O programa R e suas aplicações em Herpetologia

Diogo B. Provete[1,2] e Adriano S. Melo[1,3]

[1] *Departamento de Ecologia, Instituto de Ciências Biológicas, Universidade Federal de Goiás. Caixa Postal 131, CEP 74001-970, Goiânia, GO, Brasil. E-mail: dbprovete@gmail.com*
[2] *Programa de Pós-Graduação em Ecologia e Evolução, Instituto de Ciências Biológicas, Universidade Federal de Goiás. Caixa Postal 131, CEP 74001-970, Goiânia, GO, Brasil. E-mail: asm.adrimelo@gmail.com*
[3] *Autor para correspondência.*

INTRODUÇÃO

O programa R é, ao mesmo tempo, um programa de análises estatísticas e um ambiente de programação orientado à análise de dados e exploração gráfica (Ihaka e Gentleman, 1996; Venables e Ripley, 2002). O R é uma implementação livre de uma linguagem muito utilizada por estatísticos na década de 1990, chamada S. Ainda hoje existe um programa comercial baseado nesta linguagem, o S-PLUS. Desde sua criação em 1993 e o lançamento da primeira versão pública para usuários em 2001, a comunidade de usuários da linguagem R vem crescendo exponencialmente (*veja* Muechen, 2012).

O R possui pouquíssimas opções de menu, o que significa que à primeira vista o usuário encontrará uma tela vazia e um *prompt* no qual irá inserir as linhas de comando. Esta característica, familiar aos usuários que trabalharam com computadores antes do MS Windows ou que hoje trabalham com sistemas operacionais similares a Unix, é uma das grandes vantagens e, ao mesmo tempo, uma de suas desvantagens. Praticamente não existem botões para serem clicados e, portanto, o usuário tem de escrever o nome das funções que quer usar bem como as informações requeridas por ela. No entanto, uma grande vantagem comparado a outros softwares é a possibilidade de saber exatamente o seu funcionamento, já que outros programas comerciais são uma espécie de "caixa preta". Outra vantagem que pode auxiliar muito os iniciantes é a disponibilidade de pacotes que fornecem interfaces gráficas, como o R commander (http://socserv.mcmaster.ca/jfox/Misc/Rcmdr), JGR (http://rforge.net/JGR), RWard (http://rkward.sourceforge.net), Rattle (http://rattle.togaware.com) e editores como Tinn-R (www.sciviews.org/Tinn-R) e RStudio (www.rstudio.org). Estas ferramentas podem tornar o uso do R um pouco mais amigável para iniciantes, embora ainda seja necessário aprender alguns comandos (ou funções).

A maior desvantagem de programas com linha de comando é o tempo de aprendizagem, que inicialmente costuma ser bastante longo. Por outro lado, uma vez superada esta fase, o trabalho se torna altamente produtivo e, por que não dizer, prazeroso! Diversas outras características tornam o R um programa extremamente interessante e versátil, como a disponibilidade de listas de e-mail para esclarecimento de dúvidas (www.R-project.org/mail.html; https://listas.inf.ufpr.br/cgi-bin/mailman/listinfo/r-br), blogs (www.r-bloggers.com), livros e apostilas gratuitas (http://cran.r-project.org/other-docs.html#nenglish), e a possibilidade de desenvolvimento de novas análises e rotinas de programação. De fato, muitas análises novas em ecologia, evolução, sistemática e áreas afins são publicadas em periódicos de alto impacto que trazem frequentemente uma função desenvolvida em R.

Exemplo de uso

Talvez mais de 99% do trabalho em R é feito de forma parecida com:

```
>valor<-100
>log(x=valor, base=10)
>[1] 2
```

onde `valor` é um objeto (neste caso um vetor) que inclui o valor 100; o símbolo "<–" designa que o que está a esquerda receberá o que está a direita, ou seja, criaremos um objeto chamado "valor" que conterá o número 100. `log()` é a função (ou análise) que queremos fazer; x é um argumento onde inserimos a informação requerida pela análise; `base` é um argumento onde informamos uma opção de análise, neste caso a base do logaritmo, e `[1] 2` é o resultado da função que sai na tela. O `[1]` apenas indica que a posição do primeiro valor do resultado na linha tem posição `[1]`.

Cada função pode ter de 0 a dezenas de argumentos, que controlam a entrada da informação requerida pela função e suas opções. No exemplo acima, poderíamos colocar `base=2` para obter o logaritmo de base 2. O R trabalha com vetores e, no exemplo dado, temos a opção de colocar um vetor (ou sequência de valores) para o argumento x. Neste caso, o resultado seria o logaritmo de cada um dos valores componentes do vetor:

```
valor<-c(100, 200, 300)
log(x=valor, base=10)
[1] 2.000000 2.301030 2.477121
```

No exemplo acima, o objeto `valor` tem três números e é criado com a função `c()`, que significa simplesmente concatenar ou colocar junto. Como resultado, a função `log()` é aplicada a cada um dos três números.

O uso do R basicamente consiste em conhecer as funções e como usá-las. Existem milhares de funções, a maioria para análises que nem podemos imaginar. Entretanto, na prática, acabamos trabalhando com um conjunto reduzido, composto por funções básicas (como `log()` e `c()`) e outras específicas a nossa área de trabalho.

O primeiro passo para fazer algo no R é saber no nome da função adequada. Podemos descobrir isto num livro, internet, apostila, aula, lista de discussão etc. A partir daí podemos simplesmente abrir uma janela de ajuda que explica os detalhes da função incluindo a forma de uso. O comando:

```
?log
```

abrirá uma janela de seu navegador de internet contendo várias informações da função `log()`. Cada função, obrigatoriamente, deve ter um arquivo de ajuda (isto é válido para aquelas funções disponíveis no sítio do R).

Programação

Uma vantagem do R é ter muitas análises. Uma outra é que você pode escrever suas próprias funções. Isto pode parecer um tanto complicado, mas não é. Um exemplo:

```
simp<-function(dados){
soma<-sum(dados)
proporcao<-dados/soma
proporcao2<-proporcao^2
sum.prop2<-sum(proporcao2)
indice<-1-sum.prop2
return(indice)
}
```

Você pode escrever isto num editor de texto, copiar e depois colar dentro do R. Sua função para calcular o índice de diversidade de Simpson estará pronta! Para usá-la, suponha que tenha obtido abundância de cinco espécies de sapos em sua área de trabalho:

```
sapos<-c(1, 1, 5, 10, 34)
simp(dados=sapos)
[1] 0.5067282
```

Na função, note que em cada linha colocamos uma operação e guardamos o resultado com um nome. Vamos fazendo isto até que tenhamos obtido o resultado final, que na função foi chamado de `indice`. No fim, basta mandar o resultado para o usuário com a função `return()`. Os outros objetos criados dentro da função são perdidos após a finalização do cálculo (por exemplo: `sum.prop2`). Portanto, só o que mandarmos para fora da função com `return()` ficará disponível para o usuário.

Alguns detalhes da função `simp()`:

`function()` é uma função para criar funções. Dentro dos parênteses colocamos os nomes dos argumentos da função. Estes nomes aparecerão ao longo do código da função.

`{}` servem para delimitar a função, que pode conter de uma a centenas de linhas;

`sum()`, como você já deve ter percebido, é a função para soma. '/', '^', e '–' indicam, respectivamente, divisão, exponenciação e subtração, como você também já deve ter percebido.

Uso Diário

Conforme dito antes, o início com o R nem sempre é fácil. Entretanto, caso passe a usar diariamente, rapidamente você se sentirá mais confortável. Isto pois, a partir de um certo treinamento, o R passa a ser muito intuitivo.

Cada um tem seu jeito de trabalhar com R, embora algumas dicas sejam um tanto gerais:

1) Você deve prestar atenção no que está fazendo. Uma vírgula errada e você obtém um erro. Este vem acompanhado de uma mensagem que normalmente ajuda descobrir o que fizemos de errado. Tente trabalhar com o R de forma concentrada. O uso conjunto de um editor, como o RStudio que corrige automaticamente parênteses, aspas e colchetes pode ajudar também.
2) O arquivo de ajuda das funções possui um formato fixo. No entanto, a linguagem usada pelos autores das funções nem sempre é tão acessível para iniciantes. Tente ler com atenção. Além disto, uma das partes mais uteis do arquivo de ajuda são os exemplos. Copie uma linha, cole na linha de comando e veja o que acontece. Repita com a segunda linha e assim por diante de modo a entender o que cada passo faz.
3) Use sempre o R conjuntamente com um editor de textos, de preferência algum que corrija ou destaque os erros, como Tinn-R e RStudio. Você não precisará lembrar de tudo novamente quando precisar revisar a análise depois de um certo tempo. Além disso, se tiver de repetir alguma análise ela já estará pronta, bastando somente adaptar o texto (*script*) antigo para os novos dados.

Funcionalidades e Aplicações em Herpetologia

Atualmente existem cerca de 3.500 pacotes disponíveis no CRAN, cada um com 1 ou até mais de 100 funções. Estes pacotes realizam análises de várias áreas do conhecimento. Em herpetologia, o R pode ser útil em uma variedade de análises comumente utilizadas, desde análises uni e multivariadas (http://cran.r-project.org/web/views/Environmetrics.html) utilizadas em ecologia de comunidades e de populações, bioacústica (http://rug.mnhn.fr/seewave), análises filogenéticas (http://cran.r-project.org/web/views/Phylogenetics.html) e espaciais (http://cran.r-project.org/web/views/Spatial.html).

CONCLUSÃO

Apesar da aparente dificuldade inicial de utilização, o programa R apresenta muito mais vantagens do que desvantagens. Além dos endereços já mencionados, listamos abaixo alguns recursos que podem ser úteis para novos usuários. O conhecimento de uma linguagem de programação fornece uma imensa

liberdade ao profissional para realizar tarefas ou mesmo para criar análises e automatizar rotinas. Acreditamos que investir tempo na aprendizagem de uma linguagem de programação leva tempo e requer prática, mas este deve ser visto como um investimento na carreira profissional.

AGRADECIMENTOS

Os autores gostariam de agradecer à Camila Both pelo convite para escrever este artigo. Durante a redação deste manuscrito, Diogo B. Provete recebeu uma bolsa de doutorado da CAPES-DS, Adriano S. Melo recebeu auxílios de pesquisa (476304/2007-5; 474560/2009-0; 558187/2009-9) e uma bolsa de produtividade (302482/2008-3) do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS BIBLIOGRÁFICAS

Ihaka, R. e R. Gentleman. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics,* 5:299-314.

Muenchen, R.A. 2012. The popularity of data analysis software. Consultado em 23.02.2012. <http://r4stats.com/popularity>.

Venables, W.N. e B.D. Ripley. 2002. Modern Applied Statistics with S. Springer, New York. 495p.

LEITURA RECOMENDADA

Burns, P. 2009. The R inferno. Disponível em: www.burns-stat.com/pages/Tutor/R_inferno.pdf.

Crawley, M.J. 2007. The R book. Wiley, New York. 942p.

Landeiro, V.L. 2011. Introdução ao uso do programa R. Disponível em: http://cran.r-project.org/doc/contrib/Landeiro-Introducao.pdf. Material de apoio: http://ppbio.inpa.gov.br/Port/public/disciplinas2/introducaor/document_view.

Paradis, E. 2005. R for beginners. Disponível em: http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf.

Provete, D.B., F.R. Silva, e Gonçalves-Souza, T. 2011. Estatística aplicada à ecologia usando o R. Disponível em: http://cran.r-project.org/doc/contrib/Provete-Estatistica_aplicada.pdf. Material de apoio: https://sites.google.com/site/diogoprovetepage/teaching.

Zuur, A.F., E.N. Ieno, e E.H.W.G. Meester. 2009. A beginner's guide to R. Springer, New York. 218p.



Cycloramphus boraceiensis, Picinguaba, Ubatuba, SP. (Foto: M. Martins).