Demo in Introduction to R: Page 7 (Page 13 in the pdf)

> x

[1] 10.4 5.6 3.1 6.4 21.7

> y

[1] 10.4 5.6 3.1 6.4 21.7 0.0 10.4 5.6 3.1 6.4 21.7

>

> v = 2\*x + y +1 # Elementwise arithmetic x and 1 are recycled to 11 elements – but vector sums

Warning message:

In 2 \* x + y :

longer object length is not a multiple of shorter object length

> v

[1] 32.2 17.8 10.3 20.2 66.1 21.8 22.6 12.8 16.9 50.8 43.5 #The result

>

> matrix tot was filled using 2\*x, y and 1. X recycles 2.2 times. One recycles 11 times (10 times, given > > the initial 1). All are now conformable for vector addition.

>

> tot

Xx2 y one

[1,] 20.8 10.4 1

[2,] 11.2 5.6 1

[3,] 6.2 3.1 1

[4,] 12.8 6.4 1

[5,] 43.4 21.7 1

[6,] 20.8 0.0 1

[7,] 11.2 10.4 1

[8,] 6.2 5.6 1

[9,] 12.8 3.1 1

[10,] 43.4 6.4 1

[11,] 20.8 21.7 1

>

> rowSums(tot)

[1] 32.2 17.8 10.3 20.2 66.1 21.8 22.6 12.8 16.9 50.8 43.5

> v

[1] 32.2 17.8 10.3 20.2 66.1 21.8 22.6 12.8 16.9 50.8 43.5

The equation sums and the matrix sums agree.

5.4.1 Mixed vector and array arithmetic. The recycling rule

The precise rule affecting element by element mixed calculations with vectors and arrays is

somewhat quirky and hard to find in the references. From experience we have found the following

to be a reliable guide.

The expression is scanned from left to right.

Any short vector operands are extended by recycling their values until they match the size of any other operands.

As long as short vectors and arrays only are encountered, the arrays must all have the same dim attribute or an error results.

Any vector operand longer than a matrix or array operand generates an error.

If array structures are present and no error or coercion to vector has been precipitated, the result is an array structure with the common dim attribute of its array operands.