

Wrapping up R packages for the holidays

What is a function in R?

An R function is created by using the keyword *function*. The basic syntax of an R function definition is as follows:

```
function_name <- function(arg_1,..) {  
  Function_body  
}
```

The different parts of a function are:

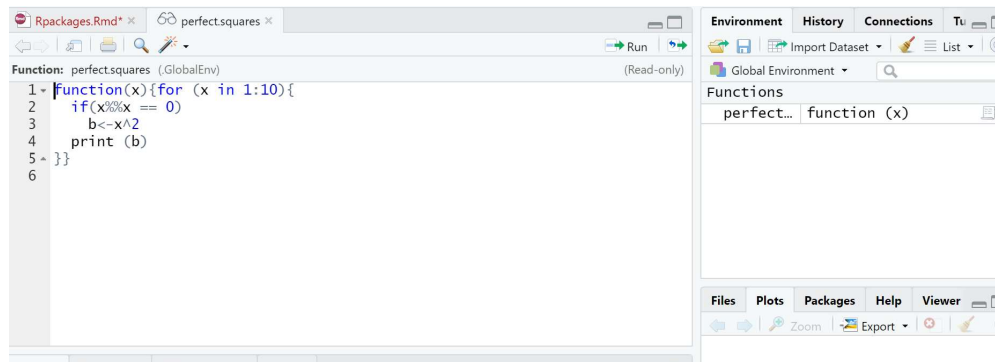
- Function Name – This is the actual name of the function. It is stored in the R environment as its name.
- Arguments – This is an placeholder to pass an input value into your function (i.e. $f(x)$).
- Function Body – The function contains the statements that determines what the function does.

Example 1:

```
#Create a function to print out the values for perfect squares.  
perfect.squares<-function(x){for (x in 1:10){  
  if(x%%x == 0)  
    b<-x^2  
    print (b)  
  }}  
  
#pass a value to the function  
perfect.squares(10)
```

```
## [1] 1  
## [1] 4  
## [1] 9  
## [1] 16  
## [1] 25  
## [1] 36  
## [1] 49  
## [1] 64  
## [1] 81  
## [1] 100
```

Here is how the function is stored as an object in the R environment:



Example 2:

```

#Create a function with more than one argument
translate<-function(x,y){
  if (x=="NA"){x=0}
  else if (x!="NA"){x=x}
  print(x)
  if (y== -1){y="NA"}
  else if (y!=-1){y=y}
  print(y)
}

```

```

#Testing the function with input values
x<-"NA"
y<--"1"

x2<-1
y2<-0

translate(x, y)

```

```

## [1] 0
## [1] "NA"

```

```

translate(x2, y2)

```

```

## [1] 1
## [1] 0

```

As an aside, in this example I could have stuck to the general structure of:

```

if (test_expression) { statement1 } else { statement2 }

```

However, the else if is useful if one wants to test more conditions.

Ifelse function in R

In order to understand the brilliance of this function let's go back to the toy example 2.

```
x3<-c(0, "NA", 1)
y3<-c(-1, "NA", 80)

translate(x3,y3)
```

```
## Warning in if (x == "NA") {: the condition has length > 1 and only the first
## element will be used
```

```
## Warning in if (x != "NA") {: the condition has length > 1 and only the first
## element will be used
```

```
## [1] "0" "NA" "1"
```

```
## Warning in if (y == -1) {: the condition has length > 1 and only the first
## element will be used
```

```
## [1] "NA"
```

The problem is revealed in the warning message. The vector has length 3 but the if statement can only evaluate one condition at a time.

The `ifelse` is a built in base R function that returns a value with the same length as the test, rather than the evaluation at the first element. The elements selected from either `yes` or `no` depend on whether the `yes` condition is true or false.

```
z<-x3
ifelse(z==0, "NA", z)
```

```
## [1] "NA" "NA" "1"
```

This function will be an important component of the mini R package I developed for this workshop.