Recognizing data types

Variables in R can contain data of various types. The most frequently used data types of variables in R programming are listed in the table below, together with a brief description:

Data type:	Description:	Example:
Character	A text character or string	"R" "R string"
Double	A decimal number	3.14
Integer	A whole number	5
Boolean	A logical value	TRUE

Hot tip

These four data types are sometimes referred to as the "atomic" or "primitive" data types as they represent the lowest level of data detail.

Unlike many other programming languages, which require the programmer to explicitly specify the data type when creating a variable, R automatically determines the variable data type according to the value it contains. The data type of a variable can be revealed by specifying its name as the argument to the built-in **typeof( )** function.

It is important to recognize that numeric variables are, by default, always created as a double data type unless an assigned integer value is suffixed by a letter **L**. For example, **number = 5L** creates an integer data type, but **number = 5** creates a double data type. More memory is allocated for the double data type, so integer values can be stored more efficiently if they are explicitly assigned to the integer data type.

R provides several built-in functions to test the data type of a variable. The name of a variable can be specified as the argument to the **is.character( )** function, which will return a Boolean value of **TRUE** or **FALSE** according to the data type of the variable. There are also **is.double( )**, **is.integer( )**, and **is.logical( )** functions that can be used in a similar manner to test the data type of a variable.

Boolean values can be assigned to a variable using either the keywords **TRUE** and **FALSE**, or simply by using the letters **T** and **F**.

Don't forget

Note that in R the Boolean values must appear in uppercase.

1. Open the RStudio Code Editor and create a variable that contains a text string value
   title <- "R for Data Analysis"

DataType.R

2. Assign a string and data type to a second variable
   result <- paste("Type of title:", typeof(title))

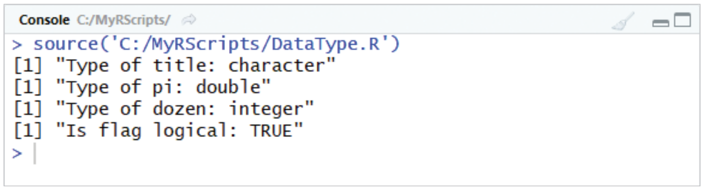
3. Output the combined string to see the variable's data type
   print(result)

4. Next, create a variable containing a double value and a variable containing an integer value
   pi <- 3.14159265
   dozen <- 12L

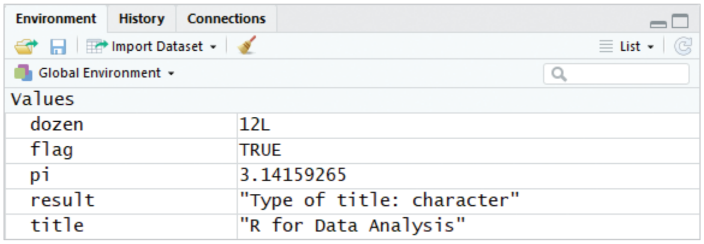
5. Output the data type of each variable in the previous step
   print(paste("Type of pi:", typeof(pi)))
   print(paste("Type of dozen:", typeof(dozen)))

6. Now, create a variable containing a logical value and output the result of a data type test on this variable
   flag <- T
   print(paste("Is flag logical:", is.logical(flag)))

7. Click the Source button in the Code Editor, or press **Ctrl** + **Shift** + **S**, to execute the script



```
> source('C:/MyRScripts/DataType.R')
[1] "Type of title: character"
[1] "Type of pi: double"
[1] "Type of dozen: integer"
[1] "Is flag logical: TRUE"
>
```



Values

dozen	12L
flag	TRUE
pi	3.14159265
result	"Type of title: character"
title	"R for Data Analysis"

Hot tip

Notice how this example includes function calls as arguments to other functions. The innermost function calls are executed first, passing their result to the outer function as their argument value.

Don't forget

The Environment tab lists the variables in alphabetical order, not in the order in which they are created.