SIT718 Real World Analytics
Week 1 Prac.

1.10 R Tutorial

There are many online tutorials available for becoming familiar with the programming language R. Once you have the basic commands down, you will find that an online search for how to do something specific can almost always find an existing solution either as part of standard packages, or as code that someone else has provided. We will start with some basic operations and then move toward being able to calculate values for the functions we came across in this first topic.

Entering commands in the console

For simple one-line commands, we can enter these straight into the console by typing the command and pressing return/enter.

If we want to retrieve a previous command, we can press the up arrow.

Basic mathematical operations

Elementary functions and basic arithmetic operations are all part of R. In most cases, these are the same as what you may have used before in programs like Microsoft Excel.

R Exercise 1 *Trying entering in the following commands (the expected results are shown for the numerical examples given).*

Operation	*Symbol*	*in R*	*Example*	*in R*	*Expected result*
Addition	$+$	+	$5 + 2$	5+2	7
Subtraction	$-$	-	6 - 8	6 - 8	-2
Multiplication	$\times$	*	3×4	3*4	12
Division	$\div$	/	$3 \div 4$	3/4	0.75
Powers	$(\cdot)^p$	^	3^4	3^4	81

Assignment of variables

We are often going to be working with general operations in terms of variables and in some cases we will require operations on vectors.

We can use almost any word or letter for a variable along with the full stop symbol "." and underscore "_". We can't use the dash (because it treats it as a minus sign) and we can't start with a number and there are some special words we can't use *for functions* because they are already used by R for other things (we can check this by typing the function and an open bracket, e.g. `max(` and seeing whether the function parameters come up in the bottom status bar.

To assign a value to a variable we use the *less than* symbol '<' and the dash '-' together (it makes an arrow). So if we want to set the value of a variable a to 3, we write

```
a <- 3
```

Try doing this and see what happens when you enter the command $a + 4$.

R Exercise 2 *Assign the following values to each variable.*

```
a <- 3
b <- 7
the.value <- 12
the.index <- 2
the.index_2 <- 3
```

Now evaluate:

```
a*b
the.value * a
a^the.index + b^the.index_2
```

Assigning a vector to a variable

In order to assign a vector to a value, we use a 'c' along with the entries of the vector inside normal brackets, separated by commas. For example,

```
a <- c(3,2,1,0)
```

We have some other special commands that we can also use. If we want a vector of a set of sequential numbers, e.g. $\langle 4, 5, 6, 7, 8 \rangle$ we can write the starting and finishing numbers separated by a colon.

```
a <- 4:8
```

Sometimes we will need to first define an empty vector, or a vector with default starting entries. This is more easily achieved using the `array()` command in R. This command has two arguments. The first one is the default entry of the array, e.g. 0, while the second argument gives the length. For example,

```
a <- array(0,3)
b <- array(1,10)
long.array <- array(2,200)
```

We can also assign arrays that have a small sub-sequence. For example, suppose we want the vector $\langle 1, 2, 3, 1, 2, 3, 1, 2, 3 \rangle$. We can assign this using a combination of the `array()` and `c()` commands.

```
my.seq <- array(c(1,2,3),200)
```

Notice that in this case, the `c(1,2,3)` acts as just one argument in the array command, even though it's a set of 3 numbers.

Basic operations on vectors

All of the basic operations we discussed, i.e. `+`, `-`, `*`, `/`, `^`, can be used on vectors. If we are operating with a vector and a normal number value (e.g. 3) these are calculated component-wise.

R Exercise 3 *Enter the following commands from the input column and check the results.*

Input	*Expected output*
`c(1,2,3) + 3`	*4 5 6*
`c(1,2,3)*4`	*4 8 12*
`c(1,2,3)^2`	*1 4 9*

We can also have operations between vectors. In most cases, these operations should be between vectors of the same length. The operation is applied between each of the corresponding components, for example

$$\langle 2,3\rangle + \langle 5,-1\rangle = \langle 2+5, 3+(-1)\rangle = \langle 7,2\rangle.$$

R Exercise 4 *Assign the following two vectors,*

```
a <- c(1,6,7,9)
b <- c(-1, 2, 1, -2)
```

Now check the following commands against the expected output.

Input	*Expected output*
`a+b`	*0 8 8 7*
`a - b`	*2 4 6 11*
`a*b`	*-1 12 7 -18*
`a/b`	*-1 3 7 -4.5*
`a^b`	*1 36 7 0.01234568*

Existing functions

The mean and the median are already pre-programmed into `R`. We can calculate the arithmetic mean of a set of numbers using `mean()` where the argument is a vector. For example,

```
mean(c(2,3,6,7))
mean(2:7)
mean(my.seq)
mean(c(4,a,1:3,the.value,long.array))
```