

# **Introduction to R Software**

## **Lists**

**Shalabh**

**Department of Mathematics and Statistics**

**Indian Institute of Technology Kanpur**

# Lists

## Example

```
> x1 <- matrix(nrow=2, ncol=2, data=1:4, byrow=T)
> x2 <- matrix(nrow=2, ncol=2, data=5:8, byrow=T)
> x1
```

|      | [,1] | [,2] |
|------|------|------|
| [1,] | 1    | 2    |
| [2,] | 3    | 4    |

```
> x2
```

|      | [,1] | [,2] |
|------|------|------|
| [1,] | 5    | 6    |
| [2,] | 7    | 8    |

```
> x1+x2
```

|      | [,1] | [,2] |
|------|------|------|
| [1,] | 6    | 8    |
| [2,] | 10   | 12   |

# Lists

R Console

```
> x1 <- matrix(nrow=2, ncol=2, data=1:4, byrow=T)
> x2 <- matrix(nrow=2, ncol=2, data=5:8, byrow=T)
> x1
      [,1] [,2]
[1,]    1    2
[2,]    3    4
> x2
      [,1] [,2]
[1,]    5    6
[2,]    7    8
> x1+x2
      [,1] [,2]
[1,]    6    8
[2,]   10   12
```

# Lists

## Example

```
> x1[2,1] <- "hello"
```

```
> x1
```

|      | [,1]    | [,2] |
|------|---------|------|
| [1,] | "1"     | "2"  |
| [2,] | "hello" | "4"  |

```
> x1 + x2
```

```
Error in x1 + x2 : non-numeric argument to  
binary operator
```

# Lists

## Example

```
R Console
> x2 <- matrix(nrow=2, ncol=2, data=5:8, byrow=T)
> x1
      [,1] [,2]
[1,]    1    2
[2,]    3    4
> x1[2,1] <- "hello"
> x1
      [,1] [,2]
[1,] "1"   "2"
[2,] "hello" "4"
> x1+x2
Error in x1 + x2 : non-numeric argument to
binary operator
```

# Lists

Lists can contain any kind of objects as well as objects of different types. For example, lists can contain matrices as objects:

## Example

```
> x1 <- matrix(nrow=2, ncol=2, data=1:4, byrow=T)
> x2 <- matrix(nrow=2, ncol=2, data=5:8, byrow=T)
```

```
> x1
```

|      | [,1] | [,2] |
|------|------|------|
| [1,] | 1    | 2    |
| [2,] | 3    | 4    |

```
> x2
```

|      | [,1] | [,2] |
|------|------|------|
| [1,] | 5    | 6    |
| [2,] | 7    | 8    |

# Lists

## Example

```
> matlist <- list(x1, x2)
```

```
> matlist
```

```
[[1]]
```

```
      [,1] [,2]
```

```
[1,]     1     2
```

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

```
[[2]]
```

```
      [,1] [,2]
```

```
[1,]     5     6
```

```
[2,]     7     8
```

R Console

```
> matlist <- list(x1, x2)
```

```
> matlist
```

```
[[1]]
```

```
      [,1] [,2]
```

```
[1,]     1     2
```

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

```
[[2]]
```

```
      [,1] [,2]
```

```
[1,]     5     6
```

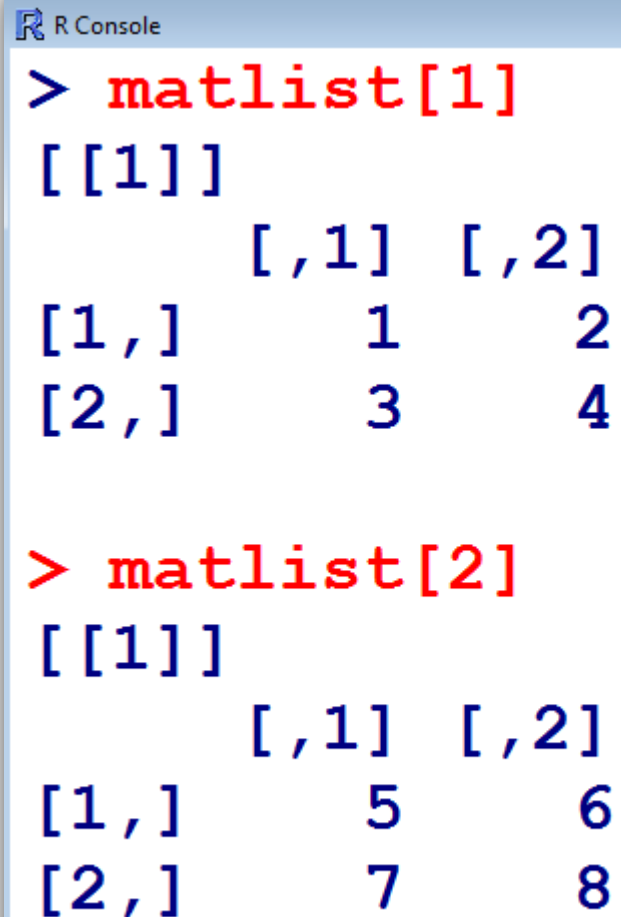
```
[2,]     7     8
```

# Lists

## Example

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

```
> matlist[2]
[[1]]
      [,1] [,2]
[1,]    5    6
[2,]    7    8
```



The image shows a screenshot of an R Console window. The title bar reads 'R Console'. The console contains two commands and their outputs. The first command is `> matlist[1]`, which outputs a list element containing a 2x2 matrix with values 1, 2, 3, and 4. The second command is `> matlist[2]`, which outputs a list element containing a 2x2 matrix with values 5, 6, 7, and 8. The output format is consistent with the R console, showing the list structure and the matrix dimensions and values.

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

> matlist[2]
[[1]]
      [,1] [,2]
[1,]    5    6
[2,]    7    8
```



# Lists

An example of a list that contains different object types:

```
> z1 <- list( c("water", "juice", "lemonade"),  
rep(1:4, each=2), matrix(data=5:8, nrow=2,  
ncol=2, byrow=T) )
```

```
> z1
```

```
[[1]]
```

```
[1] "water"      "juice"      "lemonade"
```

```
[[2]]
```

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

```
[[3]]
```

```
      [,1] [,2]
```

```
[1,]     5     6
```

```
[2,]     7     8
```

# Lists

```
R Console  
> z1 <- list( c("water", "juice", "lemonade"), rep(1:4, each=2), matrix(data=5:8, nrow=2, ncol=2, byrow=T) )
```

```
R Console
```

```
> z1  
[[1]]  
[1] "water"      "juice"       "lemonade"  
  
[[2]]  
[1] 1 1 2 2 3 3 4 4  
  
[[3]]  
      [,1] [,2]  
[1,]    5    6  
[2,]    7    8
```

# Lists

Access the elements of a list using the operator `[[ ]]`

Following commands work.

```
> z1[[1]]  
[1] "water" "juice" "lemonade"
```

Suppose we want to extract `"juice"`. The command

```
> z1[1][2] # Notice the positions of brackets  
[[1]] NULL
```

returns `NULL` instead of `"juice"`, while

```
> z1[[1]][2] # Notice the positions of brackets  
[1] "juice"
```

finally returns the desired result.

# Lists

R Console

```
> z1[[1]]
```

```
[1] "water"      "juice"      "lemonade"
```

```
>
```

```
> z1[1][2]
```

```
[[1]]
```

```
NULL
```

```
> z1[[1]][2]
```

```
[1] "juice"
```