Object Types

Vectors, Matrices, Arrays and Lists

Vectors

- A vector is simply a list of values. **R** relies on vectors for many of its operations, such as plots, basic statistics and statistical modelling.
- Values of vector can be numbers, strings, logical values or any other types, as long as they are all same type.

Vectors

- Example: set up a vector named x, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an *assignment* statement using the *function* `c()`.

In most contexts the '=' operator can be used as an alternative.

Vectors

Do it yourself:

```
> c(1,3,5)↵
> c("H", "A", "B")↵
> c(TRUE, 2, "Sky")↵
> y <- c(x, 0, x)↵
> y ↵
```

Vectors

- Vectors can be used in arithmetic expressions, in which case the operations are performed element by element.

```
> v <- 2*x + y + 1 ↵
> sum((x-mean(x))^2)/(length(x)-1) ↵
> sort(x) ↵
```

Matrix

- Matrices are usually defined in **R** by function `matrix()`

```
> matrix(vector, ncol = n, nrow = m)
```

- You can define a diagonal matrix using the `diag()` function:

```
> diag(x, ncol=n, nrow= m)
```

Matrix

Do it yourself:

- Define a matrix of 3 rows and 2 columns with following vector

```
c(1,6,5,3,2,7)
```

- Define a diagonal matrix of 5 columns and 5 rows with the diagonal values of (3,6,9.1,-0.5,0.12)

Array

- An array can be considered as a multiply subscripted collection of data entries, for example numeric.
- R allows simple facilities for creating and handling arrays, and in particular the special case of matrices.
- A vector can be used by R as an array only if it has a dimension vector as its dim attribute.

Suppose, for example, z is a vector of 1500 elements. The assignment

```
> dim(z) <- c(3,5,100)
```

Array

Do it yourself:

```
> x <- c(1,6,5,3,2,7,1,6,5,3,2,7) ↲

> array(x, dim = c(3,2,2)) ↲

# or dim(x)<- c(3,2,2)
```

Lists

- *lists* are a general form of vector in which the various elements do not need to be of the same type, and are often themselves vectors or lists.
- Lists provide a convenient way to return the results of a statistical computation.

```
list(name="Mary", spouse="Todd",
no.children=3, child.ages=c(4,7,9))
```