How can I calculate lagged values in R and provide examples?

Authored by

stats writer

June 25, 2024

Lagged values refer to the previous values of a variable in a time series. In order to calculate lagged values in R, the "lag()" function can be used. This function shifts the values of a vector by a specified number of time periods. For example, if we have a vector called "sales" with values for 12 months, using the "lag(sales, 1)" function will shift the values by one month, making the first value of the vector equal to the second value of the original vector and so on. This can be useful for analyzing trends and patterns in time series data. Other functions such as "diff()" and "tslag()" can also be used to calculate lagged values. An example of calculating lagged values in R would be using the "lag()" function on a stock price dataset to analyze the relationship between current and previous stock prices.

Calculate Lagged Values in R (With Examples)

You can use the lag() function from the package in R to calculated lagged values.

This function uses the following basic syntax:

lag(x, n=1, ...)

where:

x: vector of valuesn: number of positions to lag by

The following example shows how to use this function to calculated lagged values in practice.

Example: Calculating Lagged Values in R

Suppose we have the following data frame in R that shows the number of sales made by some store on 10

consecutive days:

```
#create data frame
df <- data.frame(day=1:10,
sales=c(18, 10, 14, 13, 19, 24, 25, 29, 15, 18))

#view data frame
df
```

day sales
1 1 18
2 2 10
3 3 14
4 4 13
5 5 19
6 6 24
7 7 25
8 8 29
9 9 15
10 10 18

We can use the lag() function from the dplyr package to create a lag column that displays the sales for the previous day for each row:

```
library(dplyr)

#add new column that shows sales for previous day
df$previous_day_sales <- dplyr::lag(df$sales, n=1)

#view updated data frame
df
```

day sales previous_day_sales
1 1 18 NA
2 2 10 18
3 3 14 10
4 4 13 14
5 5 19 13
6 6 24 19
7 7 25 24
8 8 29 25
9 9 15 29
10 10 18 15

Here's how to interpret the output:

The first value in the lag column is NA since there is no prior value in the sales column.The second value in the lag column is 18 since this is the prior value in the sales

column.The third value in the lag column is 10 since this is the prior value in the sales column.

And so on.

We can also modify the value for the n argument in the lag() function to calculate a lagged value for a different number of previous positions:

```
library(dplyr)

#add new column that shows sales for two days prior
df$previous_day_sales <- dplyr::lag(df$sales, n=2)

#view updated data frame
df
```

day sales previous_day_sales
1 1 18 NA
2 2 10 NA
3 3 14 18
4 4 13 10
5 5 19 14
6 6 24 13
7 7 25 19
8 8 29 24

9 9 15 25

10 10 18 29

Note: To create a lead column, use the lead() function from the dplyr package instead of the lag() function.