

MITOCW | MIT15_071S17_Session_1.3.04_300k

You should have already downloaded and installed R on your computer.

If you have not downloaded and installed R yet, please follow the instructions on the course website before continuing with this lecture.

Once you've installed R, go ahead and start it.

You should see the R console like we see here.

This is where we'll type commands and perform data analysis.

In this course, we'll mostly be working in the R console, but sometimes you'll also want to use what are known as "script files." We'll see an example of how to use a script file at the end of this lecture.

In your R console, you'll always type commands after the arrow, or "greater than" sign.

Let's start with some basic calculations.

So with your cursor at the arrow, let's type 8, and then the star symbol for "times," the number 6, and hit Enter.

You should see the result 48, or 8 times 6.

In this way, R can be used as a basic calculator.

We can perform many other basic calculations, like 2, "raised to the power," by typing the carrot symbol, and then the number 16, and hitting Enter.

You should see the result 65,536-- or 2 the power 16.

Note that each of the results here are labeled with a 1 in brackets.

This is just R's way of labeling the output and we can safely ignore it.

If you type something in your R console but don't finish it properly-- for example, try typing 2 and the carrot symbol, and hitting Enter-- R will show you a plus sign and will wait for you to finish the command.

You can either finish the command-- in this case, by typing a number-- or you can hit Escape, and R will take you back to the arrow sign.

So if you ever see a plus sign while working in R, it means that R is waiting for you to finish the line.

If you're not sure how to finish it, you can always just hit Escape.

A nice feature of the R console is that you can scroll through your previous commands by using the up and down arrows on your keyboard.

If you hit the up arrow three times, you should go back to the 8 times 6 command.

If you hit Enter, you can run this command again.

You could also adjust previous commands.

Hit the up arrow again to go back to the 8 times 6 command.

This time, delete the 6, type a 10, and hit Enter.

We'll use this approach to re-run or adjust commands many times in this class.

Generally, R works in terms of functions and variables.

A function can take in several arguments, or inputs, and returns an output value.

An example is the square root, or `sqrt`, function.

In your R console, type `sqrt`, and then in parentheses, the number 2, and hit Enter.

The function is `sqrt`, the input, or argument, is the number 2, and the output is 1.414214.

There are thousands of functions in R. Some of them are built into R-- like this one-- and some can be added in by installing packages, which we'll do several times in this class.

Another example of a function is the `abs`, or absolute value function, which returns the absolute value of a number.

So if we type `abs` and then in parentheses negative 65 and hit Enter, we should get the result 65 as our output.

You can get help on any function in R by typing a question mark and then the function name.

So if we type `?sqrt` and hit Enter, you should see the R Help Page for the `sqrt` function.

The help pages are often very useful, and if you want to learn more about a function, you should refer to the help page in R.

Now, get rid of the Help Page, and go back to your R console.

Suppose we now want to save the output of a function.

We can do this by saving it to a variable.

In your R console, type `SquareRoot2` and then an equals sign, and then `sqrt`, and in parentheses the number 2.

And hit Enter.

Now you don't see the output of `sqrt(2)` because we saved it to the variable named "SquareRoot2." You can see the value of a variable by typing its name and hitting Enter.

So type `SquareRoot2` exactly how you typed it before, and hit Enter.

You should see that it takes the value 1.414214.

`SquareRoot2` is a name that we created, and we could have named it many other things.

Generally, you have a lot of freedom in naming your variables, but there are a couple basic rules.

One is that you should not use spaces in variable names.

If you want to easily separate words in a variable name, popular strategies are using a mix of capital and lowercase letters-- as we did here-- or to separate the words using periods.

Another basic rule is that you should not start variable names with a number.

Keep in mind that variable names in R are case-sensitive.

Capital and lowercase letters are seen differently by R.

When we created our variable `SquareRoot2`, we used the equals sign for assignment, or to assign the value of `sqrt(2)` to the variable named `SquareRoot2`.

You could instead use a "less than" sign with a dash.

For example, let's create a new variable called `HoursYear`, and then type the less than sign and a dash, followed by 365 times 24.

If you hit Enter and then type `HoursYear` to look at its value, you should see 8,760-- or the number of hours in a year.

By doing this, the less than sign and the dash did the exact same thing as it would have done if we used the equals sign.

In this course, we'll typically use the equals sign.

But if you see this less than sign and a dash used, keep in mind that it means the same thing as an equals sign.

Lastly, if you type `ls` and then empty parentheses in your R console and hit Enter, you should see a list of all of the variables that you've created in your current R session.

Here, we've created two variables-- `HoursYear` and `SquareRoot2`.

This can be useful if you forget exactly how you typed a variable name.

These are our basic building blocks in R-- functions, like square root and absolute value, and variables, like `HoursYear` and `SquareRoot2`.