

Foundations of R Software

Lecture 24

Basics of Calculations

⋮

Functions

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Functions

- **Functions are a bunch of commands grouped together in a sensible unit**
- **Functions take input arguments, do calculations (or make some graphics, call other functions) and produce some output and return a result in a variable. The returned variable can be a complex construct, like a list.**

Functions : Components

Function Name – This is the name of the function which is stored as an object with this name.

Arguments – An argument contains the input values. A function may contain no arguments.

Function Body – Contains statements that defines what the function has to do.

Return Value – The evaluated value of the last expression in the function body.

Functions : Built-in functions

R has Built-in functions also.

Simple examples of in-built functions are

`sum()`, `prod()`, `mean()`, `max()`, `sum(x)` etc.

Functions

Syntax

```
Name <- function(Argument1, Argument2, ...)  
  {  
    expression(s)  
  }
```

where `expression(s)` is a single command or a group of commands.

You can use `=` operator also.

```
Name = function(Argument1, Argument2, ...)  
  {  
    expression(s)  
  }
```

Function arguments with description and default values

- Function arguments can be given a meaningful name
- Function arguments can be set to default values
- Functions can have the special argument '...'

Functions (Single variable)

The sign = or <- is furthermore used for defining functions:

```
> abc = function(x){  
      x^2  
}
```

A function is called with argument as

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

```
> abc(6)  
[1] 36
```

```
> abc(9)  
[1] 81
```

```
R Console  
  
> abc = function(x) {  
+       x^2  
+     }  
> abc  
function(x) {  
      x^2  
}  
  
> abc(3)  
[1] 9  
> abc(6)  
[1] 36  
> abc(9)  
[1] 81  
> |
```

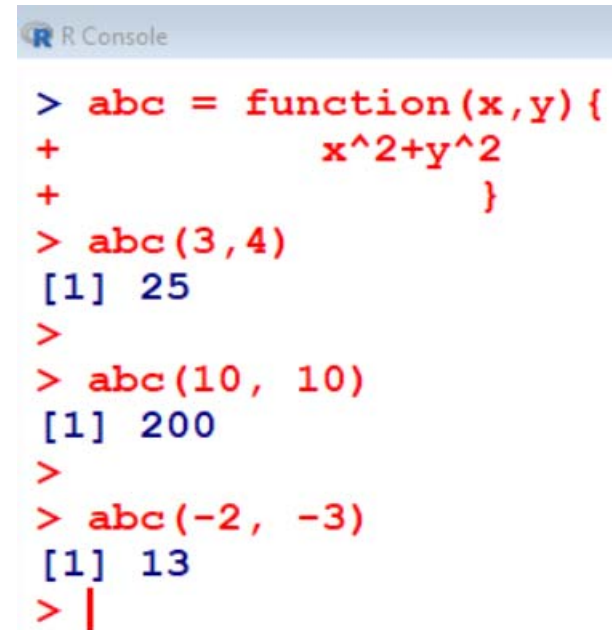
Functions (Two variables)

```
> abc = function(x,y){  
    x^2+y^2  
}
```

```
> abc(3,4)  
[1] 25
```

```
> abc(10, 10)  
[1] 200
```

```
> abc(-2, -3)  
[1] 13
```

A screenshot of an R console window with a light blue header bar containing the R logo and the text "R Console". The console shows the same R code and output as the previous blocks, with red text for commands and blue text for output.

```
> abc = function(x,y){  
+     x^2+y^2  
+ }  
> abc(3,4)  
[1] 25  
>  
> abc(10, 10)  
[1] 200  
>  
> abc(-2, -3)  
[1] 13  
> |
```

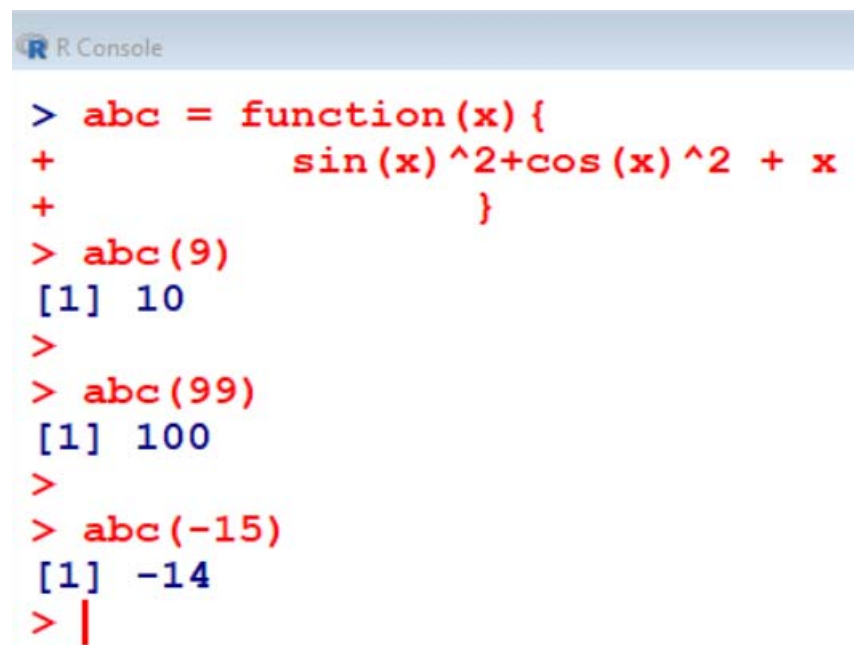
Functions- Another example

```
> abc = function(x){  
    sin(x)^2+cos(x)^2 + x  
}
```

```
> abc(9)  
[1] 10
```

```
> abc(99)  
[1] 100
```

```
> abc(-15)  
[1] -14
```

A screenshot of an R console window showing the execution of the function 'abc'. The console has a light blue header with the R logo and the text 'R Console'. The code entered is: > abc = function(x) { sin(x)^2+cos(x)^2 + x }, > abc(9), > abc(99), and > abc(-15). The corresponding outputs are: [1] 10, [1] 100, and [1] -14. The cursor is at the end of the last line.

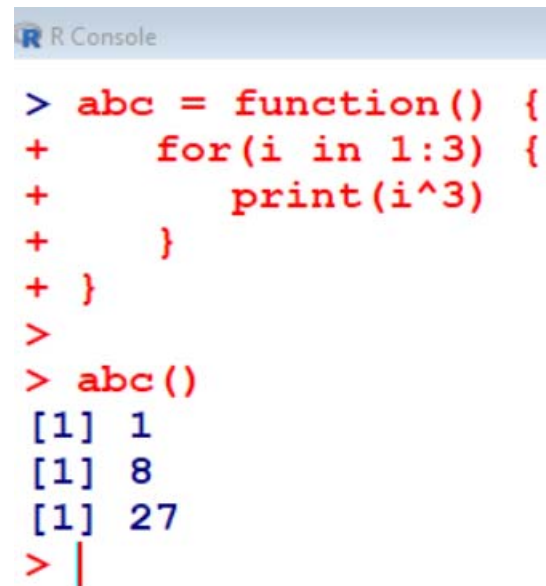
```
> abc = function(x) {  
+     sin(x)^2+cos(x)^2 + x  
+ }  
> abc(9)  
[1] 10  
>  
> abc(99)  
[1] 100  
>  
> abc(-15)  
[1] -14  
> |
```

Functions- Another example

Calling a function without an Argument

```
abc = function() {  
  for(i in 1:3) {  
    print(i^3)  
  }  
}
```

```
> abc()  
[1] 1  
[1] 8  
[1] 27
```

A screenshot of an R console window showing the execution of the function 'abc'. The console has a title bar that says 'R Console'. The input lines are in red, and the output lines are in blue. The function definition is entered first, followed by a call to 'abc()'. The output shows three lines, each starting with '[1]' followed by the values 1, 8, and 27 respectively. A cursor is visible at the end of the last input line.

```
R Console  
> abc = function() {  
+   for(i in 1:3) {  
+     print(i^3)  
+   }  
+ }  
>  
> abc()  
[1] 1  
[1] 8  
[1] 27  
> |
```