Introduction to the RStudio Server

Duncan Golicher

10/21/2018

## Introduction

Rstudio is a complete environment for working with the R language. Once you have got used to it you will find that it makes working with R far more productive than using the R console version. However the concepts involved in using RStudio will be completely new to most. The interface can seem very complex. RStudio provides an interface for working with R code, rather than an interface for running analyses directly. The code for conducting analyses will be provided for you in this course. You do not need to learn to write any Rcode yourself in order to produce graphics and run basic statistical analyses. You will also be able to reproduce more sophisticated quantitative and spatial analysis in RStudio by making minor adaptations to pre-built scripts.

## Getting started with the RStudio server

The RStudio server version runs directly through any web browser. **There is no need to install any software on your laptop, PC or tablet**

Access to the server is through the following URL. This works both on and off campus.

<http://r.bournemouth.ac.uk:8789/>

### Action: Log into the RStudio server

1. Click on the URL in a browser. Use Firefox or Chrome.
2. You will see a log in page.
3. Log in using a username and the default password **msc123**

The usernames for this course have been set up to be your first name as shown in brightspace and surname initial, all in lower case.

veronicab elizabethe garethh minniej curtisl esmel ricardol robm ttomasn georgiap andrewr molliet anuro msc1 msc2 guest phd1

If your name is not in this list, or you want to change your user name let me know. **Do not log in using someone else’s name!**. If there is any abuse of this I will show users how to change to a secure password, but we will assume trust by default.

## RStudio server concepts

Using a server can seem strange at first. The RStudio server is an integrated platform for doing the following …

1. Saving and sharing data files
2. Running analyses
3. Compiling reports
4. Connecting to data stores
5. Sharing analyses with others.

Amongst many other potential uses. It is a very powerful tool that is freely available for all those with a log in to use it. The more advanced features can be used without any programming skills through sharing scripts. However you do need to become familiar with some new concepts in order to use these.

The RStudio server is ideal for collaborative work. You have been provided with a username and a password, as this also provides you with your own permanent space on the server for saving your own work and building up a portfolio of useful analyses. Only one person can be logged in at any one time under your username. However I can always log into your user space at any time in order to help correct your errors and to give you advice.

## Finding your way around the interface

Once you are logged in you will see three sections of the interface by default. This will change to four sections when you begin using scripts in the interface.

Look carefully at the interface and learn to recognise the sections.

1. The RConsole. This is showing up on the left hand side when you first log in. The console can be used for running R code interactively. There is a tab showing up labelled “terminal” as well. You won’t use this, as it is for more advanced programming.
2. The environment, history and connections pane is at the top right of the screen. The environment tab is the one that is most used. This tab will show the data that is in the active workspace in R. The concept will only become clear after beginning to use R.
3. The files, plots, packages, help and viewer tab at the bottom right. The files tab is the most important to understand at this stage. There will be no files in your home directory yet, nor will there be any folders.

A key concept to understand when using the server is that your home directory on the server is like a directory (folder) on your PC. So ,it is rather like the university H drive. However it is all “encapsulated” on the server which is also running R. So it is distinct from your H drive and not directly linked. In order to move data files and scripts into your home directory you must upload them. You will see buttons labelled **New Folder, Upload, Delete, Rename and More**. If you click on the **More** button you will also find an option to **Export** your files. The **upload** and the **export** buttons are frequently used to move files onto the server and to directly move files off the server. It is **very important** to be aware of this concept. Files saved on the server will always be available for use later. In contrast active analyses that take place in the server memory, as opposed to the server’s hard disk space, will be temporary and will be lost between sessions.

### Using projects in RStudio

Just as we have seen previously in QGIS the use of projects is good practice. QGIS can be used without a project file, but if you do that you may find that you have to go through many steps to reload files and get back to where you left off. RStudio has a similar concept. We can use a project for each week’s work in R. Let’s start one called “intro”. Just as in QGIS each project should be associated with a single folder and all the work placed within this folder. The folder can be added when the project is first started.

### Action: Form a new project and add a new folder

1. Click on the file menu at the top left of the interface.
2. Go to New Project
3. You will see a window with three options to create a project. Choose the first option labelled **New Directory**
4. The next window will show a range of advanced options. Ignore them and just select **New Project**
5. You will now see a window with a prompt for the Directory name (and some other options). Type **“intro”** as the directory name.
6. Click create project
7. Look at the files pane in the bottom right corner. You will now see that after Home there is the word intro. You can also see a file called intro.Rproj in the folder.
8. Click on home. You can see a folder called intro in your home directory. So .. you have created a new project and placed the project file within the folder. This is just like starting a project in QGIS.
9. Click the folder again to open it.

### Using the console

The R console in the bottom left of the screen allows you to run R code interactively. In this course you will not be using this very much, as we will use pre-built R code that avoids the need to code your own. However in order to use the code you do need to gain some intuitive idea of what is happening when you do.

### Action: Make a data object (vector) in R

1. Carefully type the following code into the console **x<-rnorm(100)**. Make sure it is all lower case and typed **exactly** as shown.
2. Press enter
3. Look in the global environment window in the top right of the screen. You will see that a data object called “x” has been added.

## Data abstraction and data visualisation

The action you have taken so far involves a concept that will be completely new to most of you. We can refer to it as **data abstraction**. Most of you will be familiar with data in the form of visible numbers, text, or even maps. However programmers (and mathematicians) think of data in the abstract. We have a data object (in this case a numerical vector) called x. It consists of 100 numbers. These numbers have been simulated by R from a normal distribution. However data analysts tend not to look directly at the numbers themselves. They are much more interested in patterns in the numbers and relationships between variables.

### Action: Visualise the data using a simple histogram

1. Type **hist(x)** carefully into the R console in the bottom left of the screen and press enter.
2. Notice that the plots tab becomes visible in the bottom right. It shows a histogram of x.

So what has happened here?

1. The first line of code (**x<-rnorm(100**) generated a vector of 100 numbers drawn from a normal distribution with a mean of zero and a standard deviation of 1. This vector of numbers was placed in the computer memory. All the numbers existed, but we did not look at them.
2. We could see the vector was active in memory as it appeared in the Environment pane.
3. We then investigated the properties of the data object by plotting a simple histogram.

These concepts **will** become clearer over time when using R. Notice that at this stage no data is actually saved as a file on the hard disk. It is present in the computer’s memory.

### Action: Close the project

1. Click on the project icon (marked intro) on the top right-hand side of the top menu bar.
2. Go to “close project”
3. You will be asked if you want to save the workspace.
4. Click on save
5. You will now return to a blank workspace.
6. Click on the files tab at the bottom right and click on the intro folder
7. Notice that the folder now contains an .Rhistory file, an .RData file and a file called intro.RProj.

This was a tiny R project, but we all have to start somewhere. Programmers often begin learning how to make the language they are learning send a “Hello World” message to the console. So this was the equivalent of “Hello R World”

Let’s go back to the project.

### Action: Restore the project

1. Click on project on the top right of the menu bar.
2. You should see the word “Intro” in the window. If you had more projects then they would all be listed here. Note you could also use the open project option to go back to your project, but of course, at the moment you only have one project available.
3. Click intro
4. Go to the plots tab and notice that there is no histogram showing now.
5. Click on the history tab in the top right panel.
6. Notice that the code you wrote is shown there.
7. Click on the console panel on the right to make it active.
8. Press the up arrow on your keyboard once.
9. Notice that the last command that you entered (**hist(x)**) is now shown in the console.
10. Press enter.
11. Notice that the histogram now shows in the plots menu.

So, what’s going on?. When you answered “yes” to save the workspace both the data and the history of the R commands were saved in the folder. So the whole “analysis” can be recreated at any time. This will all become clearer over time.

## Working with markdown documents.

This course will concentrate on the use of markdown documents as a way of running R code. The advantages of using markdown are many.

1. Embedded code can be either revealed to other users to show how the results were obtained or hidden to simply produce a report with embedded figures and statistics.
2. Annotation of the results of an analysis can be embedded around the results to explain the key results.
3. Very limited knowledge of the R language and syntax is necessary to adapt markdown documents in order to analyse your own data.
4. With a little more knowledge and experience of R complex methods can be applied by altering markdown found on-line.

### Action: Produce a default markdown document.

1. Go to file on the top menu bar
2. Choose “New file”
3. Choose “R Markdown”
4. You will now see a window in which you can type in a name for the title of your analysis. By default the name is “untitled”. Change that to something like “R demo”, or anything else you feel like.
5. You will now see an untitled markdown document added to the top pane in RStudio. It is untitled, even though you’ve given it a title, as it is not saved as a document.
6. Press the “knit” button on the top right pane.
7. Now you are prompted to give the file itself a name. Call it whatever you want, maybe “R demo” again.
8. After knitting the document you will be prompted to open a window to see it.
9. Look at the document and understand what it consists of.

The steps above produce a default “demo” markdown document. Every time you start a new markdown Rstudio will start off with this one. When you use markdown for real you will either open a document ready for knitting or replace the default demo text with your own.

You should look at the logic of the document carefully. It consists of “chunks” of R code that produce output in the form of tables and figures embedded in text. The R code automatically produces output and adds it to the document after knitting. So if you have R code available that will run an analysis that you are interested in you don’t have to remember any other steps in order to run it. Simply ensure that the data that is being added to the analysis is appropriate for the type of analysis being run and you can obtain the same results with your own data. This will be the way R is used in this course.

### Action: Clean out, rinse and repeat, for practice sake

1. Close the project.
2. Delete the whole folder in the files pane by selecting it and then choosing delete.
3. Choose “clear project list” from the project menu at the top right.
4. You are now back right where you started (apart from the .RHistory file in the home directory)
5. Try all the above steps again to become familiar with the process before we begin more serious work. You will need to open a new project each week to keep all your work in good order and prevent confusion.

## Conclusion

You should now be able to start a clean project in RStudio. The code to actually run the analyses will be provided for you. So there is no need to fully understand R code to complete the unit. You will not be expected to write any code yourself. However, as you progress with R you will begin to alter code and adapt it for your own purposes. With time you will be able to write code yourself to run analyses. The key to the process is to become familiar with the general concept of data abstraction and to gain a feel for the nature of the data that is used for statistical analysis.