

Getting started in R

Lauren Kennedy

School of Psychology, University of Adelaide

2016-Version 1

1 Assumed knowledge

If you're using a university computer, you can jump right in with this guide. If you want R/R Studio on your home computer you will need to install both R and R Studio. The ITS (located in the hub) can help you with this. You should use this guide as your first step to doing statistics in R. It will go through why you should always use R studio, where things are and what they do and a few basic tricks.

2 R or R Studio?

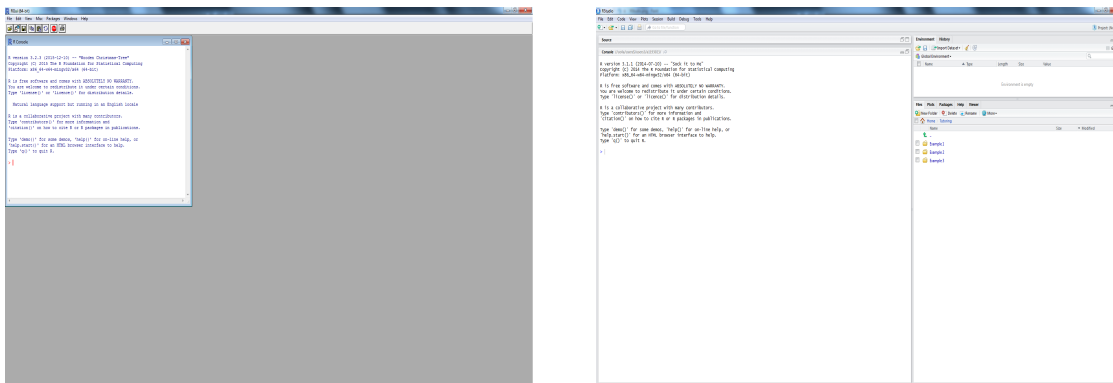
If you look on you computer (either on campus or your own computer once you've installed R and R Studio on your own computer) you will see that you've actually installed two different programs. The first is called R. If you click on R and open it up, it will look something like the left hand panel of figure 1. If you open up R Studio, it will look like the right hand side.

Hopefully you can see that R Studio provides you with a *lot* more options, menus and help guides. That's why we recommend to ALL students that they use R Studio. Why do you need both? Technically R Studio relies on R to do the actual work but R Studio provides a really clean interface. So you need to install R *AND* R Studio, but you should only ever open R Studio. Whenever anyone in your classes (or in this guide series) says to do something in R, they really mean to do it in R Studio.

Sometimes if you click on a R file (which looks something like filename.Rdata or filename.R) it might open automatically in R, *not* R Studio. We'll get into R files later, but the important thing to remember is that if you're using something the looks like 1a then you should stop and open R Studio.

3 Where to find things?

Let's take a closer look at R studio. See figure 2 for a close up. You can see we have highlighted the three panels we will use a lot of. There's the console (red), the environment/history (green) and a the bottom right panel (blue) which has a number of different tabs. We'll look at a few of the functions of each panel.



(a) Workspace of R

(b) Workspace of R studio

Figure 1: Comparison of R to R Studio

3.1 Console

In this section we will focus on the red outlined panel. This is the console, which has a couple of different uses. The most basic use is to think of R as a large calculator. If you type:

```
5 + 3
```

in the console and then press enter, you will get the answer 8. Anything that you can do in a calculator, you can also do in this console. Here are a few commands to try. Type each row into the console to see what it does. I've explained what each command does in a *comment*. A comment is a little note that R knows to skip over. We tell R that some text is a comment by putting a # before it. R will then think the rest of the row is a comment. If you're typing these out, you can skip the comments. Otherwise you can copy and paste each row into your console.

```
5 - 3 #addition
```

```
4*5 #multiplication
```

```
30/6 #division
```

```
(5+3)*2 # brackets: First add 5+3, then multiply by 2
```

3.1.1 Errors

What happens if you mess up a line? Try below to see what happens if R isn't sure what you're asking.

```
5 + / 3 #doesn't make sense
```

R returns an error in red. Don't be frightened by getting errors! It happens to everyone. The error will generally give you some information about what went wrong. In this case, it's because R was prepared to add something to 5, but then got something it wasn't expecting (a divide sign). You can fix it by taking out the divide sign. If you go back to the console and press the up key, you can get the command you just typed back. This time remove the divide sign (or the addition sign) and it will work.

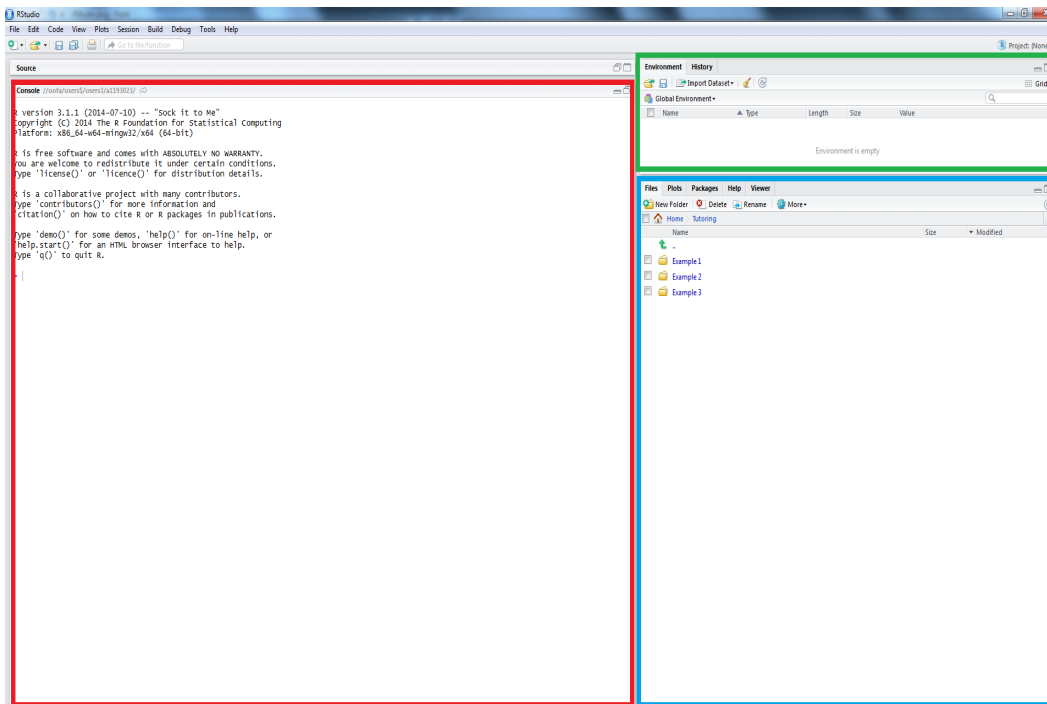


Figure 2: Workspace of R studio

3.1.2 Variables

There's one other important thing to know about the console. We can assign things to variables. A variable is a placeholder for another object. The most simple object we might store in a variable is a number. Let's say we want to do a number of operations with the number 102. We might not want to type the number every time, so we store it or *assign* it to a variable x . This means we can use x whenever we want to say 102, and R will substitute 102 in without us having to type it.

First you need to use the assign operator $<-$. If you look at it closely, you'll notice it's like an arrow. It tells R to take the things on the right of the arrow and store them under the name of the thing on the left. Here we tell R to store 102 under the name x .

```
x <- 102
```

Now that 102 is assigned to x , we can use it in a simple calculation.

```
x + 10
```

There's lots of different types of objects or things we can assign or store in variables. We'll learn more about what you can store and how to do it as we go on.

3.2 Environment/History

This panel has two tabs, the environment tab and the history tab.

3.2.1 Environment

The first tab, the environment tab, stores all of the variables or objects in the environment (the environment is a fancy term for the things that you have done in R today). If you were following along in the previous section, you should have one variable, x , in this panel. If you look at the section

highlighted yellow in the top right hand corner of figure 3, this should be set to grid view. This view tells you a bit of information about what is stored in x (i.e. it's numeric, the value is 102 etc.).

You can delete x from here too. Tick the box next to x, and then click the broom icon in the middle of top bar. This removes x from your workspace.

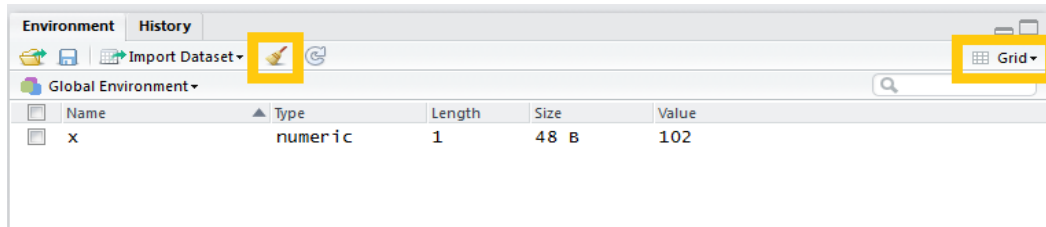


Figure 3: Environment of R studio

3.2.2 History

This panel section contains a list of all the commands that you have entered into R in this session. If you click on a line and then select the icon that says *To Console*, you can recover an older command into your console to run again or edit as needed. This history is **NOT** saved after you have exited R. To save commands you will need to learn how to create scripts (we're working on a guide of how to do that soon), but you can complete your practical without them.

3.3 The final panel: Files, looking at figures, package management and getting help

This final panel has a lot of very useful functions that you will use. We won't go into all of the things you can do, but we will look at a few common things.

3.3.1 The "Files" tab

This tab allows you to access files from your student drive. You can select R files to open from here. One of the most common files are .Rdata files, which are the files you use for your practical. For more on loading practical data, see (help loading data). There's a lot you can do with the files tab, but for now you should just know it's there.

3.3.2 The "Plots" tab

This is the place where you'll find any plots that you create. You can learn more about plotting here (help guide under construction).

3.3.3 The "Packages" tab

This one is really important. In R there are a lot of tools that are available as soon as you open it. One of the really cool things about R is that there's a network of researchers who develop tools for R users to use. They group these tools into things called *packages*. You can download most of these packages straight from R studio, but you'll need to be connected to the internet. One package you might download is written by a former Associate Professor here, Daniel Navarro. It's called the *lrs* package, and it accompanies the Learning Statistics with R book. You can install this package either by typing the following or by using the *Install* button. If you use the button, make sure the box *Install*

Dependencies is ticked (see figure 4. Dependencies are packages that the package you wish to use uses, so you need to install them too to make your package work.

```
install.packages("lsr")
```

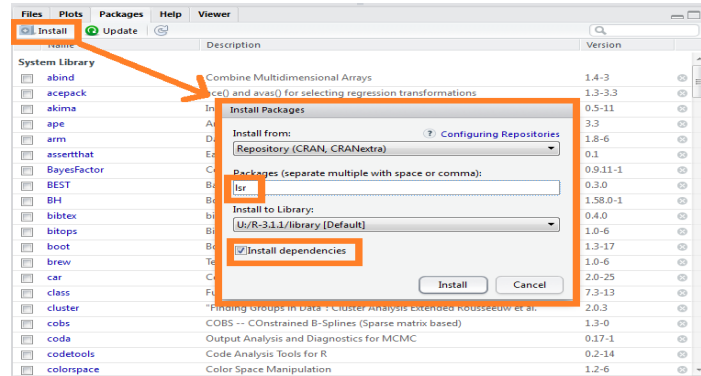


Figure 4: Installing a package

Once you have a package installed, it will be listed in this window. If you wish to use it in a current session of R, you will need to select it by ticking the box next to it or by typing:

```
library(lsr)
```

Packages generally contain functions that you can use to do certain things. For more on functions see the help guide "Fun with Functions". For now just note that there are two steps to using a package. First you have to *install* the package. You'll only need to do this once on your computer (and not at all on the university computer). Second you need to tell R you want to use it at the start of your R session. This uses the library command, and you will need to do it every time you open R and want to use that particular package.

3.3.4 The "help" tab

This one has lots and lots of information on how to do different things in R. If you're interested in how to use a specific function, you can search that function and you'll often get information and examples.

In this guide you were introduced to R Studio and where to find a lot of different functions. You should look back on this guide once you've looked at some of the others to see if you can work out some extra uses of each panel.