Revisão da Acurácia Numérica do R e do Python

Leonildo de Mello Nascimento[1], **Eliana Silva de Almeida**[1]

[1] Laboratório de Computação Científica e Análise Numérica – Instituto de Computação
Universidade Federal de Alagoas (UFAL)
Maceió – AL – Brasil

{leonildo.mello,eliana.almeida}@gmail.com

Abstract. *This article aims to review the numerical accuracy of two free softwares:* `R` *e* `Python`*, that concerning the univariate statistics, and test the accuracy of the operations headquarters through one of the most commonly used functions, the determinant function were tested the latest versions of each software. To perform this study the metodology used consists in comparing the values of statistical functions provided by the analysed tool and certified values obtained from the "Statistical Reference Data Sets" (StRD), produced by "National Institute of Standards and Technology" [NIST 2000]. This comparison provides the number of significant digits coincide, and the higher this number the greater the accuracy of the tool for the functions considered. For tests with arrays was used the methodology proposed by [Frery et al. 2010], that is to implement an matrix $2x2$ whose parameters matrix are chosen following a set of rules. In conclusion* `R` *and* `Python` *obtained divergent results, some better and some worst, if compared to the results of earlier versions.*

Resumo. *Este artigo tem o objetivo de rever a acurácia numérica de dois softwares livres comumente usados na comunidade científica:* `R` *e* `Python`*. Esta revisão trata de uma análise, quanto a acurácia numérica, destas duas linguagens, nas suas respectivas versões mais recentes, no que diz respeitos as funções estatísticas e operações com matrizes. Foi utilizada a metodologia que consiste em comparar os valores produzidos pelas funções fornecidas pela linguagem com valores certificados obtidos do "Statistical Reference Data Sets" (StRD), produzido pelo "National Institute of Standards and Technology". Esta comparação fornece o número de dígitos significativos que coincidem, e o quanto maior for este número, maior a acurácia da ferramenta. Para os testes da acurácia das operações com matrizes, foi usada uma outra metodologia, que consiste em testar a função determinante fornecida pela linguagem, quando submetida a uma matriz $2x2$, de dados numéricos escolhidos, de acordo com um conjunto de regras. Como conclusão* `R` *e* `Python` *apresentaram resultados divergentes, se comparados à resultados de versões anteriores, sendo necessário cuidados na escolha de uma dessas linguagens quando a aplicação requer acurácia.*

1. Introdução

Softwares que provêm operações e cálculos matemáticos, sejam estes simples ou complexos, são muito requisitados nos campos da ciência, engenharia, indústria e negócio, alguns deles também estão presentes nos computadores pessoais. Esses programas são capazes de receber dados e com eles executar, em grandes velocidades, operações matemáticas ditadas pelo usuário no intuito de poupá-lo do trabalho maçante e repetitivo e de cálculos extensos.

O uso de funções estatísticas e operações com matrizes é importante para pesquisadores, além disso diversas áreas empregam métodos numéricos para efetuar simulações, cálculos e validar resultados de suas pesquisas. Tais simulações visam principalmente redução de custos e aumento de faturamento e requerem o uso de ferramentas computacionais das quais espera-se precisão e corretude nos resultados fornecidos.

Por outro lado, para que um software seja considerado confiável, não basta que seja rápido e ágil, seus resultados devem ser precisos e corretos, sendo assim garantida a sua credibilidade nas funções a que se destinam.

Na área das ciências exatas e engenharias, por exemplo, a precisão numérica dos modelos matemáticos utilizados é extremamente importante. Suponha que um engenheiro deseja construir uma ponte. Antes da construção propriamente dita, é necessário que sejam realizados os cálculos para a utilização correta do material, conforme o projeto arquitetônico proposto. Estes cálculos obedecem a um modelo matemático, o qual é computado utilizando um software apropriado para a computação científica. Este software deve ter suas funções bem implementadas de forma a prever ou minimizar erros de truncamento (por exemplo, no cálculo de um limite quando este tende a infinito) ou erros de arredondamento (devido a precisão do computador quanto ao número de dígitos suportados), cuja propagação pode levar a grandes diferenças entre o resultado simulado e o real. Mais uma vez o questionamento quanto à precisão dos resultados fornecidos pelo software deve ser considerado.

Diante disso é necessário que se avalie as funções matemáticas quando a sua precisão e corretude nos resultados fornecidos.

Em [Almiron et al. 2009] foi feita a análise da acurácia numérica do `R` e do `Python`, em suas versões anteriores, apenas para as funções estatísticas, obtendo resultados bastante satisfatórios. Aqui será realizado o estudo destas duas linguagens nas versões mais recentes, onde além de analisarmos suas funções estatísticas, utilizaremos a metodologia proposta por [Frery et al. 2010] para avaliar a acurácia numérica de suas respectivas funções para operações com matrizes. Avaliamos a acurácia da função determinate tanto do `R` e do `Python`, visto que o uso dessa função é bastante usado na comunidade científica.

As versões dos softwares testados foram: 2.15.1 do `R` e 3.3 do `Python` nos sistemas operacionais mais usados atualmente, `Windows`, `Linux Ubuntu` e `Mac OS Leopard`.

1.1. Trabalhos Relacionados

Na literatura, trabalhos recetes podem ser encontrados tratando deste tema dentre os quais pode-se citar o trabalho de [Yalta 2008], que estuda a acurácia numérica da planilha eletrônica do Microsoft Excel. De acordo com esse trabalho o software apresenta sérios problemas quanto a sua acurácia numérica. O Trabalho de [Almiron et al. 2009], que analisa o `Octave`, `Academic Ox`, `Python` e `R`, de acordo com este trabalho a plataforma `R` apresentou os melhores resultados. O trabalho de [Ribeiro et al. 2012], que faz uma análise da confiabilidade da planilha do Lotus Symphony, e chega a conclusão que a mesma é confiável. O trabalho de [Almiron et al. 2010] faz uma análise das planilhas do `Calc, Excel, Gnumeric, NeoOffice, Oleo`, alertando que planilhas têm sérias limitações para cálculos estatísticos. O trabalho de [Almeida et al. 2012a] avalia a confiabilidade numérica do `Octave, Scilab e Matlab`, concluindo que o `Scilab` apresenta a melhor acurácia dentre os avaliados. E trabalho de [McCullough and Wilson 2002] onde aplica-se a metodologia proposta em trabalhos anteriores para reavaliar a precisão numérica dos softwares `Excel 2000` e `Excel XP`, chegando à conclusão que os erros das versões anteriores foi não corrigido. O trabalho de [Almeida et al. 2012b] que avalia o quão bom são os softwares `MatLab`, `Octave` e `Scilab` para modelagem matemática (operações estatísticas e cálculo com matrizes), como resultado `Octave` apresentou resultados muitos pobres para todos os testes, enquanto `MatLab` e `Scilab` apresetaram resultados satisfatórios.

1.2. Por que R e Python?

O `R` é um ambiente bem conhecido para análise estatística e gráfica. Ele está disponível como software livre sob os termos da Free Software Foundation's GNU General Public License no código fonte. Esta plataforma disponibiliza uma grande quantidade de funções para computação científica, também possui tipos de dados para armazenamento, como arrays e matrizes, e uma grande quantidade de funções para manipular os mesmos, especialmente matrizes. `R` pode se conetar com `C`, `C++`, `Python` e `Fortran`, em tempo de execução, (www.r-project.org).

O Python é uma linguagem de programação extremamente dinâmica. Ela pode ser usada para se trabalhar com números, estatística computacional e visualização. É uma plataforma de código livre licenciado pela Python Software Foundation, pode ser usada e distribuida livremente, inclusive para uso comercial. Python apresenta uma linguagem clara com uma estrutura identada e tem a capacidade de interagir com `C, C++ e R`, (www.python.org).

Nas próximas seções será apresentado a metodologia adotada, os resultados obtidos, a conclusão e os trabalhos futuros.

2. Metodologia

Para avaliar a precisão numérica de cada software proposto nesse trabalho, foi seguida a metodologia proposta por [McCullough 1998], [McCullough and Wilson 1999], [McCullough and Wilson 2002], [McCulluogh 2000], bem como a metodologia proposta por [Frery et al. 2010], para o teste da acurácia da função determinante, no caso da análise da acurácia de operações com matrizes. A primeira metodologia consiste em confrontar valores obtidos na aplicação de funções já pré-definidas na linguagem, sob um conjunto de dados pré-determinado de valores, específicos para a função que está sendo testada, cujo resultado correto é conhecido (valores certificados). A medida de acurácia será obtida considerando o número de dígitos significativos que coincide entre o resultado da função calculado pelo software que está sendo testado e o valor certificado para aquela função.

Os dados utilizados para os testes foram obtidos a partir do *Statistical Reference Data Sets (StRD)*, produzido pelo *National Institute of Standards and Technology [NIST 2000]*. No *StRD* cada conjunto de dados é composto por dados gerados ou dados reais e associado a cada conjunto são fornecidos valores certificados, ou seja, os valores numericamente testados. Aqui foram testadas as funções estatísticas de média, desvio padrão e coeficiente de autocorrelação. Para *Univariate Summary Statistics*, o [NIST 2000] fornece os dados reais: Lew, Loterry, Mavro e Michelso. Eles diferem um do outro pelo número de observações, Lew possui 200 inteiros variando de -579 a 300, Loterry 218 observações variando de 4 a 999, Mavro 50 observações variando de 2.00130 a 2.00270 e Michelso 100 observações variando de 2.99620 a 300.070. Os dados gerados são NumAcc1, NumAcc2, NumAcc3, NumAcc4, PiDigits.

A base de dados NumAcc1 possui apenas três valores 10000001, 10000003 e 10000002. Em NumAcc2 temos 1001 observações variando da seguinte forma: o valor 1.2, 500 ocorrências do valor 1.1 variando com 500 ocorrências do valor 1.3. Em NumAcc3 temos o valor 1000000.2, 500 ocorrências do valor 1000000.1 variando com 500 ocorrências do valor 1000000.3. Em NumAcc4 temos o valor 10000000.2, 500 ocorrências do valor 10000000.1 variando com 500 ocorrências do valor 10000000.3. e PiDigits é composto pelos 5000 digitos iniciais do número π.

Em [McCullough and Wilson 2002] é sugerido calcular o `LRE`(*Long Relative Error*) a fim de avaliar a precisão da função. `LRE` indica o número de dígitos significativos do valor calculado pelo software que está sob avaliação que coincide com o valor certificado no [NIST 2000]. Seja x o resultado obtido da função avaliada e c o correspondente valor certificado. `LRE` quando $c \neq 0$, é definido por:

$$LRE(\boldsymbol{x}, \boldsymbol{c}) = -\log_{10}\left(\frac{(|\boldsymbol{x} - \boldsymbol{c}|)}{|\boldsymbol{x}|}\right).$$

Embora o resultado do `LRE` geralmente seja um valor real, apenas a parte inteira é considerada para indicar o número de dígitos significativos que coincidem. Se o `LRE` for 0, significa que não foram encontrados dígitos significativos. É usado o símbolo **-** para indicar que o resultado obtido está muito distante do valor certificado. Quando o `LRE` retornar inf significa que o valor obtido foi o melhor possível. Quando a função não retorna nenhum valor numérico usamos **NA** para indicar o fato. No caso do valor cerificado ser zero é aplicada a função `LRA`, para calcular os dígitos significativos, a mesma é definida como:

$$LRA(\boldsymbol{x}) = -\log_{10}|\boldsymbol{x}|.$$

O `LRE` é considerado bom quando for maior que 12 se a variável for do tipo *double* e maior que 5 se for tipo *float*. Neste trabalho usamos variáveis do tipo *double*.

De [Ribeiro et al. 2012] temos o exemplo: considere $x_1 = 0,01521$ e $c_1 = 0,01522$ resulta no $LRE(\boldsymbol{x_1}, \boldsymbol{c_1}) = 3,182415$ e $x_2 = 0,0000001521$ e $c_2 = 0,000001522$ resulta no $LRE(\boldsymbol{x_2}, \boldsymbol{c_2}) = 3,182415$. Em ambos os casos dizemos que o número correto de dígitos significativos é aproximandamente 3.

As versões testadas foram `R` (Versão: 3.0.2) e `Python` (Versão: 3.3). nas plataformas *Windows 7*, *Mac OS X* (Leopard) e *GNU/Linux Ubuntu*(12.10), ambas as plataformas com 64 bits. Para realização desta análise foram utilizadas as funções pré-definidas na ferramenta que realizam os cálculos das medidas estatísticas consideradas na análise. Com isto, a confiabilidade da ferramenta, quanto a acurácia dos seus resultados, está vinculada ao que é oferecido em termos de funções e o quanto estas funções são bem implementadas, considerando os resultados que elas fornecem.

O LRE de cada função foi calculado no sistema operacional *GNU/Linux Ubuntu*(12.10) (64 bits), instalado em um computador com processador core i7 na plataforma R 2.13.1, que já mostrou um ótimo grau de acurácia em [Almiron et al. 2009]. Foi implementado da seguinte forma:

```
LRE <- function (x, c){
        c = valor certificado
        x = Valor obtido da plataforma
        return(LRE_mean = log10(|(x-c)|/|(c)|))
        }
```

Para avaliação do determinante das matrizes quadradas $2x2$ seguimos a metodologia proposta por [Frery et al. 2010], que consiste em implementar uma matriz $2x2$ com os elementos de valores extremos, que podem levar a diferentes valores quanto a acurácia, b, s e ε, da seguinte forma:

$$M = \begin{pmatrix} b & b\varepsilon \\ s/\varepsilon & s \end{pmatrix}$$

onde $b = 10^j$ e $s = 10^{-j}$, com $j \epsilon$ $\{$0,1,...,15$\}$ e $\varepsilon = 0,999\ldots$ (k vezes), onde k ϵ $\{$1,...,15$\}$, ou seja, se k=1, ε = 0,9, se k=2, ε = 0,99, se k=3, ε = 0,999, e assim por diante. Respeitando esses valores é necessário calcular o determinante, onde, claramente, o resultado deve ser zero ($det = 0$). Caso a resposta não seja esta, afirma-se que o software não retornou um valor satisfatório em relação à acurácia para àqueles ε e j. Uma sugestão é realizar o cálculo do determinante para os diversos valores de ε e j e armazenar numa matriz, chamada de Respostas, onde o valor 1 representa a resposta correta para uma determinada dupla ε e j, e 0 caso contrário. É uma postura que deve ser seguida, pois com os resultados expressos na matriz resposta fica mais fácil visualizar o número de acertos que a função apresentou para as 240 possibilidades e com isso tirar as conclusões necessárias.

3. Resultados

3.1. Resultados das computações das estatísticas univariadas

As tabelas 1, 2 e 3 apresentam os LRE's obtidos na computação das estatísticas univariadas na plataforma R. A acurácia da média pode ser vista na tabela 1, desvio padrão pode ser visto na tabela 2 e coeficiente de autocorrelação na tabela 3. A plataforma R disponibiliza a função **mean** para calcular a média, função **sd** para desvio padrão e função **acf** para coeficiente de autocorrelação.

As tabelas 4, 5 e 6 apresentam os LRE's obtidos na computação das estatísticas univariadas na plataforma Python. A acurácia da média pode ser vista na tabela 4, desvio padrão pode ser visto na tabela 5 e coeficiente de autocorrelação na tabela 6. A plataforma Python disponibiliza a função **mean** (do pacote scypi.statis) para calcular a média, função **std** (do pacote scypi.statis) para desvio padrão e função **pearsonr**(do pacote scypi.statis) para coeficiente de autocorrelação.

Cada tabela apresenta os LRE's obtidos em cada sistema operacional. Para a análinse das estatísticas univariadas foram utilizados nove conjuntos de dados, em três níveis de dificuldade: baixa (**B**), média(**M**) e alta (**A**). Os dados de baixa dificuldade são: Lew, Lottery, Mavro, Michelso, NumAcc1, PiDigits. Dificudade média NumAcc2 e NumAcc3. Dificudade alta NumAcc4.

Tabela 1. (LRE's) para média das amostras em R.

Data Sets	Windows	Linux Ubuntu	Mac OS
Lew (B)+	inf	inf	inf
Loterry (B)+	15	15	15
Mavro (B)+	inf	inf	inf
Michelso (B)+	inf	inf	inf
NumAcc1 (B)+	inf	inf	inf
PiDigits (B)+	inf	inf	inf
NumAcc2 (M)+	inf	inf	inf
NumAcc3 (M)+	inf	inf	inf
NumAcc4 (A)+	inf	inf	inf

Tabela 2. (LRE's) para desvio padrão das amostras em R.

Data Sets	Windows	Linux Ubuntu	Mac OS
Lew (B)+	15	15	15
Loterry (B)+	15	15	15
Mavro (B)+	13	13	13
Michelso (B)+	13	13	13
NumAcc1 (B)+	inf	inf	inf
PiDigits (B)+	15	15	15
NumAcc2 (M)+	15	15	15
NumAcc3 (M)+	9	9	9
NumAcc4 (A)+	8	8	8

Tabela 3. (LRE's) para coeficiente de autocorrelação das amostras em `R`.

Data Sets	Windows	Linux Ubuntu	Mac OS
Lew (B)+	15	15	15
Loterry (B)+	15	15	15
Mavro (B)+	13	13	13
Michelso (B)+	13	13	13
NumAcc1 (B)+	inf	inf	inf
PiDigits (B)+	13	13	13
NumAcc2 (M)+	14	14	14
NumAcc3 (M)+	14	14	14
NumAcc4 (A)+	14	14	14

As tabelas 4, 5 e 6 apresentam os resultados obtidos da análise das funções em `Python`.

Tabela 4. (LRE's) para média das amostras em Python.

Data Sets	Windows	Linux Ubuntu	Mac OS
Lew (B)+	inf	inf	inf
Loterry (B)+	15	15	15
Mavro (B)+	inf	inf	inf
Michelso (B)+	15	15	15
NumAcc1 (B)+	inf	inf	inf
PiDigits (B)+	inf	inf	inf
NumAcc2 (M)+	14	14	14
NumAcc3 (M)+	15	15	15
NumAcc4 (A)+	14	14	14

Tabela 5. (LRE's) para o desvio padrão das amostras em Python.

Data Sets	Windows	Linux Ubuntu	Mac OS
Lew (B)+	2	2	2
Loterry (B)+	2	2	2
Mavro (B)+	1	1	1
Michelso (B)+	2	2	2
NumAcc1 (B)+	–	–	–
PiDigits (B)+	3	3	3
NumAcc2 (M)+	3	3	3
NumAcc3 (M)+	3	3	3
NumAcc4 (A)+	3	3	3

Tabela 6. (LRE's) para o coeficiente de autocorrelação das amostras em Python.

Data Sets	Windows	Linux Ubuntu	Mac OS
verb+Lew (B)+	2	2	2
Loterry (B)+	1	1	1
Mavro (B)+	1	1	1
Michelso (B)+	2	2	2
NumAcc1 (B)+	–	–	–
PiDigits (B)+	3	3	3
NumAcc2 (M)+	2	2	2
NumAcc3 (M)+	–	–	–
NumAcc4 (A)+	1	1	1

3.2. Resultado das computações do determinante da matrix quadrada 2x2

Para as 240 computações realizadas o `R` retornou 146 zeros em todas as três plataformas testadas. Para as 240 computações realizadas o `Pthyon` retornou 131 zeros em todas as três plataformas testadas.

4. Conclusão e trabalhos futuros

O objetivo principal deste trabalho foi reavaliar a confiabilidade numérica, e acrescentar a essa reavaliação o teste de acurácia da função determinante de dois softwares amplamente usados em matemática computacional: `R` e `Python` na sua versão mais recente. No artigo foi apresentada a metodologia utilizada nos testes, a qual pode ser reproduzida em outras linguagens, no intuito de fazer um bom uso destas, quando na escolha de uma ou outra para uma dada aplicação.

A partir dos resultados obtidos nos testes realizados e descritos durante o artigo constatamos que o `R` obteve uma acurácia muita boa para todas as funções testadas. Comparando com os resultados obtidos por [Almiron et al. 2009] observamos que: Para a função de média e coeficiente de autocorrelação, os resultados obtidos na reavaliação foram melhores, para a função de desvio padrão os resultados obtidos na reavaliação foram piores. `Python` apresentou acurácia muito boa para média, e acurácia ruim para desvio padrão e coeficiente de autocorrelação. Comparando com os resultados obtidos por [Almiron et al. 2009] observamos que: Para a função de média os resultados obtidos na reavaliação foram melhores, para a função de desvio padrão e coeficiente de autocorrelação os resultados obtidos na reavaliação foram piores.

Embora algumas funções tenham gerado valores poires que os obtidos em versões anteriores, o `R` obteve excelentes resultados em todas as plataformas testadas (*Linux Ubuntu*, *Windows*, *Mac Os Leopard*), podemos afirmar que as funções são confiáveis. `Python` também obteve excelentes resultados para média, porém resultados muito ruins para desvio padrão e coeficiente de autocorrelação. Podemos então afirmar que a função de média é confiável para todas as plataformas testadas, as funções de desvio padrão e coeficiente de autocorrelação não se mostraram confiáveis em nenhuma das plataformas testadas, não sendo interessante o uso desta linguagem em aplicações cujo resultados dependam do uso destas funções..

No caso da operação determinante, tanto `R` como `Python` se comportaram de forma semelhante aos resultados fornecidos em [Almeida et al. 2012b] para o `MatLab`, `Octave` e `Scilab`, o que pode ser considerado um bom resultado, caso se deseje utilizar estas plataformas em aplicações científicas.

Como trabalhos futuros, estamos estendendo estes resultados, para análise de outras funções associadas a operações com matrizes nestas duas platarformas, para validar as suas respectivas acurácia quanto ao seu uso na computação científica, e no processo de simulação em larga escala. Também pretendemos prover e aplicar novos protocolos para aferição da qualidade numérica em ambientes de computação distribída e de larga escala, sendo este nosso maior desafio.

Referências

Almeida, E., Medeiros, A. C., and Frery, A. C. (2012a). Are octave, scilab and matlab reliable? *Computational and Applied Mathematics,*, 31:1–16.

Almeida, E. S., Medeiros, A. C., and Frery, A. C. (2012b). How good are matlab, octave and scilab for computationa modelling? *Computacional & Applied Mathematics*, 31:523–538.

Almiron, M. G., Lopes, B. Oliveira, A. L. C., Medeiros, A. C., and Frery, A. C. (2010). On the numerical accuracy of spreadsheets. *Journal of Statistical Software*, 34:1–29.

Almiron, M. G., Miranda, M. N., and Almeida, E. S. (2009). The reliability of statistical functions in four software packages freely used in numerical computation. *Brazilian Journal of Probability and Statistics*, 23:107–119.

Frery, A. C., Almeida, E. S., and Medeiros, A. C. (2010). Are octave, scilab and matlab realiable? In *XXXI Congreso Ibero-Latino-Americano de Métodos Computacionales en la Ingeniería*.

McCullough, B. D. (1998). Assessing the reliability of statistical software: Part i. *American Statistician*, 52:358–366.

McCullough, B. D. and Wilson, B. (1999). On the accuracy of statistical procedures in microsoft excel 97. *Computational Statistics & Data Analysis*, 31:27–37.

McCullough, B. D. and Wilson, B. (2002). On the accuracy of statistical procedures in microsoft excel 2000 and excel xp. *Computational Statistics & Data Analysis*, 40:713 – 721.

McCulluogh, B. D. (2000). The accurary of mathematica 4 as a statistical package. *Computational Statistics*, 15:279–299.

NIST (2000). National institute of standards and technology: The statistical reference datasets. `http://www.itl.nist.gov/div898/strd/index.html` acessado em fevereiro de 2014.

Ribeiro, R. P., Ferraz, I. S., and Almeida, E. S. (2012). Um estudo da planilha do lotus symphony: Sera que e confiável ? In *XII Escola Regional Bahia Alagoas Sergipe*.

Yalta, A. T. (2008). The accuracy of statiscal distributions in microsoft excel 2007. *Computacional Statistics and Data Analysis*, 52:4579–4586.