**R-VECTORS:**

### Vectors and assignment:

R operates on named data structures. The simplest such structure is the numeric vector, which is a single entity consisting of an ordered collection of numbers.

To set up a vector named x, say, consisting of five numbers, namely

6.0, 7.0, 8.0, 9.0 and 10.0, use the R command

> x<- c(6.1,7.2,8.3,9.4,10.5) or

> x = c(6.1,7.2,8.3,9.4,10.5) or

>assign("x",c(6.1,7.2,8.3,9.4,10.5)) or

>c(6.1,7.2,8.3,9.4,10.5) -> x

This is an assignment statement using the function c() which in this context can take an arbitrary number of vector arguments and whose value is a vector got by concatenating its arguments end to end.

A number occurring by itself in an expression is taken as a vector of length one.

Notice that the assignment operator (‘<-’), which consists of the two characters ‘<’ (“less than”) and ‘-’ (“minus”) occurring strictly side-by-side and it ‘points’ to the object receiving the value of the expression. In most contexts the ‘=’ operator can be used as an alternative.

The further assignment

> y <- c(x, 0, x)

would create a vector y with 11 entries consisting of two copies of x with a zero in the middle place.

Vectors and sequences Vectors or sequences of numbers can also be created, stored and manipulated. They can be created manually

x1 = c(0.1, 0.2, 0.3, 0.4, 0.5)

**Example:** if x = (1, 2, 3, 4) and y = (5, 6), then

x + 3 = (4, 5, 6, 7)

x + y = (6, 8, 8, 10)

Modern computers use 64 bits to represent numbers so some numbers (like 0.8) must be approximated.

Care is needed when testing whether 2 numbers are the same:

> x <- seq(0, 0.5, 0.1) ##generate a sequence from 0 to 0.5 in steps of 0.1

> x ##Look at x

– is x equal to (0, 0.1, 0.2, 0.3, 0.4, 0.5)?

> x==c(0, 0.1, 0.2, 0.3, 0.4, 0.5)

[1] TRUE TRUE TRUE FALSE TRUE TRUE

Rounding problems Tiny inaccuracies can accumulate:

The sample variance of a vector x is often calculated as



(sum(x^2) - n \* mean(x)^2) / (n - 1)

>x <- seq(1:100)

>x <- seq(1:100) + 10000000000