

Programming: practical 1

In the questions below, the important part is **understanding** what's going on rather than just typing the R commands. If you are unsure of what's going on, please ask.

1 Practice questions

1. Basic functions

```
v = 5
Fun1 = function() {
  v = 0
  return(v)
}
Fun1()

## [1] 0

v

## [1] 5
```

- (a) Why does the final line return 5 and not 0.
- (b) Delete line 3 in the above piece of code. Now change Fun1 to allow v to be passed as an argument, i.e. we can write Fun1(5). Call this function to make sure it works.
- (c) Now make the argument in Fun1 have a default value of 0.

2. Default arguments:

```
Fun2 = function(x=10) {
  return(x)
}

Fun3 = function(x) {
  return(x)
}
```

- (a) Why does this work:

```
Fun2()
```

but this raises an error

```
Fun3()
```

- (b) Change Fun2 so that it returns \sqrt{x} .

3. if statements.

```
Fun4 = function(x) {  
  if(x==5) {  
    y = 0  
  } else {  
    y = 1  
  }  
  return(y)  
}
```

(a) Change Fun4 so that it:

- returns 1 if x is positive;
- returns -1 if x is negative;
- returns 0 if x is zero.

4. for loops.

```
total = 0
for(i in 1:5) {
  total = total + i
}
total

## [1] 15
```

The for loop above calculates

$$\sum_{i=1}^5 i$$

- (a) What is the final value of total in the above piece of code?
 (b) Change the above loop to calculate the following summations:

$$(i) \sum_{i=1}^{20} (i + 1)$$

$$(ii) \sum_{j=-10}^{15} j$$

5. More for loops:

```
a = 2
total = 0
for(blob in a:5) {
  total = total + blob
}
```

- (a) Delete line 1. Now put the above code in a function called Fun5, where a is passed as an argument, i.e. we can call Fun5(1)
 (b) Alter the code so that the for loop goes from a to b, rather than a to 5. Allow b to be passed as an argument, i.e. we can call Fun5(1,5).
 (c) Change Fun5 so that it has default arguments of a=1 and b=10.

Solutions

Solutions are contained within the nclRprogramming package. To obtain the package, we need to use drat, which can be installed via

```
install.packages("drat")
```

Then

```
drat::addRepo("rcourses")  
install.packages("nclRprogramming", type="source")
```

The solutions can be viewed via

```
library(nclRprogramming)  
vignette("solutions1", package="nclRprogramming")
```