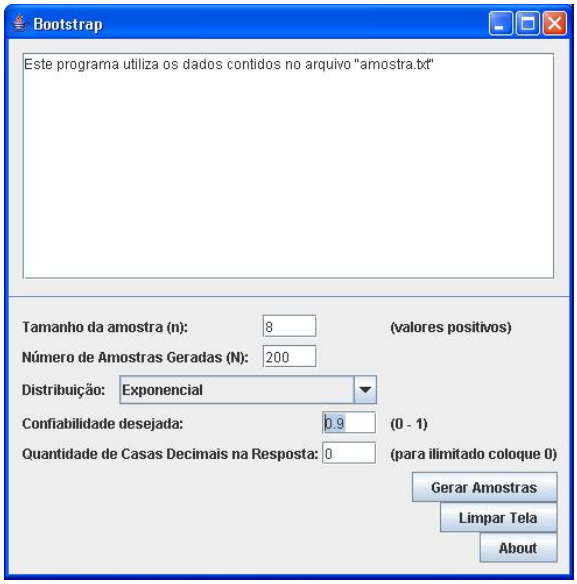

Figura 5: Tela com dados de entrada: tamanho da amostra, número de amostras, confiabilidade, distribuição e número de casas decimais

de Variáveis Aleatórias. Os resultados se mostraram consistentes aos de *Montgomery* [6]. Temos uma diferença de 0,322% no limite inferior e 0,106% para o limite superior. A média da estimativa *bootstrap* apresentou diferença de 5%, a variância em torno de 8% e desvio padrão 5,37%.

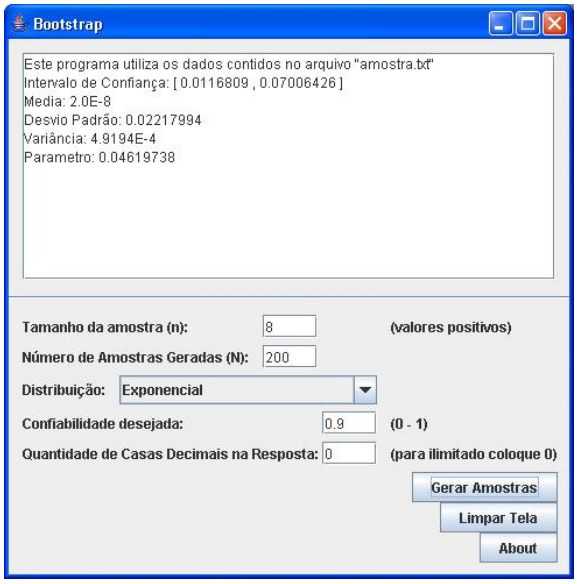

Figura 6: Tela com resultados do software para o Exemplo 1

Para o Exemplo 2 só se faz necessário apenas um ponto no programa que é a distribuição a partir da

qual calculamos a estimativa *Bootstrap*, que no caso será uma lei de probabilidade desconhecida (Sem Distribuição), como pode ser visto na Figura 5.

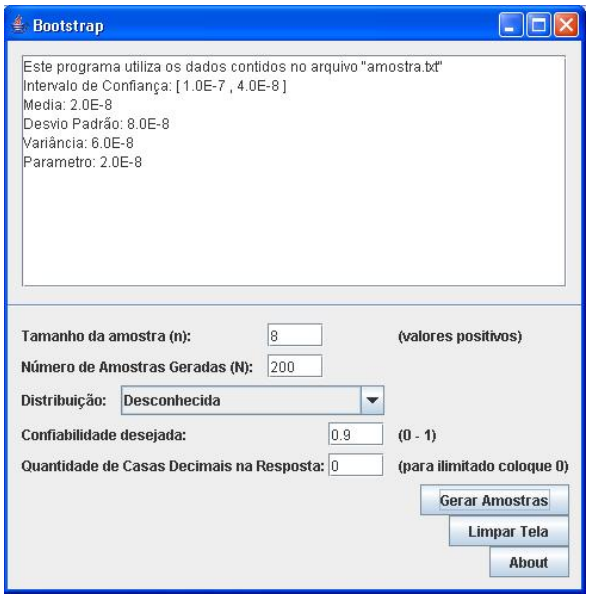

Figura 7: Tela com resultados do software para o Exemplo 2

Avaliando os resultados na Figura 5,para o limite inferior a diferença é de 1,4% e o limite superior de 6,76%. Para a média, variância e desvio padrão os valores respectivos são: 4,95%, 0,16% e 1,93%. Nesse exemplo a aproximação ficou satisfatória para a variância e o desvio padrão.

6 Trabalhos Futuros

Este trabalho foi desenvolvido de forma modularizada possibilitando, assim, a sua extensão para processar mais distribuições de probabilidade. Pode-se, também, adicionar mais funcionalidades de cálculos estatísticos, permitindo uma melhor interpretação dos resultados. Outra forma de desenvolver este trabalho é adicionando-o a outros programas, como por exemplo, o Netbook [8].

Referências

[1] G. M. Clarke, D. Cooke, A basic Course in Statistics(Edward Arnold), 1992.

[2] A. C. Davison, D.V. Hinkley, Bootstrap Methods and Their Application (Cambridge University press), 2005.

[3] B. Efron, R. J. Tibshirani, An Introduction to the Bootstrap, *Chapman & Hall/CRC*, 1993.

[4] R. Jain, The Art of Computer Systems Performance Analysis, Techniques for Experimental Design, Mesurement, Simulation and Modelling (John Wiley & Sons), 1983.

[5] P. L. Meyer, Probabilidade e Aplicações à Estatística, LTC, 1983.

[6] D. C. Montgomery, G. C. Runger, Estatística Aplicada e Probabilidade para Engenheiros, LTC, 2003.

[7] data & object factory, http://www.dofactory.com/Patterns/Patterns.aspx. Último acesso 15/10/2007.

[8] Campos, Marcília Andrade; at all. NetBook: uma ferramenta para avaliação de desempenho de sistemas de comunicação, 09/2004, SBRC 2004 - 22º Simpósio Brasileiro de Redes de Computadores, Gramado, RS, Brasil, 2004