A study of various methods to teach R programming

Specific Aims

This study seeks to assess the impact of teaching programming in R under two paradigms, one that teaches the **tidyverse** suite of packages first, henceforth referred to as "tidyverse first", and one that teaches the **base** R conventions first, henceforth referred to as "base first". Our specific goals of the study are to:

(1) Establish whether there is a relationship between the method of teaching ("tidyverse first" or "base first") and participant engagement with the material.
(2) Establish whether there is a relationship between the method of teaching ("tidyverse first" or "base first") and participant outcomes, as established by a set of assessments given before, during, and after the intervention.
(3) Establish whether there are subgroups of participants that benefit more from a specific teaching paradigm.

Background and Rationale

The programming language R was developed by Robert Gentleman and Ross Ihaka in the early 1990's at University of Auckland, New Zealand (Ihaka and Gentleman 1996). It is based on the programming language S, developed by John Chambers, Rick Becker, and Allan Wilks at Bell Laboratories (Becker and Chambers 1984). The programming language R is currently ubiquitous in the statistical research and education communities. The "tidyverse" is a suite of R packages created to help with common statistics and data science tasks that follow a consistent philosophy ("Welcome to the Tidyverse" 2019; Wickham 2017). As of tidyverse 1.2.0, the core packages include ggplot2 (Wickham 2016), dplyr (Wickham et al. 2018), tidyr (Wickham and Henry 2018b), readr (Wickham and Hester 2018), purrr (Wickham and Henry 2018a), tibble (Müller and Wickham 2018), stringr (Wickham 2018b), and forcats (Wickham 2018a). Because these packages follow a consistent philosophy and were created with the user in mind, it has been speculated that it may be easier for beginners learning R to learn this suite of packages first, before learning the underlying systems of "base R". This hypothesis, while written about in informal avenues (Robinson 2017, 2014), has not been formally tested. While there have been a few guides with suggestions on how to teach R programming published (Venables et al. 2009; Baumer, Kaplan, and Horton 2017; Çetinkaya-Rundel and Rundel 2018; Kaplan 2018; Eglen 2009; Langan and Wade 2016; Peterlin 2009), and a single randomized study that examines the impact of using certain paradigms to solve specific problems (Rafalski et al. 2019), no studies to date have studied the impact of the order that the specific methods are taught.

Study Design

We have created two sets of online modules to teach basic skills in R programming. The online modules are intended to be completed outside the classroom. Each module is designed to take approximately 15 minutes, with total completion possible in approximately 1.5 hours. Participants will be randomized upon consent. Participants randomized to the "tidyverse first" set of modules will follow the following trajectory:

1. Introduction to tibbles and column types using the **tidyverse**
2. Reading in data using the **tidyverse**
3. Introduction to data manipulation techniques using the **tidyverse**
4. Introduction to vectors, assignment, lists, and dataframes in **base** R
5. Reading in data using **base** R
6. Introduction to data manipulation techniques using **base** R

Participants randomized to the "base first" set of modules will follow the following trajectory:

1. Introduction to vectors, assignment, lists, and dataframes in **base** R
2. Reading in data using **base** R
3. Introduction to data manipulation techniques using **base** R
4. Introduction to tibbles and column types using the **tidyverse**
5. Reading in data using the **tidyverse**
6. Introduction to data manipulation techniques using the **tidyverse**

In both cases, an assessment will be given prior to the first module, after the 3rd module, and after the 6th module.

The modules include teaching material as well as interactive coding exercises.

Participants

We hope to recruit approximately 100 participants. Initial recruitment will take place in Summer 2020. We will recruit participants via social media and email. Participation in the study is optional; there will not be class credit associated.

Engagement

To address Aim (1) we will observe the participant's engagement with the material. We will do this by observing factors such as the responses to attempted exercises in each module and time spent in each module.

Assessments

The assessments to address Aim (2) and Aim (3) will include two parts: (1) evaluations of the learning objectives
(2) the participants' perception of their learning / potential for growth.

We will evaluate the participants' learning by observing:

- the code used in the attempted exercises embedded in each module
- performance on assessment items written to target specific learning objectives

In addition to evaluating whether the learning objectives are met, we will also evaluate the participant's perception of their learning / growth potential by asking how competent they feel in the language (R), how likely do they think it is that they will master the language eventually, and how much they are enjoying the learning process.

Variables collected

In addition to the assessments completed, which include questions to assess the learning objectives as well as questions to assess the participant's perception of their learning / growth potential, we will collect demographics including:

- age
- gender
- education attainment
- first language

- primary language (reading)
- previous programming experience (general)
- previous programming experience (R specific)
- previous data experience

Timeline

Pending approval, the initial recruitment email to departments will begin in May 2020. The modules will go live in May 2020. For this first phase, data will be collected from May 2020 through December 2020.

Funding

We have received seed funding from Wake Forest University's Center for the Advancement of Learning. The funding will be used for incentives.

Budget

Item	Cost	Total
Shiny Server (Standard)	$99 / month for 4 months	$ 396
Domain	$144 for the year	$144
Incentives	$800	$800
Total		**$1340**

References

Baumer, Benjamin S, Daniel T Kaplan, and Nicholas J Horton. 2017. *Modern Data Science with R*. Chapman; Hall/CRC.

Becker, Richard A, and John M Chambers. 1984. *S: An Interactive Environment for Data Analysis and Graphics*. CRC Press.

Çetinkaya-Rundel, Mine, and Colin Rundel. 2018. "Infrastructure and Tools for Teaching Computing Throughout the Statistical Curriculum." *The American Statistician* 72 (1). Taylor & Francis: 58–65.

Eglen, Stephen J. 2009. "A Quick Guide to Teaching R Programming to Computational Biology Students." *PLoS Computational Biology* 5 (8). Public Library of Science: e1000482.

Ihaka, Ross, and Robert Gentleman. 1996. "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics* 5 (3). Taylor & Francis Group: 299–314.

Kaplan, Daniel. 2018. "Teaching Stats for Data Science." *The American Statistician* 72 (1). Taylor & Francis: 89–96.

Langan, Dean, and Angie Wade. 2016. "Guidance for Teaching R Programming to Non-Statisticians." In *Http://Iase-Web. Org/Documents/Anzcots/Ozcots_2016_Proceedings*, 9:141–46. Statistical Society of Australia Inc.(SSAI).

Müller, Kirill, and Hadley Wickham. 2018. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Peterlin, Primoz. 2009. "Using R for Data Analysis and Graphing in an Introductory Physics Laboratory."

Rafalski, Timothy, P Merlin Uesbeck, Cristina Panks-Meloney, Patrick Daleiden, William Allee, Amelia Mcnamara, and Andreas Stefik. 2019. "A Randomized Controlled Trial on the Wild Wild West of Scientific

Computing with Student Learners." In *Proceedings of the 2019 Acm Conference on International Computing Education Research*, 239–47. ACM.

Robinson, David. 2014. "Don't Teach Built-in Plotting to Beginners (Teach Ggplot2)." *Variance Explained*. http://varianceexplained.org/r/teach_ggplot2_to_beginners/.

———. 2017. "Teach Tidyverse to Beginners." *Variance Explained*. varianceexplained.org/r/teach-tidyverse.

Venables, William N, David M Smith, R Development Core Team, and others. 2009. "An Introduction to R." Network Theory Limited.

"Welcome to the Tidyverse." 2019. *Tidyverse*. RStudio. https://tidyverse.tidyverse.org/articles/paper.html.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer.

———. 2017. *Tidyverse: Easily Install and Load the 'Tidyverse'*. https://CRAN.R-project.org/package=tidyverse.

———. 2018a. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

———. 2018b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Romain Francois, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2018a. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

———. 2018b. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, and Jim Hester. 2018. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.