Programming: Practical 1
Dr Colin S. Gillespie

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.

Course package

First we'll install the course R package.

```
##Windows & Linux users
install.packages("nclRprogramming",
                 repos="http://R-Forge.R-project.org")
##Apple users
install.packages("nclRprogramming",
                 repos="http://R-Forge.R-project.org",
                 type="source")
```

To load the package, use

```
library(nclRprogramming)
```

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 the third line 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) Include an additional argument, w. Make the function return v+w. Experiment with different calling patterns, for example:

```
Fun1(1, 2)
Fun1(v = 1, 2)
Fun1(1, w = 2)
Fun1(w = 2, v = 1)
```

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 x×x.

(c) Now change Fun2 so that is takes in take two arguments, x and y. Both arguments should have default value of 10. The function should return x×y.

3. if statements.

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

(a) Call Fun4 a few times with different values of x.

(b) Change Fun4 so that it returns 0, if x is greater than 5.

(c) Change Fun4 so that it returns 0, if x is greater than or equal to 5.

(d) Change Fun4 so that it:

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

(e) Suppose that x is a vector. Change the function so that it returns 0 if the mean(x) is negative and 1 otherwise.

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:

$$\text{(i)} \sum_{i=1}^{20}(i+1) \quad \text{(ii)} \sum_{j=-10}^{15} j$$

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 this package:

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