R_Factors

Mr. Sachin B.

Introduction

Factors are the data objects which are used to categorize the data and store it as levels which are nothing but limited number of different values. Factor in R is also known as a categorical variable that stores both string and integer data values as levels. They are useful in data analysis for statistical modeling. **Syntax:** factor(x = character(), levels, labels = levels, exclude = NA, ordered = is.ordered(x), nmax = NA)

Creating Factor (Direct Approach)

```
fact <- factor(c("A","B","A","B","B","A","A"))
print(fact)
```

```
## [1] A B A B B A A
## Levels: A B
```

Creating Factor (Indirect Approach)

```
# Create gender.vector
gender.vector <- c("Male","Female","Female","Male","Female")

print(gender.vector)
```

```
## [1] "Male"   "Female" "Female" "Male"   "Female"
```

```
class(gender.vector)
```

```
## [1] "character"
```

```
# Convert gender.vector to a gender.factor
gender.factor <- factor(gender.vector)
print(gender.factor)
```

```
## [1] Male   Female Female Male   Female
## Levels: Female Male
```

```
# Checking It is Factor or not?
is.factor(gender.factor)
```

```
## [1] TRUE
```

```
# class of Factor
class(gender.factor)
```

```
## [1] "factor"
```

```
#No. of Levels of Factor
nlevels(gender.factor)
```

```
## [1] 2
```

```
#Levels of Factor
levels(gender.factor)
```

```
## [1] "Female" "Male"
```

Structure of Factor

```
gender.factor
```

```
## [1] Male   Female Female Male   Female
## Levels: Female Male
```

```
str(gender.factor)
```

```
##  Factor w/ 2 levels "Female","Male": 2 1 1 2 1
```

Re-Level

```
gender.factor
```

```
## [1] Male   Female Female Male   Female
## Levels: Female Male
```

```
as.numeric(gender.factor)
```

```
## [1] 2 1 1 2 1
```

```
gender.factor<-relevel(gender.factor,ref = "Male")
gender.factor
```

```
## [1] Male   Female Female Male   Female
## Levels: Male Female
```

```r
as.numeric(gender.factor)
```

```
## [1] 1 2 2 1 2
```

Ordering a Categorical Variable

```r
#Checking Order of gender.factor
gender.factor[1] < gender.factor[2]
```

```
## Warning in Ops.factor(gender.factor[1], gender.factor[2]): '<' not meaningful
## for factors
```

```
## [1] NA
```

```r
pant <- c("XL","L","XL","XXL","L","XL")
pant.factor <- factor(pant,ordered = TRUE,levels = c("L","XL","XXL"))
pant.factor
```

```
## [1] XL  L   XL  XXL L   XL
## Levels: L < XL < XXL
```

```r
pant.factor[1] > pant.factor[2]
```

```
## [1] TRUE
```

Accessing elements of a Factor

```r
# Create Factor
gender.factor <- factor(c("Male","Female","Female","Male","Female"))
gender.factor
```

```
## [1] Male   Female Female Male   Female
## Levels: Female Male
```

```r
# Access the 3rd element of gender.factor
gender.factor[3]
```

```
## [1] Female
## Levels: Female Male
```

```r
# Access the 2nd and 4th elements of gender.factor
gender.factor[c(2,4)]
```

```
## [1] Female Male
## Levels: Female Male
```

```r
# Access all the elements except the 3rd element of gender.factor
gender.factor[-3]
```

```
## [1] Male   Female Male   Female
## Levels: Female Male
```

```r
# Access elements using Logical Vector
gender.factor[c(TRUE, FALSE, FALSE, TRUE,TRUE)]
```

```
## [1] Male   Male   Female
## Levels: Female Male
```

Modify elements of a Factor

```r
# Print Factor using Predefined Levels
gender.factor <- factor(c("Male","Female","Female","Male","Female"), levels = c("Male","Female", "Rathe
gender.factor
```

```
## [1] Male   Female Female Male   Female
## Levels: Male Female Rather-Not-Say
```

```r
# Modify 2nd Element
gender.factor[2] <- "Rather-Not-Say"
gender.factor
```

```
## [1] Male           Rather-Not-Say Female         Male           Female
## Levels: Male Female Rather-Not-Say
```

```r
# Modify 2nd Element
gender.factor[c(3,4)] <- c("Male","Female")
gender.factor
```

```
## [1] Male           Rather-Not-Say Male           Female         Female
## Levels: Male Female Rather-Not-Say
```

```r
# Trying to Assign Value outside Levels
gender.factor[3] <- "Transgender"
```

```
## Warning in `[<-.factor`(`*tmp*`, 3, value = "Transgender"): invalid factor
## level, NA generated
```

```r
gender.factor
```

```
## [1] Male           Rather-Not-Say <NA>           Female         Female
## Levels: Male Female Rather-Not-Say
```

```
# Solving Problem of Value outside Levels
levels(gender.factor) <- c(levels(gender.factor), "Transgender")    # add new level
gender.factor
```

```
## [1] Male           Rather-Not-Say <NA>           Female         Female
## Levels: Male Female Rather-Not-Say Transgender
```

```
gender.factor[3] <- "Transgender"
gender.factor
```

```
## [1] Male           Rather-Not-Say Transgender    Female         Female
## Levels: Male Female Rather-Not-Say Transgender
```

Renaming a Factor levels

```
factor(gender.factor,levels = c("Male","Female","Transgender","Rather-Not-Say"),labels = c("Gen_Male","G
```

```
## [1] Gen_Male           Gen_Rather-Not-Say Gen_Transgender    Gen_Female
## [5] Gen_Female
## Levels: Gen_Male Gen_Female Gen_Transgender Gen_Rather-Not-Say
```

Concatinating 2 Factors

```
x1 <- factor(sample(1:2,10,replace=T))
x1
```

```
##  [1] 1 2 2 2 1 1 1 2 1 2
## Levels: 1 2
```

```
x2 <- factor(sample(2:3,10,replace=T))
x2
```

```
##  [1] 3 2 2 2 3 3 2 3 2 2
## Levels: 2 3
```

```
x3<-factor(c(as.character(x1),as.character(x2)))
x3
```

```
##  [1] 1 2 2 2 1 1 1 2 1 2 3 2 2 2 3 3 2 3 2 2
## Levels: 1 2 3
```

```
## Why to use as.character
x4<-factor(c(x1,x2))
x4
```

```
##  [1] 1 2 2 2 1 1 1 2 1 2 2 1 1 1 2 2 1 2 1 1
## Levels: 1 2
```

Generating Factor Level

Syntax:- gl(n,k,length=n*k,labels=seq_len(n),ordered=False)

- n -> An Integer giving the no. of levels.
- k -> An Integer giving the no. of replication.
- length -> An Integer giving Length of Result.
- labels -> An optional vector of Labels for the resulting factor levels.
- ordered -> A Logical indication whether Result should be ordered or not.

```
gl(2,3,10)
```

```
##  [1] 1 1 1 2 2 2 1 1 1 2
## Levels: 1 2
```