Introduction to R Software

Swayam Prabha

Lecture 30

Factors – Examples and Operations

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from

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

Factors

The `factor` function encodes the vector of discrete values into a factor.
Usage

```
factor(x = character(), levels, labels =
levels, exclude = NA, …)
```

`levels` : Determines the categories of the factor variable.

Default is the sorted list of all the distinct values of `x`.

`labels` : (Optional) Vector of values that will be the labels of the

categories in the `levels` argument.

`exclude` : (Optional) It defines which levels will be classified as

NA in any output using the factor variable.

Factors

Example:

Suppose we roll a dice seven times and observe the outcome in the vector `y`.

```
> y <- c(1, 4, 3, 5, 4, 2, 4)
```

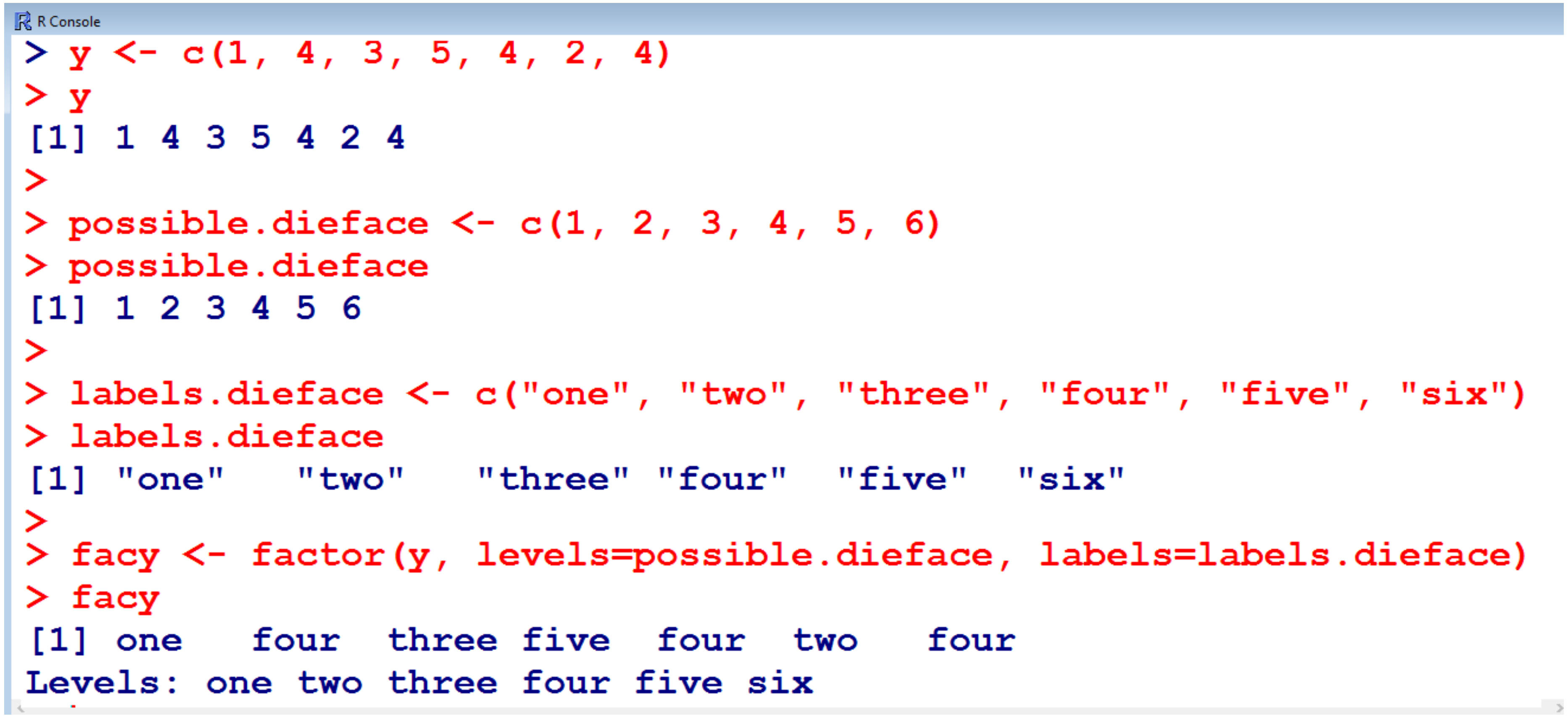
Possible values of upper face of die are 1 to 6 and we store them in a vector `possible.dieface`

```
> possible.dieface <- c(1, 2, 3, 4, 5, 6)
```

Label the rolls by the words “one”, “two”, ..., “six” and put them in the vector `labels.diefaces`:

```
> labels.dieface <- c("one", "two", "three",
"four", "five", "six")
```

Factors

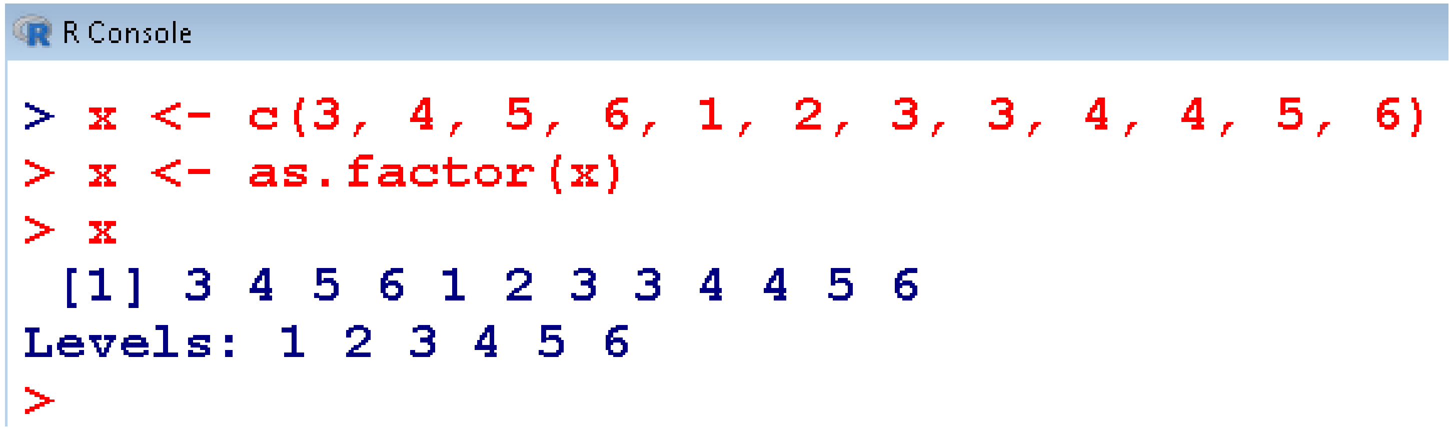


```
> y <- c(1, 4, 3, 5, 4, 2, 4)
> y
[1] 1 4 3 5 4 2 4
>
> possible.dieface <- c(1, 2, 3, 4, 5, 6)
> possible.dieface
[1] 1 2 3 4 5 6
>
> labels.dieface <- c("one", "two", "three", "four", "five", "six")
> labels.dieface
[1] "one"   "two"   "three" "four"  "five"  "six"
>
> facy <- factor(y, levels=possible.dieface, labels=labels.dieface)
> facy
[1] one   four  three five  four  two   four
Levels: one two three four five six
```

Factors

A vector can be turned into a factor with the command `as.factor`:

```
> x <- c(3, 4, 5, 6, 1, 2, 3, 3, 4, 4, 5, 6)

> x <- as.factor(x)
> x
 [1] 3 4 5 6 1 2 3 3 4 4 5 6
Levels: 1 2 3 4 5 6
```



```
> x <- c(3, 4, 5, 6, 1, 2, 3, 3, 4, 4, 5, 6)
> x <- as.factor(x)
> x
 [1] 3 4 5 6 1 2 3 3 4 4 5 6
Levels: 1 2 3 4 5 6
>
```

Factors

Example

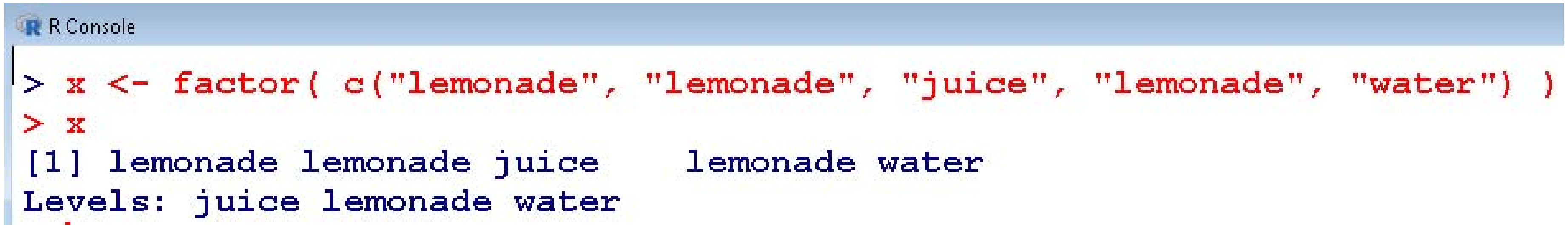
```
> x <- factor( c("lemonade", "lemonade",
"juice", "lemonade", "water") )

>x
[1] lemonade  lemonade  juice  lemonade  water
Levels: juice lemonade water
```

The single levels are ordered alphabetically:

```
juice --- lemonade --- water
```

Factors

Example



```
> x <- factor( c("lemonade", "lemonade", "juice", "lemonade", "water") )
> x
[1] lemonade lemonade juice    lemonade water
Levels: juice lemonade water
```

Factors

`class` function :

All objects in R have a class and function `class` reports it.

For simple vectors, this is just the mode, e.g. `"numeric"`, `"logical"`, `"character"`, `"list"`, `"matrix"`, `"array"`, `"factor"` and `"data.frame"`.

A special attribute class of the object is used to allow for an object-oriented style of programming in R.

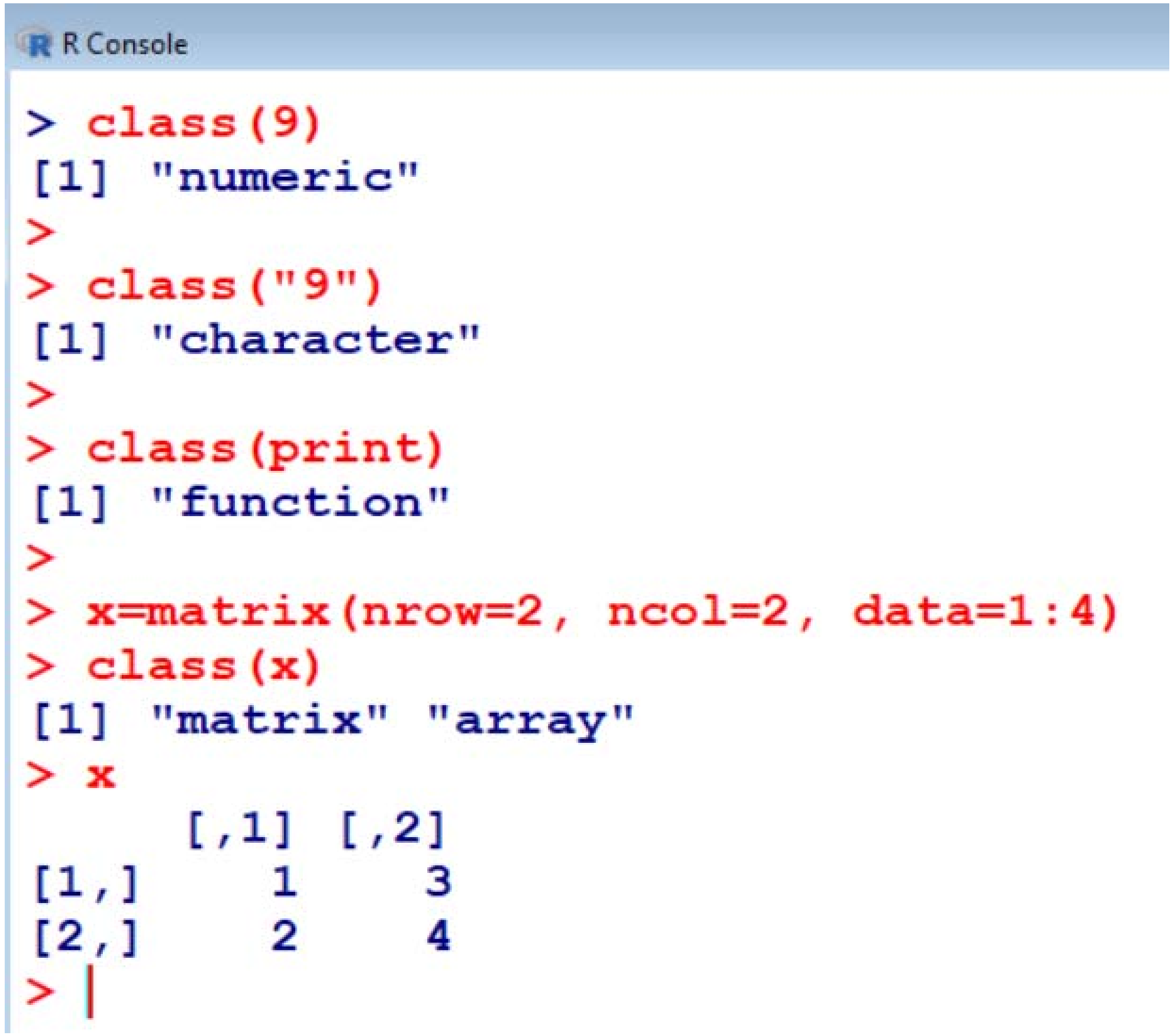
Factors

`class` **function :**

```
> class(9)
[1] "numeric"

> class("9")
[1] "character"

> class(print)
[1] "function"

> x=matrix(nrow=2, ncol=2, data=1:4)
> class(x)
[1] "matrix" "array"
```

Factors

`class` function :



```
> class(9)
[1] "numeric"
>
> class("9")
[1] "character"
>
> class(print)
[1] "function"
>
> x=matrix(nrow=2, ncol=2, data=1:4)
> class(x)
[1] "matrix" "array"
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> 
```

Factors

`unclass` function

For example if an object has class `"data.frame"`, it will be printed in a certain way, the `plot()` function will display it graphically in a certain way etc.

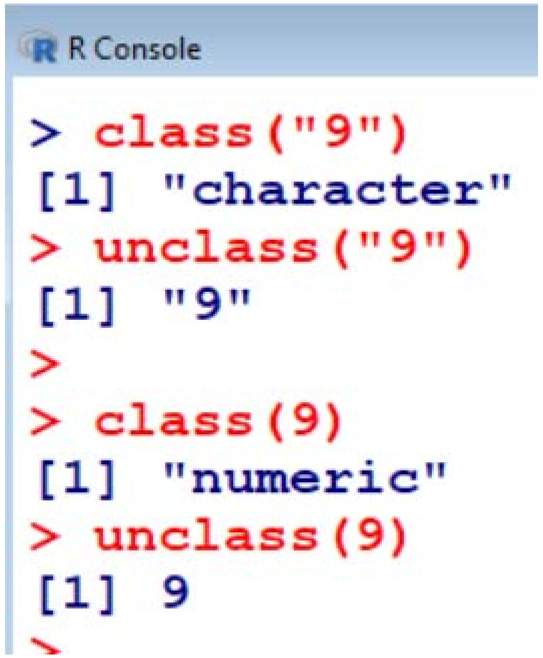
`unclass()` is used to temporarily remove the effects of class.

Use `help("unclass")` to get more information.

Factors

Examples: `unclass` function

```
> class("9")

[1] "character"

> unclass("9")

[1] "9"


> class(9)

[1] "numeric"

> unclass(9)

[1] 9
```



```
> class("9")
[1] "character"
> unclass("9")
[1] "9"
>
> class(9)
[1] "numeric"
> unclass(9)
[1] 9
>
```

Factors

Examples: `unclass` **function**

```
> colours = c("blue",  "green", "red" )
> colours
[1] "blue"  "green" "red"

> brands = c("A","A","B","B","B","B","C")
> brands
[1] "A" "A" "B" "B" "B" "B" "C"

> brands_fac = factor(brands)
> brands_fac
[1] A A B B B B C
Levels: A B C
```

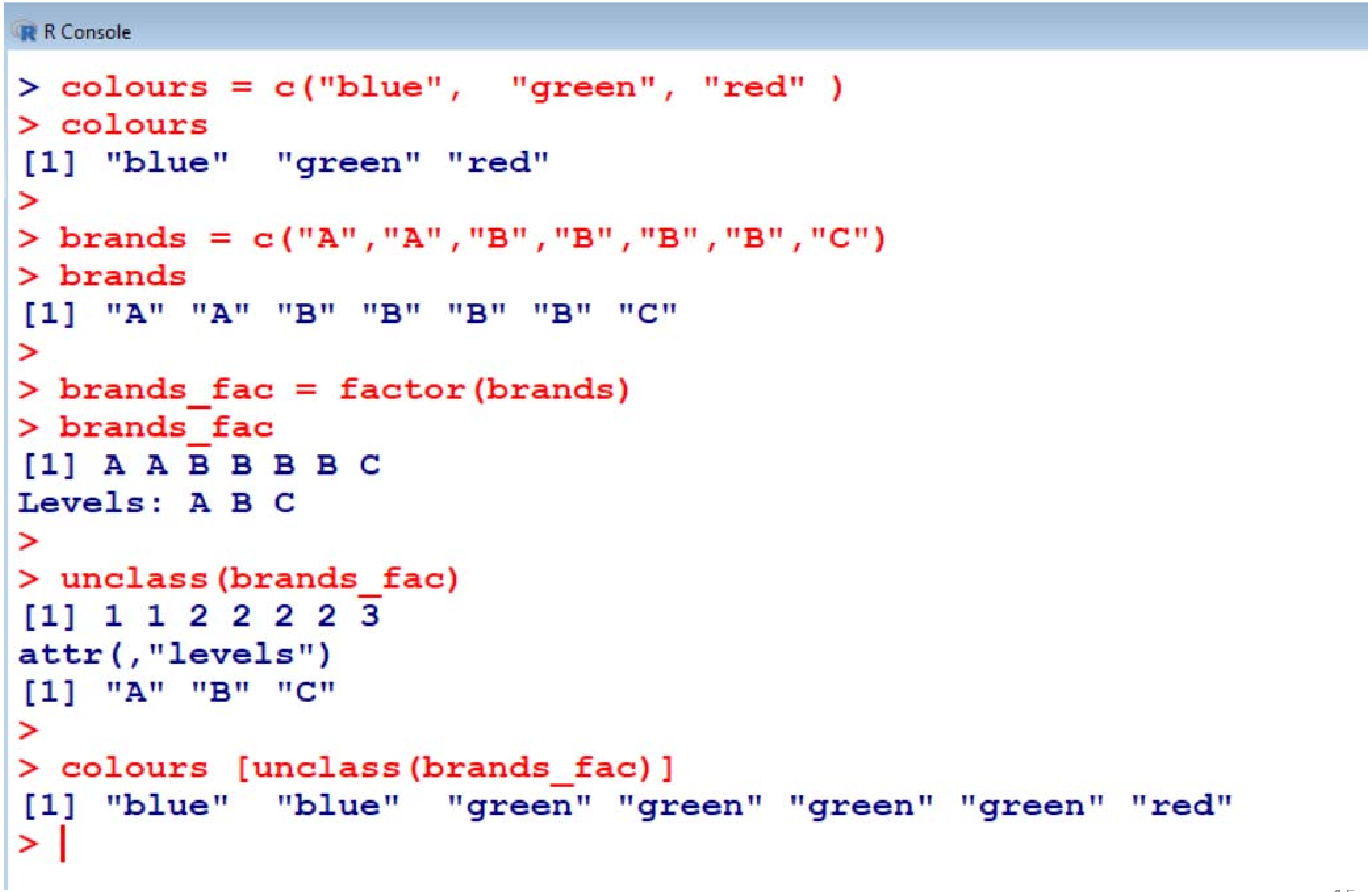
Factors

Examples: `unclass` **function**

```
> unclass(brands_fac)
[1] 1 1 2 2 2 2 3
attr(,"levels")
[1] "A" "B" "C"

> colours [unclass(brands_fac)]
[1] "blue"  "blue"  "green" "green" "green"
"green" "red"
```

Factors

Examples: `unclass` function



```
> colours = c("blue",  "green", "red" )
> colours
[1] "blue"  "green" "red"
>
> brands = c("A","A","B","B","B","B","C")
> brands
[1] "A" "A" "B" "B" "B" "B" "C"
>
> brands_fac = factor(brands)
> brands_fac
[1] A A B B B B C
Levels: A B C
>
> unclass(brands_fac)
[1] 1 1 2 2 2 2 3
attr(,"levels")
[1] "A" "B" "C"
>
> colours [unclass(brands_fac)]
[1] "blue"  "blue"  "green" "green" "green" "green" "red"
> |
```

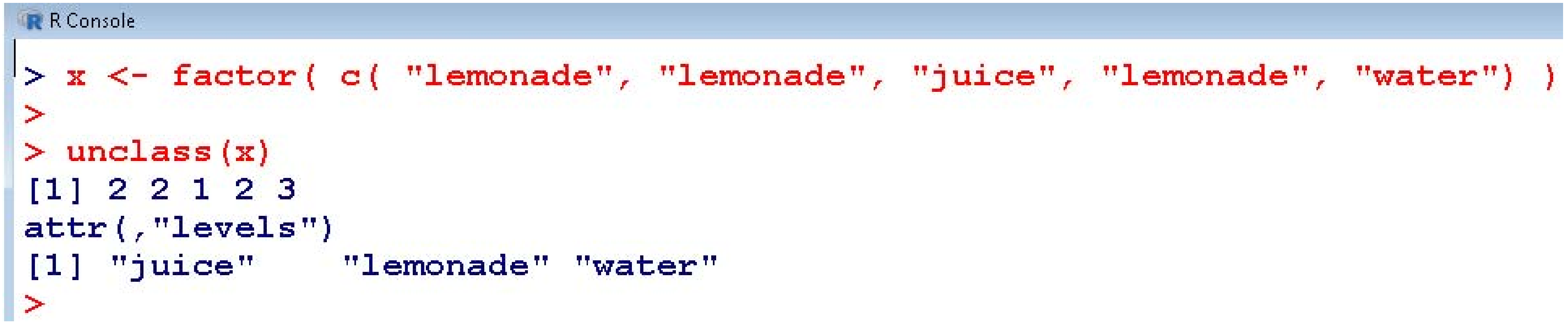
Factors

Examples: `unclass` function

The command `unclass` shows, an integer is assigned to every factor level:

```
> x <- factor( c( "lemonade", "lemonade",
"juice", "lemonade", "water") )

> unclass(x)
[1] 2 2 1 2 3
attr(,"levels")
[1] "juice" "lemonade" "water"
```

Factors



```
> x <- factor( c( "lemonade", "lemonade", "juice", "lemonade", "water") )
>
> unclass(x)
[1] 2 2 1 2 3
attr(,"levels")
[1] "juice"    "lemonade" "water"
>
```

Factors

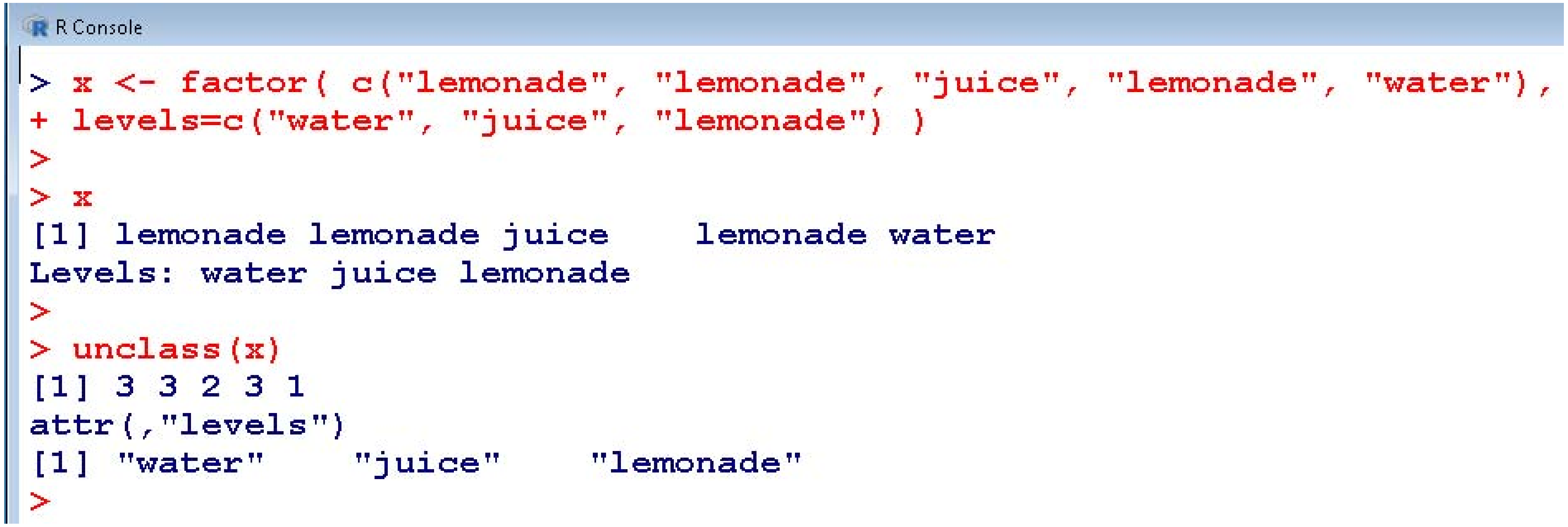
If a different assignment is desired, the parameter `levels` can be used:

```
> x <- factor( c("lemonade", "lemonade",
"juice", "lemonade", "water"),
levels=c("water", "juice", "lemonade") )

> x
[1] lemonade lemonade juice    lemonade water
Levels: water juice lemonade
```

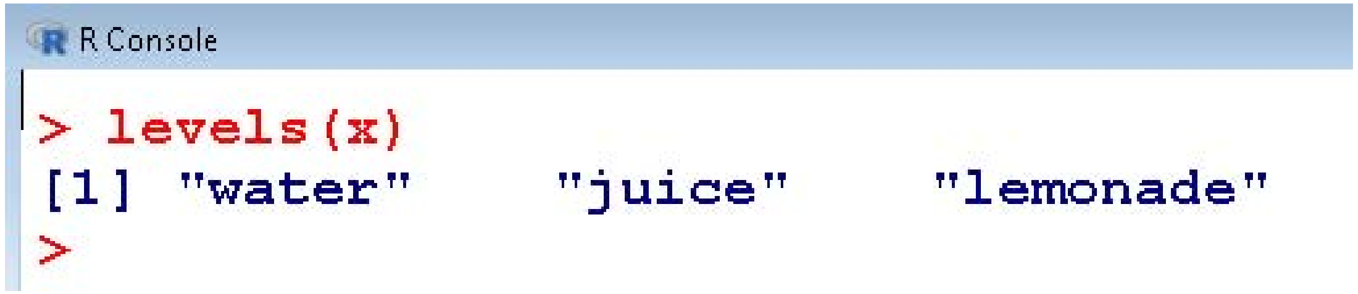
Factors

```
> unclass(x)
[1] 3 3 2 3 1
attr(,"levels")
[1] "water"    "juice"    "lemonade"

> levels(x)
[1] "water"    "juice"    "lemonade"
```

Factors



```
> x <- factor( c("lemonade", "lemonade", "juice", "lemonade", "water"),
+ levels=c("water", "juice", "lemonade") )
>
> x
[1] lemonade lemonade juice    lemonade water
Levels: water juice lemonade
>
> unclass(x)
[1] 3 3 2 3 1
attr(,"levels")
[1] "water"    "juice"    "lemonade"
>
```



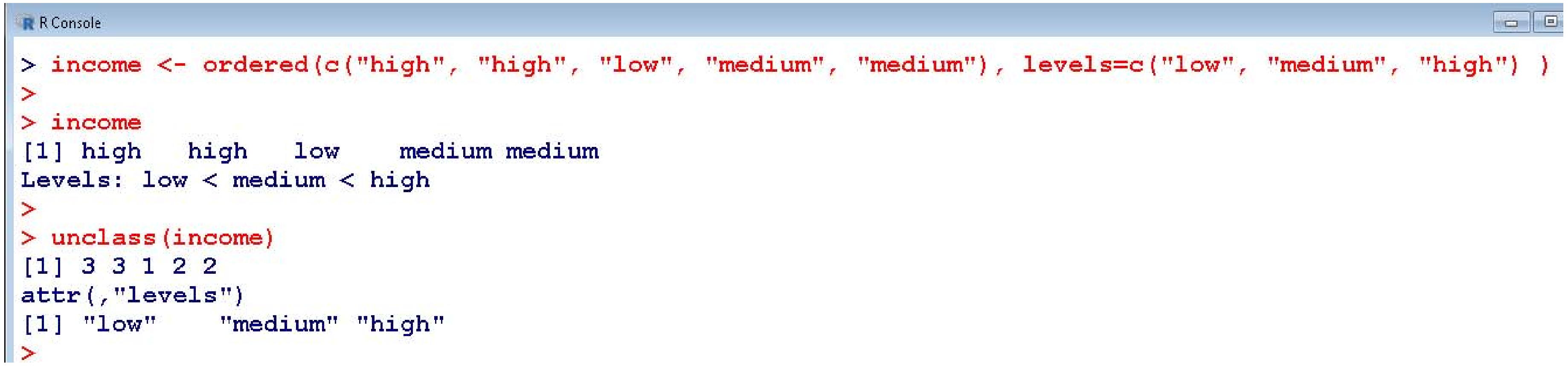
```
> levels(x)
[1] "water"    "juice"    "lemonade"
>
```

Factors

Example for an ordered factor:

```
> income <- ordered(c("high", "high", "low",
"medium", "medium"), levels=c("low", "medium",
"high") )

> income
[1] high   high   low    medium medium
Levels: low < medium < high

> unclass(income)
[1] 3 3 1 2 2
attr(,"levels")
[1] "low"    "medium" "high"
```

Factors



```
> income <- ordered(c("high", "high", "low", "medium", "medium"), levels=c("low", "medium", "high") )
>
> income
[1] high   high   low    medium medium
Levels: low < medium < high
>
> unclass(income)
[1] 3 3 1 2 2
attr(,"levels")
[1] "low"    "medium" "high"
>
```