

Fun with Functions in R

Lauren Kennedy

School of Psychology, University of Adelaide

2016-Version 1

1 Assumed knowledge

This guide assumes that you have downloaded R and have had a look through the "Getting Started with R guide". It is designed as a specific tutorial in what functions do in R. We'll use a few different functions to illustrate how they work, but the main emphasis will be the logic behind them.

2 What is a Function?

Functions are time saving and expertise saving devices. To understand how they work, let's imagine we want to make a cake. We know that we'd take some items (ingredients), and then perform a number of actions on those items (crack the eggs, sift the flour, mix and bake) and we would get a product (cake). If we were fancy cooks, we might have a Thermomix (an expensive cooking device that you feed ingredients and it returns food). With a thermomix, all you need to do is pour the ingredients in the bowl, select the type of food you'd like to make, and then enjoy the outcome. Once you select the type of food, the machine knows to follow a certain procedure to produce the outcome that you want.

Functions in R work in a similar way. Let's say we want to create a scatter graph. First we need to get our ingredients or *input* ready. In the following two lines we create two variables, a and b. These variables are more complicated than the ones we've previously used because each variable contains an ordered list of numbers. Sometimes people call them a vector. Once you've typed the two lines into the console, you can see that they are available by looking in the environment panel.

```
a <- c(1,3,5,6,7,8,9,10)
b <- c(2,3,5,6,7,7,8,8)
```

So now we have two vectors to input into a function. One thing we might like to do is plot them against one another (i.e. the first element of a against the first element of b, etc.). We can do this in R by using the *plot* function. If you want to know more about how to use the plot function, you can type the following or use the help menu in the bottom right hand panel.

```
?plot()
```

You'll get a help guide that looks similar to figure 1 in the bottom right hand panel. This help guide tells you a little about what the function does (in description), how you should use it, and what sorts of input it needs.

So now we need to give it two variables, one to plot on the x axis, and one to plot on the y axis. Conveniently, we just created two such variables in a and b. There's also a lot of other things that

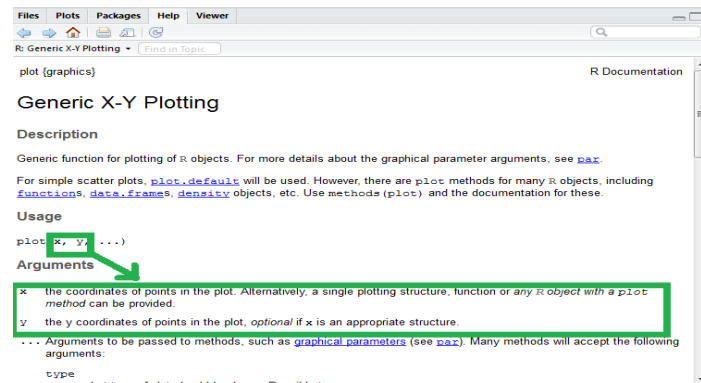


Figure 1: Help guide for the plot function

the function can use, but they are optional. These are like the sprinkles on the cake. They're great to have, but you can still have a cake without them. The next line gives our two variables to the function. The function takes those two variables, and following some steps that it has, produces a plot (figure 2).

```
plot(x=a, y=b)
```

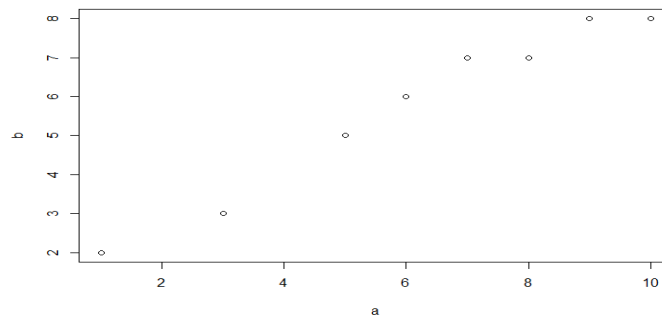


Figure 2: Plot produced

Now, if we had used a different function other than *plot*, we would have gotten something other than a plot (just like if we'd followed a recipe for a stirfry and not a cake). Similarly, if we'd given the plot function some input that it wasn't expecting, we might have ended up with something funny looking, or the function may have told us something is wrong (imagine substituting gravel for flour in your cake). Both the function name and the input are important to get right.

There's lots of functions in R. We can use them to make plots (nicer ones than this one), to calculate correlations, to do a t-test or an ANOVA or a linear regression. In this section you learnt how to use the plot function to make a simple figure in R. You learnt that functions are named, and require specific types of input. If the input is correct, then the function returns some output. We will introduce lots of different functions in the following guides.