

Tutorial 2

John O'Sullivan

1. Creating vectors:

- Create a vector called x of the odd numbers from 1 to 9. Try doing it three different ways:
 - (i) using the concatenate function `c()`
 - (ii) using `vector()` to create an empty vector of the right length, and then square brackets to input each number
 - (iii) using `seq()`

2. Manipulating vectors:

- (i) Append the numbers 11, 12, 13 and 14 to the end of your vector x
- (ii) Then remove all of the numbers in it that are divisible by 3
- (iii) The vector x should now be of length 6. Multiply your vector by 3, then add the vector $z = 6:5$. Why does this not give an error?
- (iv) Store the result in a new vector w

3. Subset:

- (i) Write some code to determine if any of the values in w are less than 30
- (ii) Write some code (using **subset**) to determine all the values in w that are divisible by 3
- (iii) Use **which** to find which elements of w are greater than 20

4. Matrices:

- (i) Create a matrix via the command `M <- matrix(1:12, 4, 3)`
- (ii) Write code to access:
 - (a) the first row of M
 - (b) the second column of M
 - (c) the elements that are in the second and third row and second column of M - save this object as `vec1`
- (iii) Write code to calculate the mean of `vec1`

5. The `apply()` function:

- (i) What does the command `apply(M, 1, sum)` do?
- (ii) How do you get column sums of M using `apply()`?
- (iii) Replace the value in the bottom right-hand corner of M with the value **NA**
- (iv) What happens if you re-run the `apply` command from part (i)?
- (v) Add an extra argument to the `apply` command (hint: look at the help for **sum**) which removes the NA value

6. Challenge question:

The `find.k()` function is at the end of Lecture 2 Code. Run this function to see what it does, then answer the questions below.

- (i) Take the `find.k()` function and change it so that it looks for runs of 2 values of k in a row (e.g., for the vector $x = c(1, 3, 2, 3, 6, 2, 3, 3, 4, 2)$, for $k = 3$, your function should find the two 3s together (starting at position 7) but *not* count the 3s which are on their own (in positions 2 and 4))
- (ii) Change it again to find runs of k of any given length n

7. `ixmap` package:

There is code involving the `ixmap` package at the end of Lecture 2 Code. Download the picture from the Topic 2 folder on Moodle, then try to run this code. You will need to set your working directory in order to do this.

- (i) Install and load the **pixmap** package
- (ii) Check that you can create the image and manipulate it as in the lecture code. Try your own manipulations and see what you can create
- (iii) Look for some ppm files online, and try to read these in. Can you manipulate these in the same way? (You can find some ppm files [here](#), but look for others online too.)