The EMaC R Manual

Alex Sciuto

Last Updated Febuary 2020

Contents

1 How to Install R **5**

2 Introduction to Programming in R **7**
2.1 Variables in R 7
2.2 Logical Operators & if-Statements 9

3 Introduction to Data Wrangling **11**

4 Methods **13**

5 Applications **15**
5.1 Example one 15
5.2 Example two 15

6 Final Words **17**

Chapter 1

How to Install R

This is the EMaC manual for how to wrangle, analyze, and visualize data in **R**. I will review a few key packages, in addition to explaing their key functions, with real data examples.

There are two things you need to do to install R on your computer. First, you would need to install the latest version of R which you would download and install from this link: R-download. Once you run and install R onto your computer, you would need to install R-Studio. R studio is the graphic user interface (GUI) where you can place all of your R-code. Follow this link: R-Studio Desktop Download. Once you have these two programs installed, all you need to do is launch R-studio Desktop and you are ready to go.

Chapter 2

Introduction to Programming in R

Before you can do data wrangling, statiscs, and visualization using R for cognitive psychology research, you need to learn the basic syntax in R. This chapter will introduce the basics.

2.1 Variables in R

R can manipulate and wrangle all kinds of data, these data could be stored in a handful of variables that we can then work with. The first one we will be talking about is numeric.

2.1.1 Numeric

a numeric is a number (including decimals) that can be stored within a variable. Here is an example:

```
x = 2
```

Here, we assigned the numeric value `2` to the variable `x`. Thus, if we were to do arithemtic, `x` will be treated as `2`. Moreover, there are specific functions in R we can use in order to do arithmetic with numeric variables. Here are a few examples:

Arithemtic	R-Input	R-Output
Addition:	`x + 2`	`4`
Substraction:	`x - 2`	`0`

Arithemtic	R-Input	R-Output
Division:	`x / 2`	`1`
Multiplication:	`x * 2`	`4`
Exponent:	`x ^ 2`	`4`

2.1.2 Character

A character is a combination of characters (either letters and/or numbers) that can be stored within a variable. Moreover, it is very important that you place the character between quotation marks. Here is an example:

```
x = "Cognitive Psychology"
```

Here, we assigned the character value `"Cognitive Psychology"` to the variable `x`. There are ways to manipulate character type variables, and also modify them. However, we will touch on that later.

2.1.3 Logical

A logical can only have two possible values, `TRUE` and `FALSE` that can be stored within a variable. This is not typically a varaible type that you will need to assign variable values to. However, the logical type variable is extremely important becuase a lot of functions in R return a logical variable. We will dive into some of these functions

2.1.4 Vector

A vector is multiple variables stored into one data-set. Vectors can contain any variable type: character, numeric, string, and logical. When you are declaring a a vector type variable, you typically have to use the concatenate function `c()`. Here is an example:

```
x = c(1,2,3,4)
```

```
x = c("I", "like", "Cognitive", "Psychology")
```

```
x = c(TRUE, FALSE, FALSE, TRUE)
```

To reference a vector varaible, you need to use `[]` and inside the brackets, you have to specify the index of where the given variable in the vector is. Here is an example

```
x = c("10", "20", "30", "40")
x[4]
```

```
## [1] "40"
```

Here, we assigned the numbers: `10, 20, 30, 40` to the vector `x`. `"Cognitive Psychology"` to the variable `x`. There are ways to manipulate character type variables, and also modify them. However, we will touch on that later.

2.2 Logical Operators & if-Statements

Now that you have a basic understanding of how variables work in R, the next step is to learn how to use logical operators in order to specifiy what are the specific conditions under which you want your variables to be manipulated. The main way to do this in R is by using logical operators.

```
x = 2
```

Here, we assigned the numeric value `2` to the variable `x`. Thus, if we were to apply logical operators to `x`, `x` will be treated as `2`. Here are examples of all of the logical operators:

Logical Operators	Description	R-Input	R-Output
<	less than	x < 10	TRUE
<=	less than or equal to	x <= 2	TRUE
>	greater than	x > 1	TRUE
>=	greater than or equal to	x >= 3	FALSE
==	exactly equal to	x == 9	FALSE
!=	not equal to	x != 10	FALSE
!x	Not x	!x	FALSE
x | y	x OR y	x == 10 | x != 10	TRUE
x & y	x AND y	x == 10 &x != 10	FALSE

Now that you have a basic understanding of logical operators, we can use them in order to specify code to manipulate variables in a particular way. For this example, we will be working with the function `if_else()`. However, one question you may have is what is a function in R? I will describe the structure of functions, starting with a simple one.

function: if you give an input, the function will do some computation and return an output	arguments: this is the structure of how the input is recieved, every function has preference	output: this is what the function returns	input	
max(): this function finds the maximum number in a numeric vector	(c(1,2,3,4)): parameters are the things you input into teh function to get a specified output	less than	x < 10	TRUE

‘‘

Chapter 3

Introduction to Data Wrangling

Here is a review of existing methods.

Chapter 4

Methods

We describe our methods in this chapter.

Chapter 5

Applications

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Chapter 6

Final Words

We have finished a nice book.