

Functions in R

Zina Ibrahim

Introduction to Functions in R

- R has many built-in functions
 - e.g. `length`, `dim`, `sqrt`, etc...
- The great thing about R (and other scriptable languages)
 - If you don't like how a function works, you can change it by writing your own.
 - If a function is missing, you can write it.

Functions in R

- A **data type** called 'function'
 - It is designed to perform the task you program it to.
- Has **arguments** (list of data to go in)
- Has a **body** (some code)
- You assign a **name** just like for any other variable
- You run it like this : `name()`
- It **returns** some results

Calculating the Cube Root of a Signed Number: Method 1

```
> x < -8  
> sign(x) * abs(x)^(1/3)  
[1] -2
```

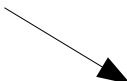
- **Problem:** inelegant and cumbersome to type every time.
- **Solution:** define your own cubeRoot function.

Calculating the Cube Root of a Signed Number: Method 2

```
CubeRoot <- function(num)
{
  answer <- sign(num) * abs(num)^(1/3)
  return (answer)
}
```

Calculating the Cube Root of a Signed Number: Method 2

The name of the function



```
CubeRoot <- function(num)
{
    answer <- sign(num) * abs(num)^(1/3)
    return (answer)
}
```

Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

```
CubeRoot <- function(num)
```

```
{
```

```
    answer <- sign(num) * abs(num)^(1/3)
```

```
    return (answer)
```

```
}
```

Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

Argument (input to the function)

```
CubeRoot <- function(num)
```

```
{
```

```
  answer <- sign(num) * abs(num)^(1/3)
```

```
  return (answer)
```

```
}
```


Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

Argument (input to the function)

```
CubeRoot <- function(num)
```

Delimits the
start of the
function body

```
{
```

```
  answer <- sign(num) * abs(num)^(1/3)
```

```
  return (answer)
```

```
}
```

Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

Argument (input to the function)

```
CubeRoot <- function(num)
```

Delimits the start of the function body

```
{
```

```
  answer <- sign(num) * abs(num)^(1/3)
```

```
  return (answer)
```

```
}
```

Statement(s) doing the work of the function

Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

Argument (input to the function)

```
CubeRoot <- function(num)
```

Delimits the start of the function body

```
{
```

```
  answer <- sign(num) * abs(num)^(1/3)
```

```
  return (answer)
```

```
}
```

Statement(s) doing the work of the function

The R function that creates returns the value computed by the function

Calculating the Cube Root of a Signed Number: Method 2

The name of the function

The R function that creates functions

Argument (input to the function)

```
CubeRoot <- function(num)
```

Delimits the start of the function body

```
{
```

```
  answer <- sign(num) * abs(num)^(1/3)
```

```
  return (answer)
```

Statement(s) doing the work of the function

```
}
```

Delimits the end of the function body

The R function that creates returns the value computed by the function

General Form

```
FunctionName <- function(arg1, arg2, ..)
{
    Statement 1
    Statement 2
    .....
    return (value)    ##Optional
}
```

Using Your Function

- Open your favorite editor
- Type in your function
- Save your file (e.g. cr.R)
- Navigate to the directory where the R file lives
- Load your function into R, e.g.
> `source("cr.R")`
- Use your function, e.g.
> `cubeRoot(-3)`

Variable Scope in Functions

- Once created, a function is a black box.
- Any variable defined inside the function (including the arguments) is local to the function and will not be visible to the outside world (i.e. R console)

cr.R

```
CubeRoot <- function(num)
{
  answer <- sign(num)
  * abs(num)^(1/3)
  return (answer)
}
```

R Console

```
> CubeRoot(-9)
[1] -2.080084

> num
Error: object 'num' not found

> answer
Error: object 'answer' not found
```

Why Use Functions?

- Re-use:
 - Perform the same task repeatedly with different data as input without having to change the function's code
- Modularity:
 - Your program becomes modular.
 - a number of task-specific functions that get called whenever they are needed.
 - Cleaner code!
 - You become more productive with time.
 - You can share functions!