MAS115 R programming 2016-17
Lab Class 4

1 While and repeat loops

1.1 The `while` loop

Using a `while` loop

As well as `for` loops, **R** provides `while` loops which allow you to repeat a statement until a particular condition becomes `FALSE`. Run the following code and try to figure out what it does

```
# A while loop
nobs <- 10
x <- runif(nobs)
while((mean(x) < 0.49) || (mean(x) > 0.51)) {
  nobs <- 2 * nobs
  cat("Mean is", mean(x), "\n")
  cat("Need to try number = ", nobs, "\n")
  x <- runif(nobs)
}
nobs
mean(x)
```

Syntax

The format to construct a while loop is as follows:

```
while(condition) statement
```

This construction will cause `statement` to be repeated until `condition` is `FALSE`. For it to work then you need the condition to take a single value (i.e. not be a vector of length > 1). The `statement` can be compound as in the example above in which case you need to enclose it in brackets `{ }`

1.2 The `repeat` loop

Using a `repeat` loop

R's final method of producing loops is to use the command `repeat`. Try the following code and discover what it does:

```
# A repeat loop
i <- 0
repeat {
  i <- i+1
  x <- rnorm(1)
  cat("Try", i, "realisation is", x, "\n")
  if(x > 1.2) break
```

```
}
i
```

Syntax

The syntax for a `while` loop is as follows:

```
repeat   statement
```

If it is used then `statement` will be repeated until the flow is transferred out of the loop using the `break` statement. Typically we will therefore combine `repeat` loops with `if` statements which determine when we want to break out of the loop.
Note: If you can't remember the `break` statement then look back at last week's practical sheet.

Subtle differences between `while` and `repeat` The different loops available in **R** can normally be transferred between e.g. you can rewrite a `for` loop as a `while` or `repeat` loop and so on. It is however worth pointing out a small difference between the computer's response when writing a `while` loop and a `repeat` loop:

- In the case of the `while` loop, the loop will only be entered if the initial condition is `TRUE`. Hence it is possible that the loop will never be performed.
- In the case of the `repeat` loop, the loop will always be entered at least once since it will only be exited when the `break` command is encountered.

1.3 Killing `while` and `repeat` loops if you've made a mistake

Sometimes when you write either a `while` or a `repeat` loop you will make an error in your code which means that the condition required to exit the loop will never be attained. In such cases then your computer will try and run through the loop forever and you will need to stop the code running manually yourself. In order to do this on your Windows machine (or if you're using a Mac at home) then you will need to click in the R console window (where the code is actually running) and then press the `Esc` key. Try this on the following code:

```
# A loop which never ends
while(TRUE) {
cat("Trapped inside a never-ending loop", "\n")
}
```

2 Lab class tasks

2.1 Cumulative sums of random variables

Consider a sequence of random variables $X_1, X_2, \ldots,$ where each $X_i \sim f$ for some distribution f. Suppose we are interested in finding out how many of the random variables we need until the sum exceeds 100 i.e. we want to find out the number n such that

$$\sum_{i=1}^{n-1} X_i \leq 100 \quad \text{but} \quad \sum_{i=1}^{n} X_i > 100.$$

This number n will itself be a random variable as it depends upon the values of the individual X_i's.

Write a loop (of some kind) to create a value of n with the individual X_i's following an Exponential distribution with mean 5 (look at `?rexp`). Note: you need to be careful with defining an Exponential random variable as some people use the rate while others define the scale.

- Now place the loop created above inside a `for` loop to create 1000 realisations of n.
- Plot a histogram of n. Look at `?hist` for how to create a histogram in R if you haven't already come across this command.

2.2 Death at a chemical plant (a bit more pseudo-code)

2.2.1 The problem

A fluid dynamics model describes concentration of a pollutant in a region following release from a point source,

$$C(y, z) = \frac{Q}{2\pi u_{10}\sigma_z\sigma_y} \exp\left[-\frac{1}{2}\left\{\frac{y^2}{\sigma_y^2} + \frac{(z-h)^2}{\sigma_z^2}\right\}\right], \tag{1}$$

where the variables have the following meanings: C: air concentration of pollutant; Q: release rate; u_{10}: wind speed at 10m above ground; σ_y, σ_z: diffusion parameters in horizontal and vertical directions; h: release height; (y, z): coordinates along wind direction and above ground.

We are given $Q = 100$, $h = 50$m, but u, σ_z, σ_y are uncertain, with

$$\log u_{10} \sim N(2, 0.1) \qquad \log \sigma_y^2 \sim N(10, 0.2) \qquad \log \sigma_z^2 \sim N(5, 0.05).$$

We are interested in questions such as: what is the distribution of $C(100, 40)$? What is the 95th percentile of $C(100, 40)$?

2.2.2 Solution technique

- Sample unknown input parameters from their distributions.
- Evaluate the function to obtain output values from its distribution.
- Given suitably large sample (say $N = 5000$) we can get a good approximation to the distribution by drawing a histogram of the output values. Similarly the 95th percentile from distribution of $C(100, 40)$ can be estimated by the 95th percentile from sample of simulated values of $C(100, 40)$.

2.2.3 Pseudo-code

1. INPUT y, z, N.

2. CREATE C as a vector of length N.

3. SET $Q \leftarrow 100, h \leftarrow 50$.

4. FOR $i = 1, 2, \ldots, N$:

   (a) Sample a set of input values:

   - Sample $u_{10,i}$ from $\log N(2, 0.1)$
   - Sample $\sigma^2_{y,i}$ from $\log N(10, 0.2)$
   - Sample $\sigma^2_{z,i}$ from $\log N(5, 0.05)$

   (b) Evaluate the model output C_i. Set

   $$C_i \leftarrow \frac{Q}{2\pi u_{10,i}\sigma_{z,i}\sigma_{y,i}} \exp\left[-\frac{1}{2}\left\{\frac{y^2}{\sigma^2_{y,i}} + \frac{(z-h)^2}{\sigma^2_{z,i}}\right\}\right]$$

   ENDFOR

5. Return $C_1, C_2, \ldots, C_N$.

Notes:

- You can sample from a $\log N(2, 0.1)$ using the `rlnorm()` function. For example to sample $u_{10,i}$ from $\log N(2, 0.1)$ we would use the command `u<-rlnorm(1,meanlog = 2,sdlog = sqrt(0.1))`.

- When implementing in R, it may be easier to break down the calculation of C_i into more than one step, just because the formula is quite long.

- R can also work out quantiles easily using the `quantile()` command. Take a look at the help file for this.

3 Homework — due practical class week 5

Your solutions must be clearly structured and be written in such a way that they are readable in the way that a standard set of notes is. Explanation, code and output should be grouped together in an appropriate manner. Do **NOT** simply submit raw R code and output without any real world explanation.

1. **Cumulative sums**
   (a) Write pseudo-code for the 'cumulative sums' problem described above.
   (b) Implement your program in R; show your code, and a small sample of the values of n, for example by using `head()`.
   (c) What is the probability that $n > 30$?

2. **Death at a chemical plant**
   (a) Implement the pollutant diffusion model in R; show your code, and a small sample of the output.
   (b) What is the mean value of $C(100, 40)$?
   (c) What is the 95th percentile of $C(100, 40)$?

3. **Triangular numbers**
   The kth triangular number, T_k, is

$$\sum_{j=1}^{k} j = \frac{k(k+1)}{2}.$$

   Use a loop to search for triangular numbers that are also perfect squares; keep going until you have found seven of them. For each of the seven, record k and T_k, and also the number that T_k is the square of.