

R Small Group: Class 5

Amy Allen & Dayne Filer

July 12, 2016

Using this document

- Code blocks and R code have a grey background (note, code nested in the text is not highlighted in the pdf version of this document but is a different font).
- `#` indicates a comment, and anything after a comment will not be evaluated in R
- The comments beginning with `##` under the code in the grey code boxes are the output from the code directly above; any comments added by us will start with a single `#`
- While you can copy and paste code into R, you will learn faster if you type out the commands yourself.
- Read through the document after class. This is meant to be a reference, and ideally, you should be able to understand every line of code. If there is something you do not understand please email us with questions or ask in the following class (you're probably not the only one with the same question!).

Class 5 expectations

1. Know how to write and run a basic function in R
2. Understand function environments and how functions find things
3. Understand the “do not repeat yourself” (DRY) principle

Functions

So far we've used a lot of functions that already exist in R. We can also add our own functions, which on of the great strengths of R. User-written functions take the following structure.

```
myfunction <- function(arg1, arg2, ... ){  
  do some stuff  
  return(object)  
}
```

Let's make a simple function that just returns the argument that it is given.

```
first_function <- function (x){  
  return(x)  
}
```

Now if we hand a value to `first_function` it should return that value

```
first_function(9)  
## [1] 9
```

This function isn't really very useful, so let's try make a function that squares a value and then adds one to it.

```
second_function <- function(x){  
  ans <- x^2+1  
  return(ans)  
}  
second_function(9)  
## [1] 82
```

Function Environment