

Advanced programming in R: Functions

In this lecture

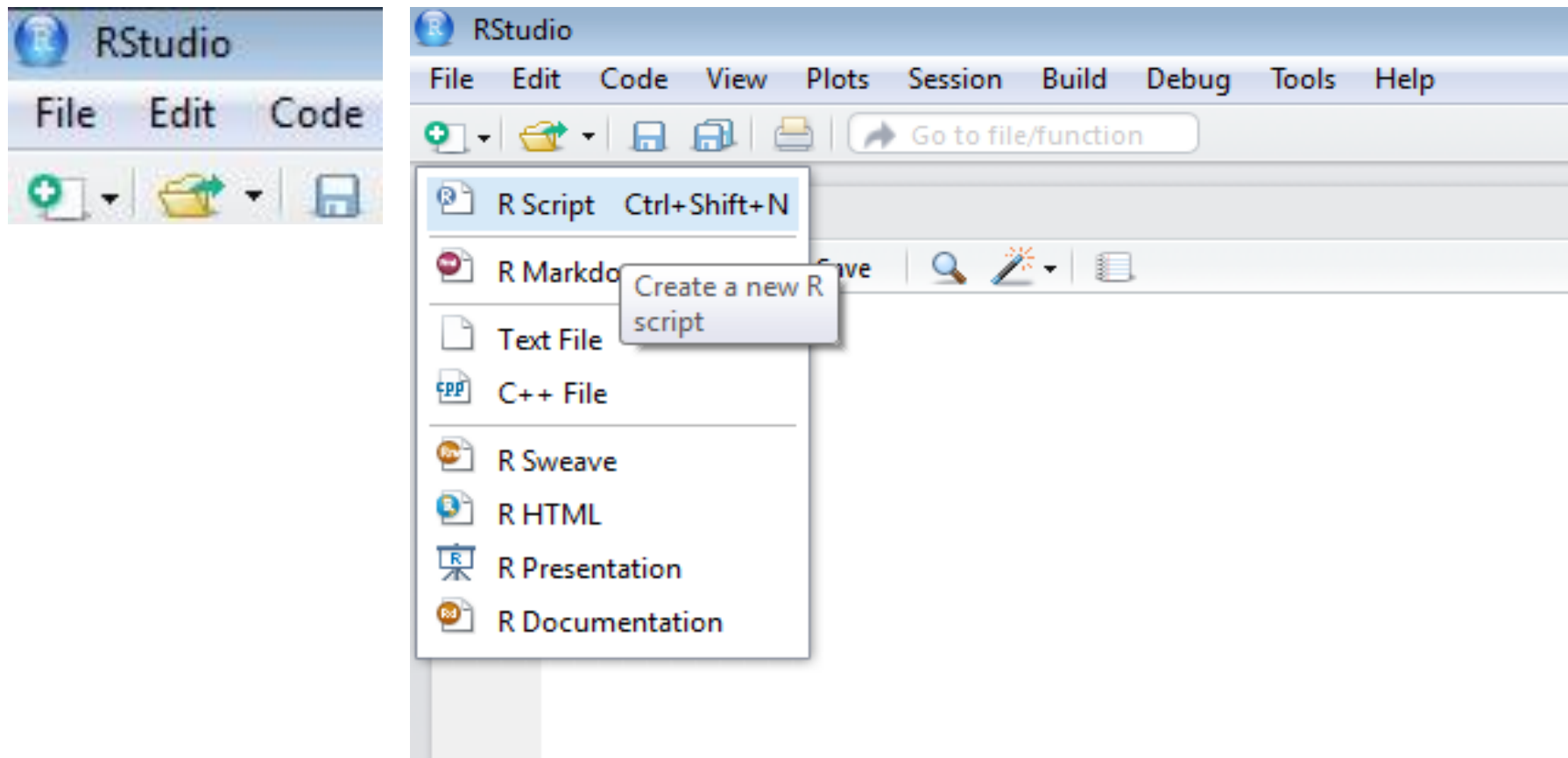
- Functions
- Source
- Call

Functions in R

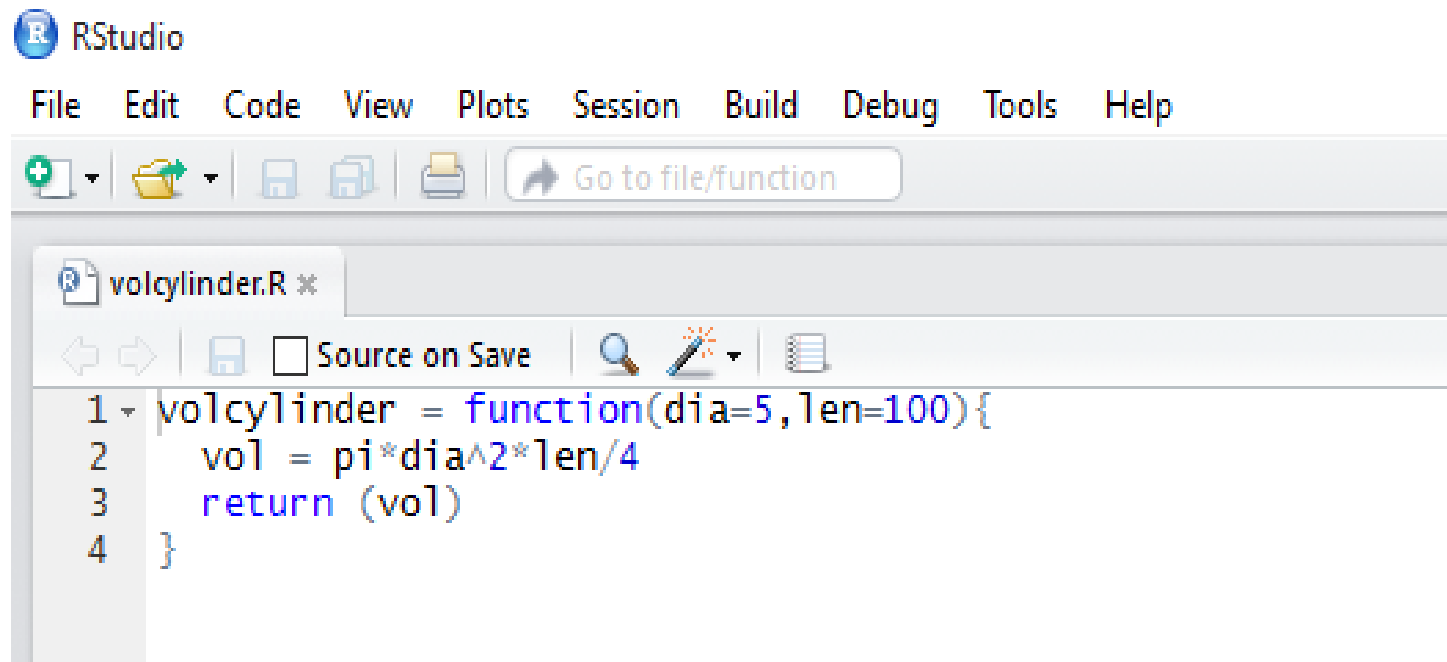
- A function accepts input arguments and produces output by executing valid R commands present in the function.
- Function name and file names need not be the same.
- A file can have one or more function definitions.
- Functions are created using the command `function()`

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

Creating a function file



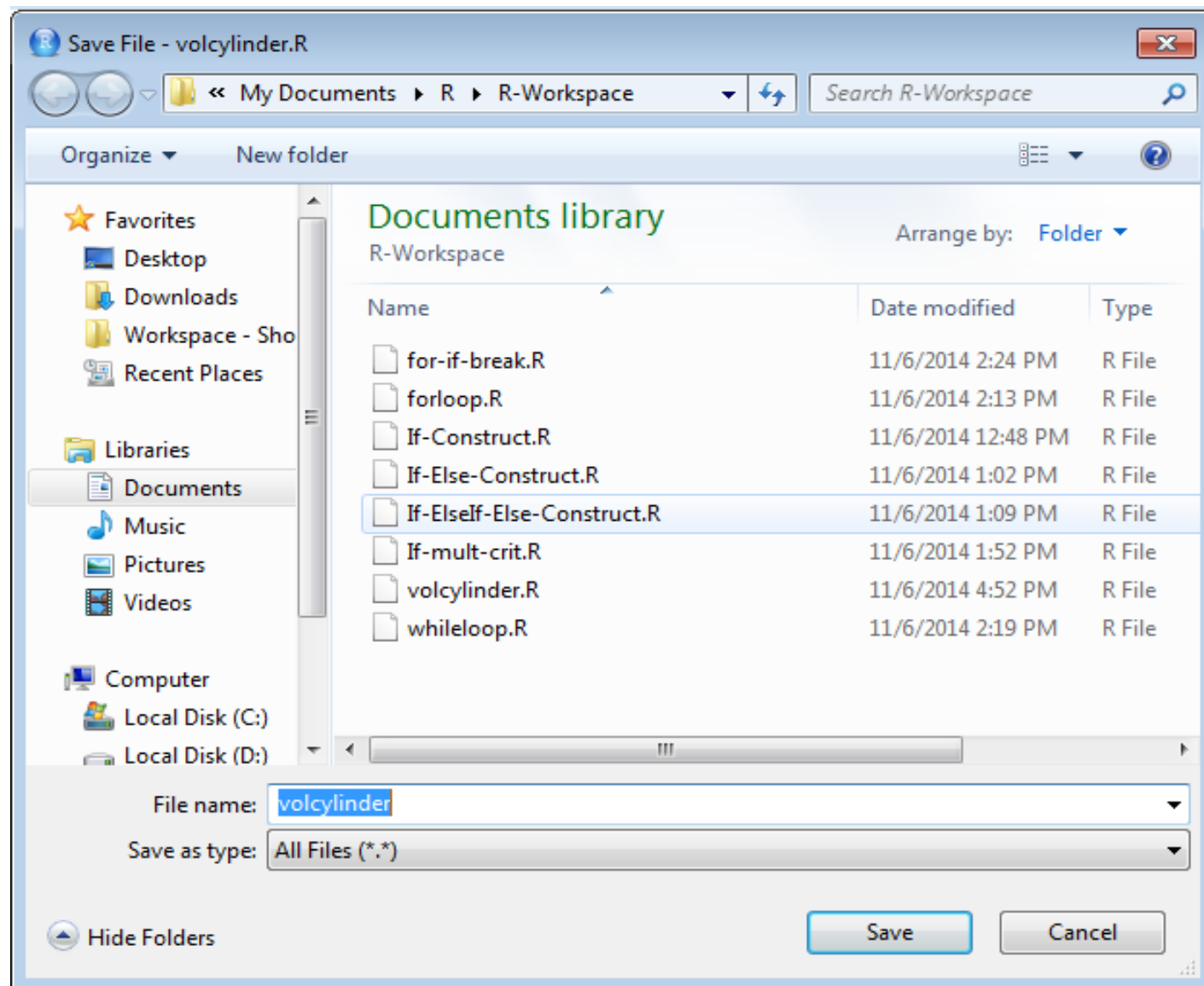
Creating a function file



The screenshot shows the RStudio application window. The title bar reads 'RStudio'. The menu bar includes 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Tools', and 'Help'. The toolbar contains icons for creating a new file, saving, and a search bar labeled 'Go to file/function'. The file explorer shows a single file named 'volcylinder.R *'. The source editor displays the following R code:

```
1 volcylinder = function(dia=5,len=100){  
2   vol = pi*dia^2*len/4  
3   return (vol)  
4 }
```

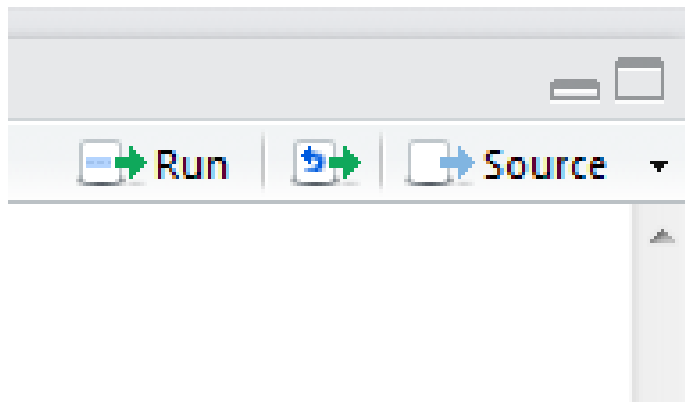
Saving the function file



Loading the functions

Function files have to be loaded before invoking
(execution)

Loading a function file



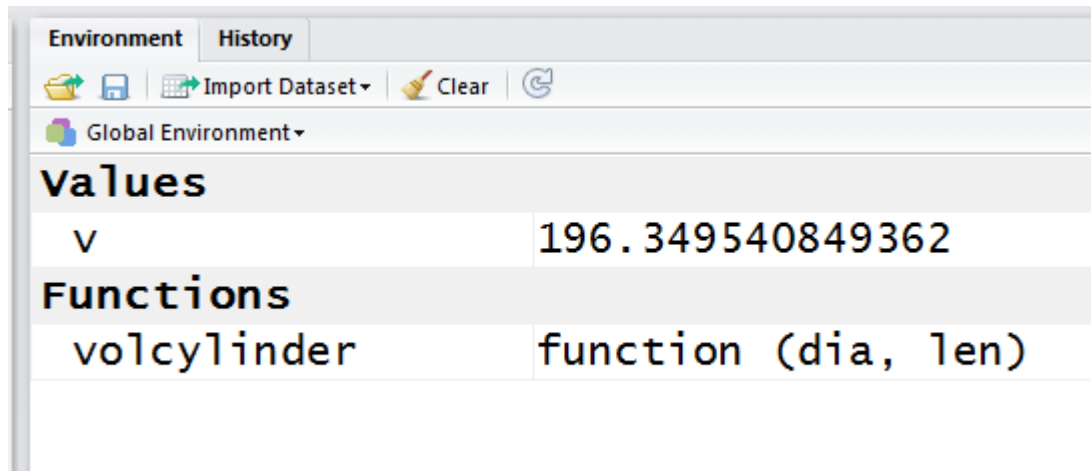
The function file can also be loaded using the following command
> source('~/.R-Workspace/volcylinder.R')

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

Invoking the function from console

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

Variable Browser



The screenshot shows the RStudio Variable Browser window. It has tabs for 'Environment' and 'History'. Below the tabs are icons for file operations and a 'Global Environment' dropdown. The main area is divided into 'values' and 'Functions' sections. The 'values' section shows a variable 'v' with the value '196.349540849362'. The 'Functions' section shows a function 'volcylinder' with the definition 'function (dia, len)'.

Environment	
History	
Global Environment	
values	
v	196.349540849362
Functions	
volcylinder	function (dia, len)

Passing arguments to functions

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

```
> vol = volcylinder(5,10)
> vol
[1] 196.3495
```

```
> vol = volcylinder()
> vol
[1] 1963.495
```

```
> vol = volcylinder(len = 10, dia = 5)
> vol
[1] 196.3495
```

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)
```

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

```
Error in print(rad) : argument "rad" is missing, with no default
```

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

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"
```