

## Lecture 3 - R Software

**Ashutosh Rajput**  
Department of Mathematics

Rajdhani College  
University of Delhi

## Combining Objects

The `c()` function attempts to combine objects in the most general way. For example, if we combine a matrix and a vector, the result is a vector.

```
> c(matrix(1:4,ncol=2),1:3)
[1] 1 2 3 4 1 2 3
```

Note that the `list()` function preserves the identity of each of its elements:

```
> list(matrix(1:4,ncol=2),1:3)
[[1]]
  [,1] [,2]
[1,]  1   3
[2,]  2   4
[[2]]
[1] 1 2 3
```

When the `c()` function is applied to lists, it will return a list:

```
> c(list(matrix(1:4,ncol=2),1:3),list(1:5))
```

```
[[1]]  
  [,1] [,2]  
[1,]  1  3  
[2,]  2  4
```

```
[[2]]  
[1] 1 2 3
```

```
[[3]]  
[1] 1 2 3 4 5
```

To break down anything into its individual components, use the `recursive=TRUE` argument of `c()`:

```
> c(list(matrix(1:4,ncol=2),1:3),recursive=TRUE)
```

```
[1] 1 2 3 4 1 2 3
```

- Note:
- A list of currently defined variables can be generated with `ls()`.
  - Variables may be removed with the `rm()` function.

Most mathematical operators in **R** work component-wise on arrays. Unlike mathematical arrays, **R** arrays can sometimes be combined even when they are not the same size.

For example: An array of length 3 can be added to an array of length 6 (and the answer is an array of length 6).

To make it possible, **R** will expand the shorter array by reusing entries that start from the left; the array [1, 2, 3] would be expanded to [1, 2, 3, 1, 2, 3].

For example:

```
> a = c(1, 2, 3)
```

```
> b = c(5, 5, 5, 5, 5, 5)
```

```
> 4+a
```

```
[1] 5 6 7
```

```
> a+b
```

```
[1] 6 7 8 6 7 8
```

```
> sin( a*pi/2 )
```

```
[1] 1.000000e+00 1.224606e-16 - 1.000000e+00
```