How can I use the “split column into multiple columns” function in R, and what are some examples of its usage?

Authored by

stats writer

May 12, 2024

The "split column into multiple columns" function in R allows users to split a single column in a data frame into multiple columns based on a chosen delimiter. This function is useful for organizing and analyzing data that may be stored in a single column but contains multiple values or categories. For example, if a data frame contains a column with full names (i.e. first name and last name) separated by a space, the split function can be used to divide this column into two separate columns for first name and last name. Other examples of its usage include splitting strings based on specific characters or separating data into different categories. This function is a powerful tool for data manipulation and can greatly improve the efficiency of data analysis in R.

Split Column Into Multiple Columns in R (With Examples)

You can use one of the following two methods to split one column into multiple columns in R:

Method 1: Use str_split_fixed()

library(stringr)

df <- str_split_fixed(df$original_column, 'sep', 2)

Method 2: Use separate()

library(dplyr)
library(tidyr)

df %>% separate(original_column, c('col1', 'col2'))

The following examples show how to use each method

in practice.

Method 1: Use str_split_fixed()

Suppose we have the following data frame:

```
#create data frame
df <- data.frame(player=c('John_Wall', 'Dirk_Nowitzki', 'Steve_Nash'),
points=c(22, 29, 18),
assists=c(8, 4, 15))

#view data frame
df

player points assists
1 John_Wall 22 8
2 Dirk_Nowitzki 29 4
3 Steve_Nash 18 15
```

We can use the str_split_fixed() function from the stringr package to separate the 'player' column into two new columns called 'First' and 'Last' as follows:

```
library(stringr)
```

```
#split 'player' column using '_' as the separator
df <- str_split_fixed(df$player, '_', 2)

#view updated data frame
df

player points assists First Last
1 John_Wall 22 8 John Wall
2 Dirk_Nowitzki 29 4 Dirk Nowitzki
3 Steve_Nash 18 15 Steve Nash
```

Notice that two new columns are added at the end of the data frame.

Feel free to rearrange the columns and drop the original 'player' columns if you'd like:

```
#rearrange columns and leave out original 'player' column
df_final <- df

#view updated data frame
df_final

First Last points assists
1 John Wall 22 8
```

2 Dirk Nowitzki 29 4

3 Steve Nash 18 15

Method 2: Use separate()

The following code shows how to use the separate() function from the tidyr package to separate the 'player' column into 'first' and 'last' columns:

```
library(dplyr)library(tidyr)

#create data frame
df <- data.frame(player=c('John_Wall', 'Dirk_Nowitzki', 'Steve_Nash'),
points=c(22, 29, 18),
assists=c(8, 4, 15))

#separate 'player' column into 'First' and 'Last'
df %>% separate(player, c('First', 'Last'))
```

First Last points assists

1 John Wall 22 8

2 Dirk Nowitzki 29 4

3 Steve Nash 18 15

For example, if the first and last names were separated

by a comma, the separate() function would automatically split based on the location of the comma:

```
library(dplyr)
library(tidyr)

#create data frame
df <- data.frame(player=c('John,Wall', 'Dirk,Nowitzki', 'Steve,Nash'),
points=c(22, 29, 18),
assists=c(8, 4, 15))

#separate 'player' column into 'First' and 'Last'
df %>% separate(player, c('First', 'Last'))

First Last points assists
1 John Wall 22 8
2 Dirk Nowitzki 29 4
3 Steve Nash 18 15
```

You can find the complete online documentation for the separate() function .