Introduction to R Software

Swayam Prabha

Lecture 21

Vector Indexing

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

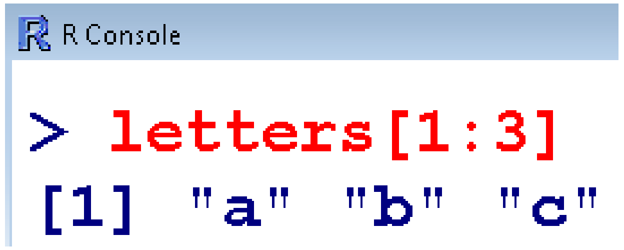
Slides can be downloaded from

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

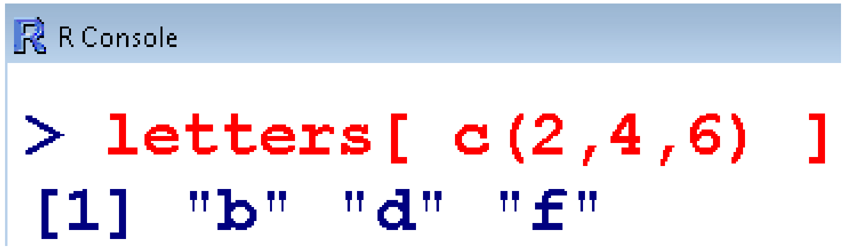
Vector Indexing

A vector of positive integers (`letters` and `LETTERS` return the 26 lowercase and uppercase letters, respectively).

```
> letters[1:3]
[1] "a" "b" "c"

> letters[ c(2,4,6) ]
[1] "b" "d" "f"

> LETTERS[1:3]
[1] "A" "B" "C"

> LETTERS[ c(2,4,6) ]
[1] "B" "D" "F"
```



```
> letters[1:3]
[1] "a" "b" "c"
```



```
> letters[ c(2,4,6) ]
[1] "b" "d" "f"
```

Vector Indexing

❑ **A logical vector**

```
> x <- 1:10
>x
[1] 1 2 3 4 5 6 7 8 9 10

> x[ (x > 5) ]
[1] 6 7 8 9 10
```

Vector Indexing

❑ **A logical vector**

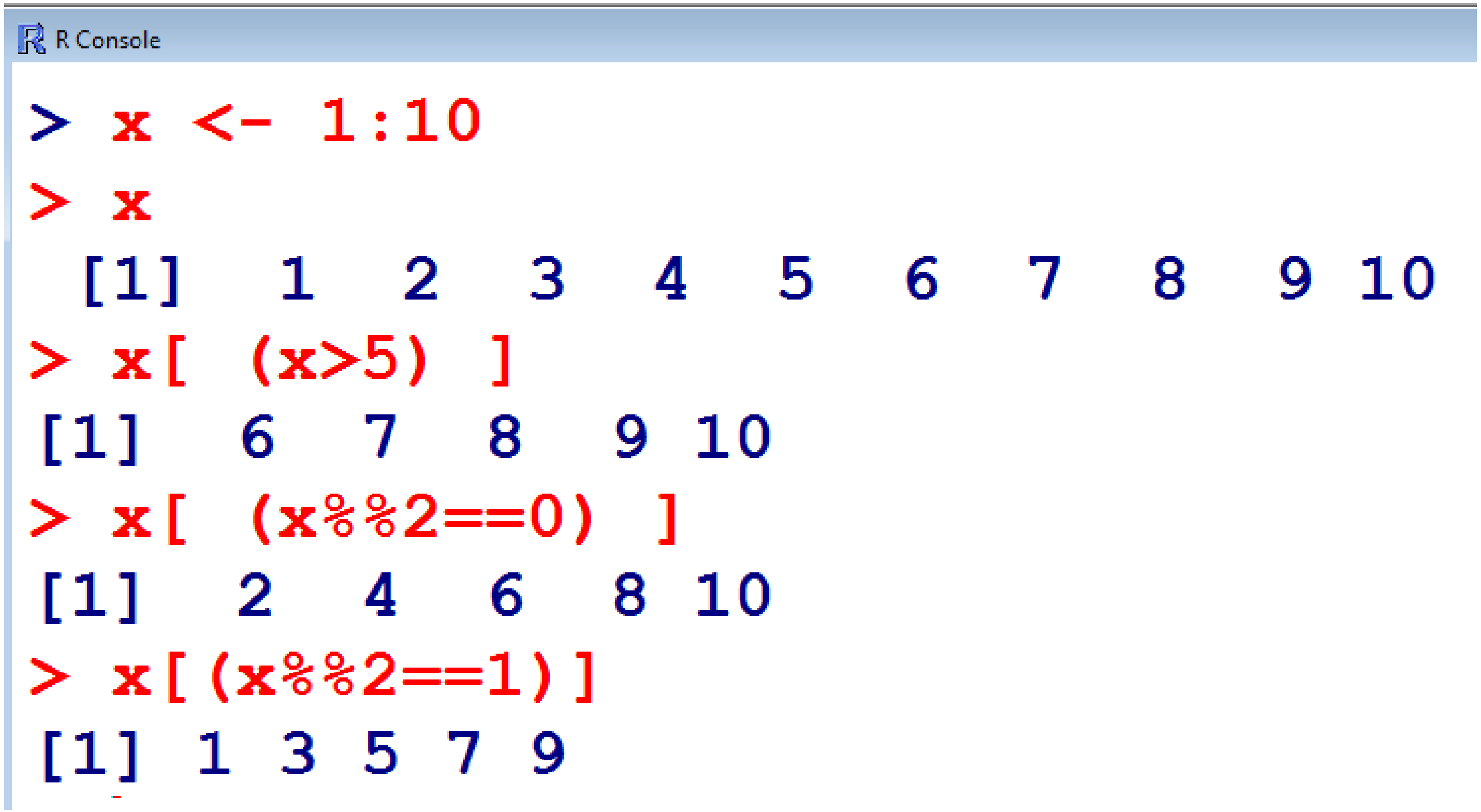
```
> x <- 1:10
>x
[1] 1 2 3 4 5 6 7 8 9 10


> x[ (x%%2==0) ] #%% indicates x mod y
[1] 2 4 6 8 10   #values for which x mod 2 is 0
                 #remainder is zero
```

Vector Indexing

❑ **A logical vector**

```
> x <- 1:10
>x
[1] 1 2 3 4 5 6 7 8 9 10


> x[ (x%%2==1) ]
[1] 1 3 5 7 9     #values for which x mod 2 is 1
                  #remainder is 1
```

Vector Indexing



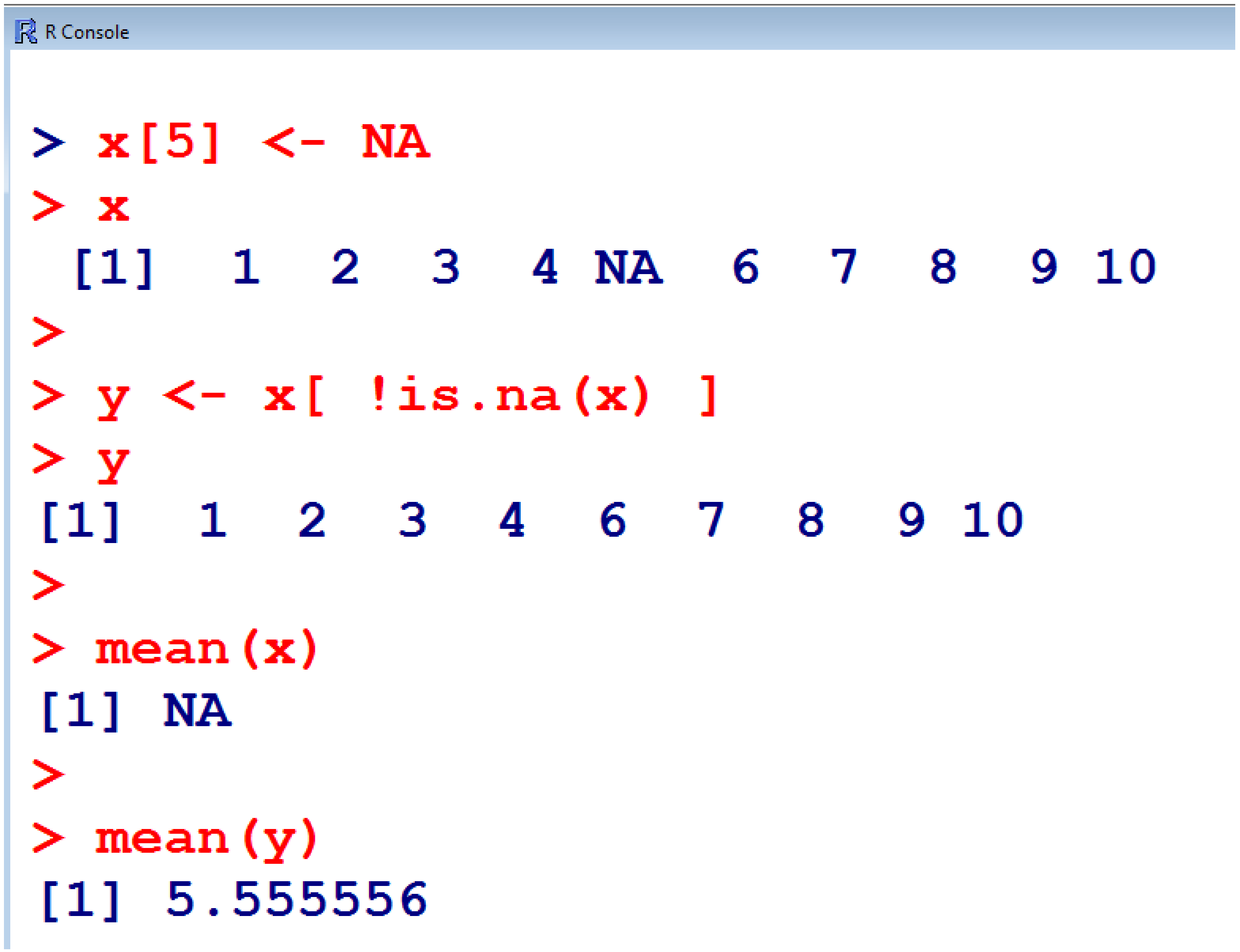
```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x[ (x>5) ]
[1]  6  7  8  9 10
> x[ (x%%2==0) ]
[1]  2  4  6  8 10
> x[(x%%2==1)]
[1] 1 3 5 7 9
```

Vector Indexing

- A logical vector

```
> x[5] <- NA

> x
[1] 1 2 3 4 NA 6 7 8 9 10


> y <- x[ !is.na(x) ]  #! Means negation

> y

[1] 1 2 3 4 6 7 8 9 10 # 5 is missing


> mean(x)
[1] NA


> mean(y)
[1] 5.555556
```

Vector Indexing


R Console

```
> x[5] <- NA
> x
 [1]  1  2  3  4 NA  6  7  8  9 10
>
> y <- x[ !is.na(x) ]
> y
[1]  1  2  3  4  6  7  8  9 10
>
> mean(x)
[1] NA
>
> mean(y)
[1] 5.555556
```

Vector Indexing

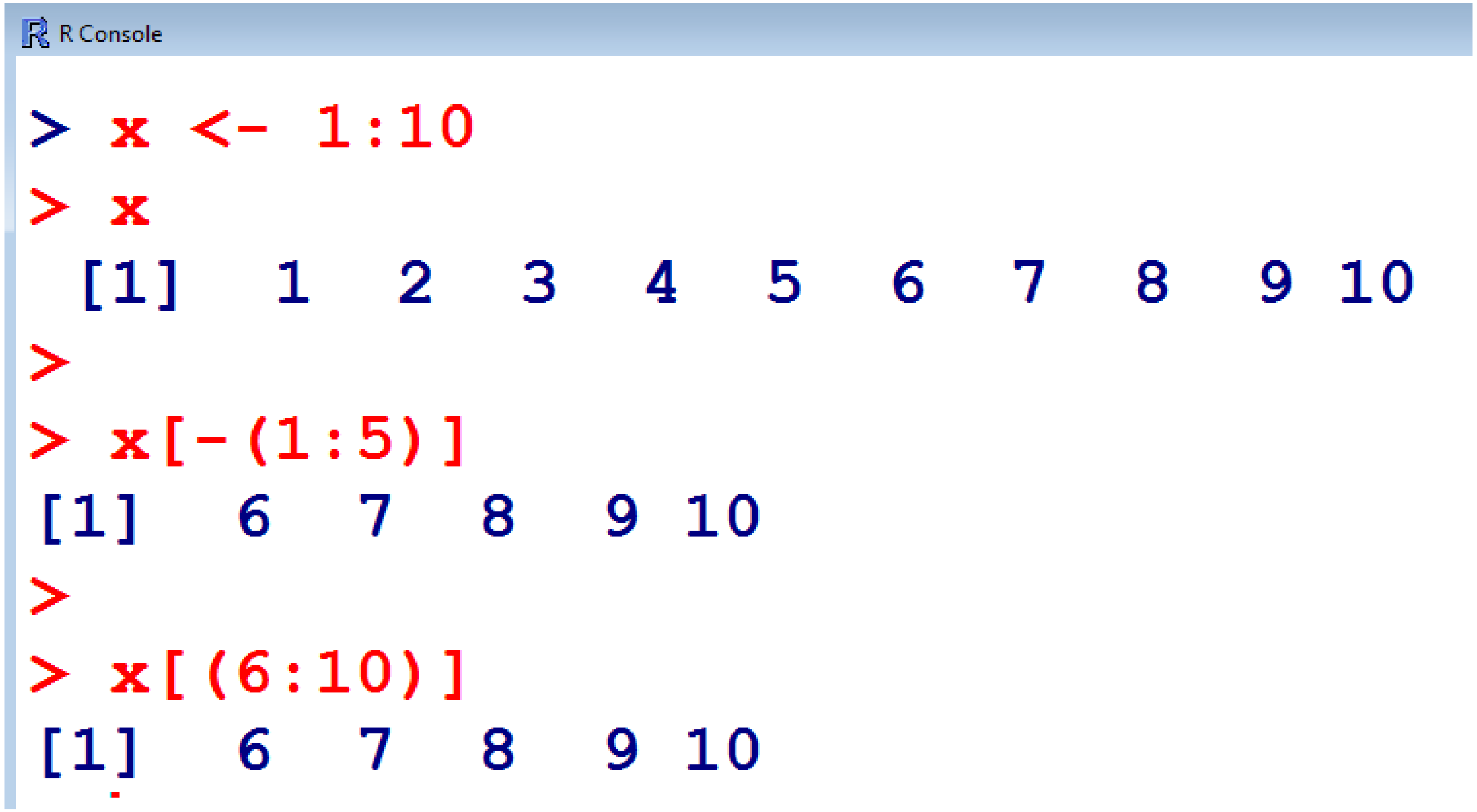
- Vector of negative integers

```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10

> x[-(1:5)]
[1]  6  7  8  9 10
```

has the same outcome as

```
> x[(6:10)]
[1]  6  7  8  9 10
```

Vector Indexing



```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
>
> x[-(1:5)]
[1]  6  7  8  9 10
>
> x[(6:10)]
[1]  6  7  8  9 10
```

❑ String vector

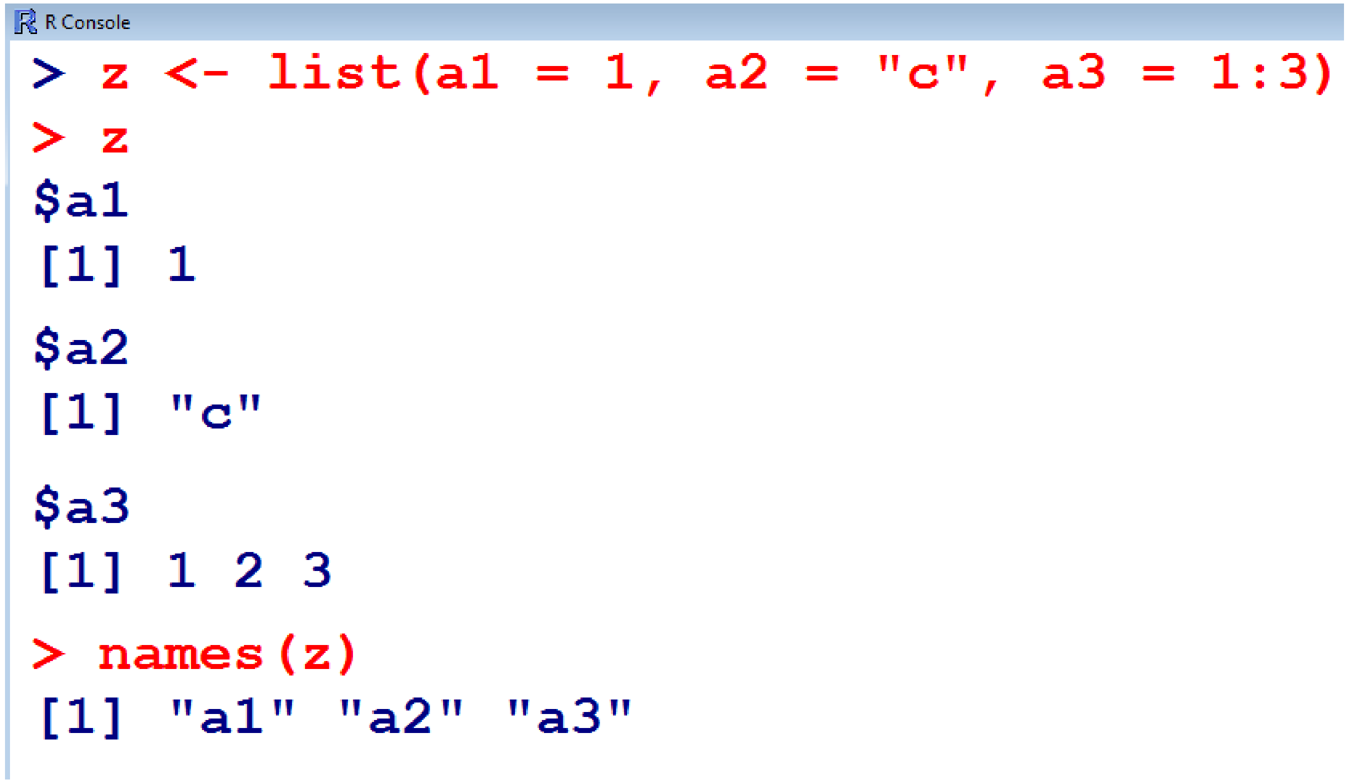
The elements of a vector can be named.

Using these names, we can access the vector elements.

names is used for functions to get or set the names of an object

```
> z <- list(a1 = 1, a2 = "c", a3 = 1:3)
> z
$a1
[1] 1
$a2
[1] "c"
$a3
[1] 1 2 3

> names(z)
[1] "a1" "a2" "a3"
```

❑ String vector



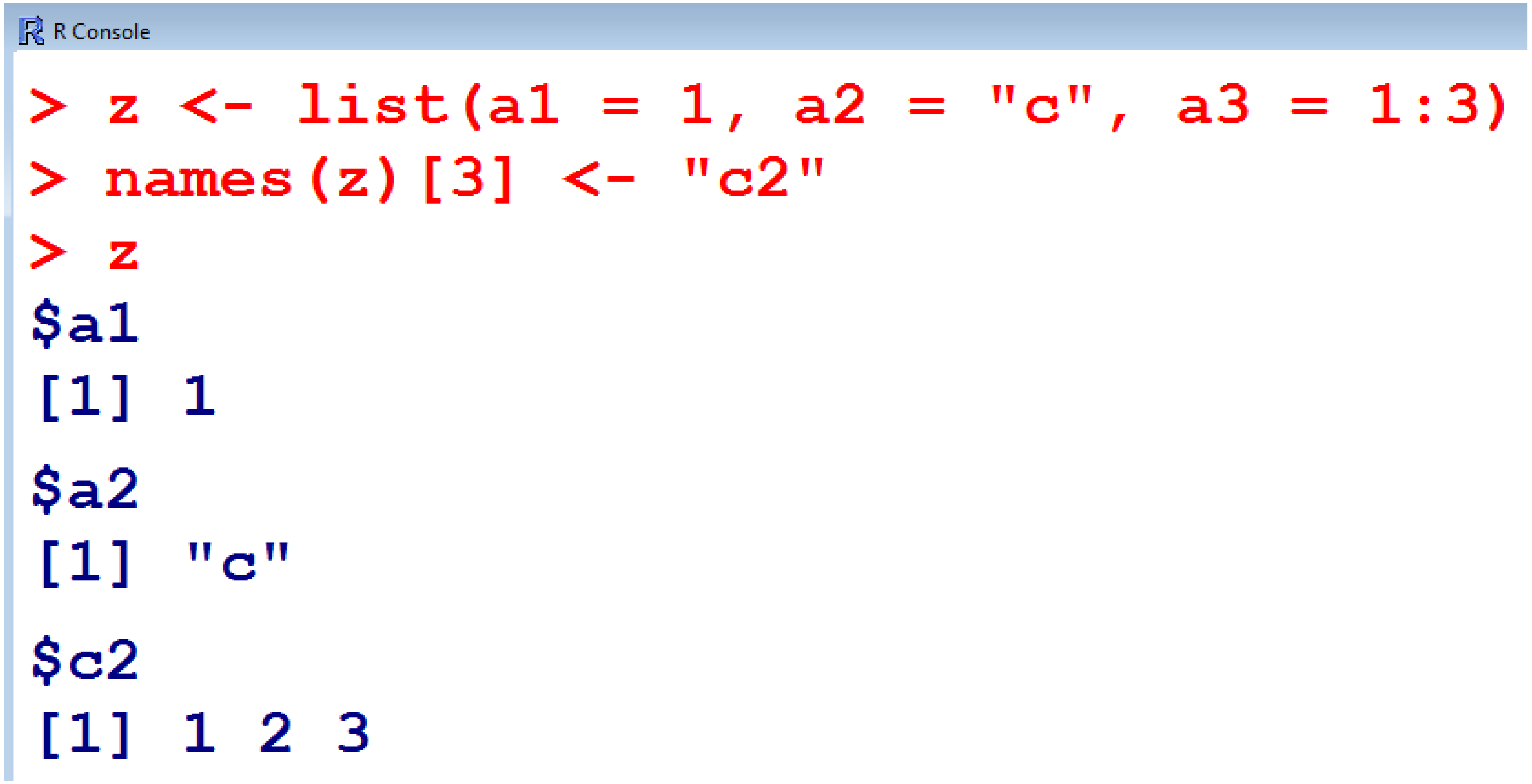
```
> z <- list(a1 = 1, a2 = "c", a3 = 1:3)
> z
$a1
[1] 1

$a2
[1] "c"

$a3
[1] 1 2 3

> names(z)
[1] "a1" "a2" "a3"
```

❑ String vector

Suppose want to change just the name of the third element.

```
> z <- list(a1 = 1, a2 = "c", a3 = 1:3)

> names(z)[3] <- "c2"

> z
$a1
[1] 1
$a2
[1] "c"
$c2
[1] 1 2 3
```

❑ String vector



```
> z <- list(a1 = 1, a2 = "c", a3 = 1:3)
> names(z)[3] <- "c2"
> z
$a1
[1] 1

$a2
[1] "c"

$c2
[1] 1 2 3
```

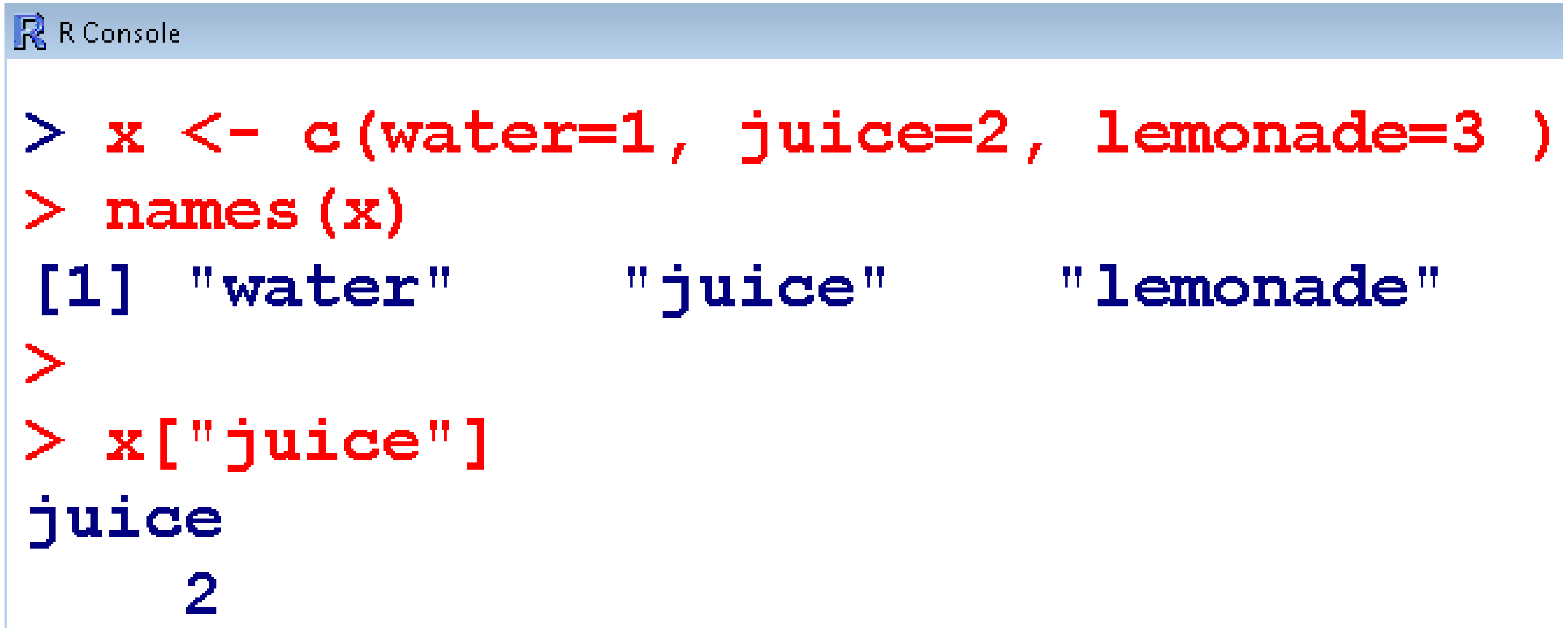
❑ String vector

Example

`names` is used for functions to get or set the names of an object

```
> x <- c(water=1, juice=2, lemonade=3 )

> names(x)
[1] "water"    "juice"    "lemonade"


> x["juice"]
juice
    2
```

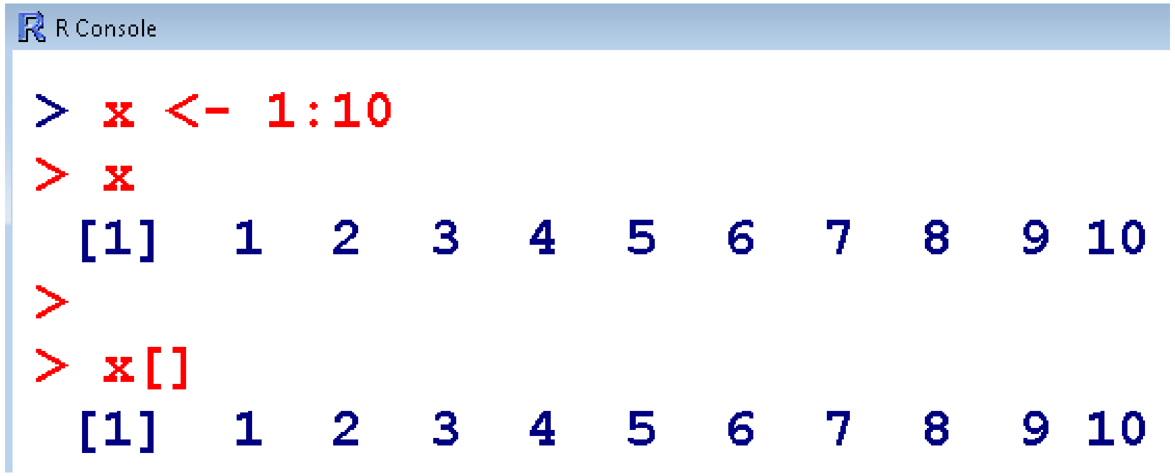
❑ String vector



```
> x <- c(water=1, juice=2, lemonade=3 )
> names(x)
[1] "water"     "juice"     "lemonade"
>
> x["juice"]
juice
    2
```

❑ Empty index

```
> x <- 1:10

>x
 [1] 1 2 3 4 5 6 7 8 9 10

> x[]
 [1] 1 2 3 4 5 6 7 8 9 10
```



```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
>
> x[]
 [1]  1  2  3  4  5  6  7  8  9 10
```

❑ Matrices created from Lists

List can be heterogeneous (mixed modes).

We can start with a heterogeneous list,

give it dimensions, and

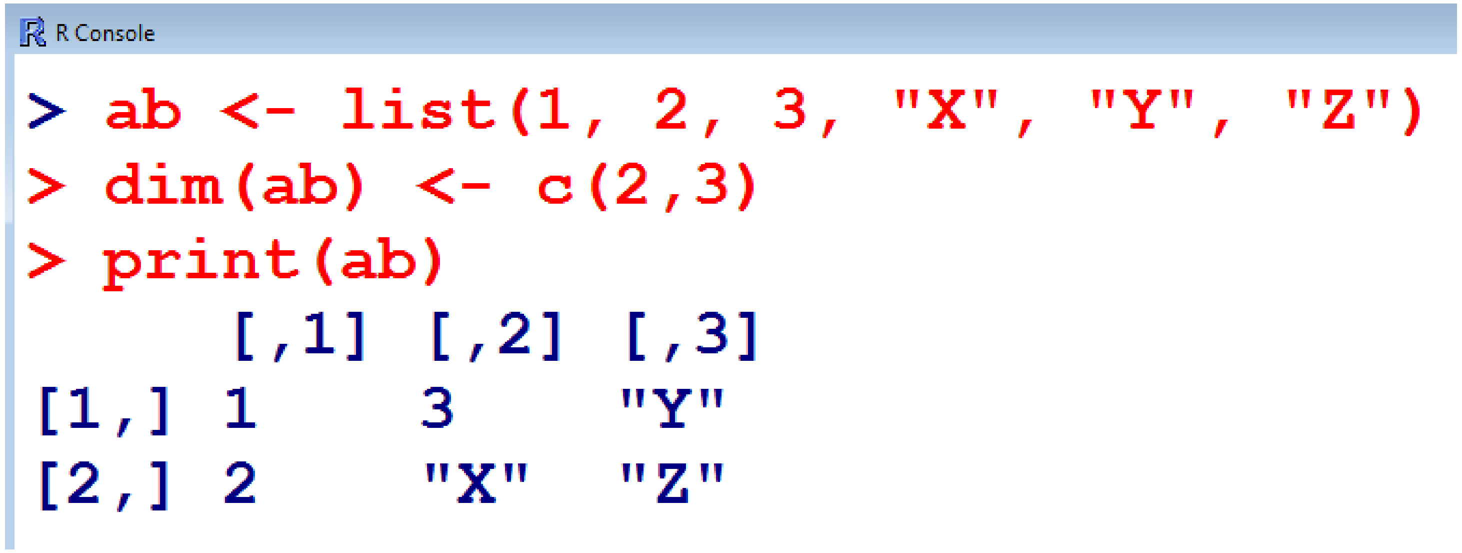
thus create a heterogeneous matrix

that is a mixture of numeric and character data:

Example

```
> ab <- list(1, 2, 3, "X", "Y", "Z")
> dim(ab) <- c(2,3)
> print(ab)
     [,1] [,2] [,3]
[1,] 1    3    "Y"
[2,] 2    "X"  "Z"
```

❑ Matrices created from Lists



```
> ab <- list(1, 2, 3, "X", "Y", "Z")
> dim(ab) <- c(2,3)
> print(ab)
     [,1] [,2] [,3]
[1,] 1    3    "Y"
[2,] 2    "X"  "Z"
```