

# **Introduction to R Software**

## **Factors**

**Shalabh**

**Department of Mathematics and Statistics**

**Indian Institute of Technology Kanpur**

# Factors

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

```
factor(x)
```

where `x` is a vector of strings or integers.

```
factor(x, levels)
```

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

# Factors

## Example

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

```
>x
```

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

```
Levels: juice lemonade water
```

The single levels are ordered alphabetically:

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

# Factors

## Example

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

# Factors

`unclass` 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

## `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

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

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

```
> unclass(x)
```

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

```
attr(,"levels")
```

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

# Factors



R Console

```
> unclass(x)
```

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

```
attr(,"levels")
```

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

# Factors

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

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

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

# Factors

```
> unclass(x)
```

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

```
attr(,"levels")
```

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

```
> levels(x)
```

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

# Factors

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

```
R Console
> 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

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

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

```
> x <- as.factor(x)
```

```
> x
```

```
[1] 4 5 1 2 3 3 4 4 5 6
```

```
Levels: 1 2 3 4 5 6
```

R Console

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

```
>
```

```
> x <- as.factor(x)
```

```
>
```

```
> x
```

```
[1] 4 5 1 2 3 3 4 4 5 6
```

```
Levels: 1 2 3 4 5 6
```