How can I utilize the across() function in dplyr to manipulate multiple columns at once?

Authored by

stats writer

June 27, 2024

The across() function in dplyr allows for manipulation of multiple columns at once in a data frame. This function can be utilized by providing a set of columns or a range of columns to perform the desired operation on. It is particularly useful for performing repetitive tasks, such as applying the same function to multiple columns, or creating new columns based on existing ones. By using the across() function, data manipulation tasks can be streamlined and more efficient.

Use the across() Function in dplyr (3 Examples)

You can use the function from the package in R to apply a transformation to multiple columns.

There are countless ways to use this function, but the following methods illustrate some common uses:

Method 1: Apply Function to Multiple Columns

```
#multiply values in col1 and col2 by 2
df %>%
mutate(across(c(col1, col2), function(x) x*2))
```

Method 2: Calculate One Summary Statistic for Multiple Columns

```
#calculate mean of col1 and col2
df %>%
summarise(across(c(col1, col2), mean, na.rm=TRUE))
```

Method 3: Calculate Multiple Summary Statistics for Multiple Columns

```
#calculate mean and standard deviation for col1 and col2
df %>%
summarise(across(c(col1, col2), list(mean=mean, sd=sd), na.rm=TRUE))
```

The following examples show how to each method with the following data frame:

```
#create data frame
df <- data.frame(conf=c('East', 'East', 'East', 'West', 'West', 'West'),
points=c(22, 25, 29, 13, 22, 30),
rebounds=c(12, 10, 6, 6, 8, 11))

#view data frame
df

conf points rebounds
1 East 22 12
2 East 25 10
3 East 29 6
```

4 West 13 6
5 West 22 8
6 West 30 11

Example 1: Apply Function to Multiple Columns

The following code shows how to use the across() function to multiply the values in both the points and rebounds columns by 2:

```
library(dplyr)

#multiply values in points and rebounds columns by 2
df %>%
mutate(across(c(points, rebounds), function(x) x*2))
```

```
conf points rebounds
1 East 44 24
2 East 50 20
3 East 58 12
4 West 26 12
5 West 44 16
6 West 60 22
```

Example 2: Calculate One Summary Statistic for Multiple Columns

The following code shows how to use the across() function to calculate the mean value for both the points and rebounds columns:

```
library(dplyr)

#calculate mean value of points an rebounds columns
df %>%
summarise(across(c(points, rebounds), mean, na.rm=TRUE))

points rebounds
1 23.5 8.833333
```

Note that we can also use the is.numeric function to automatically calculate a summary statistic for all of the numeric columns in the data frame:

```
library(dplyr)

#calculate mean value for every numeric column in data frame
df %>%
summarise(across(where(is.numeric), mean, na.rm=TRUE))
```

points rebounds
1 23.5 8.833333

Example 3: Calculate Multiple Summary Statistics for Multiple Columns

The following code shows how to use the across() function to calculate the mean and standard deviation of both the points and rebounds columns:

```
library(dplyr)

#calculate mean and standard deviation for points and rebounds columns
df %>%
summarise(across(c(points, rebounds), list(mean=mean, sd=sd), na.rm=TRUE))
```

```
points_mean points_sd rebounds_mean rebounds_sd
1 23.5 6.156298 8.833333 2.562551
```

Note: You can find the complete documentation for the across() function .

Additional Resources

The following tutorials explain how to perform other common functions using dplyr: