#3

User-written functions

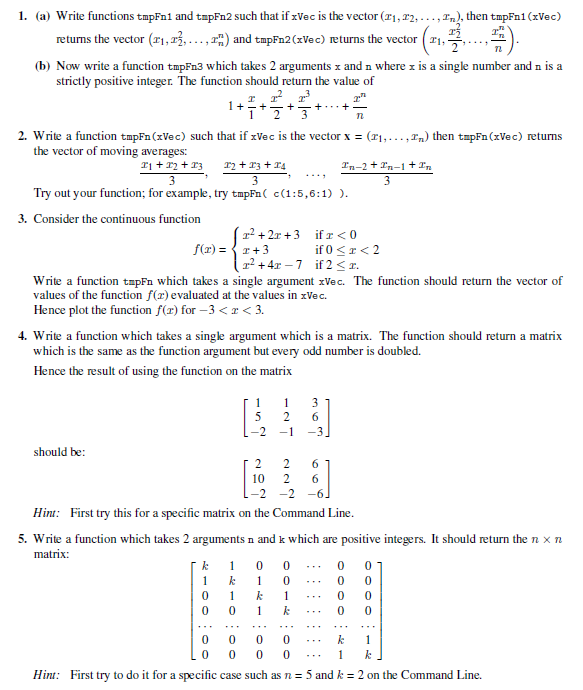
One of R’s greatest strengths is the user’s ability to add functions. In fact, many of the functions in R are functions of existing functions. The structure of a function looks like this:

myfunction <- function(*arg1*, *arg2*, ... ){

*statements*

return(*object*)

}



1. A

tmpFn1 <- function(xVec) {

xVec^(1:length(xVec))

}

tmpFn2 <- function(xVec)

{

n <- length(xVec) (xVec^(1:n))/(1:n)

}

(b) tmpFn3 <- function(x, n)

{

1 + sum((x^(1:n))/(1:n))

}

(b) tmpFn3 <- function(x, n)

{

1 + sum((x^(1:n))/(1:n))

}

1. tmpFn <- function(xVec)

{

n <- length(xVec) ( xVec[ -c(n-1,n) ] + xVec[ -c(1,n) ] + xVec[ -c(1,2) ] )/3

}

Or

tmpFn <- function(xVec)

{

n <- length(xVec) ( x[1:(n-2)] + x[2:(n-1)] + x[3:n] )/3

}

Note that

tmpFn( c(1:5,6:1) )

should return the vector (2, 3, 4, 5, 5.333, 5, 4, 3, 2).

3.tmpFn <- function(x)

{

ifelse(x < 0, x^2 + 2\*x + 3, ifelse(x < 2, x+3, x^2 + 4\*x - 7))

}

tmp <- seq(-3, 3, len=100) plot(tmp, tmpFn(tmp), type="l")

4.tmpFn <- function(mat)

{

mat[mat%%2 == 1] <- 2 \* mat[mat%%2 == 1] mat

}

5. tmp <- diag(2, nr = 5)

tmp[abs(row(tmp) - col(tmp)) == 1]

<- 1 tmp

Now for the function for the general case: tmpFn <- function(n, k)

{

tmp <- diag(k, nr = n) tmp[abs(row(tmp) - col(tmp)) == 1]

<- 1 tmp

}