

CODE SHARING IN STATISTICS WORKSHOPS

Dean Langan and Angie Wade
UCL Great Ormond Street Institute of Child Health,
University College London, United Kingdom
d.langan@ucl.ac.uk

In statistical programming workshops, communication has a different dynamic to other classroom-based statistics courses. Many of the traditional methods of communication (e.g. verbal, via a projector or printed notes) are insufficient for communication of the programming language. For example, learners may need to describe their exercise solution or a programming error to the class. To address this problem, the Centre for Applied Statistics Courses (CASC) at University College London (UCL) developed a platform for code sharing in workshops for R and Stata programming. On this platform, learners are provided with a live version of the teachers' code and learners can submit sections of their own code for group discussion. We present anonymous feedback and web analytics to show how the platform is used in practice.

INTRODUCTION

Classroom-based courses, as opposed to online courses, can have better pedagogical outcomes and are able to cultivate a social community, allowing effective and immediate verbal communication (Zhang et al. 2004). This is not only a benefit of classroom-based courses in statistical theory, but also of applied statistical workshops held in a computer room. However, many statistical software packages are associated with their own programming language such as R (R Core Team 2017) or Stata (StataCorp 2017) and this can make effective communication in these workshops more complex. Programming code is best shared in written electronic form.

The Centre for Applied Statistics Courses (CASC) at University College London (United Kingdom) (www.ucl.ac.uk/stats-courses) provide introductory and advanced statistical software workshops, where teachers have experienced first-hand the complexity of teaching statistical programming languages. These workshops are primarily marketed to adult learners seeking professional development from a variety of fields. Learners often lack a strong background in statistics or programming, but require a basic knowledge for their research. These workshops include 'Introduction to R' for beginners and 'Further Topics in R' for intermediate and advanced learners. Additionally, an 'Introduction to Stata' workshop has more recently been developed (since October 2017). These workshops typically include (1) live demonstrations of programming via a projector screen and (2) intermittent practical exercises, and our courses have previously had communication issues in both areas:

Live Demonstrations

The learner can choose to observe demonstrations passively, or manually copy and run on their own computer. For example, during the workshops in R, the teacher will demonstrate by typing code manually in the software package *Rstudio* (Rstudio Team 2015) and running it so that learners can observe the output. Manual coding while teaching, Levy (2016) explains, is more effective because it forces the teacher to slow down to a more suitable pace and ensures they are flexible and responsive to questions the learners might have. Sharing pre-written code before the start of the course, although useful, does not allow for this flexibility. However, from anonymous online feedback, it emerged that some individuals found the pace too fast because they took an active approach to learning, choosing to copy and run the code manually on their own computer while failing to observe key parts of the demonstration. These learners may have benefited from better access to the teachers code. Learning is less efficient when attendees multitask (Ellis et al. 2010), but there is a common misconception that it increases productivity.

Exercises

Live demonstrations are typically followed by a practical exercise and, after allowing time for completion, the solutions are discussed as a group. This discussion relies on the learners effectively communicating their code to the teacher and the rest of class without the aid of a projector

screen. For example, the learner may describe an error they observed, or describe novel code they wrote when completing an exercise that the teacher may not have anticipated. While stimulating class discussion is important, it was often obstructive to the flow of the course. The teacher may have needed to approach the learner's computer to see their code directly and achieve a better understanding and this process this may have taken time. The teacher may then have difficulty communicating the learner's problem and solution with the rest of the class.

Given issues in these two areas, we considered whether there are any tools available online for code sharing to facilitate this type of communication. In the next section, we briefly review existing code sharing tools. By making use of these existing technologies, we developed our own platform. We present anonymous online feedback collected from learners and web analytics to show how the platform is used in practice. We also discuss ideas and teaching styles that may help utilise the platform and make recommendations for teachers of similar statistical programming workshops.

ONLINE TOOLS FOR CODE SHARING

A number of generic tools are available online for code sharing, including *firepad* (<https://firepad.io>) and *codeshare* (<http://codeshare.io>). These tools have primarily been designed for programmers to collaborate remotely in any programming language, by allowing multiple users to make edits to a shared text file online. These tools can be adapted for use in computer programming workshops. The limitation of this code sharing tool is that the teacher would ideally prefer to teach and write code within the statistical software package (e.g. *Rstudio*), and share their code live, not through a generic web platform.

Teachers could share their code using cloud-based storage such as *Dropbox* (<https://www.dropbox.com>) or *Google Drive* (<https://www.google.com/drive>). Each time the teacher saves their code locally, the file can be automatically synced online and learners have access through a web address. Cloud applications can be installed on the teacher's computer making the syncing process automatic. However, we use an add-on package available in R named *rdrop2* so that the platform can be updated from any machine without any additional installation (Ram & Yochum 2017). The shortfall of syncing code to *Dropbox* is that it's not strictly live, so learners may be viewing a slightly out-of-date version.

Another tool we found that could be adapted, specifically for R programming workshops, is *Rstudio Server* (Rstudio Team 2015). The teacher could conduct demonstrations from an online version of *Rstudio* on a remote server, allowing multiple users to view the software from their own computers with read-only or full access R code. As a side note, example data could also be shared live between teachers and learners via the server during the workshop. *Rstudio Server* is only available on the Linux platform at the time of writing, and there are no similar tools that exist for other common statistical software packages to our knowledge.

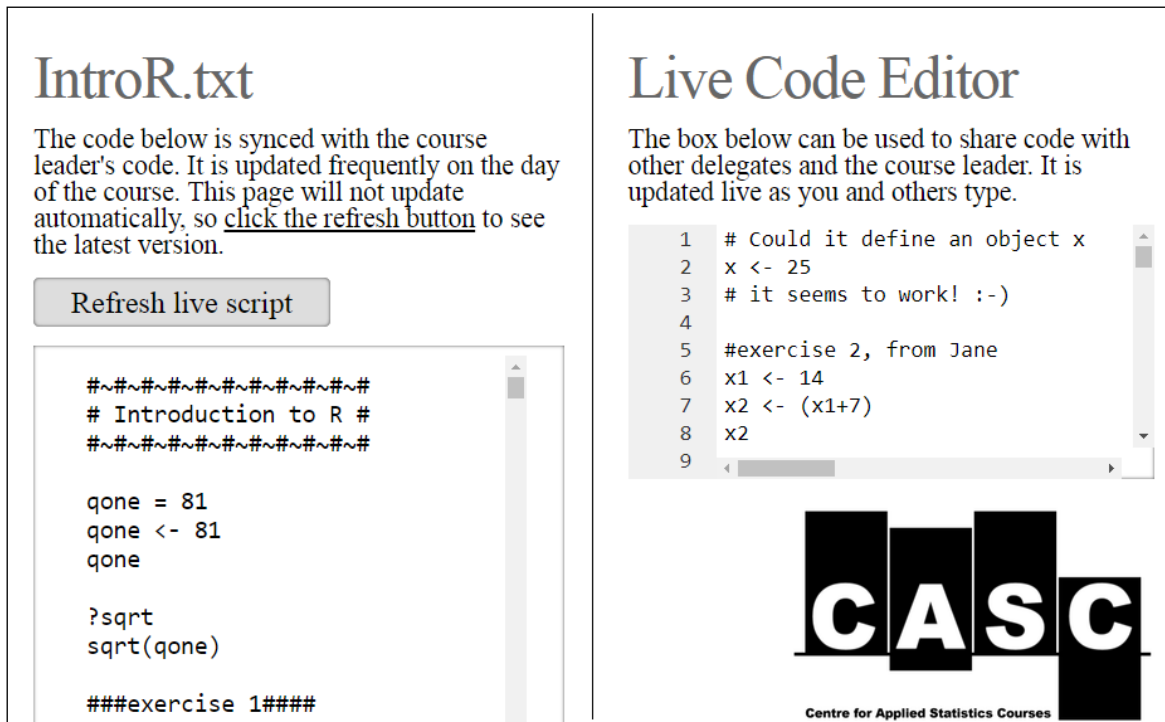
OUR PLATFORM

Individually, the tools above are not sufficient for bi-directional code sharing between both the teacher and learners. Therefore, we developed our own platform by combining two of the tools above. Our platform uses (1) *Dropbox* so that the teacher can share a read-only version of their code to the rest of the class and (2) *firepad* so that learners can submit code to the teacher and other learners in the workshop.

An illustration of the platform is given in figure 1, containing code from one of our 'Introduction to R' courses. On the left hand side (figure 1), learners can view a read-only version of the teacher's script file and the latest version can be viewed after the learner clicks the 'Refresh live script' button. On the right hand side, *firepad* is embedded into the platform, which is essentially a shared text editor so anyone with access can amend/correct code that others have submitted.

HOW OUR PLATFORM IS USED

At the beginning of a workshop, learners are introduced to the platform and asked to access it from their own computers using a provided web address. It is important they are familiar with the platform from the beginning and understand how it can benefit them during the course. The live code editor (on the right hand side, figure 1) can be confusing and daunting at first, so the learners are asked to try it out by simply typing their names.



IntroR.txt

The code below is synced with the course leader's code. It is updated frequently on the day of the course. This page will not update automatically, so [click the refresh button](#) to see the latest version.

Refresh live script

```

#####
# Introduction to R #
#####

qone = 81
qone <- 81
qone

?sqrt
sqrt(qone)

###exercise 1####

```

Live Code Editor

The box below can be used to share code with other delegates and the course leader. It is updated live as you and others type.

```

1 # Could it define an object x
2 x <- 25
3 # it seems to work! :-)
4
5 #exercise 2, from Jane
6 x1 <- 14
7 x2 <- (x1+7)
8 x2
9

```

CASc

Centre for Applied Statistics Courses

Figure 1. Online platform for sharing statistical programming code.

Note: the sizing of the windows have been changed for presentation in this paper.

The teacher's published code (on the left hand side, figure 1) has proved useful during demonstrations so that learners do not feel the need to manually type the code on their own computers, but rather are able to engage more passively with the demonstration if they prefer. This is particularly useful in more advanced programming workshops, where demonstrations involve more lengthy sections of code. The learner can to deviate from the material more efficiently and explore what might happen if they make small changes. Learners can also review code no longer be visible on the projector.

The platform has been particularly useful during practical exercises. For example, in our course 'Further Topics in R', learners are asked to sort a dataset in order of a variable after this has been demonstrated on a different example dataset. The teacher recommends adapting the teacher's code to complete this task. This reflects how a lot of R users program in practice; programming infrequently involves writing code manually from scratch and instead involves adapting similar code from a previous analysis or perhaps an example/tutorial found online. It must be stressed that manual coding is also a useful skill, and not all exercises need to adhere to this format.

During practical exercises, the teacher offers one-to-one help. Aside from the more obvious benefits of this approach, this allows the teacher to identify any common mistakes or interesting solutions that the teacher may not have anticipated that could be shared with the rest of the class. Towards the end of the exercise, learners are encouraged to submit their solution for group discussion through the live code editor (right hand side, figure 1). This provides learners with extrinsic motivation during exercises (Ormrod 2006), given they may get positive recognition of their work. The teacher can instigate a group discussion making use of the code submitted by one or more learners. This introduces an element of communal problem based learning to the course, which has grounds in communal constructivism (Savery & Duffy 1995). Typically without a platform for code sharing, the teacher would simply demonstrate their own pre-written solution.

EVALUATION AND FEEDBACK

We asked learners through an anonymous online feedback questionnaire two questions about the platform, with possible responses on a five point Likert scale. 20 responses were collated from

four workshops involving a total of approximately 80 learners (Aug-Nov 2017, 2x ‘Introduction to R’, 1x ‘Further Topics in R’ and 1x ‘Introduction to Stata’). Responses are presented in table 1. It is worth noting that even for those that did not find the platform useful, it is unlikely to have been detrimental to their learning since the platform is optional.

	1 (not very useful)	2	3	4	5 (very useful)
Was the platform helpful? (n=20)	0 (0%)	1 (5%)	6 (30%)	6 (30%)	7 (35%)
	1 (never)	2	3	4	5 (very often)
How often did you make use of the platform? (n=20)	1 (5%)	5 (25%)	6 (30%)	5 (25%)	3 (15%)

Table 1. Responses from an anonymous feedback questionnaire collated from four workshops.

In the same feedback questionnaire, learners were also asked for general comments about the workshop they attended and all three unprompted comments about the platform were: “the live feed was excellent”, “very useful live refresh aspect of the R script” and “the live streaming code was a great idea”.

We monitored platform usage during three different workshops, recording each time one of the 20 learners clicked the button ‘Refresh live script’ (Figure 2). In total, the page was refreshed 224 times in the ‘Introduction to R’, 426 times during ‘Further Topics in R’ and 219 times during ‘Introduction to Stata’. Results show the platform is used more during ‘Further Topics in R’, since it is more advanced and involves more complex code. The workshops began at 10.00am each day and finished at 4.30pm. Lunchtime was between 12.45pm and 1.45pm and breaks roughly between 11.15am-11.30am and 3.00pm-3.15pm; explaining why there was little usage during these periods.

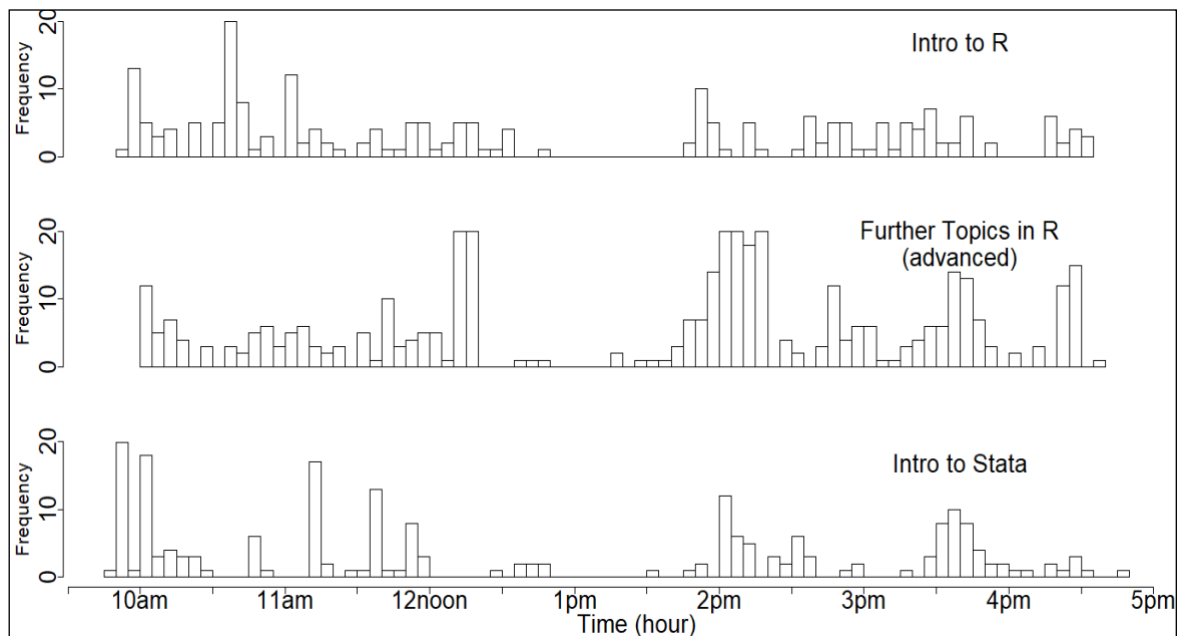


Figure 2. Web analytics from three workshops; Introduction to R, Further Topics in R (advanced) and Introduction to Stata.

Each bar/bin represents a period of ten minutes.

The refresh button was clicked frequently during the beginning of the course when learners are introduced to the platform. However, this perhaps doesn’t reflect real usage given that the teacher had not written/demonstration code at this stage. The platform was also used frequently during exercises. For example, during ‘Introduction to Stata’ (bottom graph), the exercises began at

11.30am, 2pm after lunch and 3.30pm. Within the advanced course, learners used the platform more during exercises towards the end for the day, where exercises involve lengthier solutions and learners have more to reflect on from previous sections. The platform in all three courses was frequently refreshed shortly after lunch (after 1.45pm), perhaps because learners wanted to review the material from the morning session or simply check they are viewing the latest version.

CONCLUSION

In this paper we introduced a platform for sharing code between the teacher and learners (and vice versa) in statistical programming courses. The platform has been implemented successfully into short-courses in R and Stata for adult learners as described in this paper. Feedback has so far been positive and proves the platform is frequently used, particularly during the more advanced course (Further Topics in R) and practical exercises.

We recognised that communication of a programming language in a typical class-room environment is limited, as code is best shared in written electronic form. Online courses do not suffer from these limitations in the same way, as they naturally allow for more appropriate written forms of communication such as email and forums. Those that advocate class-room based learning should recognise that communication is not always optimised simply by the teacher and learners being in the same room together.

The platform could also be used for statistical programming courses in other languages such as SAS and Python and may have applications with non-statistical programming languages too. For those interested in setting up a similar platform for their own course, more detailed instructions are available for download in pdf format via following link: tinyurl.com/ICOTS-Japan

REFERENCES

- Ellis, Y., Daniels, B., & Jauregui, A. (2010). The effect of multitasking on the grade performance of business students. *Research in Higher Education Journal*, 8, 1.
- Levy, M.A. (2016). Teaching R to 200 people in a week. Presented at the International R user conference, Stanford, CA. Retrieved from <http://environmentalpolicy.ucdavis.edu/node/394>
- Ormrod, J.E. (2006). *How Motivation Affects Learning and Behavior*. Pearson Allyn Bacon Prentice Hall, www.education.com/reference/article/motivation-affects-learning-behavior.
- R Core Team (2017). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. www.R-project.org
- Ram, K., & Yochum, C. (2017). rdrop2: Programmatic Interface to the 'Dropbox' API. R package version 0.8.1. <https://CRAN.R-project.org/package=rdrop2>
- RStudio Team (2015). *RStudio: Integrated Development for R*. RStudio, Inc., Boston, MA. www.rstudio.com
- Savery, J. R., & Duffy, T. M. (1995). Problem based learning: An instructional model and its constructivist framework. *Educational technology*, 35(5), 31-38.
- StataCorp (2017). *Stata Statistical Software: Release 15*. College Station, TX: StataCorp LLC
- Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker Jr, J. F. (2004). Can e-learning replace classroom learning? *Communications of the ACM*, 47(5), 75-79.