SOFTWARE QUE TRABALHA COM ESTATÍSTICA COMPUTACIONAL

Luan Roberto Mendonça Castro[1]
Rafael Viana Sousa do Nascimento[2]
Victor Matheus de Sá Melo[3]
Cassius Gomes de Oliveira[4]

Ciência da Computação

cadernos de graduação
ciências exatas e tecnológicas

ISSN IMPRESSO 1980-1777
ISSN ELETRÔNICO 2316-3135

RESUMO

Desde de que o serviço de internet chegou ao Brasil, o número de usuários do serviço tem crescido rapidamente. a implantação de novas tecnologias e de preços acessíveis têm tornado o acesso a essa forma de comunicação muito mais facil e rapido. Pesquisas em 2016 demonstram que o país possui 116 milhões de pessoas conectadas e que utilizam a maior parte das vezes o celular para navegar, não se limitando apenas a internet móvel, mas também, ao uso do serviço de pontos de acesso de internet fixa, segundo a mesma pesquisa, 63,3 milhões de pessoas se mantêm offline, sendo que os motivos é a falta de interesse na ferramenta ou não sabem utilizar. É notável também, que as empresas provedoras devem investir em tecnologia, técnicos e em uma infraestrutura que atenda a demanda crescente de usuários. Através de uma pesquisa em redes sociais utilizando a plataforma do google e também gerando gráficos, podemos analisar a satisfação do brasileiro em relação ao serviço prestado.

PALAVRAS-CHAVE

Infraestrutura, Internet, País, Provedores, Redes Sociais, Satisfação, Tecnologia.

ABSTRACT

Since the internet service arrived in Brazil, the number of users of the service has grown rapidly. the introduction of new technologies and affordable prices have made access to this form of communication much easier and faster. Research in 2016 shows that the country has 116 million people connected and who use the mobile phone to navigate, not only the mobile Internet, but also the use of fixed internet access points service. According to the same survey, 63.3 million people remain offline, the reasons being the lack of interest in the tool or do not know how to use. It is also notable that the provider companies must invest in technology, technicians and an infrastructure that meets the increasing demand of users. Through a search in social networks using the platform of google and also generating graphs, we can analyze the satisfaction of the Brazilian in relation to the service provided.

KEYWORDS

Infrastructure. Internet. Country. Providers. Social Networks. Satisfaction. Technology.

1 INTRODUÇÃO

"Software livre" é um conceito importante no mundo da computação. Quando o software é livre, seu código fonte está universalmente disponível e pode ser livremente alterado para adaptá-lo a necessidades específicas. Assim sendo, o software livre é de fato gratuito, porém não se deve usar esta denominação para referir-se a plataformas computacionais sem custo.

O software gratuito (freeware) pode ser usado sem necessidade de compra ou pagamento, porém não oferece necessariamente acesso ao código fonte, por isso não pode ser alterado nem ter tal código estudado; unicamente pode-se utilizá-lo tal como foi disponibilizado. Fica, assim, estabelecida a diferença entre software livre e software gratuito. As plataformas R e Ox são, respectivamente, software livre e gratuito só para fins acadêmicos.

O artigo tem como objetivo demonstrar ao leitor o uso de duas plataformas computacionais apropriadas para computação científica, notadamente para simulação estocástica, análise estatística de dados e produção de gráficos. Essas plataformas são de grande valia para o trabalho cotidiano de estatísticos, matemáticos aplicados, físicos, químicos, engenheiros, economistas e profissionais de áreas afins. Elas devem ser vistas como complementares e não como substitutas, dado que cada uma tem vantagens relativas bem definidas. Diferente de outras plataformas muito disseminadas, tanto o programa R e o programa Ox são confiáveis e recomendáveis até para aplicações críticas.

A linguagem de programação Ox, a primeira das plataformas abordadas, é uma linguagem matricial de programação com orientação a objetos que foi de-

senvolvida por Jurgen Doornik. Sua sintaxe é similar à da linguagem C, como será ilustrado por meio de exemplo. Ela contém uma ampla lista de implementações numéricas de grande utilidade e é distribuída gratuitamente para uso acadêmico, havendo uma versão comercial para uso não-acadêmico. Uma de suas vantagens mais marcantes é a sua eficiência.

Programas bem escrito em Ox às vezes chegam a ser competitivos, em termos de tempo de execução, com programas escritos em linguagens de mais baixo nível, como, e.g., C e FORTRAN. A principal utilidade da linguagem Ox reside em utilizações computacionalmente intensivas, como, e.g., simulações de Monte Carlo. Essas simulações são de grande valia na avaliação do desempenho de procedimentos estatísticos de estimação e teste em amostras de tamanho típico. Em particular, são úteis para avaliações de robustez e da qualidade de aproximações, notadamente aproximações assintóticas.

A plataforma R, por sua vez, é um ambiente para análise de dados, programação e gráficos. Ela é distribuída gratuitamente mesmo para uso não-acadêmico e seu código fonte encontra-se disponível para inspeção e alteração, se desejável. Ela é semelhante à plataforma comercial S-PLUS, ambas sendo baseadas na linguagem S de programação, que foi desenvolvida por John Chambers e colaboradores. Sua maior utilidade, a nosso ver, reside na análise de dados e na produção de gráficos com qualidade de publicação.

Outra virtude de R é que, por ser uma plataforma muito utilizada no meio acadêmico, existe uma grande variedade de pacotes desenvolvidos para as mais diversas aplicações. Uma diferença entre as duas plataformas consideradas reside em suas formas de distribuição. Ox é distribuída gratuitamente apenas para uso acadêmico e seu código fonte não se encontrando publicamente disponível. Por outro lado, R é software livre.

2 LINGUAGEM R

R é uma linguagem e um ambiente para computação estatística, também para preparação de gráficos de alta qualidade. É um projeto GNU similar a linguagem e ambiente S-PLUS e, ainda que haja diferenças significativas entre eles, grande parte do código desenvolvido para um funciona no outro.

R oferece uma grande variedade de técnicas estatísticas (modelos lineares e não-lineares, testes estatísticos clássicos, modelos de séries temporais, classificação e agrupamento, entre outros) e gráficas, é altamente extensível.

R é uma coleção integrada de facilidades de software para manipulação de dados, realização de cálculos e preparação de gráficos, que inclui

- Tratamento efetivo de dados e facilidades de armazenamento;
- Operadores para cálculos em matrizes multidimensionais;
- Ferramentas de diversos níveis para análises de dados;
- Facilidades gráficas para análise de dados;
- Uma linguagem de programação bem definida, simples e eficaz que inclui ex-

pressões condicionais, laços, funções recursivas definidas pelo usuário e recursos de entrada e saída.

R pode ser usado como uma calculadora de grande capacidade. Vamos à seguinte sessão:

```
1  $ R
2  > 2
3  [1] 2
4  > 2+2
5  [1] 4
6  > sqrt (2)
7  [1] 1.414214
8  > exp ( sqrt (2))
9  [1] 4.11325
10 > sin ( exp ( sqrt (2)))
11 [1] -0.8258217
12 > sinh ( exp ( sqrt (2)))
13 [1] 30.56439
14 > sinh ( exp ( sqrt (2 - 1 i *2)))
15 [1] -20.96102 -6.575177 i
16 > q ()
17 Save workspace image ? [ y / n / c ]: n
```

Iniciamos uma sessão (em Linux) chamando, a partir de qualquer caminho, o sistema R (linha 1). Entre as linhas 1 e 2, teremos uma saída com informações da versão do R, sua data de lançamento e outros dados (aqui omitidos). A linha 2 passa ao R uma entrada constante e R a imprime (linha 3); a saída de dados numéricos é precedida por defeito pelo indicador da linha, neste caso [1], já que R supõe que pode haver mais de uma linha de dados. Na linha 4 pedimos ao R que calcule 2 + 2, e o resultado é impresso na linha 5. Nas linhas 6, 8, 10 e 12 solicitamos a realização de outros cálculos, e seus respectivos resultados são exibidos nas linhas 7, 9, 11 e 13.

R trabalha com números complexos; a unidade complexa $\sqrt{-1}$ é denotada na entrada por 1i, e as linhas 14 e 15 mostram uma operação com complexos e seu resultado, respectivamente. Ao terminar uma sessão (linha 16) R nos perguntará se desejamos guardar as variáveis e funções definidas (linha 17) para uso futuro; se assim o fizermos, salvaremos também os comandos que foram emitidos na sessão. Se desejamos exportar os comandos para um arquivo de texto, podemos fazê-lo com savehistory(file = "arquivo.txt"), para depois recuperá-los com loadhistory(file = "arquivo.txt").

Para ter uma ideia da capacidade gráfica do R podemos usar os seguintes comandos, que ativarão as demonstrações incluídas na distribuição básica:

```
1 > demo ( " graphics " )
2 > demo ( " image " )
3 > demo ( " persp " )
4 > demo ( " recursion " )
```

A linha 1 ativa a demonstração de algumas capacidades gráficas do R, incluindo o uso de cores. A linha 2 ativa a demonstração dos recursos de uso de imagens para visualização de dados multidimensionais. A linha 3 mostra alguns recursos do R para visualização de funções multidimensionais em perspectiva. A linha 4 mostra como R implementa um método adaptativo para calcular integrais numéricas.

2.1 AMOSTRAS UNIVARIADAS

A plataforma R oferece diversas funções para o cálculo de estatísticas descritivas, como a média, a mediana, estatísticas de ordem, medidas de dispersão, assimetria e curtose. Para ilustrar o uso destas funções será utilizado o conjunto de dados iris, disponível no R. Este conjunto de dados consiste em 151 linhas com seis colunas cada uma. A primeira linha, de tipo texto, descreve o conteúdo de cada coluna. As cinco primeiras colunas correspondem a medidas realizadas sobre flores, e a última, que é de tipo texto, categoriza em uma de três espécies cada flor medida.

A primeira coluna está rotulada Sepal.Length; para ver os valores basta emitir o seguinte comando:

```
[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8
[13] 4.8 4.3 5.8 5.7 5.4 5.1 5.7 5.1 5.4 5.1 4.6 5.1
[25] 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
[37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6
[49] 5.3 5.0 7.0 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2
[61] 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
[73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0
[85] 5.4 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7
[97] 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
[109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0
[121] 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9
[133] 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
[145] 6.7 6.7 6.3 6.5 6.2 5.9
```

Se queremos ter acesso às variáveis diretamente, sem necessidade de fazer referência ao conjunto de dados (iris), podemos colocar as variáveis na lista de objetos definidos com o comando

```
> attach(iris)
```

Para calcular a média amostral da variável Sepal.Length basta fazer

```
> mean(Sepal.Length)
[1] 5.843333
```

A mediana amostral é obtida com

```
> median(Sepal.Length)
[1] 5.8
```

Para calcular os quartis fazemos

```
> quantile(Sepal.Length)
0%  25% 50% 75% 100%
4.3   5.1   5.8   6.4   7.9
```

A função quantile() admite como argumento opcional um vetor de valores no intervalo [0, 1], retornando os percentis da amostra nesses pontos. Se, por exemplo, queremos calcular os decis deveríamos entrar quantile (iris$Sepal.Length, v), onde v é o vetor que contém os valores (i/10)$1 \leq i \leq 9$. Podemos fazê-lo manualmente, ou utilizar uma função do R para gerar este vetor auxiliar.

```
> quantile(Sepal.Length, seq(.1,.9,.1))
10% 20% 30% 40% 50% 60% 70% 80% 90%
4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90
```

Já que usaremos este vetor várias vezes, é conveniente guardá-lo em uma variável de nome mais curto e manejável com o comando:

```
> l_s <- Sepal.Length
```

As últimas versões do R admitem "=" como comando de atribuição, em vez do mais exótico (porém mais utilizado, até agora) "<-".

R também oferece funções para calcular medidas de dispersão como variância, desvio padrão e desvio médio absoluto, tal como é mostrado a seguir.

```
> var(l_s) [1] 0.6856935
> sd(l_s) [1] 0.8280661
> mad(l_s) [1] 1.03782
```

O máximo, o mínimo e o tamanho da amostra podem ser obtidos com

```
> max(l_s) [1] 7.9
> min(l_s) [1] 4.3
> length(l_s) [1] 150
```

Para calcular estatísticas de ordem superior, como assimetria e curtose, é necessário carregar o pacote e1071, que provê as funções skewness() e kurtosis().

```
1  > install . packages ( " e1071 " )
2  > library ( e1071 )
3  > skewness ( l_s )
4  [1] 0.3086407
5  > kurtosis ( l_s )
6  [1] -0.6058125
```

A linha 1 é necessária para baixar uma biblioteca que não está disponível localmente. R usará a conexão a Internet para obtê-la. Se o comando é dado para uma biblioteca já instalada, R verificará se há uma versão mais atual e, se houver, a instalará.

R permite construir gráficos com facilidade. Por exemplo, para construir um boxplot é necessário apenas emitir o comando

```
> boxplot(l_s, horizontal=T)
```

De fato, para gerar o arquivo que armazena o gráfico mostrado na Figura 2.1 é necessário ativar o dispositivo de saída, fazer o gráfico e desativar o dispositivo. A sequência de instruções é

```
> postscript("box_plot.eps")
> boxplot(l_s, horizontal=T)
> dev.off()
```

Tal como comentamos anteriormente, o Boxplot é particularmente útil para realizar uma comparação visual rápida entre várias amostras. Para isso, basta emitir o comando com os nomes das amostras separadas por comas.

Outro gráfico importante é o histograma, cuja versão mais simples pode ser construída com o seguinte comando:

```
> hist(Petal.Length, main="", freq=FALSE,
xlab="Largura de Pétalas", ylab="Proporções")
```

R oferece uma grande variedade de parâmetros para controlar o aspecto com que os histogramas em particular e, todos os gráficos em geral, são produzidos e exibidos.

O diagrama stem-and-leaf é obtido a partir do comando

```
> stem(Petal.Length)
```

```
The decimal point is at the |
1 | 012233333334444444444444
1 | 55555555555556666666777799
2 |
2 |
3 | 033
3 | 55678999
4 | 000001112222334444
4 | 55555555666777778888999999
5 | 000011111111223344
5 | 55566666677788899
6 | 0011134
6 | 6779
```

2.2 AMOSTRAS MULTIVARIADAS

R trata com facilidade dados multivariados, isto é, onde para cada indivíduo temos um vetor de observações. A notação que utilizaremos para denotar um conjunto

de n vetores k-dimensionais é y = (y1, . . . , yn), com yi R k . Este tipo de dados aparece naturalmente em estudos onde se mede mais de um atributo para cada indivíduo como, por exemplo, em antropométrica onde se registram o peso, a estatura, a idade e diversas medidas corporais de cada pessoa. Este tipo de análise está recebendo atualmente muita atenção, já que é um passo importante na cadeia de operações conhecida como *Knowledge Discovery in Databases* (KDD).

Para obter uma visão geral de um conjunto de dados deste tipo podemos emitir o seguinte comando:

```
> summary(iris)
 Sepal.Length    Sepal.Width     Petal.Length
Min. :4.300     Min. :2.000     Min. :1.000
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
Median :5.800   Median :3.000   Median :4.350
Mean :5.843     Mean :3.057     Mean :3.758
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
Max. :7.900     Max. :4.400     Max. :6.900
Petal.Width     Species
Min. :0.100     setosa :50
1st Qu.:0.300   versicolor:50
Median :1.300   virginica :50
Mean :1.199
3rd Qu.:1.800
Max. :2.500
```

A matriz de covariância descreve relações entre variáveis, assim como sua variância:

```
> var(iris[1:150, 1:4])
             Sepal.Length  Sepal.Width  Petal.Length
Sepal.Length   0.68569351  -0.04243400    1.2743154
Sepal.Width   -0.04243400   0.18997942   -0.3296564
Petal.Length   1.27431544  -0.32965638    3.1162779
Petal.Width    0.51627069  -0.12163937    1.2956094
              Petal.Width
Sepal.Length    0.5162707
Sepal.Width    -0.1216394
Petal.Length    1.2956094
Petal.Width     0.5810063
```

Nota-se que eliminamos a última coluna, que não contém valores reais, mas rótulos. Analogamente, é possível obter a matriz de correlações:

```
> cor(iris[1:150, 1:4])
             Sepal.Length  Sepal.Width  Petal.Length
Sepal.Length    1.0000000  -0.1175698    0.8717538
```

```
Sepal.Width        -0.1175698    1.0000000    -0.4284401
Petal.Length        0.8717538   -0.4284401     1.0000000
Petal.Width         0.8179411   -0.3661259     0.9628654
                   Petal.Width
Sepal.Length        0.8179411
Sepal.Width        -0.3661259
Petal.Length        0.9628654
Petal.Width         1.0000000
```

Um gráfico muito interessante para se ver simultaneamente o comportamento de todos os pares de variáveis de um conjunto multivariado é o diagrama de pares, que é obtido com

```
> pairs(iris)
```

A função *stars*() também é muito utilizada:

```
> stars(Iris)
```

3 LINGUAGEM OX

Ox é uma linguagem de programação matricial orientada a objetos que, utilizando uma sintaxe muito parecida com as de C e de C++, oferece uma enorme gama de recursos matemáticos e estatísticos. Para a preparação deste curso utilizou-se a versão 3.40 para Linux.

Do ponto de vista da precisão numérica, Ox é uma das mais confiáveis plataformas para computação científica. A versão que não oferece interface gráfica está disponível gratuitamente para uso acadêmico e de pesquisa. Ox está organizado em um núcleo básico e em bibliotecas adicionais. É possível chamar funções de Ox a partir de programas externos, bem como ter acesso a executáveis compilados externamente ao Ox.

Um primeiro programa em Ox poderia ser o seguinte:

```
 #include // include Ox standard library header
main() // function main is the starting point
{
decl m1, m2; // declare two variables, m1 and m2
m1 = unit(3);// assign to m1 a 3 x 3 identity matrix
m1[0][0] = 2; // set top-left element to 2
m2 =<0,0,0,1,1,1> ; //m2 is a 2 x 3 matrix, the first
                     // row consists of zeros, the
                    // second of ones
print("two matrices", m1, m2); // print the matrices
}
```

Ao executá-lo, teremos como saída

```
frery@frery$ oxl primero
```

```
Ox version 3.40 (Linux) (C) J.A. Doornik, 1994-2004
two matrices
2.0000          0.0000          0.0000
0.0000          1.0000          0.0000
0.0000          0.0000          1.0000

0.0000          0.0000          0.0000
1.0000          1.0000          1.0000
```

A fim de ilustrar a similaridade de sintaxes entre C e Ox, veremos, a seguir, o exemplo apresentado em uma pesquisa *Econometric and statistical computing using* Ox, onde são comparados programas com o mesmo propósito (gerar uma tabela de equivalência entre graus Celsius e Fahrenheit). Esta similaridade de sintaxes é, de fato, uma vantagem da linguagem Ox; conhecimento de C auxilia sobremaneira no aprendizado de Ox e, para aqueles que não têm domínio de C, o aprendizado de Ox conduz a uma familiaridade inicial com a linguagem C.

Primeiramente, o código C:

```
1  /* ********************************************
2   * PROGRAM : celsius . c
3   *
4   * USAGE : To generate a conversion table of
5  * temperatures ( from Fahrenheit to Cel
6   * sius ). Based on an example in the
7   * Kernighan & Ritchie 's book .
8   *
9  ******************************************** */
10
11  # include < stdio .h >
12
13  int main ( void )
14  {
15  int fahr ;
16
17  printf ( " \ nConversion Ã table Ã ( F Ã to Ã C )\ n \ n " );
18   printf ( " \ t %3 s Ã %5 s \ n " , " F " , " C " );
19
20   /* Loop over temperatures */
21   for ( fahr = 0; fahr <= 300; fahr += 20)
22   {
23  printf ( " \ t %3 d Ã %6.1 f \ n " , fahr , 5.0*( fahr
24   -32)/9.0 );
25   }
26
```

```
27 printf ( " \ n " );
28
29 return 0;
30 }
```

e a sua saída depois de compilado em ambiente Linux, usando o compilador gcc:

```
1 Conversion table ( F to C )
2
3            F C
4           0 -17.8
5          20 -6.7
6          40 4.4
7          60 15.6
8           80 26.7
9          100 37.8
10          120 48.9
11         140 60.0
12          160 71.1
13         180 82.2
14         200 93.3
15         220 104.4
16         240 115.6
17         260 126.7
18         280 137.8
19         300 148.9
```

A seguir o código equivalente em Ox

```
1 /* ******************************************
2 * PROGRAM : celsius . ox
3 *
4 * USAGE : To generate a conversion table of
5 * temperatures ( from Fahrenheit to Cel
6 * sius ). Based on an example in the
7 * Kernighan & Ritchie 's book .
8 ****************************************** */
9
10 # include < oxstd .h >
11
12 main ()
13 {
14 decl fahr ;
15
```

```
16 print ( " \ nConversion Ã table Ã ( F Ã to Ã C )\ n \ n " );
17 print ( " \ t Ã Ã F Ã Ã Ã Ã Ã Ã Ã C \ n " );
18
19 // Loop over temperatures
20 for ( fahr = 0; fahr <= 300; fahr += 20)
21 {
22 print ( " \ t " , " %3 d " , fahr );
23 print ( " Ã Ã Ã " , " %6.1 f " , 5.0*( fahr -32)
24 /9.0 , " \ n " );
25 }
26
27 print ( " \ n " );
28 }
```

e a sua saída

```
1 Ox version 3.40 ( Linux ) ( C ) J . A . Doornik , 1994 –
2 2004
3
4 Conversion table ( F to C )
5
6           F C
7         0 -17.8
8         20 -6.7
9          40 4.4
10         60 15.6
11         80 26.7
12        100 37.8
13        120 48.9
14         140 60.0
15        160 71.1
16        180 82.2
17        200 93.3
18        220 104.4
19        240 115.6
20        260 126.7
21        280 137.8
22        300 148.9
```

Nos documentos de ajuda incluídos com as diversas distribuições do Ox existe uma grande variedade de exemplos, assim como na detalhada documentação que acompanha esta plataforma.

3.1 MODELOS PARAMÉTRICOS

Um modelo estatístico paramétrico é uma família de distribuições de probabilidade indexadas (determinadas) por um vetor p dimensional θ sobre o qual só sabemos que pertence a um conjunto $\Theta \subset R$ p. Os dados nos servirão para termos uma idéia do valor parâmetro θ.

A variável aleatória não trivial mais simples é a que pode adotar só dois valores: 1, com probabilidade $0 \leq p \leq 1$, e 0 com probabilidade $1 - p$. Dizemos que esta variável aleatória tem distribuição Bernoulli com probabilidade p de êxito.

A distribuição da soma de m variáveis aleatórias independentes e identicamente distribuídas, cada uma com distribuição Bernoulli com probabilidade p de êxito, é uma variável aleatória que pode adotar n + 1 valores, $0 \leq k \leq n$, cada um com probabilidade:

$$\Pr(Y = k) = \binom{k}{n}^k (1-p)^{n-k},$$

onde $(k|n) = n!/(k!(n-k)!)$, diremos que a variável aleatória Y obedece a distribuição binomial com parâmetros n e p.

A média e a variância de uma variável aleatória com distribuição binomial com parâmetros n e p são, respectivamente, np e $np(1-p)$. E imediato que uma variável aleatória com distribuição binomial com parâmetros n = 1 e p segue distribuição Bernoulli com probabilidade de êxito p.

Consideremos uma situação onde um bom modelo para as observações é a distribuição binomial. Suponhamos que a probabilidade p de êxito individual seja muito pequena, com a qual a probabilidade de observar qualquer evento distinto de zero será, também, muito pequena. Para compensar esta situação, suponhamos que sejam realizadas muitas observações (repetições) independentes, isto é, que n seja grande. É possível provar, usando somente ferramentas analíticas, que

$$\lim_{\substack{p>0 \\ n> \\ np>0}} \Pr(Y = k) = \lim_{\substack{p>0 \\ n> \\ np>0}} \binom{n}{k} p^k (1-p)^{n-k} = \frac{0^k}{k!} e^{-0}$$

Esta lei de probabilidade é denominada distribuição de Poisson com parâmetro $\theta > 0$. Uma variável aleatória que obedece a distribuição de Poisson com parâmetro θ tem média e variância iguais a θ.

As distribuições mencionadas até agora são discretas, isto é, os valores que as variáveis aleatórias cuja distribuição está caracterizada por elas são finitos ou, como máximo, contáveis (enumeráveis).

A distribuição uniforme sobre o intervalo (a, b) é aquela que a cada intervalo $(b, c) \subset (a, b)$ atribui probabilidade

$$\Pr\big(Y \in (b,c)\big) = \frac{c-b}{b-a}$$

Para o caso particular a = 0 tem-se que a esperança de uma variável aleatória com esta distribuição é b/2 e sua variância é b/12.

Uma variável aleatória Y com distribuição normal ou gaussiana de média μ ∈ R e variância σ 2 > 0 tem sua distribuição caracterizada pela densidade.

$$f(y;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

Para o caso particular a = 0 tem-se que a esperança de uma variável aleatória com esta distribuição é b/2 e sua variância é b/12.

Uma variável aleatória Y com distribuição normal ou gaussiana de média μ R e variância σ 2 > 0 tem sua distribuição caracterizada pela densidade.

$$f(y;\mu,\sigma) = \frac{1}{\beta^{a}\Gamma(a)}y^{a-1}\exp\left(-\frac{y}{\beta}\right)\mathrm{IR}+\ (y)$$

Onde IA denota a função indicadora do conjunto A. Esta situação denota-se Y Γ(α, β). Esta densidade está disponível na plataforma R por meio da função dgamma. A esperança de uma variável aleatória com esta distribuição é αβ, sua variância sendo αβ².

A variável aleatória Y segue uma lei triangular com parâmetro α > 0 se a sua densidade é dada por:

$$f(y;a) = \begin{cases} 0\ se\ y < -a \\ a^{-1}(1+a^{-1}y)\ se - a \le y < 0 \\ a^{-1}(1-a^{-1}y)\ se\ 0 \le y \le a \\ 1\ se\ y > a \end{cases}$$

A sua função de distribuição acumulada é dada por:

$$F(y;a) = \begin{cases} 0\ se\ y < -a \\ \frac{(a+y)^2}{2a^2}\ se - a \le y < 0 \\ \frac{1}{2}\left(1-\frac{y(y-2a)}{a^2}\right)se\ 0 \le y \le a \\ 1\ se\ y > a \end{cases} \qquad (3.7)$$

A inversa da função de distribuição acumulada é dada por:

$$F^{-1}(y;a)\begin{cases} a\left(\sqrt{2u}-1\right)se\ 0 < u\ \le \frac{1}{2} \\ a\left(1-\sqrt{2(1-u)}\right)\ se\ \frac{1}{2} < u \le 1 \end{cases}$$

A variável aleatória Y segue uma lei de Weibull-Gnedenko com parâmetros α ≠ 0 e β > 0 se a sua densidade é dada por:

$$f(y;a,\beta) = |a|\beta y^{a-1}\exp\left(-\beta y^{a}\right)\mathrm{IR}+\ (y)$$

Esta situação é denotada Y W(α, β). A variável aleatória Y segue uma lei Erlang com parâmetro α N se a sua densidade é dada por:

$$f(y;a) = \frac{1}{\Gamma(a)}y^{a-1}e^{-y}\mathrm{IR}+\ (y).$$

É possível ver que a sua função de distribuição acumulada é:

$$F(y;a) = 1 - e^{-y}\left(1 + \sum_{1 \le i \le a-1} \frac{y^i}{i!}\right)$$

4 CONCLUSÃO

O desenvolvimento do presente artigo possibilitou uma análise de como é demonstrada a metodologia da Estatística nos dois programas R e Ox. Além da sua precisão computacional com todos os seus comandos, atalhos e afins percebe-se que traz uma grande vantagem ao profissional que os usa, conforme o mesmo busque o conhecimento para trabalhar com tais ferramentas, fugindo ainda mais do erro humano, sendo cada vez mais preciso com as respostas dos problemas estatísticos.

REFERÊNCIAS

APOSTILA, **Curso de extensão:** Software Estatístico Livre R. Disponível em: https://www.ime.uerj.br/~mrubens/slae/Texto_SLAE2.pdf. Acesso em: 14 ago. 2018.

LOPES, Hélio. **Introdução à simulação estocástica usando R. INF2035**. Rio de janeiro: PUC, 2013. Disponível em: http://www-di.inf.puc-rio.br/~lopes//inf2035/Aula2Sim.pdf. Acesso em: 12 ago. 2018.

R(linguagem de programação). **Wikipédia, a enciclopédia livre**, 2017. Disponível em: https://pt.wikipedia.org/wiki/R_(linguagem_de_programa%C3%A7%C3%A3o). Acesso em: 25 set. 2018.

Data do recebimento: 21 de julho de 2016
Data da avaliação: 9 de novembro de 2016
Data de aceite: 12 de dezembro de 2017

1 Graduando em Ciência da Computação – UNIT. E-mail: luan.mendonca@souunit.com.br
2 Graduando em Ciência da Computação – UNIT, E-mail: rafael.viana@souunit.com.br
3 Graduando em Ciência da Computação – UNIT. E-mail: victor.matheus97@souunit.com.br
4 Mestre; Professor da Universidade Tiradentes – UNIT. E-mail: cassius.gomes@souunit.com.br