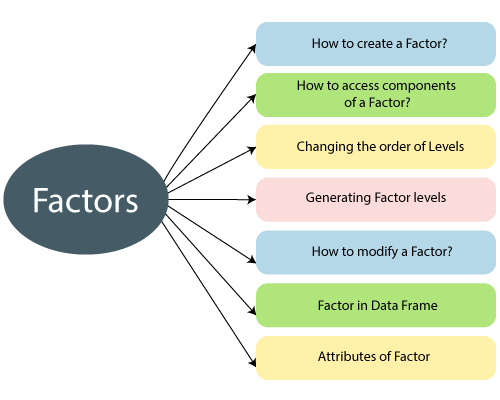
R factors

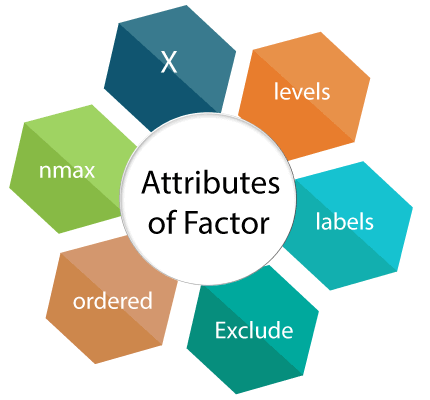
The factor is a data structure which is used for fields which take only predefined finite number of values. These are the variable which takes a limited number of different values. These are the data objects which are used to categorize the data and to store it on multiple levels. It can store both integers and strings values, and are useful in the column that has a limited number of unique values.



Factors have labels which are associated with the unique integers stored in it. It contains predefined set value known as levels and by default R always sorts levels in alphabetical order.

Attributes of a factor

There are the following attributes of a factor in R



1. **X**  
   It is the input vector which is to be transformed into a factor.
2. **levels**  
   It is an input vector that represents a set of unique values which are taken by x.
3. **labels**  
   It is a character vector which corresponds to the number of labels.
4. **Exclude**  
   It is used to specify the value which we want to be excluded,
5. **ordered**  
   It is a logical attribute which determines if the levels are ordered.
6. **nmax**  
   It is used to specify the upper bound for the maximum number of level.

How to create a factor?

In R, it is quite simple to create a factor. A factor is created in two steps

1. In the first step, we create a vector.
2. Next step is to convert the vector into a factor,

R provides factor() function to convert the vector into factor. There is the following syntax of factor() function

1. factor\_data**<-** factor(vector)

Let's see an example to understand how factor function is used.

**Example**

1. # Creating a vector as input.
2. data **<-** c("Shubham","Nishka","Arpita","Nishka","Shubham","Sumit","Nishka","Shubham","Sumit","Arpita","Sumit")
4. print(data)
5. print(is.factor(data))
7. # Applying the factor function.
8. factor\_data**<-** factor(data)
10. print(factor\_data)
11. print(is.factor(factor\_data))

**Output**

[1] "Shubham" "Nishka" "Arpita" "Nishka" "Shubham" "Sumit" "Nishka"

[8] "Shubham" "Sumit" "Arpita" "Sumit"

[1] FALSE

[1] Shubham Nishka Arpita Nishka Shubham Sumit Nishka Shubham Sumit

[10] Arpita Sumit

Levels: Arpita Nishka Shubham Sumit

[1] TRUE

Accessing components of factor

Like vectors, we can access the components of factors. The process of accessing components of factor is much more similar to the vectors. We can access the element with the help of the indexing method or using logical vectors. Let's see an example in which we understand the different-different ways of accessing the components.

**Example**

1. # Creating a vector as input.
2. data **<-** c("Shubham","Nishka","Arpita","Nishka","Shubham","Sumit","Nishka","Shubham","Sumit","Arpita","Sumit")
4. # Applying the factor function.
5. factor\_data**<-** factor(data)
7. #Printing all elements of factor
8. print(factor\_data)
10. #Accessing 4th element of factor
11. print(factor\_data[4])
13. #Accessing 5th and 7th element
14. print(factor\_data[c(5,7)])
16. #Accessing all elemcent except 4th one
17. print(factor\_data[-4])
19. #Accessing elements using logical vector
20. print(factor\_data[c(TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,TRUE,FALSE,FALSE,FALSE,TRUE)])

**Output**

[1] Shubham Nishka Arpita Nishka Shubham Sumit Nishka Shubham Sumit

[10] Arpita Sumit

Levels: Arpita Nishka Shubham Sumit

[1] Nishka

Levels: Arpita Nishka Shubham Sumit

[1] Shubham Nishka

Levels: Arpita Nishka Shubham Sumit

[1] Shubham Nishka Arpita Shubham Sumit Nishka Shubham Sumit Arpita

[10] Sumit

Levels: Arpita Nishka Shubham Sumit

[1] Shubham Shubham Sumit Nishka Sumit

Levels: Arpita Nishka Shubham Sumit

Modification of factor

Like data frames, R allows us to modify the factor. We can modify the value of a factor by simply re-assigning it. In R, we cannot choose values outside of its predefined levels means we cannot insert value if it's level is not present on it. For this purpose, we have to create a level of that value, and then we can add it to our factor.

Let's see an example to understand how the modification is done in factors.

**Example**

1. # Creating a vector as input.
2. data **<-** c("Shubham","Nishka","Arpita","Nishka","Shubham")
4. # Applying the factor function.
5. factor\_data**<-** factor(data)
7. #Printing all elements of factor
8. print(factor\_data)
10. #Change 4th element of factor with sumit
11. factor\_data[4] **<-**"Arpita"
12. print(factor\_data)
14. #change 4th element of factor with "Gunjan"
15. factor\_data[4] **<-** "Gunjan"    # cannot assign values outside levels
16. print(factor\_data)
18. #Adding the value to the level
19. levels(factor\_data) **<-** c(levels(factor\_data),"Gunjan")#Adding new level
20. factor\_data[4] **<-** "Gunjan"
21. print(factor\_data)

**Output**

[1] Shubham Nishka Arpita Nishka Shubham

Levels: Arpita Nishka Shubham

[1] Shubham Nishka Arpita Arpita Shubham

Levels: Arpita Nishka Shubham

Warning message:

In `[<-.factor`(`\*tmp\*`, 4, value = "Gunjan") :

invalid factor level, NA generated

[1] Shubham Nishka Arpita Shubham

Levels: Arpita Nishka Shubham

[1] Shubham Nishka Arpita Gunjan Shubham

Levels: Arpita Nishka Shubham Gunjan

Factor in Data Frame

When we create a frame with a column of text data, R treats this text column as categorical data and creates factor on it.

**Example**

1. # Creating the vectors for data frame.
2. height **<-** c(132,162,152,166,139,147,122)
3. weight **<-** c(40,49,48,40,67,52,53)
4. gender **<-** c("male","male","female","female","male","female","male")
6. # Creating the data frame.
7. input\_data**<-** data.frame(height,weight,gender)
8. print(input\_data)
10. # Testing if the gender column is a factor.
11. print(is.factor(input\_data$gender))
13. # Printing the gender column to see the levels.
14. print(input\_data$gender)

**Output**

height weight gender

1 132 40 male

2 162 49 male

3 152 48 female

4 166 40 female

5 139 67 male

6 147 52 female

7 122 53 male

[1] TRUE

[1] male male female female male female male

Levels: female male

Changing order of the levels

In R, we can change the order of the levels in the factor with the help of the factor function.

**Example**

1. data **<-** c("Nishka","Gunjan","Shubham","Arpita","Arpita","Sumit","Gunjan","Shubham")
2. # Creating the factors
3. factor\_data**<-** factor(data)
4. print(factor\_data)
6. # Apply the factor function with the required order of the level.
7. new\_order\_factor**<-** factor(factor\_data,levels = c("Gunjan","Nishka","Arpita","Shubham","Sumit"))
8. print(new\_order\_factor)

**Output**

[1] Nishka Gunjan Shubham Arpita Arpita Sumit Gunjan Shubham

Levels: Arpita Gunjan Nishka Shubham Sumit

[1] Nishka Gunjan Shubham Arpita Arpita Sumit Gunjan Shubham

Levels: Gunjan Nishka Arpita Shubham Sumit

Generating Factor Levels

R provides gl() function to generate factor levels. This function takes three arguments i.e., n, k, and labels. Here, n and k are the integers which indicate how many levels we want and how many times each level is required.

There is the following syntax of gl() function which is as follows

1. gl(n, k, labels)
2. n indicates the number of levels.
3. k indicates the number of replications.
4. labels is a vector of labels for the resulting factor levels.

**Example**

1. gen\_factor**<-** gl(3,5,labels=c("BCA","MCA","B.Tech"))
2. gen\_factor

**Output**

[1] BCA BCA BCA BCA BCA MCA MCA MCA MCA MCA

[11] B.Tech B.Tech B.Tech B.Tech B.Tech

Levels: BCA MCA B.Tech