



Using R-Shiny to Teach Quantitative Research Methods

Brendan Halpin, Dept of Sociology, University of Limerick

QUANT-SOC-IE@LISTSERV.HEANET.IE online seminar, Nov 26 2020

What and why?

R-shiny for interactive pedagogical apps

- R-Shiny is an R library for writing interactive web pages
- Full access to the power of the R statistical programming language
- I have developed a small suite of R-Shiny tools for teaching quantitative research methods.
 - demonstrations of ideas
 - self-teaching exercises and
 - self-marking assessments

Slides at <http://teaching.sociology.ul.ie/shiny/quantsoc.pdf>

Easy?

- If you use R already, writing Shiny web pages is a relatively easy extension
- R very general and very powerful, so there are lots of possibilities but many require extensive programming
- But programming interactive web pages has important differences from conducting a data analysis
- Nonetheless, easy and quick to create simple and robust tools

Small examples

Small examples

- A very simple example

`http://teaching.sociology.ul.ie:3838/apps/normsd/`

- Uses Shiny input sliders to manipulate an R graph, showing a normal distribution.

Programming

- Programming is not within scope for today, but a quick illustration
- Two elements, a User-Interface function and a Server function
- Combined in a `shinyApp(ui, server)` function call

User interface

```
ui <- fluidPage(  
  headerPanel("Normal distribution: defined by mean and standard deviation"),  
  mainPanel(  
    plotOutput("normPlot"),  
    sliderInput("mean", "Mean:", value = 0.0, min = -10.0, max=10.0, step=0.1),  
    sliderInput("sd", "Standard Deviation:", value = 1.0, min = 0.0, max=10.0, step=0.1),  
    p("Use the sliders to vary the mean (this affects where the distribution is centred)  
      and the standard deviation (this affects how spread out it is)."))
```

```
server <- function(input, output) {  
  output$normPlot <- renderPlot({  
    x <- seq(-10, 10, length=1000)  
    y <- dnorm(x, input$mean, input$sd)  
    plot(x, y, type="l", xlab="X", ylab="Probability Density",  
         main="Normal Distribution", axes=TRUE, ylim=c(0.0, 0.4))  
  })  
}
```

More complex examples

More complex examples: leverage

- Leverage of outliers in linear regression:

<http://teaching.sociology.ul.ie:3838/influence/>

- Basic X/Y data set is stable (until you press "Refresh"), but the outlier reacts immediately to the sliders
- R draws the scatter plot, fits and draws the regression line, calculates Cook's distance

Making data and analyses available

- More recently, I've been using Shiny to make data or example analyses available to students who can't access labs
- Access to COVID-19 data:
`http://teaching.sociology.ul.ie:3838/so4087/ecdcplots`
- Deterministic SIR model:
`http://teaching.sociology.ul.ie:3838/so4087/sir/mod1`
- Agent-based SIR model:
`http://teaching.sociology.ul.ie:3838/apps/sirdsen`

Storing data

Even more complexity: storing data

- Persistent data storage is the next level
 - More power
 - More programming complexity
- A class-room exploration of the binomial distribution:
<http://teaching.sociology.ul.ie:3838/heads/>
 - Students toss a coin four times, enter the result
 - All submitted data is amalgamated and echoed back

Backend database

- Requires a database that is updated by, but is independent of, the invocation of the app
- Each student invokes their own instance, updating in parallel
- Full SQL database is possible but heavy duty
- Lightweight `sqlite3` program emulates a database but saves to a local file: more manageable
- A constraint of this particular app: database must be re-set by the instructor before each class

Coin toss in pedagogical context

- Pedagogically, the app pairs with <http://teaching.sociology.ul.ie:3838/binsim/>, which simulates the binomial distribution, but doesn't store data persistently.

Self-learning exercises

Self-learning exercises

- Present a problem or exercise with random values
- Check the correctness of the result the student calculates
- Give a worked example with the same values
- Repeat as often as the student wishes
- Example: read the table of the standard normal distribution
<http://teaching.sociology.ul.ie:3838/so5041/ass2/q1/>.

More challenging example

- An example that requires more work:
<http://teaching.sociology.ul.ie:3838/so5041/ass2/q5/>.
- Requires several calculations, ideally in a spreadsheet (previously covered in lab)

Considerations

- Self-learning exercises should present a pedagogically useful task
- Randomisation should be guaranteed not to throw up values that make the task meaningless or unduly difficult, but give a good range
- The worked example needs to accommodate variations in the solution
- E.g., in the SND example, different steps are required depending on what side of the mean X falls

Assessments

Assessments: storing and assessing individual data

- Assessments are like self-learning exercises
 - Unique values per student
 - Automatic marking and feedback
- Differences
 - Only one version of the question (with fixed random values per student)
 - Marking and feedback delayed, not immediate
 - Results are stored persistently

- Problems need to be equally difficult for all students
- Ensuring that the numbers are random within bounds
 - for a question on the normal distribution, for instance, set sensible bounds on the X value
 - for a t-test construct it such that the t-stat falls in a given range (everyone rejects the null, for instance), etc.
- Design the question around the answer and the work required to find it

Tying randomness to the student

- Use student ID to set the random seed.
- Better, combine ID with other information (the question number, the module code, the year)
 - the year in order that a student who is repeating the year will not get the same values
 - the question number in order that different questions' random values will not be related

Verification of ID

- Verifying the student's identity (within reason) is important
- Not possible to stop them getting unfair help
- Important to stop other people submitting answers for them without their involvement or permission
- Best solution: link the Shiny server to a password-authentication system
 - Possible with paid premium version of R-Shiny
 - Not easily possible for the free version.

Cryptographic hash

- Alternative to passwords: cryptographic hash based on the ID and other info
- Give each student a private link based on this hash
- Student submits ID, ID is re-hashed and result compared
- As long as the links remain private each student controls their own assignment
 - The hash is not guessable

A small example

ID	link
12345	http://teaching.sociology.ul.ie:3838/exass/?key=fa7013a5daa77ed7605495312a2d73ce
54321	http://teaching.sociology.ul.ie:3838/exass/?key=236f13ccd50f0e573b1f743708668fc6
77777	http://teaching.sociology.ul.ie:3838/exass/?key=9ed6336de5f2e4a6caa30a757ff103b8

The links above take us to a specimen assessment exercise, with three questions. To validate the link the student enters their ID, and the questions appear fully, plus the most recent responses.

- Assessments are very like a self-learning exercise, but with some extra features:
 - persistent data storage
 - marking the questions and summing the score
 - functionality to allow multiple entries and take the best answer, if relevant
- Like the self-learning exercises, assessments need code:
 - that does the correct calculation, and assigns marks
 - that does feedback (though to be delivered differently)

Coding issues

Assessment question structure

- Current working framework
 - main file sets up the page, and defines functions for reading and writing the persistent data to the SQL database
 - Reads one subfile per question

- Question subfiles define the question details
 - Caption
 - Create the random data based on ID
 - To insert values in the question text and to write it
 - Required inputs (answers) and the related database fields needed
 - To show any required graph or table
 - To calculate the correct answers for the question (defining a tolerance), assess the result, and short feedback text showing how the question is correctly answered
- Most of this is normal R functions, easily tested and debugged outside Shiny

Marking and feedback

- Marking and feedback occurs outside of Shiny
- An R program uses the functions to talk to the SQL database, and the functions for data creation, question setting and marking/feedback
- Writes results to a CSV file containing the answers, the correct answers, points awarded and worked example
- To be e-mailed to each student individually

Example feedback

Sample feedback for Q1, student 12345

ID	Question	Answer submitted	Correct answer	Outcome	Point
12345	Qn 1 P between X1 and X2	0.11	0.254685736884538	Incorrect	0
Explanation					
$z_1 = (X_1 - \text{mean}) / \text{SD} = (92.0 - 102.0) / 16.3 = -0.613$ $z_2 = (X_2 - \text{mean}) / \text{SD} = (103.0 - 102.0) / 16.3 = 0.061$ From the table of the standard normal distribution (or a computer function): - proportion below $z_1 = -0.613$ is 0.270 (equals proportion above $-z_1$, if $z_1 < 0$) - proportion above $z_2 = 0.061$ is 0.476 $P = 1 - 0.476 - 0.270 = 0.255$					
Question text					
If a normal distribution has mean and standard deviation respectively 102.0 and 16.3, what proportion of the distribution lies between 92.0 and 103.0?					

Reflection

How to write effective apps

- Not always clear, for demos or for self-learning exercises
- Requires serious pedagogical thinking, and creativity
 - select key concepts/learning outcomes
 - think hard about how to present them, and whether the interactivity can help learning
- What are the typical gaps in understanding, what are the common pitfalls?

Use of self-learning apps

- Self-learning apps seem to work very well, in conjunction with assessments
- Tasks closely aligned to what we cover in class, derived from assessments
- As teachers, we have plenty experience in setting assessments to match what is being taught
- Students engage because it helps with the assessment task
- My impression is that most students get full or near full marks on most questions, because of the opportunity for structured independent learning

Pedagogical advantages

- The key advantage of these apps is computer-supported practice
 - students interactively and independently acquire relevant technical skills
 - immediate feedback of a nature that a textbook can't provide
- Allows setting of more frequent but smaller homework tasks
 - synchronous with the learning done in class
 - more beneficial than a smaller number of larger assessments
- Quick feedback is reassuring for students
 - and makes the instructor aware of emerging problems sooner
- Being able to replace in-term assessments and exams made last semester so much easier

Implementation and technical issues (appendix)

Required skills for implementation

- If you know R, you can leverage R-Shiny immediately
 - But interactive web pages are different animals from sequential data analyses
 - Completeness of R makes lots possible but risks excessive complexity
- Complex things, e.g., processing assessment results, are tricky
- IOW good data analysis programming skills are needed
 - But if you're teaching QM, perhaps you do have them

Some Technical Challenges

- Device compatibility: something that looks OK on big and small screens
- Aesthetics and ergonomics: make it look good, be as easy to use as possible, be pedagogically effective
- Protecting the server from attack or overload
- Making the data storage for assessments more secure
- Making the assessment and feedback code more robust
- Making feedback more digestible

Shiny server

- Shiny apps run on a specialised server
- Can install on a dedicated machine
 - Must be visible from the public net
 - Not hugely difficult but requires some minor non-standard IT skills
- Free version has almost all the functionality you can need, premium version has extra bells & whistles (and support):
<https://rstudio.com/products/shiny/download-server/>.
- Can host apps on <https://www.shinyapps.io/>
 - Run by RStudio (the organisation behind Shiny, tidyverse and the RStudio R-interface)
 - There are various offers, between free and very expensive

Finally...

- If anyone wants to look at the code, I am happy to share: e-mail me
- OTOH the assessment code is rough and ready:
 - I can solve the problems I cause myself
 - But I can't guarantee that I can even explain it you, let alone support you using it!

Slides at <http://teaching.sociology.ul.ie/shiny/quantSOC.pdf>