**R FACTORS**

Factors

Attributes of Factors in R Language

x: It is the vector that needs to be converted into a factor.

Levels: It is a set of distinct values which are given to the input vector x.

Labels: It is a character vector corresponding to the number of labels.

Exclude: This will mention all the values you want to exclude.

Ordered: This logical attribute decides whether the levels are ordered.

nmax: It will decide the upper limit for the maximum number of levels.



Creating a Factor in R Programming Language

The command used to create or modify a factor in R language is – factor() with a vector as input.

The two steps to creating an R factor :

Creating a vector

Converting the vector created into a factor using function factor()

R - Factors GeeksforGeeks

Factors are used to categorize data. Examples of factors are:

Demography: Male/Female

Music: Rock, Pop, Classic, Jazz

Training: Strength, Stamina

To create a factor, use the factor() function and add a vector as argument:

**Example**

# Create a factor

music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))

# Print the factor

music\_genre

You can see from the example above that that the factor has four levels (categories): Classic, Jazz, Pop and Rock.

To only print the levels, use the levels() function:

**Example**

music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))

levels(music\_genre)

You can also set the levels, by adding the levels argument inside the factor() function:

**Example**

music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"), levels = c("Classic", "Jazz", "Pop", "Rock", "Other"))

levels(music\_genre)

**Factor Length**

Use the length() function to find out how many items there are in the factor:

Example

music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))

length(music\_genre)