R3

Basic functions

Covered in R3

- Functions
- Log and exponential
- Square root
- Trigonometric functions
- Factorial and Choose

1 Introduction

In the previous chapter we covered the common arithmetic operators. In this chapter we're going to look at functions for simple mathematical operations (such as logs and trig functions) that we can find on a scientific calculator.

There are literally hundreds of functions included in the standard R program that cover mathematical operations, statistical analysis, graphing and many other purposes.

We can also get extra functions for specific features (such as time series analysis, twitter data mining or prettier graphics) by downloading something called packages. We'll cover how to do this in a later chapter.

The form of a function

All functions in R operate on the following principle:

<function name> (<value1>, <value2>, … , <option1>, <option2>, …)

arguments

You have the name of the function, followed by brackets. Inside the brackets you put the arguments of the function in a specified order separated by commas. There are two types of arguments: the values you're going to put into the function and then the options. If we omit any options then R will use the default settings.

Note that both brackets are needed even if there are no arguments. For example, in an earlier video, we introduced the command for quitting R which was quit( ) or q( ). It has no arguments – but it's still got both brackets, which RStudio will remind you by automatically inserting the closed bracket.

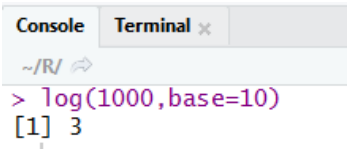
2 Log, exponential and square root

Let's start with the log function.

The function name is, unsurprisingly, log. The value is, unsurprisingly, the value we wish to find the log of and there is only one option which is the base of the log.

```
log(1000, base = 10)
```

```
3
```



We see the answer is 3 as 10^3 = 1000.

Again note the **index number** [1] at the beginning of the output line – which, as mentioned previously, gives the index number of the first answer on that output line.

Recall that R ignores spaces in commands, whereas spaces make life a bit easier for us humans to read the commands. So we could put spaces after the comma, or around the equals or anywhere that helps.

We could abbreviate argument names (as long as the abbreviation is unique)

```
log (1000, b=10)
```

```
3
```

or (being quite risky for an actuary) we could omit the option names altogether.

Just make sure you've got all the arguments in the brackets in the correct order!

```
log (1000, 10)
```

```
3
```

Let's now omit the option altogether.

```
log(1000)
```

```
6.907755
```

R uses its default option setting which is base e, *ie* natural log (in this case ln 1000).

Non-numeric outputs

Let's just remind ourselves of some **unusual outputs** that we might get.

```
log(0)
```

```
–Inf
```

This obviously stands for –infinity.

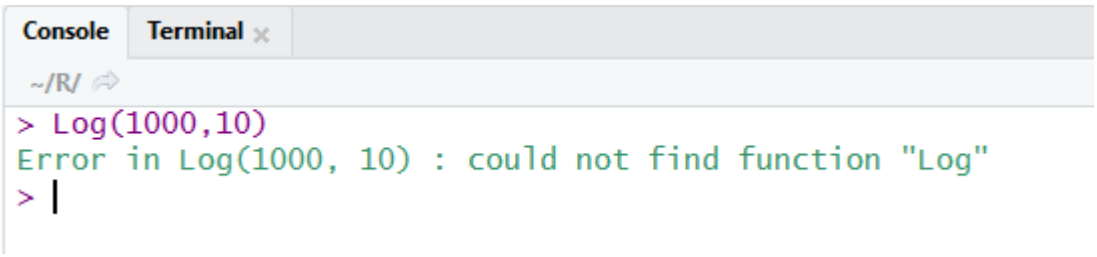
```
log(-5)
```

```
NaN
```

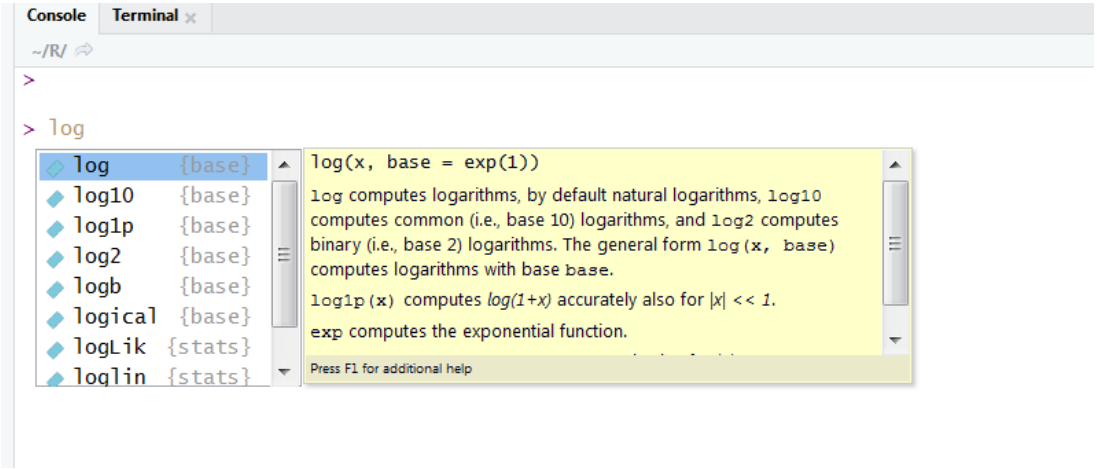
Recall that this stands for “not a number” as the log function is undefined for negative numbers.

Errors in commands

Finally recall that R is **case sensitive**. So let’s try that command again (you can use the up cursor key to get this command to reappear). If you type log with a capital L you’ll get this output:



So remembering the names of functions and their arguments is important and we’ll look at some more ways to do this in the next chapter. However, you have probably already noticed that as you type in functions in RStudio, it tries to predict what you are trying to do and also displays information about the function selected. You can scroll down through the suggestions to find the one you’re looking for.



If the function you are looking for is selected before you have finished typing, you can press enter to jump to the arguments.

Exponential

The exponential function is exp(x)

The only argument is the value you're finding the exponential of. It doesn't have any options.

exp(-5)

0.006737947

Square root

We covered powers in the previous chapter so we could calculate the square root either by taking the power of a half:

36^(1/2)

6

Or we can use the dedicated sqrt function:

sqrt(36)

6

3 Trigonometric functions

As you'd expect these are given by sin( ), cos( ) and tan ( ).

The only argument is the value you're inputting and there are no options.

```
sin(30)
```

```
-0.9880316
```

As you can see the function assumes the input is in radians not degrees ($\text{radians} = \text{degrees} \times \frac{\pi}{180}$).

We can enter pi as pi in R:

```
pi
```

```
3.141593
```

```
sin(pi/6)
```

```
0.5
```

```
cos(pi)
```

```
-1
```

```
tan(pi/2)
```

```
1.633178e+16
```

This should be undefined. Why is it not? Because the value stored for pi is a little limited. There are ways of getting more accurate values that we won't cover here. Similarly:

```
sin(pi)
```

```
1.224606e-16
```

This should be zero but is again hindered by the rounded value of pi.

The inverse trig functions arcsin, arcosine and arctangent are: asin( ), acos( ) and atan( ).

Again there are no options.

```
acos(1)
```

```
0
```

Hyperbolic trigonometric functions

Hyperbolic functions are defined in a similar way:

sinh(x)	cosh(x)	tanh(x)
asinh(x)	acosh(x)	atanh(x)

4 Factorials and combinations

To calculate the factorial we, unsurprisingly, use the function factorial(x). The only argument is the value we're finding the factorial of. There are no options:

factorial(6)

720

Let's find the factorial of a different number. Recall that we can use the up arrow ↑ to cycle through the previous commands. Use this to bring up the previous factorial command and change it to calculate 0!

factorial(0)

1

Finally to calculate combinations, ${}^{n}C_{k}$, we use the choose function:

choose(n, k)

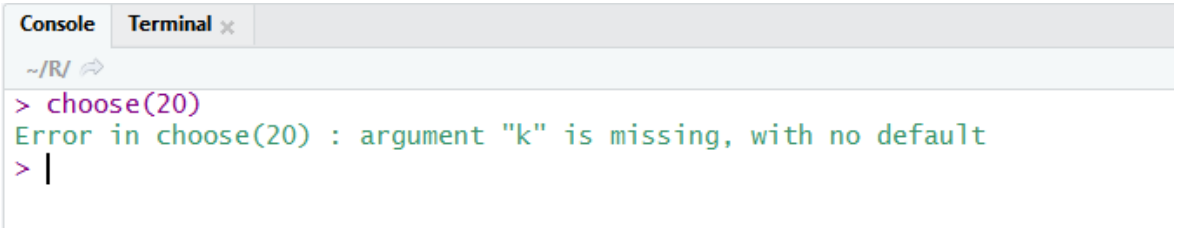
This has two arguments which are the values n and k. There are no options. For example:

choose (10,3)

120

Errors in commands

What happens if we enter only one of the two values? Say choose(20)?



R tells us helpfully that we've missed an argument and there's no default. Which is a polite way of telling us we've made a mess of things.

5 Summary

Key terms

Function	An R command that has inputs = <value1>, <value2>, … and also has options for how it works. It has the following form: <function name> (<value1>, <value2>, … , <option1>, <option2>, …)
Arguments	The values and options of a function
Default option	If an option is omitted in a function it will take the default

Key commands

log(x, base = b)	$\log_b x$	default base= exp{1} , *ie* it gives $\ln x$
log(x)	$\ln x$	
exp(x)	e^x	
sqrt(x)	$\sqrt{x}$	
sin(x), cos(x), tan(x)	trig functions sine, cosine and tangent, in radians	
asin(x), acos(x), atan(x)	inverse trig functions, in radians	
sinh(x), cosh(x), tanh(x)	hyperbolic trig functions	
asinh(x), acosh(x), atanh(x)	inverse hyperbolic trig functions	
factorial(x)	$x!$	
choose(n, k)	${}^nC_k$	

6 Have a go

You will only get proficient at R by practising:

1. Without referring to the previous chapter, use R to calculate the following:

   $3\times(2+5)$

   4^{-5}

   $8 \div 0$

2. Without referring to this chapter, use R to calculate the following:

   $\log(100)$

   $\ln 5$

   $\log_2 32$

   $\ln(-10)$ what does this output mean?

   $\log 0$ what does this output mean?

   e^{-3}

   $\frac{-3+\sqrt{7}}{4}$

   $\sin(\frac{\pi}{4})$, $\cos(\frac{\pi}{6})$, $\tan(\frac{\pi}{2})$

   $\sin^{-1}(1)$, $\cos^{-1}(0)$, $\tan^{-1}(½)$

   $\sinh(5)$, $\cosh^{-1}(3)$

   $10!$

   ${}^{8}C_4$