USING R TO TEACH STATISTICS

Charles C. Taylor
Department of Statistics, University of Leeds, Leeds LS2 9JT, U.K.
charles@maths.leeds.ac.uk

Before 2003, the Department of Statistics at the University of Leeds exposed its Maths students to MINITAB, SPSS and SAS as they progressed from first to third year. As a result of a unanimous decision in a departmental meeting, these were all replaced with just R. This short contribution reflects on the ups and downs of the last decade, and will also cover attempts to teach (rather than merely illustrate) statistics using R, and a comparison of different versions in a small designed experiment.

BACKGROUND

In 2003 the Department of Statistics at the University of Leeds decided to teach exclusively using R. Previously we had used various statistical packages (MINITAB, SPSS, SAS and Splus) – and even MAPLE – but our students were graduating without much confidence in practical data analysis. Even though there were few job adverts which even mentioned R (or S/Splus) at the time, we felt that more ability (and confidence) to program in one language could provide the basis for easy conversion to others. Other points in favour included excellent graphics capabilities; ease of extension; ease of installation – on a variety of computing platforms. It was (and is) also free, which meant it could be easily installed onto personal equipment.

We also recognized some more negative aspects:

- no menu-driven interface (as MINITAB had) and so finding the function you need is not always easy
- less extensive import/export capabilities
- programming language requires some knowledge of syntax
- takes a while to learn, and feel confident in

and – despite more recent developments - many of these are still valid.

To illustrate some key features, we can consider a set of tasks which may be given to a first year class in probability and statistics. Simulate data from an exponential distribution; summarize the data numerically; draw a histogram, and experiment with different parameters. We can compare the suggestions to be given for MINITAB, with those for R.

Table 1: comparison between MINITAB and R

MINITAB	R
Simulate 100 values using the menus: Calc → Random Data → Exponential and store the result in column C1	You need to TYPE (ignore anything after the #): `x=rexp(100, 1)` # generates 100 values from an exponential distribution with mean 1, which are stored in vector (column) x
Calc → Column Statistics OR Stats → Basic → Statistics	`mean(x); sd(x); summary(x)` # give mean, standard deviation and other summaries of x
Graphs → Histogram (also consider Options)	`hist(x)` # plots histogram of data in vector x

During the use of the MINITAB menus the student will notice the other distributions (e.g. normal) when simulating random data, other options in the Column Statistics menu, and the Options available for the histogram are also quite intuitive. By contrast, after typing the R commands (including encountering unhelpful R statements after any mistyped characters, or spelling mistakes) the student has no indication of how to generate data from other distributions (for example, `rnorm()`, `runif()`, `rbinom()`, `rt()`), and little idea of what options there are in the `hist()` function. In order to make long-term progress, we have encouraged the students to get to grips with the help pages. The `example()` function is also recommended, but a first look at the

In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* Voorburg, The Netherlands: International Statistical Institute. iase-web.org

`help()` page for `hist` is quite daunting for the student new to R. The following table shows the first part of the help page.

Table 2: Help for histograms in R

```
hist            package:graphics            R Documentation

Histograms

Description:

    The generic function 'hist' computes a histogram of the given data
    values.  If 'plot=TRUE', the resulting object of 'class
    "histogram"' is plotted by 'plot.histogram', before it is
    returned.

Usage:

    hist(x, ...)

    ## Default S3 method:
    hist(x, breaks = "Sturges", freq = NULL, probability = !freq,
        include.lowest = TRUE, right = TRUE,
        density = NULL, angle = 45, col = NULL, border = NULL,
        main = paste("Histogram of" , xname),
        xlim = range(breaks), ylim = NULL,
        xlab = xname, ylab,
        axes = TRUE, plot = TRUE, labels = FALSE,
        nclass = NULL, ...)
.......
```

Student experiences of the change were generally favourable. Those who repeated the year, and so were able to compare R to MINITAB, had a preference for R. They were all pleased that R was freely available and easy to install. Keeping records, saving work and writing up (using WORD or LaTeX) also worked well. However, there was a tendency for staff to continue to (over) spoon-feed students, who in turn rarely took any initiatives (for example, to use R to check calculations for smaller problems they were doing "by hand"). Perhaps more importantly, there was little attempt to *co-ordinate* the teaching of R (so that we could build on material covered in the core modules), which resulted in a continuing lack of confidence for many of the graduating students.

AN EXPERIMENT TO TEACH STATISTICS WITH R

As part of the desire to co-ordinate teaching of R, a member of staff undertook a major project, in which an R library would be created to *teach statistics* (as well as R) through a standard R interface. This was intended to include interactive sessions, with automated marking of work, and optional extra information and examples for those who needed it. The first-year class (of about 300 students) were given fewer lectures and required to do more self/online-learning with the use of this package (`Rteacher`). A number of difficulties were encountered. Firstly, students had to install a package which was only available as a zipped file, also requiring many other R libraries to be loaded. The scale of the project was underestimated, resulting in beta-version code, with the students being the first to discover problems ("From time to time I will upload a newer version of the Rteacher package. I recommend you always use the latest version as it will contain bug-fixes.") Students were unfamiliar with and generally not enthusiastic about the self-learning approach, leading them to be less motivated. Finally, since the library was installed on each

computer, the answers to any (assessed) questions were also available (though only industrious students knew where to look).

To overcome this last problem the following year, the next version (Rteacher2) made use of Shiny (see examples at https://shiny.rstudio.com/gallery/), with everything running off a server in a web browser. The scale of the project was again underestimated, since many aspects had to be written from scratch. Due to lack of time, it was necessary to pay for a (shiny) server, and this then required issuing the class with a bespoke set of usernames and passwords. However, in this year, there were no installation problems, though for many connections the interface was too slow, and the latex2html processing of the lecture notes very cumbersome. The student's dislike of self-learning was not resolved, and the project never developed the supplementary material for students who needed to practice or learn more on a particular topic. On the plus side, tutors were grateful that the marking of homeworks was automated. Perhaps most crucially with regard to the desire to co-ordinate teaching in R – although students gained experience with R syntax, output and several commands – they never experienced any standard R interface; even leading some to believe they had used a program called "Rteacher". With poor student performance (and an associated student dislike of statistics) in both academic years, this project was dropped, but with several important (and painful) lessons learned.

TRIAL ON R VERSION

Apart from the standard (regular) program, there are now several R interfaces (RStudio, Rattle GUI, R Commander, Tinn-R, and RKWard) which I will refer to as versions. The most commonly used is RStudio, which is generally perceived as being a bit more user-friendly with predictive text, and a layout which includes a panel listing the structure of current objects. It was desired to make a comparison to ascertain which was preferred.

During the first lecture of a conversion-type Masters level module in Statistics, a class of 28 students were invited to take part in a trial on the *version* of R. It was explained that at any time the students could "drop out" of the trial, but that half of them would be allocated to one version of R (RStudio), and half to the other "regular version"'. Sixteen students (initially) volunteered to take part, and they were randomly assigned to one of two groups, with the students informed by email which version they should use.

In the practical sessions no distinction was taken between the two groups – indeed, it was not evident at this point which student was in the study, since all students were free to choose either version and no attempt was made to track of whether anyone dropped out. The outcomes were measured, for all students on the module, in

- the awarded mark on each of two pieces of assessed coursework
- the final exam mark
- an (optional) questionnaire.

The last of these was completed by 19 students, with no sure way of associating individuals with either of the two groups.

The simplest comparisons can be made in the performance of the students in the coursework and the final exam. These are shown in Figure 1, in which it appears that the more able students choose not to volunteer for the trial --- perhaps because they had used R (and already done some statistics) before, and did not want to be told which version to use. The boxplot also suggests that those using the regular version of R did slightly better in the assessed work than those using RStudio, but the sample sizes are too small to make confident conclusions. The scatterplot shows a moderate correlation ($r = 0.5$) between the assessed work and the exam mark, as would be expected.

The questionnaire invited free-text comments. Those which mentioned specific versions of R included:

- *Having used Rstudio before I did sometimes struggle with the regular version. With Rstudio if you know roughly the function you want to use, starting to type will bring up suggestions and also hovering shows syntax which is good. I think perhaps the thing I missed about Rstudio was being able to type just a few letters of a variable name then use the tab key*
- *Sometimes the Rstudio will collapse during the use. While the regular R will seldom come to this.*

- *I didnt like the regular one because after using studio version I get used to having my right list of variables and easily can view and access everything, while in the classic version its only one screen drive me crazy once error appeared.*
- *R studio is more friendly to the user.*
- *There was not anything in particular that I like about this [regular] version. I think it was unfortunate having used Rstudio before and being used to this. I think if I had started from fresh with the other I think I would have found it a bit easier to use.*
- *the R studio can show the structure of the data more clearly and you can write the script and run it in the same screen. I think it is convenient. But when you use plot function, i think the R is faster than Rstudio.*
- *I liked the R studio version because it was easier to visualise everything in one go (eg. tables without having to open them separately, or the help page open directly in R studio)*
- *I think it is better to use these two tools at the same.*

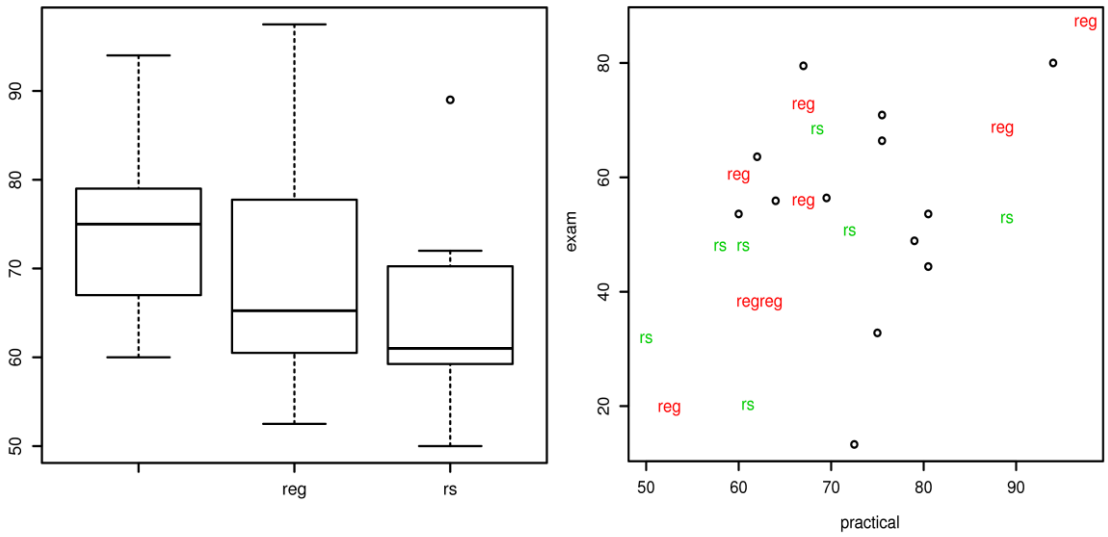

Figure 1: Left: boxplots of student coursework marks for those not in the trial (12 students), those allocated to the regular version of R (8 students) and those allocated to RStudio (8 students). Right: scatterplot of practical vs exam marks for each student, coded by type.

CONCLUSION

Based on rather limited experience, it seems that students prefer to start to learn R with RStudio, though there may be some stability issues for larger datasets. The most recent first-year experiments have consolidated the use of RStudio, together with the package `swirl`, which has posed few problems. This package focuses on learning R (rather than learning statistics), and allows an easy way for students to practice at their own speed, using a standard RStudio (or regular R) interface.

REFERENCES

Minitab 17 Statistical Software (2010). State College, PA: Minitab, Inc. http://www.minitab.com/.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. http://www.R-project.org/

RStudio, Inc (2013). *Shiny. Easy web applications in R*. http://www.rstudio.com/shiny/

RStudio Team (2015). *RStudio: Integrated Development for R*. RStudio, Inc., Boston, MA. http://www.rstudio.com/.