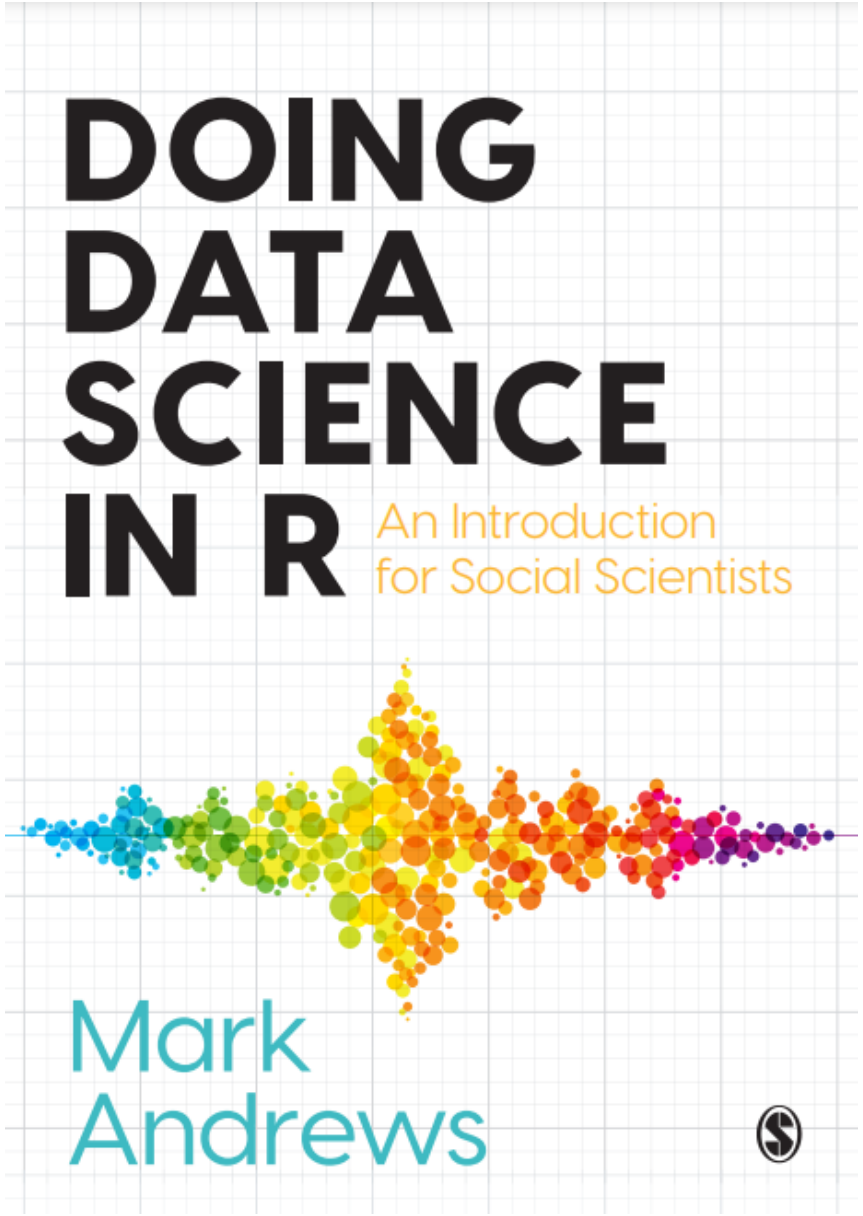

Doing Data Science in R: An Introduction for Social Scientists

Chapter 6

Programming in R

Introduction

We'll begin with functions because of the major role they play in programming in R generally.

We will then consider conditionals, which allow us to execute different blocks of code depending on whether certain conditions are true.

We will then turn to iterations, also known as loops. This will lead on to functionals, which can often take the place of traditional loops in R

Functions

Functions in R allow us to create custom commands to perform specific calculations or carry out specific tasks

We can use functions to carry out any calculations or any procedure that we could perform using any other R code.

Default values for arguments

In some cases, however, we may prefer to allow some input arguments to take default values.

Optional arguments

A function can also have optional arguments indicated by ... in the arguments list in the function definition

Function return values

Functions, however, can have arbitrarily many statements and expressions in their body.

When there are multiple statements, the value of the last expression is the value that is returned.

Conditionals

Conditionals allows us to execute some code based on whether some condition is true or not. Consider the following simple example:

```
library(tidyverse)
# Make a data frame
data_df <- tibble(x = rnorm(10),
                  y = rnorm(10))
write_data <- TRUE
if (write_data) {
  write_csv(data_df, 'tmp_data.csv')
}
```

Iterations

In R, there are two types of iterations or loops, which we informally refer to as for loops and while loops.

for loops

while loops

Functionals

Functionals are functions that take a function and a vector as input and return a new vector.

They play an important role in programming in R, often taking the place of for loops.

Functionals

Lapply; which takes two required arguments, a vector or list and a function, and then applies the function to each element in the vector or list and returns a new list

sapply and vapply; The sapply function works like lapply but will attempt to simplify the list as a vector or a matrix if possible

Functionals

Mapply; With mapply, we can iterate over any number of lists of input arguments simultaneously

Filter, Find, and Position; The Filter functional takes a predicate, which is a function that returns a logical value, and a vector or list, and returns those elements of the list for which the predicate is true.

Functionals with purrr

purrr provides additional functional tools beyond those in base R. We can load purrr with library(purrr), but it is also loaded by library(tidyverse).

One of the main tools in purrr is map and its variants. It is very similar to lapply

purrr-style anonymous functions

We saw above that we can use anonymous functions in lapply, sapply, etc., functionals.

When we have two or more sets of input arguments, we can use map2 and pmap, respectively.

Functionals with purrr

- The walk function in purrr is like map but is used with functions that are called just for their side effect

- The purrr package also provides us with functions to effectively emulate base R's Filter.