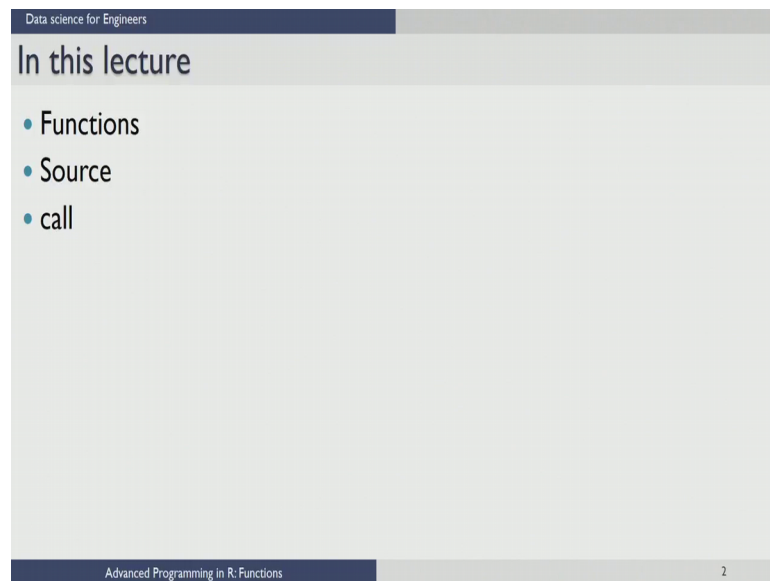


Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 08
Advanced programming in R: Functions

Welcome to the lecture 7 in the R module of the course Data Science for Engineers.

(Refer Slide Time: 00:21)



In this lecture we are going to introduce you to the functions in R. We are going to explain how to load or source the functions and how to call or invoke the functions.

(Refer Slide Time: 00:32)

The slide is titled "Functions in R" and is part of a presentation for "Data science for Engineers". It contains the following bullet points:

- A function accepts input arguments and produces output by executing valid R commands present in the function.
- Function name and file names need not be the same.
- A file can have one or more function definitions.
- Functions are created using the command `function()`

Below the list, a code snippet is shown in a box:

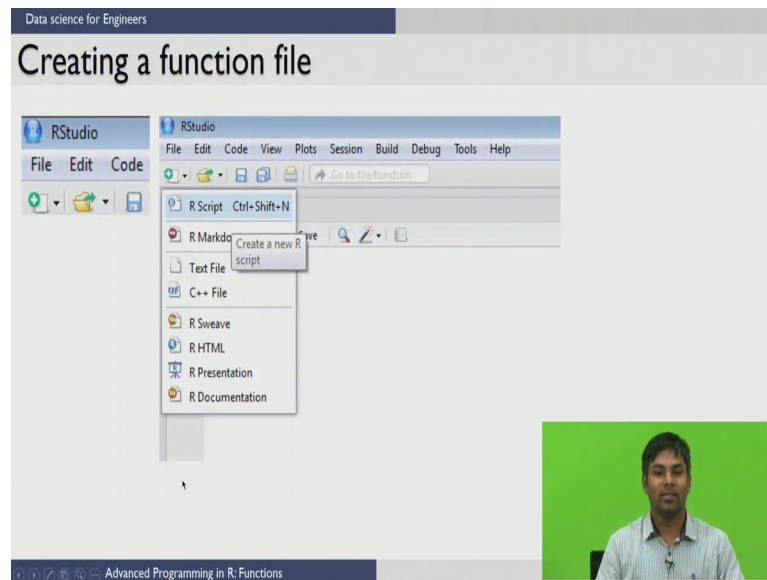
```
f = function(arguments) {  
  statements  
}
```

A small video inset in the bottom right corner shows a man with a beard and glasses, wearing a light blue shirt, speaking against a green background.

Functions are useful when you want to perform certain tasks many number of times. A function accepts input arguments and produces the output by executing valid R commands that are inside the function.

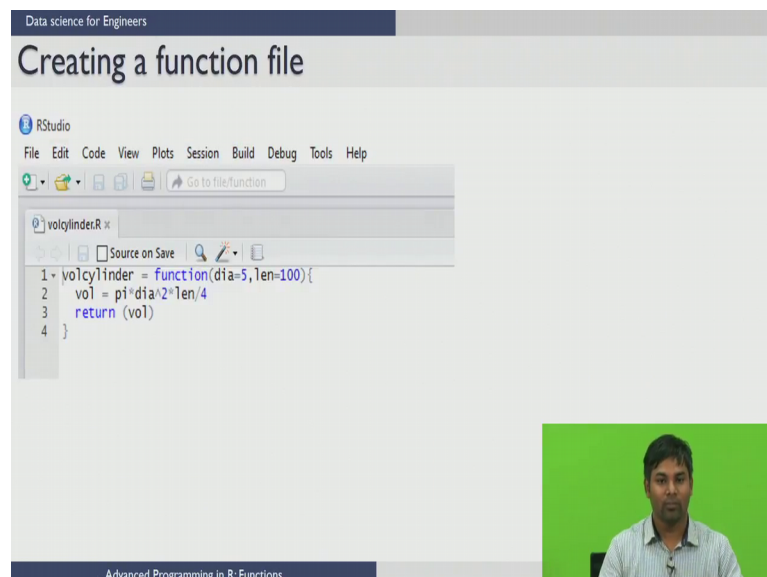
In R when you are creating a function the function name and the file in which you are creating the function need not be same and you can have one or more function definitions in a single R file. Functions are created in R by using the command `function`. The general structure of the function file is as follows `f` is equal to `function` of arguments and then you have statements that are needed to be executed. This `f` is the function name when you write this command this means that you are creating a function with name `f` which takes certain arguments and executes the following statements.

(Refer Slide Time: 01:31)



Let us see how to create a function file. Creating a function file is similar to opening an R script which we have already seen. You can either use file button in the toolbar or you can use the plus button just below the file tab to create an R script, once you create an R script you can save it with whatever name you want.

(Refer Slide Time: 01:56)

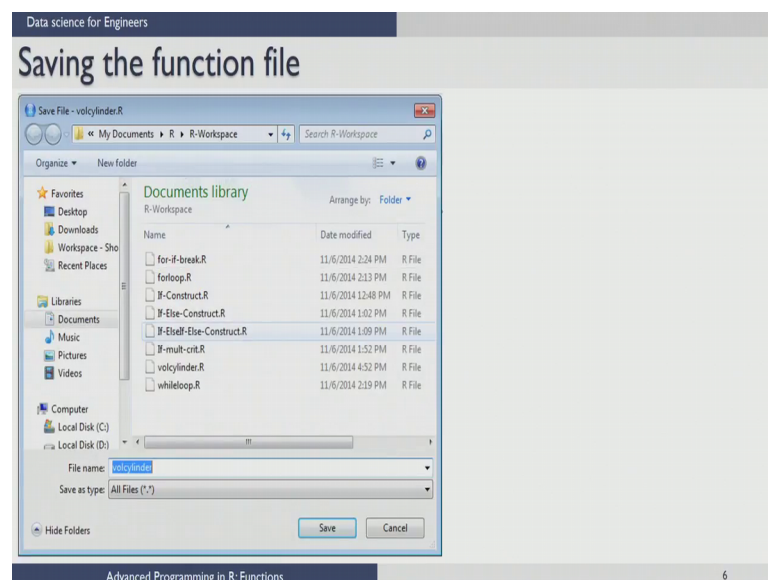


For example, we have saved the R file as vol cylinder. Now, once you save you are ready to write functions, now I want to create a function which calculates the volume of a cylinder which takes in the arguments the diameter and length. So, to create a function

by name volume cylinder I have to have the function named as volume cylinder function and the arguments that are needed to be passed are the diameter of the cylinder and the length of the cylinder.

If you notice here we are passing this values of 5 and 100 as a default arguments for this function. Once you have diameter and length you can calculate the volume by the formula $\pi d^2 l$ by 4 then what you need to return is the volume variable that is calculated inside the function. Once you have written the R statements that are needed to be executed in a function file, you can save that file. So, we are saving it as vol cylinder dot R.

(Refer Slide Time: 02:56)



Once you save this. So, you need to load the functions before you invoke or execute them in R. To load a function you need to click on the source button that is available in the R script menu.

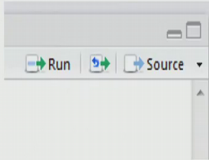
(Refer Slide Time: 03:02)

Data science for Engineers

Loading the functions

Function files have to be loaded before invoking (execution)

Loading a function file



The function file can also be loaded using the following command
> source('~/.R/R-Workspace/volcylinder.R')

Note: Clicking the "Source" button will not execute the function, it will only load the function file. After loading, the function can be executed by invoking the function

Advanced Programming in R: Functions 7

Clicking the source button will not execute the function; it will only load the function file and make it ready for invoking.

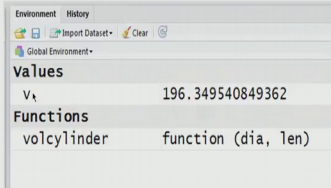
(Refer Slide Time: 03:26)

Data science for Engineers

Invoking the function from console

```
> source('~/.R/R-Workspace/volcylinder.R')
> v = volcylinder(5,10)
> v
[1] 196.3495
>
```

Variable Browser



| Values | |
|--------|------------------|
| v | 196.349540849362 |

| Functions | |
|-------------|---------------------|
| volcylinder | function (dia, len) |

Advanced Programming in R: Functions 8

Once you load the function, you can invoke the function from the console as follows you want the volume to be saved in the variable v and then you are calling this function vol cylinder with the arguments 5 and 10. So, this will run the function to calculate the volume and returns the volume. In the variable browser you can also see value of volume

and you can also see that the function volume cylinder is available with two arguments dia and length.

(Refer Slide Time: 03:56)

The slide is titled "Passing arguments to functions" and is part of a presentation for "Data science for Engineers". It lists three points about passing variables as arguments to functions:

- Passed in the same order as in function definition
- Names of the arguments can be used to pass their values in any order
- Default values are used if some or all arguments are not passed

Below the list, there are three R code snippets demonstrating the use of the `volcylinder` function:

| | |
|----------------------------------------------------------------------------|-----------------------------------------------------------|
| <pre>> vol = volcylinder(5,10) > vol [1] 196.3495</pre> | <pre>> vol = volcylinder() > vol [1] 1963.495</pre> |
| <pre>> vol = volcylinder(len = 10, dia = 5) > vol [1] 196.3495</pre> | |

The slide footer includes navigation icons and the text "Advanced Programming in R: Functions" and the number "9".

Now, there are several ways you can pass the arguments to the function. Generally in R the arguments are passed to the function in the same order as in the function definition. If you do not want to follow any order what you can do is you can pass the arguments using the names of the arguments in any order. If the arguments are not passed the default values are used to execute the function.

Now, let us see the examples for each of these cases when you pass the arguments 5 and 10 the first argument is diameter and second argument is length according to the definition of the function. So, it will take in the same way, but when you want not to follow any order you can pass the arguments by the names in any order. So, for example, I want to pass length as a first argument you can specify length is equal to 10 and diameter is equal to 5 and you can still see the result is same even though you pass the arguments in the different order.

So, point to keep in mind is you can pass the arguments in any order by specifying its name. When you do not pass any arguments here it takes the default values of 5 and 100 which are default diameter and length and then calculates the volume.

(Refer Slide Time: 05:26)

Data science for Engineers

Lazy evaluations of functions in R


- Functions are lazily evaluated, which means that if some arguments are missing, the function is still executed as long as the execution doesn't involve these arguments

```
> volcylinder = function(dia, len, rad){  
+ vol = pi*dia^2*len/4  
+ return(vol)}  
>  
> vol = volcylinder(dia = 5, len = 10)  
> vol  
[1] 196.3495
```

Argument `rad` is missing, but the function is executed

```
> volcylinder = function(dia, len, rad){  
+ vol = pi*dia^2*len/4  
+ print(rad)  
+ return(vol)}  
>  
> vol = volcylinder(dia = 5, len = 10)  
Error in print(rad) : argument "rad" is missing, with no default
```

Here `rad` is used in the function body, which throws up error





In R the functions are executed in a lazy fashion, when we say lazy what it mean is if some arguments are missing the function is still executed as long as the execution does not involve those arguments. We will see an example for this. We have the same function we have defined now an extra argument radius in the function and the volume calculation does not involve this argument radius in this calculation.

Now, when you pass this arguments dia and length even though you are not passing this radius the function will still execute because this radius is not used in the calculations inside the function. But R is clever enough, if you do not pass the argument and then use it in the definition of the function it will throw an error saying that this rad is not passed and it is being used in the function definition.

(Refer Slide Time: 06:28)

Data science for Engineers

Summary of function file creation and execution

1. Open a function file by clicking . First line of a function file should be `function_name = function (inputs)`. Type the necessary and valid R statements/commands to be executed
2. Save the function file
3. Load the function file by pressing 
4. Invoke the function with the right number of inputs to execute the function

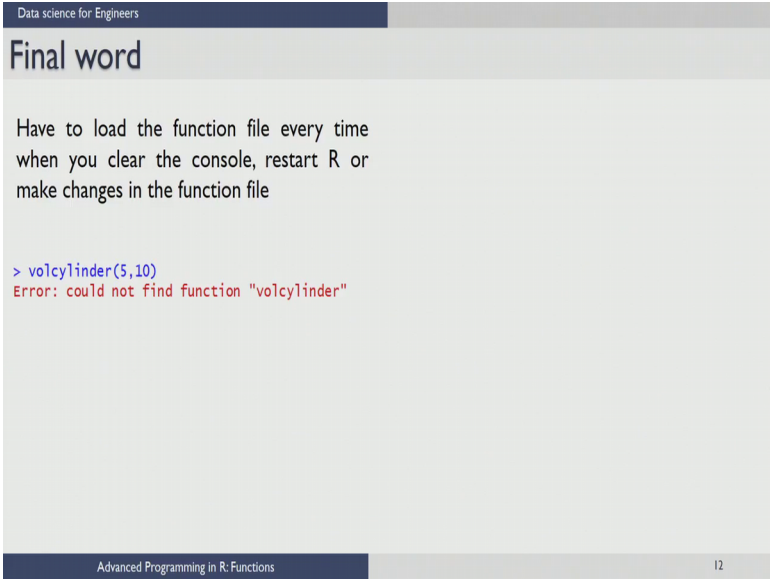
Advanced Programming in R: Functions

11

In summary these are the steps in creating a function file in R and executing. First we need to open or create a function file by clicking a that the plus symbol or file tab in the toolbar you have to define the function in this fashion function name, keyword function and the input arguments.

All the statements that are typed inside the function has to be valid R statements to be executed, and you need to save the function file before executing you need to load the function file by using the source button once you load the function file you can invoke or call the function file with the right number of inputs so that you will execute the function properly and you will get the required result.

(Refer Slide Time: 07:23)



Data science for Engineers

Final word

Have to load the function file every time when you clear the console, restart R or make changes in the function file

```
> volcylinder(5,10)  
Error: could not find function "volcylinder"
```

Advanced Programming in R: Functions 12

A final word we need to load the function file every time you change something inside the function definition either you restart R studio or make changes in the function file. If you do not do that either you get an error or you will not get correct outputs which you are expecting, because you would have changed something in the function definition and not saved the original version. Once you save the original version also you have to invoke the function before you use.

In the next lecture we are going to explain the functions which are having multiple inputs and multiple outputs.

Thank you.