R10

Objects

Covered in R10

- Types of objects
- Vectors and matrices
- Arrays, data frames, lists and factors

1 Types of objects

R is an object orientated language. It stores information in data structures called objects. Everything in R is an object (even unassigned numbers are treated as objects with no name) and we perform operations on these objects.

In the previous chapter we looked at the types of data that can be stored in objects. These were numeric (*ie* real numbers such as 2.7 or pi), character (*ie* qualitative data such as policyholders names), logical (*ie* TRUE, FALSE and NA), complex (*ie* complex numbers such as $3+2i$) and raw (which was raw data bytes).

In this chapter we're going to look at six different types (or classes) of objects (*ie* data structures) that we can store the data in, which are:

- vectors
- matrices
- arrays
- data frames
- lists
- factors.

The type of object is called its **class**. We can find the class of an object by using class(<object>). The class of object tells R how functions interact with it. For example using a print command on a vector object just displays its contents but using it on a function returns its output:

```
A<-5

print(A)

print(log(A))
```

2 Vectors and matrices

The default object type is a vector, which is a one-dimensional ordered collection of data of the *same type*. You can find out whether an object is a vector by using is.vector(<object>). The dimension is called length and can be found by using the length(<vector object>) command.

Everything we have worked with so far has been a vector. Even unassigned values are considered vectors with a single element. For example:

```
is.vector(5)
```

```
TRUE
```

```
length(5)
```

```
1
```

Unlike a vector in maths, which contains only numbers, vectors in R can contain any of the five types of data that we met in the previous chapter (numeric, character, logical, complex and raw).

Examples of vectors would include:

$$\begin{pmatrix} 3.7 \\ -1.4 \end{pmatrix} \quad \begin{pmatrix} \text{"bob"} \\ \text{"larry"} \\ \text{"ginger"} \end{pmatrix} \quad \begin{pmatrix} \text{TRUE} \\ \text{TRUE} \\ \text{FALSE} \\ \text{TRUE} \end{pmatrix} \quad \begin{pmatrix} 2-3i \\ 5i \\ 4+8.7i \end{pmatrix} \quad \begin{pmatrix} 7 \end{pmatrix}$$

However R displays vectors horizontally rather than vertically. We will look at vectors in detail in the next chapter.

A **matrix** is a two-dimensional object containing data of the same type. It is essentially composed of several vectors of the same length. You can find out whether an object is a matrix by using is.matrix(<object>). The dimensions are called rows and columns and the numbers of each can be found by using the nrow(<matrix object>) and ncol(<matrix object>), respectively. Alternatively, you could use the dimensions command dim(<object>) to get both the number of rows and columns.

Again, unlike matrices in maths which contain only numbers, matrices in R can contain any of the five types of data (numeric, character, logical, complex and raw). Examples include:

$$\begin{pmatrix} 3 & -2.1 \\ 4.9 & 8.6 \end{pmatrix} \quad \begin{pmatrix} \text{"barry"} & \text{"alice"} \\ \text{"harry"} & \text{"belinda"} \\ \text{"larry"} & \text{"chelsea"} \end{pmatrix}$$

More on matrices in a later chapter.

3 Arrays, data frames, lists and factors

An **array** is an n-dimensional object containing data of the same type. For example it might have 3 dimensions length, width and height. You can find out whether an object is an array by using is.array(<object>). We can find the dimensions of an array by using the dimensions command dim(<object>).

A **data frame** is a two-dimensional object (like a matrix). However, whilst each column (*ie* vector) contains data of the same type the different columns (*ie* vectors) can be a different data type. This will be most useful for statistical analysis where each row represents a single observation (*eg* a single policyholder). For example, a data frame could include policyholders' names, their ages and their smoker status:

$$\begin{pmatrix} \text{Alfie} & 34 & \text{TRUE} \\ \text{Belinda} & 28 & \text{FALSE} \\ \text{Charlie} & 31 & \text{FALSE} \\ \text{Delilah} & 38 & \text{TRUE} \end{pmatrix}$$

You can find out whether an object is a data frame by using is.data.frame(<object>). We will look at data frames in more detail in a later chapter.

A **list** is a one-dimensional ordered collection of data (like a vector) but the data items don't have to be the same type. We can have lists of things like vectors, matrices and data frames and even lists! An example might be:

$$\begin{pmatrix} 3.7 \\ 2-3i \\ \text{"Bob"} \\ \text{FALSE} \end{pmatrix} \quad \begin{pmatrix} \begin{pmatrix} \text{"Open"} & \text{TRUE} \\ \text{"Closed"} & \text{FALSE} \end{pmatrix} \\ \begin{pmatrix} 3.1 \\ 4.3 \end{pmatrix} \end{pmatrix}$$

You can find out whether an object is a list by using is.list(<object>). Like a vector, the dimension is called length and can be found by using the length(<list object>) command.

Factors are vectors of characters where the entries are categorical data (*eg* gender, insurance group, country). Each entry can only take one of a specified number of categories (*eg* male/female, or groups 1-15, or UK, US, *etc*). We call these categories the *levels* of the factor. By default, R will assign the levels alphabetically (so female=1 and male=2). If the categorical data are ordinal (*eg* high/medium/low), then we use an ordered factor.

We have to be a bit careful when importing data into R as it often assumes that character data are factors (for example policyholder's names). So we might need to use coercion to tell R what type of data values they are. We met coercion in the last chapter.

4 Summary

Key terms

Object	Something which stores data which R can perform commands on
Vector	A one-dimensional, ordered collection of data of the same type
Matrix	A two-dimensional object containing data of the same type
Array	An n-dimensional object containing data of the same type
Data frame	A two-dimensional object, each column containing data of the same type – different columns can be of different data types
List	A one-dimensional, ordered collection of data (of different types)
Factor	vector of characters where the entries are categorical data (*eg* gender, insurance group, country)

Key commands

class(<object>)	Displays the class of an object
is.<object type>(<object>)	Logical test of whether <object> has the <object type> Returns TRUE or FALSE. Includes: is.vector( ), is.matrix( ), is.array( ), is.data.frame( ), is.list( ) and is.factor( ).
dim(<object>)	Displays the dimensions of a matrix, data frame or array
nrow(<object>)	Displays the number of rows in a matrix or data frame
ncol(<object>)	Displays the number of columns in a matrix or data frame

There is not a "Have a go" section in this chapter as we explore the key types of objects in more detail in the next few chapters.