

How can I prevent ifelse() from converting a date to numeric when using the R programming language?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I prevent ifelse() from converting a date to numeric when using the R programming language?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153191>

The `ifelse()` function in the R programming language has the capability to convert a date variable into a numeric value, which can cause unexpected results in data analysis. To prevent this, one can use the `as.Date()` function to explicitly convert the date variable back to its original format before using the `ifelse()` function. This will ensure that the date remains in its original format and prevent any unwanted changes. Additionally, using the `Date` class and specifying the format of the date can also help prevent the conversion of dates to numeric values. By implementing these techniques, one can effectively prevent the `ifelse()` function from converting dates to numeric, ensuring accurate and reliable data analysis.

R: Prevent ifelse() from Converting Date to Numeric

The `ifelse()` function in base R converts date objects to numeric objects by default.

To prevent this from happening, you can use one of the following methods as an alternative:

Method 1: Use `as.character()` in Base R

```
df$date <- as.Date(ifelse(df$date < '2022-01-20',  
as.character(df$date+5),  
as.character(df$date)))
```

Method 2: Use `if_else()` in dplyr

```
df$date <- if_else(df$date < '2022-01-20', df$date+5,  
df$date)
```

Method 3: Use fifelse() in data.table

```
df$date <- fifelse(df$date < '2022-01-20', df$date+5,  
df$date)
```

The following examples show how to use each method in practice with the following data frame in R:

```
#create data frame
```

```
df <- data.frame(date=as.Date(c('2022-01-05',  
'2022-01-17', '2022-01-22',  
'2022-01-23', '2022-01-29', '2022-02-13'))),  
sales=c(22, 35, 24, 20, 16, 19))
```

```
#view data frame
```

```
df
```

```
date sales
```

```
1 2022-01-05 22
```

```
2 2022-01-17 35
```

```
3 2022-01-22 24
```

```
4 2022-01-23 20
```

```
5 2022-01-29 16
```

```
6 2022-02-13 19
```

Example 1: Use as.character() in Base R

The following code shows how to use the as.character() function with the ifelse() function from base R to perform an if else statement on the date column in the data frame and produce a date as a result:

```
#if date is before 2022-01-20 then add 5 days  
df$date <- as.Date(ifelse(df$date < '2022-01-20',  
as.character(df$date+5),  
as.character(df$date)))
```

```
#view updated data frame  
df
```

```
date sales
```

```
1 2022-01-10 22
```

```
2 2022-01-22 35
```

```
3 2022-01-22 24
```

```
4 2022-01-23 20
```

```
5 2022-01-29 16
```

```
6 2022-02-13 19
```

If any value in the date column was before 2022-01-20, we added five days to the date.

Notice that the date column has retained its date format instead of being converted to a numeric format.

Example 2: Use if_else() in dplyr

The following code shows how to use the if_else() function from dplyr to perform an if else statement on the date column in the data frame and produce a date as a result:

```
library(dplyr)
```

```
#if date is before 2022-01-20 then add 5 days
```

```
df$date <- if_else(df$date < '2022-01-20', df$date+5,  
df$date)
```

```
#view updated data frame
```

```
df
```

```
date sales
```

```
1 2022-01-10 22
```

```
2 2022-01-22 35
```

```
3 2022-01-22 24
```

```
4 2022-01-23 20
```

```
5 2022-01-29 16
```

```
6 2022-02-13 19
```

If any value in the date column was before 2022-01-20, we added five days to the date.

Notice that the date column has retained its date format instead of being converted to a numeric format.

Example 3: Use fifelse() in data.table

The following code shows how to use the fifelse() function from data.table to perform an if else statement on the date column in the data frame and produce a date as a result:

```
library(data.table)
```

```
#if date is before 2022-01-20 then add 5 days
```

```
df$date <- fifelse(df$date < '2022-01-20', df$date+5,  
df$date)
```

```
#view updated data frame
```

```
df
```

```
date sales
```

```
1 2022-01-10 22
```

```
2 2022-01-22 35
```

```
3 2022-01-22 24
```

4 2022-01-23 20

5 2022-01-29 16

6 2022-02-13 19

If any value in the date column was before 2022-01-20, we added five days to the date.

Once again, the date column has retained its date format instead of being converted to a numeric format.

Note: For extremely large data frames, the dplyr and data.table methods will be faster than the base R method.