Functions

CSC121

Mark Kazakevich

Last time

- Finished learning about variables and assignment statements

- We learned what a **function** can do:
  - Take some input
  - Do something with that input
  - Produce some output

- Saw some examples of functions (and using them)

Today

- Talk a bit about the motivation behind functions
- See how R runs functions when they are “called”
- Talk about “return values” and how to use them in the console
- Talk a little more about features of RStudio

Functions seem useful!

- Without using functions in R, all we have is a few math operators:
  - +, -, *, //, %%, etc.
- But we want to do more than that!
- The functions we've used so far demonstrate that indeed we can do more than just basic operations in R

But notice something...

- When I told you about functions like sqrt(x) or abs(x), how much time did you spend thinking about **how** they work?

- Most likely, not much.
  - You probably just thought "Ok, cool. R can find the square root." and just used the function to do your work

- That's a good thing! And….
  - It's actually part of what functions are all about!

Functions are meant to hide the details

- We don't know what the computer actually does to calculate the square root of a number
- If we needed to know how every function works before we can use it, we would never get any work done!
- Right now, we want to use functions to do interesting things
- When we see something like sqrt(x), we assume it's going to work as expected:
  - We assume the people who made R also made a good square root function for it

So how do functions in R work?

How functions work

The three parts to a what a function does:

1. Take some input
2. Do something with that input
3. Produce some output

This is a broad description for any function (in math, programming, etc.)

Let's make this description more specific to R, by changing parts 1 and 3 a bit

How functions work

The three parts to a what a **function in R** does:

1. **'Call' the function with some input**
2. Do something with that input
3. **Produce a 'Return value'**

- Because we don't know what each function does with the input (part 2) at the moment, we're going to focus on the first and last parts

Part 1: “Calling” the function

- Whenever we type a function in the R Console and press enter, we are **‘calling’** that function

```
> sqrt(9)
```

“I called the function ‘sqrt’ with the value 9”

- The ‘9’ is called an **argument** to the function

Arguments

- The value that we pass to the function is called an ‘argument’
- Each argument has a value

```
> sqrt(9)
```

Argument value: 9

Arguments (con't)

- The argument can be a **literal** value

```
> sqrt(9)
```

Argument value: 9

- Or, it can be an expression

```
> sqrt(10 + 6)
```

Argument value: 10 + 6 = 16

- Because arguments can be expressions, we have to evaluate them before the function can work with them

Arguments (con't)

- Functions can have more than one argument, and we'll look at those next week

Two steps in calling a one-argument function

```
function(argument)
```

Step 1: Evaluate the argument given to the function to produce a value

Step2: Call the function with that value

function(argument)

```
sqrt(9)
```

Step 1: Evaluate the argument given to the function to produce a value

9

Step 2: Call the function with that value

The function sqrt(x) is called with the argument 9

R Console:

```
> sqrt(9)
```

Press ‘Enter’ to call the function

We’re not worried about what value it produces at the moment;
We’re just concerned with calling the function.

function(argument)

```
sqrt(15 + 10)
```

Step 1: Evaluate the argument given to the function to produce a value

15 + 10 = 25

Step 2: Call the function with that value

The function sqrt(x) is called with the argument 25

R Console:

```
> sqrt(15 + 10)
```

Press ‘Enter’ to call the function

```
a <- 48
sqrt(a + 1)
```

Steps for the function call (not the variable assignment):

Step 1: Evaluate the argument given to the function to produce a value

a + 1 = 48 + 1 = 49

Step 2: Call the function with that value

The function sqrt(x) is called with the argument 49

R Console:

```
> a <- 48
> sqrt(a + 1)
```

Press ‘Enter’ to call the function

So that's part 1

The three parts to a what a **function in R** does:

1. **'Call' the function with some input**
2. Do something with that input
3. **Produce a 'Return value'**

- Because we don't know what each function does with the input (part 2) at the moment, we're going to focus on the first and last parts

Part 3: Produce a “Return value”

- In the console, after we press Enter and the function is called, we saw that we get back a value

```
> sqrt(9)
> [1] 3
```

  - The ‘3’ that we get back after calling sqrt(9) is called the **return value** to the function

Return Values

- The return value of a function is the value that the function **evaluates** to.

- For example, we say `sqrt(9)` **evaluates** to 3

- We can use the return value of the function the same way we use the value we get from an expression
  - We can assign it to a variable
  - We can use it as an argument to another function

```
p <- sqrt(9)
```

A variable assignment and a function call!

Overall, this is a variable assignment, so….

Step 1: Evaluate right hand side to get a value

`sqrt(9)` # a function call! Let's do our function call steps

- Step 1: Evaluate the argument to the function to get a value

  9

- Step 2: Call the function with with that value

  The function sqrt(x) is called with the argument 9

  Now we get a return value from the function

  3 # the square root of 9 is 3

  The right side of the assignment statement evaluates to 3

  Now back to step 2 of the variable assignment

Step 2: Assign the value to the variable name on the left hand side

p is assigned the value 3

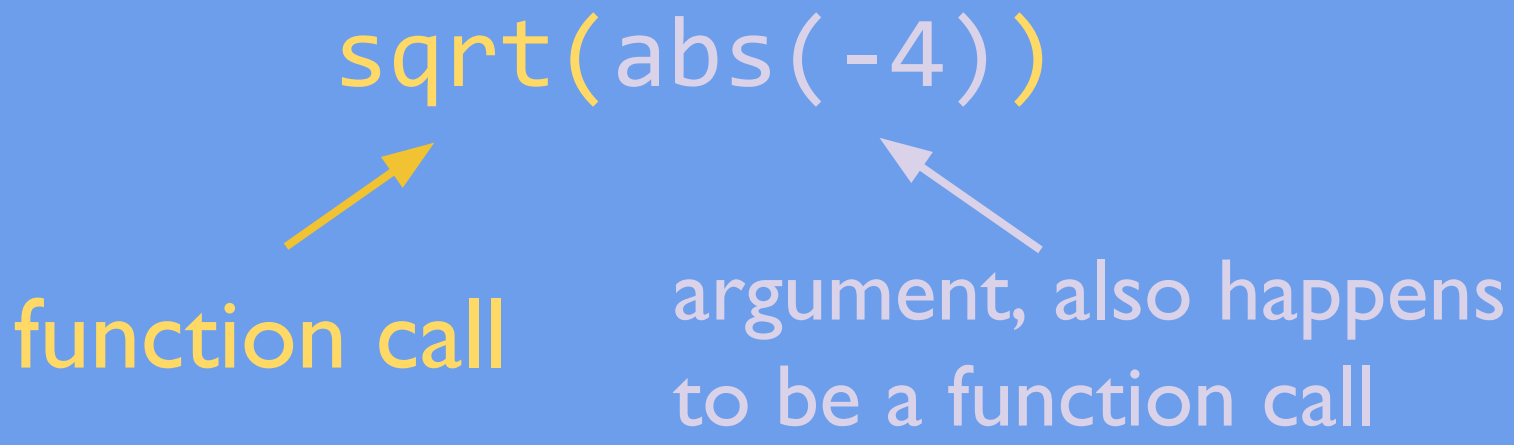
```
t <- sqrt(abs(-4))
```

Again, this is a variable assignment

Step 1: Evaluate right hand side to get a value

`sqrt(abs(-4))`

The argument to the sqrt function is also a function call!



`sqrt(abs(-4))`

Step 1: We have to evaluate all the arguments to the `sqrt` function before we call it

So, we need to evaluate `abs(-4)`

abs(x) returns the absolute value of x

So `abs(-4)` = 4

So the argument to the `sqrt` call is 4.

Step 2: We call the function sqrt with the value 4

And finally, we get our return value:

`sqrt(4)` = 2

The final result in the console

```
> t <- sqrt(abs(-4))
> t
> [1] 2
```

Today, we talked about Parts 1 and 3

The three parts to a what a **function in R** does:

1. **'Call' the function with some input**
2. Do something with that input
3. **Produce a 'Return value'**

We'll talk about Part 2 next week

To the console!