Introduction to R Software

Swayam Prabha

Lecture 19

Repeats, Sorting and Mode

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from

http://home.iitk.ac.in/~shalab/sp

Repeats

Command `rep` is used to replicates the values in a vector.

Syntax `rep(x)` replicates the values in a vector `x`.

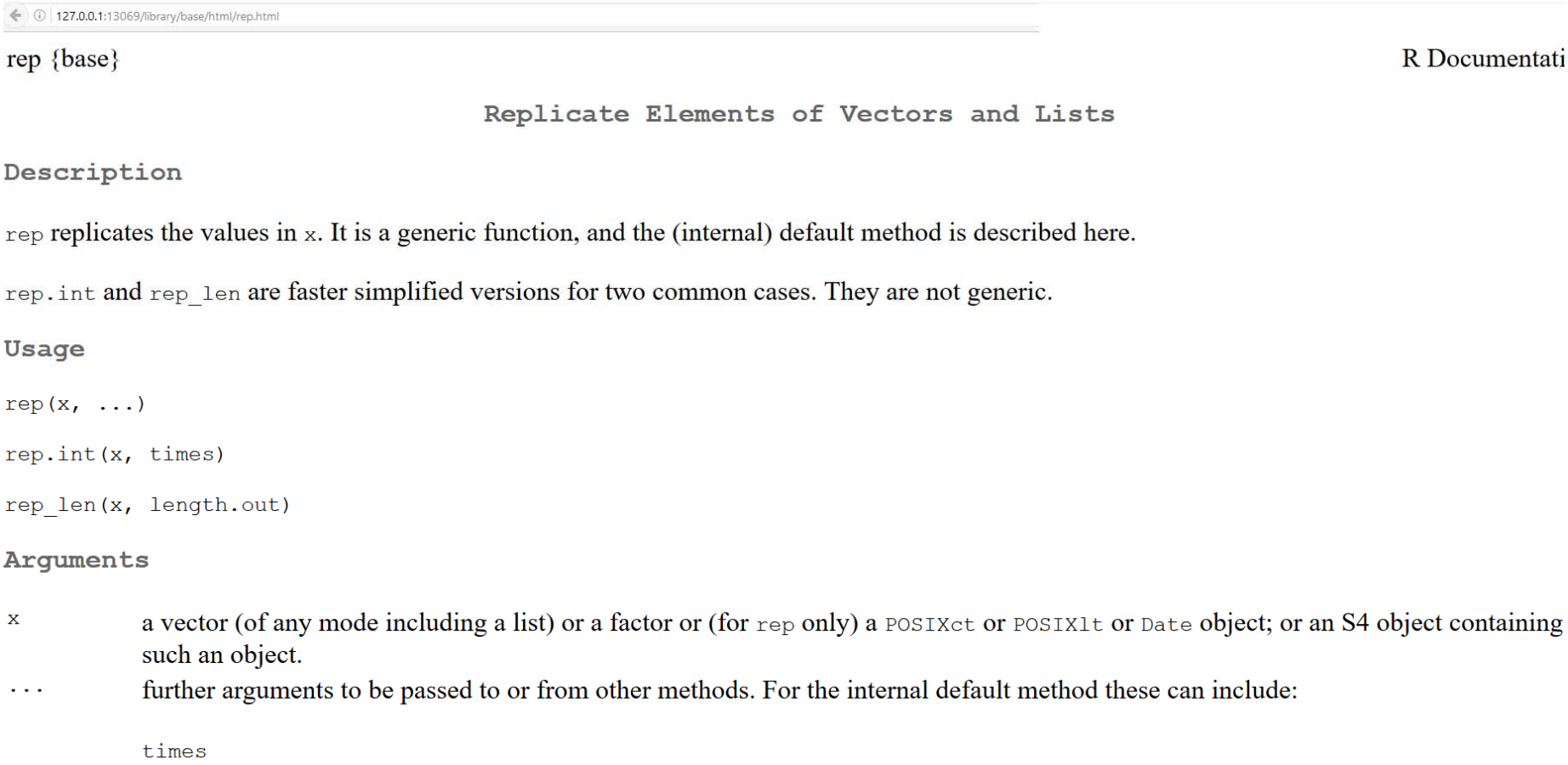
`rep(x, times=n )` # Repeat `x` as a whole `n` times

`rep(x, each=n )` # Repeat each cell `n` times

Repeats

Help for the command `rep`

```
> help("rep")
```



127.0.0.1:13069/library/base/html/rep.html

rep {base} R Documentati

Replicate Elements of Vectors and Lists

Description

`rep` replicates the values in `x`. It is a generic function, and the (internal) default method is described here.

`rep.int` and `rep_len` are faster simplified versions for two common cases. They are not generic.

Usage

```
rep(x, ...)

rep.int(x, times)

rep_len(x, length.out)
```

Arguments

`x`	a vector (of any mode including a list) or a factor or (for `rep` only) a `POSIXct` or `POSIXlt` or `Date` object; or an S4 object containing such an object.
`...`	further arguments to be passed to or from other methods. For the internal default method these can include:
	`times`

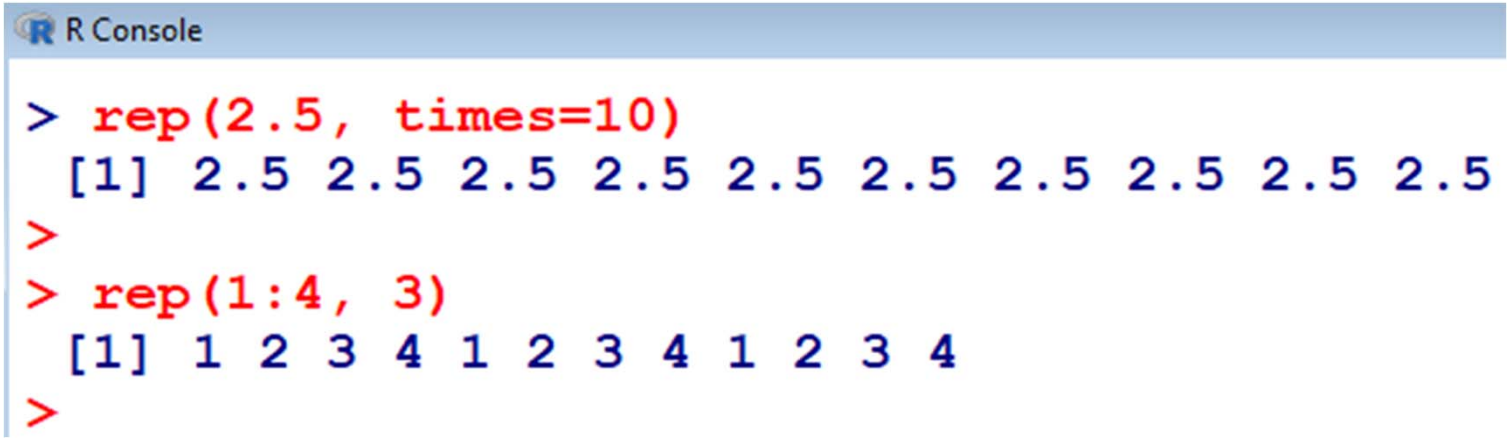
Repeats

The command `rep`

Repeat an object *n*–times:

```
> rep(2.5, times=10)
 [1] 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5

> rep(1:4, 3)
  [1] 1 2 3 4 1 2 3 4 1 2 3 4
```



Repeats

Repeat an object *n*–times:

```
rep(x, times = n )
```

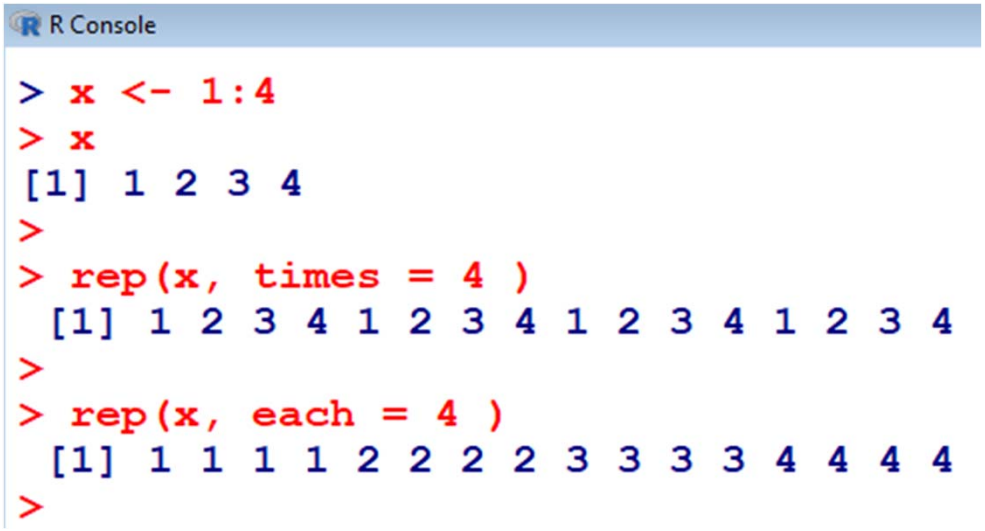
Repeat each cell *n*–times:

```
rep(x, each = n )
```

```
> x <- 1:4
> x
[1] 1 2 3 4

> rep(x, times = 4 )
[1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4

> rep(x, each = 4 )
[1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4
```

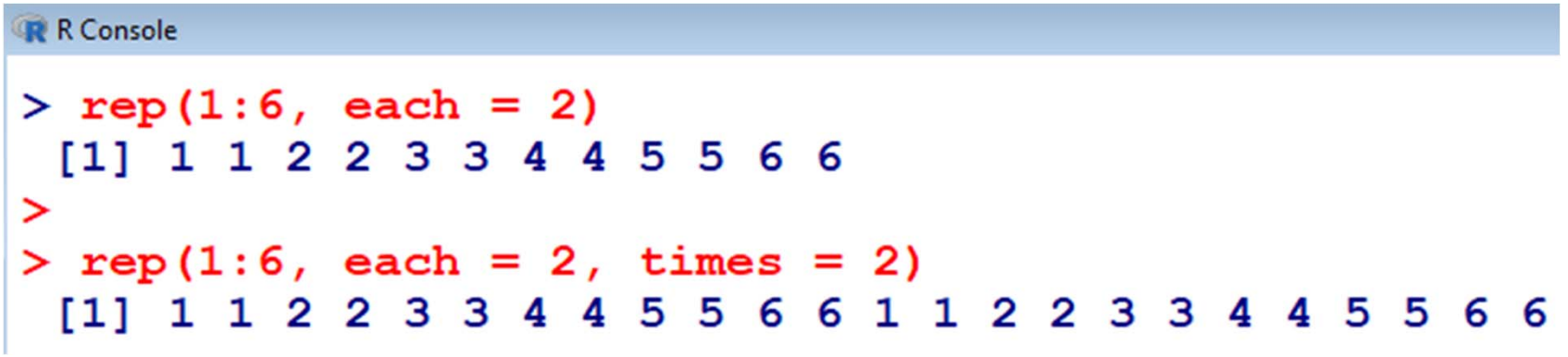


Repeats

Every object is repeated several times successively:

```
> rep(1:6, each = 2)
 [1] 1 1 2 2 3 3 4 4 5 5 6 6
```

```
> rep(1:6, each = 2, times = 2)
[1] 1 1 2 2 3 3 4 4 5 5 6 6 1 1 2 2 3 3 4 4 5 5 6 6
```

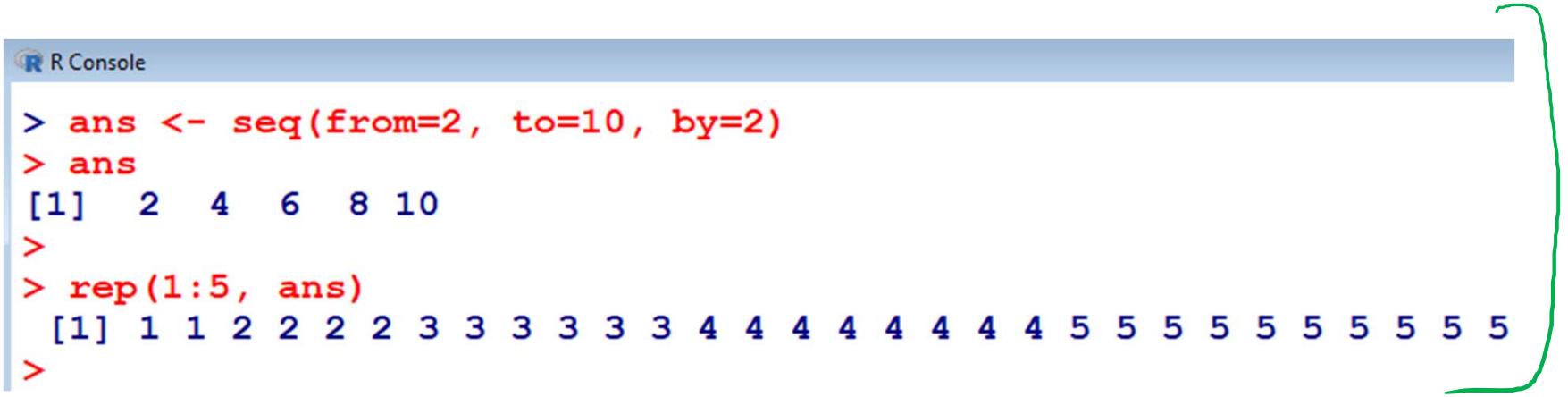


Repeats

Every object is repeated a different number of times:

```
> ans <- seq(from=2, to=10, by=2)
> ans
  [1] 2 4 6 8 10

> rep(1:5, ans)
  [1] 1 1 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5 5
```



Repeats

```
> x <- matrix(nrow=2, ncol=2, data=1:4, byrow=T)

> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4

> rep(x, 4)
 [1] 1 3 2 4 1 3 2 4 1 3 2 4 1 3 2 4
```

Repeats

R Console

```
> x <- matrix(nrow=2, ncol=2, data=1:4, byrow=T)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
>
> rep(x, 4)
 [1] 1 3 2 4 1 3 2 4 1 3 2 4 1 3 2 4
```

Repeats

Repetition of characters

```
> rep(c("a", "b", "c"), 2)
[1] "a" "b" "c" "a" "b" "c"
```

```
> rep(c("toffee", "orange", "cake"), 2)
[1] "toffee" "orange" "cake"   "toffee" "orange"
"cake"
```

Repeats

R Console

```
> rep(c("a", "b", "c"), 2)
[1] "a" "b" "c" "a" "b" "c"
>
> rep(c("toffee", "orange", "cake"), 2)
[1] "toffee" "orange" "cake"   "toffee" "orange" "cake"
> |
```

Sorting

`sort` function sorts the values of a vector in ascending order (by default) or descending order.

Syntax

```
sort(x, decreasing = FALSE, ..,)
sort(x, decreasing = FALSE, na.last = NA, ...)
```

`x`	Vector of values to be sorted
`decreasing`	Should the sort be increasing or decreasing
`na.last`	for controlling the treatment of `NA`s. If `TRUE`, missing values in the data are put last; if `FALSE`, they are put first; if `NA`, they are removed.

Sorting

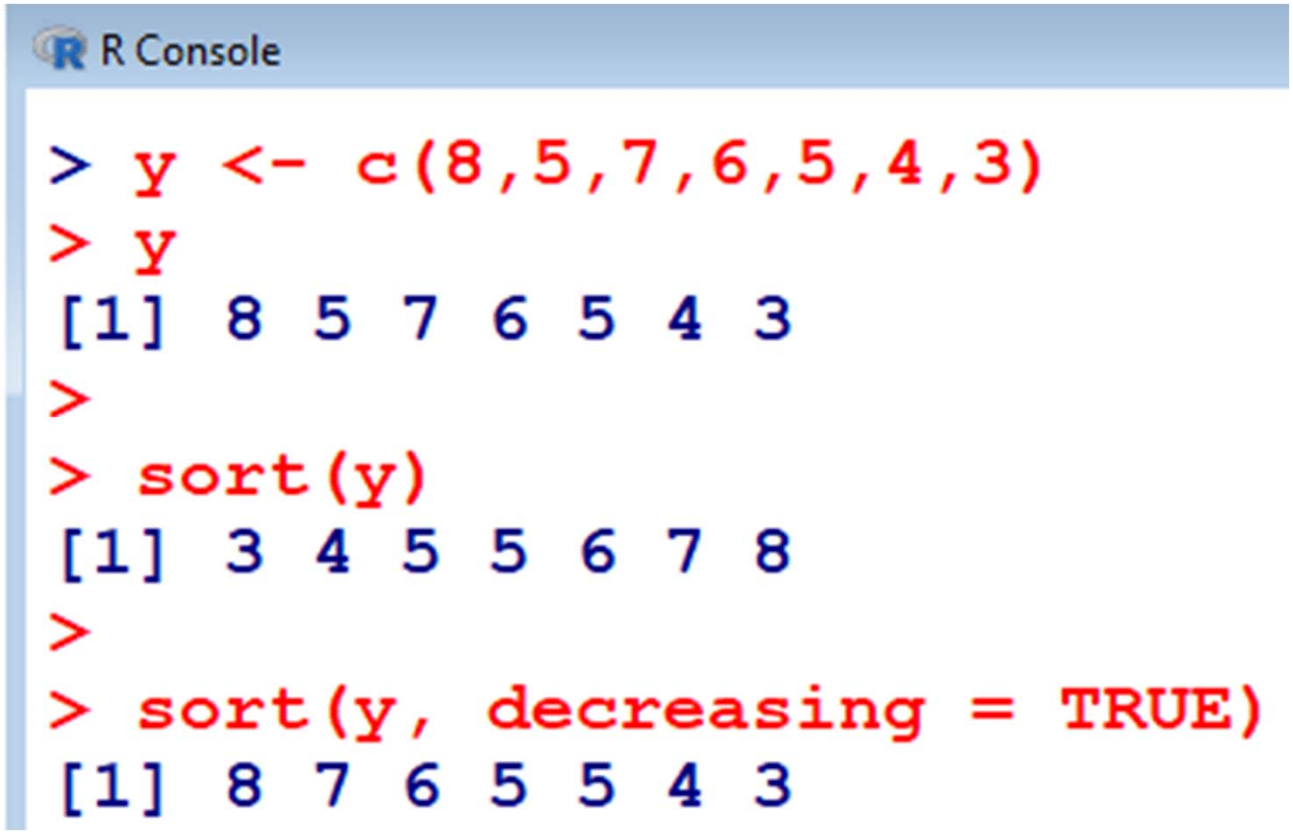
Example

```
> y <- c(8,5,7,6,5,4,3)
> y
[1] 8 5 7 6 5 4 3


> sort(y)
[1] 3 4 5 5 6 7 8


> sort(y, decreasing = TRUE)
[1] 8 7 6 5 5 4 3
```

Sorting


R Console
> y <- c(8,5,7,6,5,4,3)
> y
[1] 8 5 7 6 5 4 3
>
> sort(y)
[1] 3 4 5 5 6 7 8
>
> sort(y, decreasing = TRUE)
[1] 8 7 6 5 5 4 3

Mode

Every object has a mode.

The mode indicates how the object is stored in memory: as a

- ✓ **number,**
- ✓ **character string,**
- ✓ **list of pointers to other objects,**
- ✓ **function etc.**

Mode

`mode` function gives us such information.

Syntax

```
mode()
```

Mode

Object	Example	Mode
Number	`1.234`	numeric
Vector of numbers	`c(6, 7, 8, 9)`	numeric
Character string	`"India"`	character
Vector of character strings	`c("India", "CANADA")`	character
Factor	`factor(c("MP", "UP"))`	numeric
List	`list("India","CANADA")`	list
Data frame	`data.frame(x=1:2, y=c("India", "CANADA"))`	list
Function	`print`	function

Mode

Example

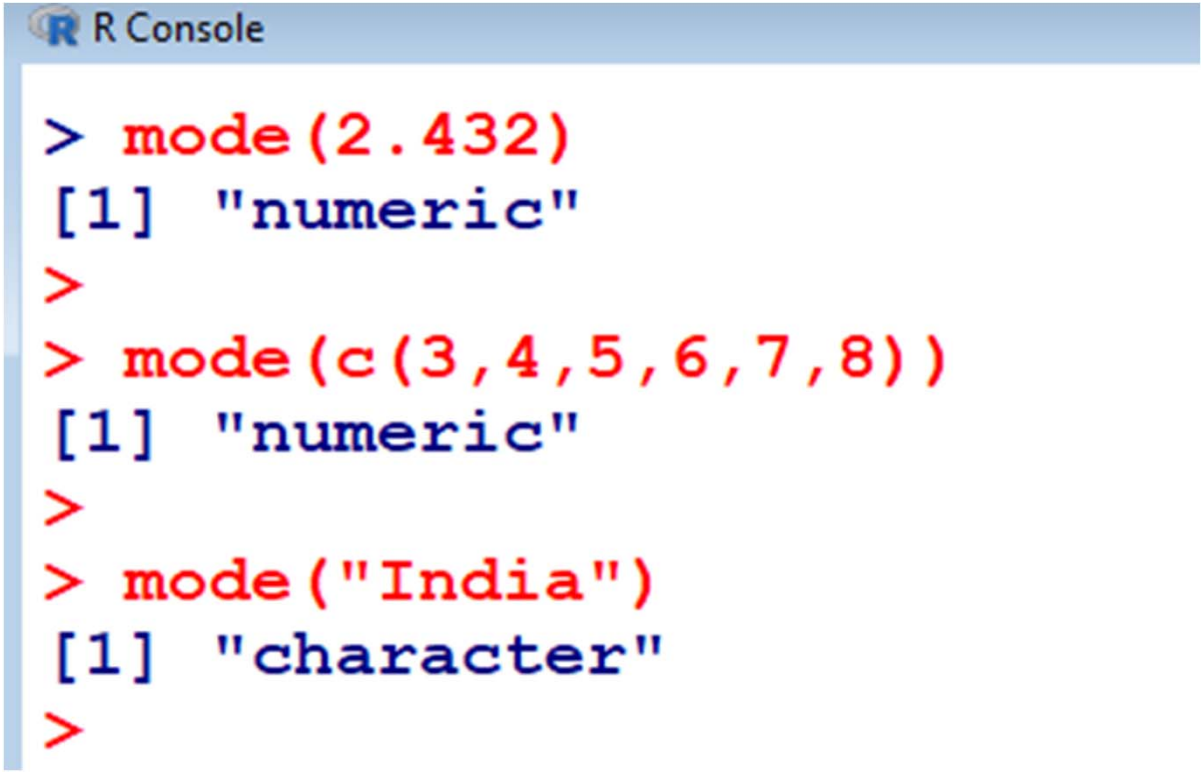
```
> mode(2.432)
[1] "numeric"

> mode(c(3,4,5,6,7,8))
[1] "numeric"

> mode("India")
[1] "character"

> mode(c("India", “CANADA”))
[1] "character"
```

Mode


R Console
> mode(2.432)
[1] "numeric"
>
> mode(c(3,4,5,6,7,8))
[1] "numeric"
>
> mode("India")
[1] "character"
>

Mode

Example

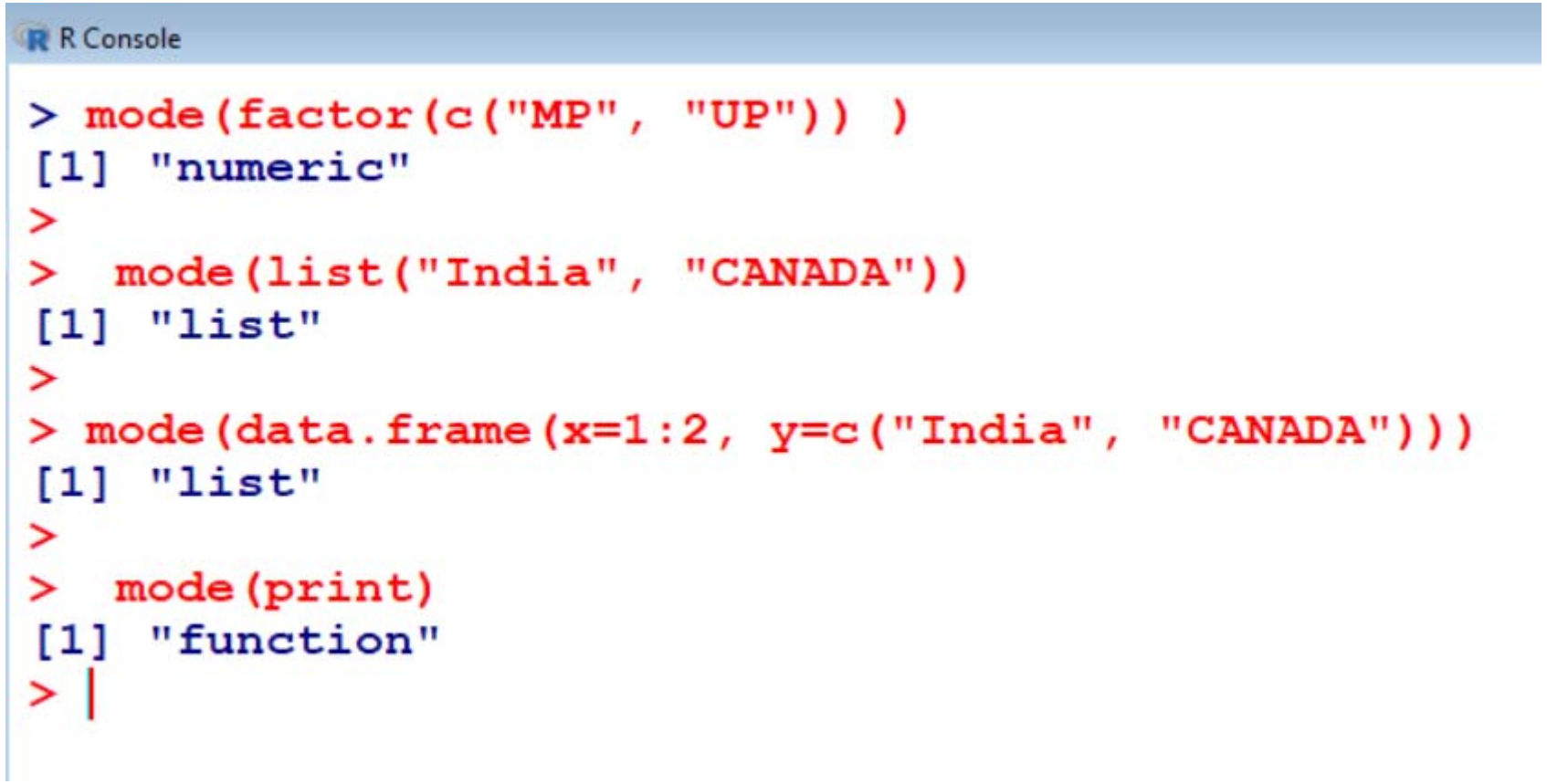
```
> mode(factor(c("MP", "UP")) )
[1] "numeric"

>  mode(list("India", "CANADA"))
[1] "list"

> mode(data.frame(x=1:2, y=c("India", "CANADA")))
[1] "list"

>  mode(print)
[1] "function"
```

Mode


R Console
> mode(factor(c("MP", "UP")))
[1] "numeric"
>
> mode(list("India", "CANADA"))
[1] "list"
>
> mode(data.frame(x=1:2, y=c("India", "CANADA")))
[1] "list"
>
> mode(print)
[1] "function"
>