Factor in R: Categorical & Continuous Variables

What is Factor in R?

Factors are variables in R which take on a limited number of different values; such variables are often referred to as categorical variables.

In a dataset, we can distinguish two types of variables: **categorical** and **continuous**.

- In a categorical variable, the value is limited and usually based on a particular finite group. For example, a categorical variable can be countries, year, gender, occupation.
- A continuous variable, however, can take any values, from integer to decimal. For example, we can have the revenue, price of a share, etc..

Categorical Variables

R stores categorical variables into a factor. Let's check the code below to convert a character variable into a factor variable. Characters are not supported in machine learning algorithm, and the only way is to convert a string to an integer.

Syntax

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

Arguments:

- **x**: A vector of data. Need to be a string or integer, not decimal.
- **Levels**: A vector of possible values taken by x. This argument is optional. The default value is the unique list of items of the vector x.
- **Labels**: Add a label to the x data. For example, 1 can take the label `male` while 0, the label `female`.
- **ordered**: Determine if the levels should be ordered.

Example:

Let's create a factor data frame.

```
# Create gender vector
gender_vector <- c("Male", "Female", "Female", "Male", "Male")
class(gender_vector)
# Convert gender_vector to a factor
factor_gender_vector <-factor(gender_vector)
class(factor_gender_vector)
```

Output:

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

It is important to transform a **string** into factor when we perform Machine Learning task.

A categorical variable can be divided into **nominal categorical variable** and **ordinal categorical variable**.

Nominal Categorical Variable

A categorical variable has several values but the order does not matter. For instance, male or female categorical variable do not have ordering.

```
# Create a color vector
color_vector <- c('blue', 'red', 'green', 'white', 'black', 'yellow')
# Convert the vector to factor
factor_color <- factor(color_vector)
factor_color
```

Output:

```
## [1] blue   red    green  white  black  yellow
## Levels: black blue green red white yellow
```

From the factor_color, we can't tell any order.

Ordinal Categorical Variable

Ordinal categorical variables do have a natural ordering. We can specify the order, from the lowest to the highest with order = TRUE and highest to lowest with order = FALSE.

Example:

We can use summary to count the values for each factor.

```
# Create Ordinal categorical vector
day_vector <- c('evening', 'morning', 'afternoon', 'midday', 'midnight',
'evening')
# Convert `day_vector` to a factor with ordered level
factor_day <- factor(day_vector, order = TRUE, levels =c('morning', 'midda
y', 'afternoon', 'evening', 'midnight'))
# Print the new variable
factor_day
```

Output:

```
## [1] evening   morning   afternoon midday
midnight  evening
```

Example:

```
## Levels: morning < midday < afternoon < evening < midnight
# Append the line to above code
# Count the number of occurence of each level
summary(factor_day)
```

Output:

```
##   morning    midday afternoon   evening  midnight
##         1         1         1         2         1
```

R ordered the level from 'morning' to 'midnight' as specified in the levels parenthesis.

Continuous Variables

Continuous class variables are the default value in R. They are stored as numeric or integer. We can see it from the dataset below. mtcars is a built-in dataset. It gathers information on different types of car. We can import it by using mtcars and check the class of the variable mpg, mile per gallon. It returns a numeric value, indicating a continuous variable.

```
dataset <- mtcars
class(dataset$mpg)
```

Output

```
## [1] "numeric"
```

Prev (/r-matrix-tutorial.html)

Report a Bug

Next (/r-data-frames.html)

YOU MIGHT LIKE:

BLOG

(/best-udemy-

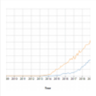
course.html) (/best-udemy-course.html)

70 BEST Udemy Courses in 2020

(/best-udemy-course.html)

DEVOPS

(/kubernetes-vs-



docker.html) (/kubernetes-vs-docker.html)

Kubernetes vs Docker: Must Know Differences!

(/kubernetes-vs-docker.html)

R PROGRAMMING

(/r-dplyr-tutorial.html) (/r-

dplyr-tutorial.html)

R Dplyr Tutorial: Data Manipulation(Join) & Cleaning(Spread)

(/r-dplyr-tutorial.html)

BLOG

(/imacros-

alternative.html) (/imacros-alternative.html)

10 Best iMacros Alternatives in 2020

(/imacros-alternative.html)

BLOG

(/ibm-rational-

clearquest.html) (/ibm-rational-clearquest.html)

IBM Rational ClearQuest Tool Tutorial

(/ibm-rational-clearquest.html)

BLOG

(/ivr-tools.html) (/ivr-tools.html)

17 BEST IVR Tools in 2020

(/ivr-tools.html)

R Programming

6) R Data Frames (/r-data-frames.html)

7) Lists in R (/r-lists-create-select.html)

8) R Sort a Data Frame (/r-sort-data-frame.html)

9) R Dplyr Tutorial (/r-dplyr-tutorial.html)

10) Merge Data Frames in R (/r-merge-data-frames.html)

11) Functions R Programming (/r-functions-programming.html)

(https://www.facebook.com/guru99com/)
(https://twitter.com/guru99com)
(https://www.linkedin.com/company/guru99/)
(https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ)
(https://forms.aweber.com/form/46/724807646.htm)

About

About Us (/about-us.html)
Advertise with Us (/advertise-us.html)
Write For Us (/become-an-instructor.html)
Contact Us (/contact-us.html)

Career Suggestion

SAP Career Suggestion Tool (/best-sap-module.html)
Software Testing as a Career (/software-testing-career-complete-guide.html)

Interesting

eBook (/ebook-pdf.html)
Blog (/blog/)
Quiz (/tests.html)
SAP eBook (/sap-ebook-pdf.html)

Execute online