

## 3 Inbuilt functions

### 3.1 What are functions?

R contains a huge number of inbuilt functions. In fact, in the previous section, we used a number of inbuilt functions such `c()`, `length()`, `ls()`, `seq()` and `subset()`. Let's see how we used two of these.

```
> x <- c(1,2,3,4)
> length(x)
[1] 4
```

On the first we line used `c` with four **arguments**—the numbers 1 to 4—separated by commas (this is known as a **function call**). The function took the four elements in its arguments and combined them into a vector. This vector is **returned** by the function and we assigned it to the variable `x`.

On the second line, we supply the newly created `x` vector to the `length` function. This function calculates and returns the length of the vector. Because we haven't assigned the returned value to another variable the result is printed to the screen.

This simple example demonstrates a number of features of functions in R.

- Functions are **called** by specifying their names followed by a pair of round brackets. The round brackets contain the **arguments** separated by commas.
- Functions do some work or a calculation and **return** an object. This object could be a single variable or collection of variables, e.g. a vector.
- Functions can take different numbers of arguments (including no arguments, like the call to `ls`).
- We can **assign** the result of a function to a variable for later use. If we do not do this the returned value will be outputted to the screen (and lost).

The last point is important. In general, we need to think of functions as separate pieces of code. In general, we can only communicate with functions via their arguments and their returned values. We will return to this point in Section 7.1. Finally note that the `length` function takes only one argument. If we try to pass in two vectors, we get an error.

```
> length(x, x)
Error in length(x, x) : 2 arguments passed to 'length' which requires
1
```

So how do we know how many arguments a function takes? Well one way to find out is to use the **documentation**. For example we can use the `help` or `? length` function as follows.

```
> help(length)
> ? length
```

This will open up a window with documentation for the function. The ‘Usage’ section lists the way the function can be called. The documentation in R is of high quality and it is good practice to use it. However, be warned that the style needs a bit of getting used to. It is likely that you will initially find the help pages more confusing than enlightening. But persevere! We would recommend you look up every command you use. By looking at the help pages for very simple functions, you will get used to their language.

### Exercise 3.1

1. Use the documentation to view the alternative ways to use the `seq` function. Can you call it using the `length.out` argument instead of specifying the `to` argument?

**Solution:**

Use the documentation

```
> help(seq)
```

## 3.2 Numerical functions

Another example of an numerical inbuilt function is the exponential function.

```
> a <- 3
> exp(a)
[1] 20.08554
```

To look up the details of this function, we can use the `help` command again. For example, to see the help page on the exponential function (and logarithms) just issue the command.

```
> help(exp)
```

It should be noted that most inbuilt R functions operate on vectors in an element-wise fashion so that the function is applied to each element in turn.

```
> v <- c(1, 10, 100, 1000, 10000, 100000)
> log10(v)
[1] 0 1 2 3 4 5
```

So here the function `log10()` is applied to each element of the vector `v`.

When in the help screen use the arrow keys to scroll up and down or the space bar for page down. To quit the help page type ‘q’ for quit.

Other common numerical functions include `sum()`, `abs()`, `sqrt()`, `cos()`, `sin()`, `tan()`.

### Exercise 3.2

1. Use the documentation to understand what the following functions do: `abs()`, `sqrt()`, `cos()`, `sin()`, `tan()`
2. Calculate the value of the Golden Ratio,  $(1 + \sqrt{5})/2$  but this time use the inbuilt function `sqrt`

```
> (1 + sqrt(5)) / 2  
[1] 1.618034
```

## 3.3 Statistical functions

R contains many functions for data analysis. Here is a list of some of the most commonly used for descriptive statistics:

- `mean(x)` - mean of vector `x`
- `sd(x)` - standard deviation of vector `x`
- `median(x)` - median of vector `x`

There are functions for all standard (and not so standard) statistical tests, but we will not describe those here.