

# **Introduction to R Software**

## **Factors**

**Shalabh**

**Department of Mathematics and Statistics**

**Indian Institute of Technology Kanpur**

# Categorical variables

## Quantitative variables

### Example:

Height (in meters) – 1.65, 1.76, ...

## Qualitative variables

### Example:

Gender – Male, Female

Performance – Excellent, Good, Average, Bad ...

# Categorical variables

## Categorical variables

### Example:

**X : Gender – Male, Female**

**X = 0 if a person is male**

**X = 1 if a person is female**

### Example:

Performance	Excellent	Average	Good	Bad	Labels
X	1	2	3	4	Numeric codes

**The categories are stored internally as numeric codes, with labels to provide meaningful names for each code.**

# Factors

**Factors represent categorical variables and are used as grouping indicators.**

# Factors

## Example:

Suppose we denote the three colours of balls in a basket by following numbers:

**Red** = 1, **Blue** = 2, **Green** = 3

Suppose we draw five balls with following colours:

**Red**, **Green**, **Green**, **Blue**, **Red**

This outcome of colours can be coded by numbers

Performance	Excellent	Average	Good	Bad	Labels
X	1	2	3	4	Numeric codes

# Factors

**Each character is mapped to a code.**

**Factors represent categorical variables and are used as grouping indicators.**

**The categories are stored internally as numeric codes, with labels to provide meaningful names for each code.**

# Factors

The order of the labels is important.

First label is mapped to code 1.

Second label is mapped to code 2 and so on.

The values of the codes are always restricted to  $1, 2, \dots, k$ , to represent  $k$  discrete categories.

Here “Red” is mapped to code 1,

“Blue” is mapped to code 2 and

“Green” is mapped to code 3.

# Factors

We have a vector of character strings or integers.

R's term for a categorical variable is a factor.

In R, each possible value of a categorical variable is called a level.

A vector of levels is called a factor.

A categorical variable is characterized by a (here: finite) number of levels called as factor levels.



# Factors

To define a factor, we start with

- a vector of values,
- a second vector that gives the collection of possible values, and
- a third vector that gives labels to the possible values.

# Factors

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

```
factor(x)
```

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

If the vector contains only a subset of possible values and not the entire values, then include a second argument that gives the possible levels of the factor:

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

# Factors

## 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

Look into **help**

**> help("factor")**

factor {base}

R Documentation

## Factors

### Description

The function `factor` is used to encode a vector as a factor (the terms ‘category’ and ‘enumerated type’ are also used for factors). If argument `ordered` is `TRUE`, the factor levels are assumed to be ordered. For compatibility with S there is also a function `ordered`.

`is.factor`, `is.ordered`, `as.factor` and `as.ordered` are the membership and coercion functions for these classes.

### Usage

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

```
ordered(x, ...)
```

```
is.factor(x)  
is.ordered(x)
```

# Factors

## Example:

Suppose we roll a die 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)
```

# Factors

## Example:

We wish to label the rolls by the words “one”, “two”, ..., “six”.

We put these labels in the vector `labels.diefaces`:

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

Construct the factor variable `facy` using the function `factor`:

```
> facy <- factor(y, levels = possible.dieface,  
labels = labels.dieface)
```

# Factors

## Example:

Observe the difference between a character vector and a factor.

```
> facy
```

```
[1] one four three five four two four
```

```
Levels: one two three four five six
```

Note that

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

# Factors

```
R Console
> 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
```