Author/s:
Finch, S;Gordon, I;Patrick, C

Title:
Taking the aRghhhh out of teaching statistics with R: Using R Markdown

Date:
2021-07-01

Citation:
Finch, S., Gordon, I. & Patrick, C. (2021). Taking the aRghhhh out of teaching statistics with R: Using R Markdown. Teaching Statistics, 43 (S1), pp.S143-S147. https://doi.org/10.1111/test.12251.

Finch Sue (Orcid ID: 0000-0003-4261-0504)

Taking the aRghhhh out of teaching statistics with R: Using R Markdown

Sue Finch[1], Ian Gordon[1] and Cameron Patrick[1]

[1] Statistical Consulting Centre, University of Melbourne

Correspondence

Sue Finch, Statistical Consulting Centre, University of Melbourne, Parkville, Australia 3010
email: sfinch@unimelb.edu.au

This is the author manuscript accepted for publication and has undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/test.12251

Taking the aRghhhh out of teaching statistics with R: Using R Markdown

Sue Finch[1], Ian Gordon[1] and Cameron Patrick[1]

[1] Statistical Consulting Centre, University of Melbourne

Correspondence

Sue Finch, Statistical Consulting Centre, University of Melbourne, Parkville, Australia 3010
email: sfinch@unimelb.edu.au

1 | CHOOSING SOFTWARE FOR TEACHING STATISTICS

Instructors have a potentially wide range of choices of statistical software for teaching statistics at the tertiary level. The growth of open source code-based options has raised debate about what kind of software is most suitable. As Chance et al[1] noted:

> *"Despite the endless capabilities that technology offers, instructors should be careful about using sophisticated software packages that may result in the students spending more time learning to use the software than applying it."*

Technological tools have the potential to both enhance and impede student learning.

R is a popular code-based statistical computing environment[2, 3] which provides access to a very wide range of statistical methods from the simplest and most traditional to new innovations as they emerge. Users need to develop skills in writing code and develop knowledge of appropriate functions and packages to call to carry out data wrangling, analysis and visualisation. Typically, users write code in a console window, often using RStudio[4] an integrated development environment which provides a structured environment for managing the workflow.

Many popular commercial statistical software packages are menu based. In learning to use such software, students need to learn how to access appropriate menus for analytic work and to choose from a range of options usually offered within a menu. Analytic options are thereby usually more transparent when using menu-based software.

In debating the merits of two software packages, R and Minitab® Statistical Software[5], Gunn and Morphett[6] discussed the concept of pedagogical affordances as it relates to learning statistics:

> *"Pedagogical affordances ... are what the technology (potentially) offers in terms of learning. ... affordances need to be both transparent and accessible to the user: a software package may afford sophisticated computation, but only for a user who can perceive how it may be used."*

Pedagogical affordances, hereafter referred to as learning opportunities, are important to the quality of learning[7]. Gunn and Morphett argued that the code-based statistical software offered fewer opportunities and involved a greater cognitive load in both obtaining statistical results and in interpreting the output. They echoed Chance et al's point about the risks in the trade-off of computational (coding) work and statistical thinking.

In this paper, we describe a case study of teaching an introductory statistics subject with R after many years of teaching with a range of different menu-based software packages. We discuss how we dealt with the serious concerns raised above.

2 | AN INTRODUCTORY TERTIARY LEVEL STATISTICS SUBJECT

Statistics for Research Workers (hereafter SRW) is an introductory statistics subject taken by graduate students and applied users from inside and outside the tertiary sector. The content is both theoretical and applied, and ranges from descriptive statistics to simple linear models. Importantly, SRW is taught intensively – a full semester subject, usually taught in 3 weeks or less and often in 6 to 8 full days. In various guises, SRW has been taught for a few decades, with appropriate developments and innovations, sometimes 3 times per year. The students vary substantially in their mathematical and statistical background, and in their experience with statistical software. A software package suitable for this subject is one which students will be able to use for their own applied analysis.

SRW has been taught using

- Minitab
- IBM SPSS[8]
- R with R Commander[9] which provides a GUI
- R with R Studio, a programming environment, and R Markdown[10] (which is a file format for creating dynamic documents).

An R Markdown document provides code for specifying the type and style of document and includes chunks of R code to generate statistical output including graphs.

Our preference historically has been to use Minitab. It was developed by statisticians at Penn State University (USA) with statistical pedagogy in mind, and hence in general the menus are coherent, and the behaviour of the software is predictable. Over time, SRW was offered using SPSS and R due to demand; most recently demand for R has outstripped demand for SPSS.

IBM SPSS, a commercial product, and R, free software, have had widespread uptake in business and educational settings. However statistical training has not taken centre place in their design. As a result, there is a lack of consistency in the menus, coding or language used to organise and reference statistical concepts and methods. In our experience, this can be an impediment to learning.

Our first approach with R (in 2012) used the R Commander package because it offered a menu, and this was potentially a way of managing "risk" in teaching statistics using R, particularly in an intensive subject where a lot of time can be lost if things go wrong.

Using R Commander had several drawbacks. It was an atypical way of using R, although it did provide an opportunity to gently introduce coding to students. Learning some coding was required, as R Commander did not include all the functionality needed, and the graphs did not meet the standards of good statistical practice[11]. Students were provided with improved script templates for graphs; in 2016 the R package ggplot2[12] was introduced. For the increasing fraction of students familiar with R Studio, the use of R Commander felt like a step backwards. So, in late 2019, we taught SRW for the first time using RStudio and R Markdown, relinquishing the use of a user-driven menu.

3 | DESIGN CONSIDERATIONS WHEN TEACHING WITH R

Our need to prioritise statistical learning while supporting the development of coding skills raised questions about how to structure student learning activities, and to consider features of R and RStudio that could be exploited to emphasize good analytic practice and statistical thinking. We considered what might be lost without a statistical menu structure, and what important features were present in (well-designed) menus. The re-design included development of generic R scripts and the revision of how exercises and answers were provided to students.

An important feature in the design of generic R scripts was the use of structure and language consistent with the statistical content of the subject. The generic scripts were organised in a folder structure to reflect a meaningful pedagogical structure, similar to a well-designed menu (Figure 1). The R code was curated, and instructions and guidance were included in the scripts. The generic code was written in a way that signalled the structure of variables and data (Figure 1), and the scripts provided modelled good analysis – for example, code for visualisations was included with analysis.

Figure 1 includes code based on the R tidyverse collection of R packages[13]. In providing this type of curated code, the goal was not to introduce students to the full functionality of the packages in tidyverse, but rather to gradually start with relevant aspects of the functionality. In this case, students needed an explanation of the use of 'piping' – the operator '%>%' first developed in the maggritr package[14]. Our observations were that they coped with this well.

Figure 1: Example of menu structure and code snippet

Over the course of the subject, students were also provided with reference to the range of cheatsheets available for R packages. These supplemented the generic scripts

and allowed students to pursue development of coding skills as time and inclination allowed.

Students completed practice exercises in a computer laboratory with individual access to a PC with a standardised software setup. This ensured they were working with the same, up-to-date version of R and the required packages.

The practice exercises in SRW are designed to develop statistical thinking about the analytic process in context and require both the analytic results and an applied interpretation. Traditionally students had developed their own practices for preserving the output and recording results when carrying out the practice exercises. In re-design of the provision of the practice exercises we aimed to preserve the statistical pedagogic aims by providing a framework that encouraged integration of analytic output with interpretation. In addition, we aimed to make preservation of individuals' analytic work and extended responses systematic and relatively seamless, to establish practices to support reproducible research, and to reduce coding load where possible.

We provided the exercises in R Markdown files that provided code to load the relevant packages and data, so that to answer an exercise, students needed to add code for analytics within code chunks and add their own text responses. Figure 2 provides an example of (part of) an exercise. Students could step through code in a transparent way, see results immediately below the code, and focus on the coding needed for analysis. At the same time, steps for setting up the framework for analysis were modelled. Students would need to identify the relevant code from the generic scripts and edit it to suit the data context and question. When a user runs a function to "knit"[15] an R Markdown file, a document is generated based on the code provided; html documents were generated in this case. (Figure 3 shows the result of a "knitted" document.) Students could knit the exercise if they preferred to review the requirements as a document without code. Students were not required to submit their answers to practice exercises but were encouraged to systematically preserve them as html documents.

````
---
title: "Exercise"
output:
  html_document:
    theme: yeti
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, fig.height = 2)
```

```{r, include=FALSE}
library(tufte)
library(ggplot2)
library(signmedian.test)
library(summarytools)
```

An investigator wished to determine whether epinephrine (adrenaline) has the effect of
elevating plasma cholesterol levels in humans.  Twelve adult males were selected and given both
a placebo and the drug.  Blood samples were taken following injection of the placebo and again
after injection of epinephrine. Analysis of the blood samples resulted in the  data epin.Rdata.

<!-- The next code chunk will read in the data file from your working directory. -->
```{r, echo=TRUE}
load("epin.Rdata")
```

(a) Are these samples paired or independent?  Explain.

EDIT THIS TEXT TO WRITE YOUR ANSWER.

(b) Produce an appropriate visual display (or displays) of the data. What do you conclude from
your display(s)?

<!-- For paired data it is desirable to plot the differences, so we need to derive them
first. Write the code in the chunk below. -->

```{r, echo=TRUE}
```

<!-- Choose an appropriate visual display. -->

```{r, echo=TRUE}
```

EDIT THIS TEXT TO WRITE YOUR ANSWER.
````

The data are loaded for the students.

Students edit the text to add their answer.

Code is added in the chunk to find the differences.

Code is added to produce a visualisation.

Figure 2: Example of part of a practice exercise

The design also included model answers in R Markdown files, which modelled applied context-based analysis and interpretation. In the answers provided, the functionality of R Markdown was extended as students progressed. An example of a "knitted" model answer to the (part) exercise in Figure 2 is shown in Figure 3; this is an html document.

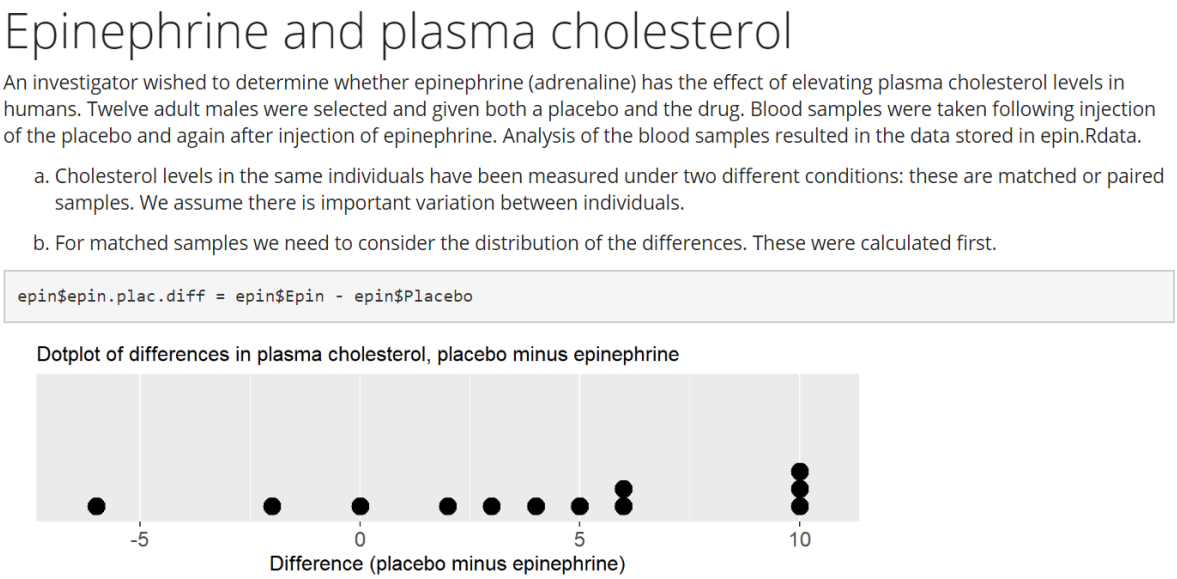


Figure 3: Part of an html document generated as a model answer for the part of the practice exercise in Figure 2

4 | FEEDBACK

Given the redesign, we collected feedback from students three days (half-way in this case) into the subject. In particular, we sought open-ended feedback to allow us to respond to concerns that could be addressed immediately or in future revisions of the subject. The focus was on evaluating the use and features of R with R Markdown.

In total, 36 of the 44 participants responded. Of these responding, 60% had no experience with R and all but one of the remainder had used R "a bit". The vast majority (89%) agreed that R Markdown was a useful aid to their learning. Students also provided open-ended feedback on what worked well and what needed improving; most of the comments were very positive. Their experience of R Markdown included the extra scaffolding we have described, which – we believe – may have led to more favourable feedback than from the use of R Markdown without it.

In their comments, students referred to several different learning opportunities they perceived in the design of the learning activities in SRW, including the use of R Markdown. This included *ease of use and transparency*; for example; "I love it. The output is nicely formatted and easy to read." It provided *structure to problem solving*; for example: "A single flow of syntax and answers, unlike SPSS which operates in 2 [windows]." *Reproducibility was supported*; for example: "Useful way of retaining answers and scripts". There were two aspects in which learning support was described: *supporting learning coding* (e.g. "You feel like you are programming), and; *supporting statistical learning* (e.g. "Helped put theory into practice"). There were also two opportunities related to practical application. These were *supporting integration of analysis and explanation* (e.g. "enjoyed how text/instructions and code were

integrated"), and *useful in applied practice* (e.g. "Likely what we would use when actually using R – so it feels practical").

Very few comments were provided in response to the question about what needed improving. Students indicated that they needed additional support material initially, the opportunity to develop more coding skills and to find a balance between the amount of direction provided as they progressed to allow development of independent coding skills. Future refinements to consider are to scaffolding coding skills as the subject progresses and extending the functionality of R Markdown.

5 | CONCLUSION

The decision to teach SRW with R, without a menu to guide students' analytic choices, was made with an eye to the risk of the coding load substantially detracting from the primary focus of learning. There was, however, a generally positive response from students, and unprompted identification of a range of learning opportunities in their open comments.

A key feature of our approach was to mindfully overlay the structure and language of statistics in the support and teaching and learning materials that related to the software. It remains a limitation of R that, at times, the output provided is sub-optimal; for example, an independent samples *t*-test result and associated confidence interval can be obtained without an estimate of the mean difference. Structured generic scripts can assist with alerting students to deficiencies in standard output and signalling ways to deal with them.

Also important to our approach was the intent to mimic genuine analytic practices in the use of software, while lightening the load of aspects of standard file and package management. We avoided teaching all the intricacies of using R but provided scaffolding for students to develop their skills and understand principles with suitable support material and modelling in code. We curated resources on learning R coding for those interested in developing their skills after SRW, and have subsequently developed a short course on using R and R Markdown for reproducible research.

In this case study, the student response is cause for optimism. If students themselves can readily identify how the design and use of the software supports their learning, the battle may be half won.

REFERENCES

1 B. Chance, D. Ben-Zvi, J. Garfield, E. Medina. The role of technology in improving student learning in statistics. Technology Innovations in Statistics Education, 1 (2007), 1-26.

2 R. Ihaka, R. Gentleman. R: a language for data analysis and graphics. Journal of computational and graphical statistics, 5(1996), 299-314.

3 R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL https://www.R-project.org/

4 RStudio Team. RStudio: Integrated Development for R. RStudio, PBC, Boston, MA, 2020. URL http://www.rstudio.com/

5 Minitab 19 Statistical Software. [Computer software]. State College, PA: Minitab, Inc., 2019. (www.minitab.com)

6 S. Gunn, A. Morphett. To R or not to R – What should we be considering? In H. MacGillivray, M. Martin and B. Phillips (Eds.). Proceedings of the Ninth Australian Conference on Teaching Statistics, December 2016, Canberra, Australia.

7 P. Ihanainen. Affordances of pedagogy. In J.W. Moravec (Ed.). Emerging education futures: Experiences and visions from the field. Education Futures, 2019, pp.163-176.

8 IBM Corp. Released 2017. IBM SPSS Statistics for Windows, Version 25.0. Armonk, NY: IBM Corp.

9 J. Fox, M. Bouchet-Valat. Rcmdr: R Commander. R package version 2.7-1, 2020. https://socialsciences.mcmaster.ca/jfox/Misc/Rcmdr/

10 J. Allaire, Y. Xie, J. McPherson, J. Luraschi, K. Ushey, A. Atkins, H. Wickham, J. Cheng, W. Chang, R. Iannone. rmarkdown: Dynamic Documents for R. R package version 2.4, 2020. https://github.com/rstudio/rmarkdown.

11 I. Gordon, S. Finch. Statistician heal thyself: have we lost the plot? Journal of Computational and Graphical Statistics, 24(2015), 1210-1229.

12 H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

13 H. Wickham et al. Welcome to the tidyverse. Journal of Open Source Software, 4(2019), 1686, https://doi.org/10.21105/joss.01686

14 S. Milton Bache, H. Wickham. magrittr: A Forward-Pipe Operator for R. R package version 2.0.1, 2020. https://CRAN.R-project.org/package=magrittr

15 X. Yihui. knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.29, 2020.

Menu

- 01 Descriptive statistics
- 02 Graphs
- 03 Distributions
- 04 Basic inference
- 05 Sample size determination
- 06 Linear models

Script snippet

```
############################################################
# DESCRIPTIVE STATISTICS FOR NUMERICAL VARIABLES BY GROUP
############################################################

# Mean, SD and n
MYDATA %>%
  group_by(CATEGORICAL_VARIABLE) %>%
  descr(NUMERICAL_VARIABLE, stats = c("n.valid", "mean", "sd"))

# Five number summary, including median and quartiles
MYDATA %>%
  group_by(CATEGORICAL_VARIABLE) %>%
  descr(NUMERICAL_VARIABLE, stats = c("n.valid", "min", "q1", "med", "q3", "max"))
```

TEST_12251_Figure 1.tif

```
---
title: "Exercise"
output:
  html_document:
    theme: yeti
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, fig.height = 2)
```
```{r, include=FALSE}
library(tufte)
library(ggplot2)
library(signmedian.test)
library(summarytools)
```

An investigator wished to determine whether epinephrine (adrenaline) has the effect of
elevating plasma cholesterol levels in humans. Twelve adult males were selected and given both
a placebo and the drug. Blood samples were taken following injection of the placebo and again
after injection of epinephrine. Analysis of the blood samples resulted in the data epin.Rdata.

<!-- The next code chunk will read in the data file from your working directory. -->
```{r, echo=TRUE}
load("epin.Rdata")
```

(a) Are these samples paired or independent? Explain.

EDIT THIS TEXT TO WRITE YOUR ANSWER.

(b) Produce an appropriate visual display (or displays) of the data. What do you conclude from
your display(s)?

<!-- For paired data it is desirable to plot the differences, so we need to derive them
first. Write the code in the chunk below. -->
```{r, echo=TRUE}

```

<!-- Choose an appropriate visual display. -->
```{r, echo=TRUE}

```

EDIT THIS TEXT TO WRITE YOUR ANSWER.
```

The data are loaded for the students.

Students edit the text to add their answer.

Code is added in the chunk to find the differences.

Code is added to produce a visualisation.

TEST_12251_Figure 2.tif

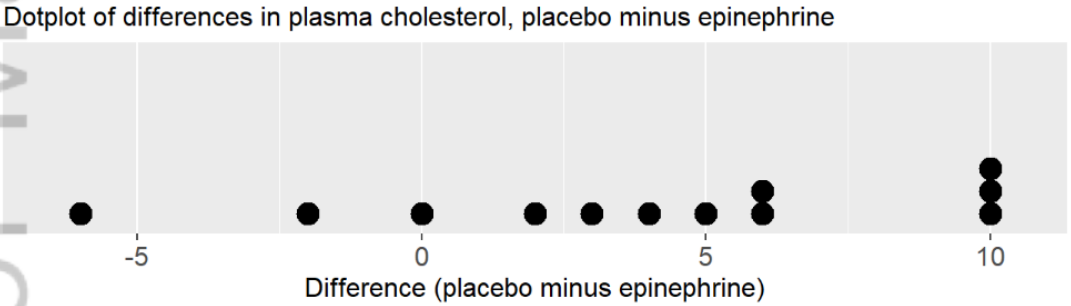
Epinephrine and plasma cholesterol

An investigator wished to determine whether epinephrine (adrenaline) has the effect of elevating plasma cholesterol levels in humans. Twelve adult males were selected and given both a placebo and the drug. Blood samples were taken following injection of the placebo and again after injection of epinephrine. Analysis of the blood samples resulted in the data stored in epin.Rdata.

a. Cholesterol levels in the same individuals have been measured under two different conditions: these are matched or paired samples. We assume there is important variation between individuals.

b. For matched samples we need to consider the distribution of the differences. These were calculated first.

```
epin$epin.plac.diff = epin$Epin - epin$Placebo
```



The dotplot above shows the distribution of the differences. In a small samples, drawing conclusions about the nature of the true underlying distribution is difficult. However, it can be seen that the majority of differences are positive, consistent with lower plamsa cholesterol for epinephrine.

TEST_12251_Figure 3.tif