An Introduction to R Statistical Computing for Horticultural Science

Alexander Q. Susko[1,3] and Zachary T. Brym[2]

Additional index words. computer programming, data analysis, data management, horticulture education

Summary. **We present the format for a workshop on introductory computer programming, which was held at the 2015 American Society for Horticultural Science (ASHS) Annual Conference in New Orleans, LA. The main workshop objective was to familiarize attendees with basic computer programming, including data structures, data management, and data analysis. The workshop used the general programming language R, though the concepts and principles presented are transferable across programming languages. Given the increased importance of statistical analysis in the agricultural sciences, the workshop was well attended. Participants appreciated the opportunity to improve their computational literacy and supported follow-up workshops like this at future ASHS events. We have released the presentation and the companion R script online.**

Robust data management and statistical analysis have become increasingly important in horticultural science. Greater complexity in experimental design as well as the rise in "-omics" technologies (e.g., genomics, metabolomics, phenomics) have required the use of computers to execute customized and sophisticated analyses that draw from a large amount of information. Computers must also meet the demand to store and curate large amounts of data generated through experimental equipment, sensors, or surveys. The amount of these data across all scientific disciplines has been expanding exponentially since the 1990s, and often requires uniform curation and detailed documentation to share and analyze across research groups (Howe et al., 2008). This uniformity in both data curation and analysis improves repeatability and allows others to make use of the data and data products to further support scientific discovery.

Many computer programs are available to perform statistical analysis. Programs vary based on their user interface (text, graphics); program language, including C (Bell Laboratories, Murray Hill, NJ), R (University of Auckland, Auckland, New Zealand), or Python (Centrum Wiskunde & Informatica, Amsterdam, The Netherlands); and licensing (proprietary, open source). Although requests for statistical computing skills in job postings have increased broadly since 2005, growth in R programming skills increased over 400%, underscoring this software's significance in data-oriented positions. In comparison, there was just a 25% increase in postings seeking SAS (SAS Institute, Cary, NC) skills during this period (Indeed.com, 2016). R uses a text-based or command-line interface, and is compiled through its own computing language, also called R, which promotes consistent style and documentation. In addition to the command-line interface, R offers a graphical user interface and an integrated development environment that makes R accessible to beginners and powerful for advanced users (Rstudio, Boston, MA). R has attracted a strong interest in the biological sciences due to its open-source nature and active development community that spans many sectors and disciplines. Such open-source software is free of charge with anyone able to participate in software development and contribute program packages containing custom functions and operations (Ihaka and Gentleman, 1996). Numerous packages exist relevant to the agricultural sciences, such as those for data management [dplyr (Wickham, 2015)], elaborate graphing [ggplot2 (Wickham, 2009)], analysis of genomic data [Bioconductor (Huber et al., 2015)], or mixed linear model analysis [lme4 (Bates et al., 2015)]. Despite the open access to R and its many resources, programming in R requires a large early investment to learning about software development in a general programming environment by users.

We designed a 90-min workshop to introduce horticultural scientists to basic computer programming with R to help beginning users navigate this learning curve. This workshop was inspired by the data management education methods developed by Data Carpentry (Teal et al., 2015a). We created a slide presentation as well as an R script that was distributed to participants to facilitate their learning during the workshop. The workshop specifically addressed the following points: data structures and workflow; how to find help and install additional packages from open-source software; and how to import, subset, and export data. The workshop occurred on 4 Aug. 2015 at the ASHS Annual Conference in New Orleans, LA, sponsored jointly by Computer Applications in Horticulture and Graduate Student working groups.

Methods

Workshop participants installed R version 3.1.3 from the R Project for Statistical Computing before attending and downloaded workshop materials (R Project, 2015; Susko and Brym, 2015). The R code developed and distributed for the workshop follows the presentation closely (Fig. 1). The code contained the same questions and examples as well as more detailed, step-by-step instructions. We provided green and red post-it notes for participants to put on their computers to indicate when exercises were complete or if help was needed. The post-it notes identified individuals that required one-on-one attention and ultimately ensured milestone concepts were reached as a group.

Results

Our workshop had ≈25 attendees, the vast majority of which were new to R computing. We first gave a brief background on R, including the software's open-source nature and introduced the large and

This paper was part of the workshop "Online Learning and Big Data in Horticulture: New Insights and Directions" presented jointly by the COMP and GRAD working groups at the 2015 ASHS Annual Conference in New Orleans, LA, on 4 Aug. 2015.

[1]Department of Horticultural Science, University of Minnesota, St. Paul, MN

[2]Department of Biology, Utah State University, Logan, UT

[3]Corresponding author. E-mail: susko004@umn.edu.

doi: 10.21273/HORTTECH03339-16

active R user community that provides programming development and troubleshooting resources. We emphasized ideas universal to programming workflow and then translated those concepts into R code (Fig. 2). This problem-solving strategy of thinking what you want the computer to do first and then translating that workflow into R is an important step for beginners. Using the introductory code as a platform, we discussed the basics of the R computing language, including data structures, syntax, and troubleshooting strategies that are critical for performing statistical analysis. The basic material emphasized best practices for scientific computing, namely writing code for people to interpret rather than for computers to execute (Wilson et al., 2014).

Throughout the workshop, we posed interactive programming questions to participants (Table 1). The style of questions was a mix of conceptual and computational content. Conceptual questions asked participants to explain what a piece of code was doing, whereas computational questions asked participants to write code to practice syntax or solve a problem. The most complicated example constituted asking participants to subset their dataset into different groups based on the values of one variable to expedite a process commonly prone to errors and slow execution when performed manually. Despite the limited amount of time available for the workshop, this capstone question exposed participants to the general workflow for working with and importing data, how to subset, and how to export results.

Participant feedback was largely positive, though many participants indicated that more time would have been beneficial for more examples to help reinforce concepts. For future workshops to introduce R programming, we recommend half- to 2-d sessions to allow for sufficient depth of materials and adequate time for practice applying the concepts learned. In addition, many open-source scientific programming resources are available for future use from Data Carpentry including: quick reference sheet (Teal et al., 2015b), self-guided student materials (Teal et al., 2015c), and teaching materials for instructors (Teal et al., 2015d).

```
#Now that you have a vector of numbers, let's do something with them.  All of
them, fast!  We can perform operations on many different numbers using a for loop.
 A for loop simply gives a condition to do something, and does it until the
condition is no longer satisfied.

#In R, we use the letter "i" to denote the ith element (number in our case) in a
vector

#Let's print everything in our vector x using a for loop.

for (i in x){
  print(i)
}

#Notice the curly brackets and indented text within the loop.  These are necessary
for success

#!!!Question!!! Can you print i times 2?

#Modify the print statement function, so that i is multiplied by 2.
```

Fig. 1. Example of the companion R computing language (University of Auckland, Auckland, New Zealand) code for the "for loops" section of the workshop. This R-code is available online (Susko and Brym, 2015).

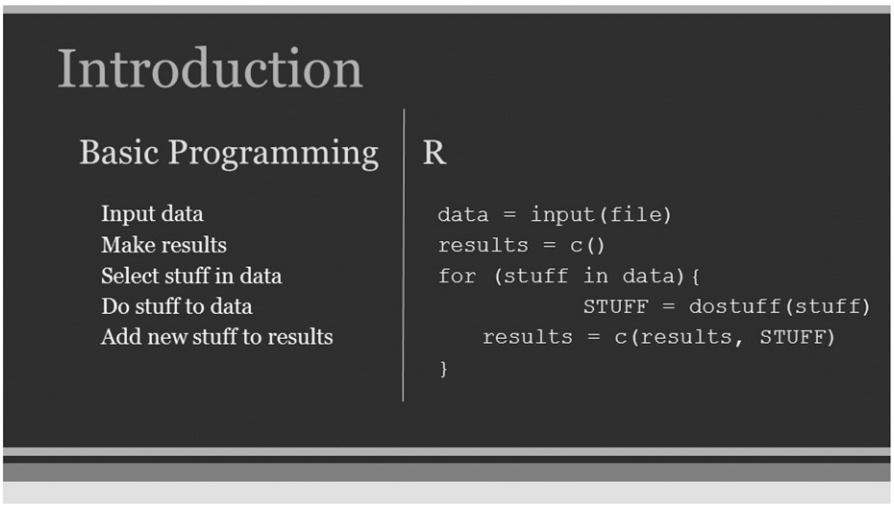

Fig. 2. Comparison of basic programming ideas in plain English and the R computing language (University of Auckland, Auckland, New Zealand). The broad goal for a data management problem is presented on the left, whereas the translation into R code is presented on the right. The full presentation is available online (Susko and Brym, 2015).

Discussion

Since the first published applications of personal computing to horticultural education, instructing proper computing techniques has become ever more important (White et al., 1990). As computing in the agricultural sciences becomes more central to analyzing experimental data, the need for education on proper programming techniques will continue to be essential. The growing size of horticultural datasets means that manually executed graphical interface software is quickly becoming less reproducible for large datasets and prone to user error associated with clicking and dragging. Additionally, the variety of data-driven research questions in horticulture may be better served by open-source software with active contributors and developers across multiple disciplines to provide novel insights on statistical problems. Open-source general programming software, such as R, improves both repeatability and flexibility of analyses through in-depth documentation and customization of programs for data analysis. Improving proficiency in techniques for computer programming and data management is an important contemporary challenge for science. All scientists should be able to produce well-documented data and repeatable analysis that can be easily shared and used for future research. As sharing and drawing meaningful conclusions from growing datasets becomes integral to the advancement of science, we hope this workshop

Table 1. Basic R computing language (University of Auckland, Auckland, New Zealand) programming prompts and acceptable answers posed during the workshop presentation.

R programming prompt posed to participants	Acceptable answer

Can you print i multiplied by 2 using a "for" loop?	for (i in x){
	print(i*2)
	}

In assigning "for" loop output to a variable, why do you only see one number?	Each time an output is produced, it is written to the same variable. This overwrites the previous output

What if you wanted to perform an operation on values in "x" that are greater than 3? Greater than or equal to 3? Not equal to 3?	#Greater than 3
	if (x[i]>3){
	y[i] <- (i*3)
	} else {
	y[i] <- (i*2)
	}
	#Greater than or equal to 3
	if (x[i]>=3){
	y[i] <- (i*3)
	} else {
	y[i] <- (i*2)
	}
	#not equal to 3
	if (x[i]=!3){
	y[i] <- (i*3)
	} else {
	y[i] <- (i*2)
	}

What differences do you see between the three versions of "elfearData" you imported?	When header = FALSE, the text header line becomes line 1. When sep = ',' columns aren't separated

presents resources and facilitates dialog for the future of computing in horticulture.

Literature cited

Bates, D., M. Mächler, B. Bolker, and S. Walker. 2015. Fitting linear mixed-effects models using lme4. J. Stat. Softw. 67:1–48.

Howe, D., M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D.P. Hill, R. Kania, M. Schaeffer, S. St Pierre, and S. Twigger. 2008. Big data: The future of biocuration. Nature 455:47–50.

Huber, W., V.J. Carey, R. Gentleman, S. Anders, M. Carlson, B.S. Carvalho, H.C. Bravo, S. Davis, L. Gatto, T. Girke, and R. Gottardo. 2015. Orchestrating high-throughput genomic analysis with Bioconductor. Nat. Methods 12:115–121.

Ihaka, R. and R. Gentleman. 1996. R: A language for data analysis and graphics. J. Comput. Graph. Stat. 5:299–314.

Indeed.com. 2016. R statistics, SAS statistics, and SPSS statistics job trends. 1 Aug. 2015. <http://www.indeed.com/jobtrends/Sas%2CR%2CSPSS.html>.

R Project. 2015. The R Project for Statistical Computing. 4 Aug. 2015. <https://www.r-project.org/>.

Susko, A.Q. and Z.T. Brym. 2015. 2015 ASHS computing workshop materials. 4 Aug. 2015. <https://figshare.com/articles/2015_ASHS_Computing_Workshop_Materials/2068122>.

Teal, T.K., K.A. Cranston, H. Lapp, E. White, G. Wilson, K. Ram, and A. Pawlik. 2015a. Data carpentry: Workshops to increase data literacy for researchers. Intl. J. Digital Curation 10:135–143.

Teal, T.K., K.A. Cranston, H. Lapp, E. White, G. Wilson, K. Ram, and A. Pawlik. 2015b. Quick reference sheet. 4 Aug. 2015. <http://www.datacarpentry.org/semester-biology/materials/r-intro/>.

Teal, T.K., K.A. Cranston, H. Lapp, E. White, G. Wilson, K. Ram, and A. Pawlik. 2015c. Self-guided student materials. 4 Aug. 2015. <http://www.datacarpentry.org/semester-biology/START-for-self-guided-students/>.

Teal, T.K., K.A. Cranston, H. Lapp, E. White, G. Wilson, K. Ram, and A. Pawlik. 2015d. Teaching materials for instructors. 4 Aug. 2015. <http://www.datacarpentry.org/lessons/>.

White, J.W., D.J. Beattie, and P. Kubek. 1990. Inquiry learning with videodiscs and computers: An innovative teaching method for horticulture courses. HortScience 25:385–388.

Wickham, H. 2009. ggplot2: Elegant graphics for data analysis. 1st ed. Springer, Berlin, Germany.

Wickham, H. 2015. dplyr: A grammar or data manipulation. 4 Aug. 2015. <https://cran.r-project.org/web/packages/dplyr/dplyr.pdf>.

Wilson, G., D.A. Aruliah, C.T. Brown, N.P. Chue Hong, M. Davis, R.T. Guy, S.H.D. Haddock, K.D. Huff, I.M. Mitchell, M.D. Plumbley, and B. Waugh. 2014. Best practices for scientific computing. PLoS Biol. 12:e1001745.