Chapter 10

Writing functions in R

Reading:


The true power of R becomes apparent once you can write your own functions. Consider the following example of a variance function:

```r
# calculate the sample variance of 'x'
variance <- function (x) {
    sum(( x - mean(x) )^2) / (length(x)-1)
}
```

Just like everything else in R a function is stored as an object. We can call our variance function as follows:

```r
# calculate the sample variance of the data vector c(3,3,3,3)
variance(rep(3,4))
[1] 0

# calculate the sample variance of a random sample of 100 N(0,1) variables.
variance(rnorm(100))
[1] 0.8174446
```

Functions are structured as follows
<name of function> <- function (<function arguments separated by commas>) {

<command 1>
<command 2>
.
.
.
<the last command is returned as the answer>
}

For example here is a function that takes no arguments and only returns the number 2.

two <- function () {
  2
}

# now call the function -- remember functions have brackets!
two()
[1] 2

Here is a function to add the objects x and y together

# return the result x+y (where you get to specify the 'x' and 'y')
add <- function (x, y) {
  x + y
}

Try the following examples

add(2,3)
[1] 5

add(c(1,2,3), c(3,1,2))
[1] 4 3 5

add(T,F)
[1] 1

add("dog","cat")
Error in x + y : non-numeric argument to binary operator
Note how the function handles numbers, vectors, TRUEs (T) and FALSEs (F) without problem (it will also work with matrices). It gives an error message if you try to “add” two words together – the calculation does not make sense.

**Remark:** In R, TRUE is considered to be the number 1 and FALSE is considered the number 0. This can be very useful in practice.

You can also specify default values for the function arguments, for example,

```r
add <- function (x, y=3) {
  x + y
}
```

If you do not supply a value for y then it will be assumed to be 3 in this function.

### 10.1 A bootstrap function

We now declare the `bootstrap.mean` function. Try to make sure you understand how the function works. Try some examples for yourself. What are the default values for the arguments in the function?

```r
# Calculate a 'confidence'\% bootstrap confidence interval for
# the population mean based on the data 'x'.
# Base the bootstrap interval on 'num.samples' replicates.
# Also plot a histogram of the bootstrap distribution of the mean.
# ---------------------------------------------------------------------
bootstrap.mean <- function (x, num.samples=1000, confidence=95) {

  # somewhere to store the means in...
  boot.means <- rep(NA, num.samples)

  # calculate the bootstrap sampling distribution
  for (i in 1:num.samples) {
    # Take a random sample of length n WITH replacement from 'transformers'.
    # Then calculate the mean of that random sample and store it in the i’th
    # entry of the vector 'boot.means'.
    boot.means[i] <- mean(sample(x, replace=T))
  }

  # let’s look at the bootstrap distribution of the sample means.
```
hist(boot.means)

# calculate 'alpha' where 100(1-alpha) is the confidence level.
alpha <- 1 - confidence/100

# return the alpha/2 and 1-alpha/2 quantiles of the
# bootstrap distribution for the mean.
quantile(boot.means, c(alpha/2,1-alpha/2))

Exercises:

1. In the function above, why do we define `boot.means` to initially be a vector of missing values?

2. Rewrite the above function so you can calculate bootstrap confidence intervals for the population standard deviation.