

Can NA be included in an ifelse statement in R?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *Can NA be included in an ifelse statement in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153308>

The "ifelse" statement in R allows for conditional execution of code, where a specified action is taken if a certain condition is met. It is used in scenarios where there are two possible outcomes based on a given condition. However, unlike other programming languages, R also allows for the inclusion of "NA" (not available) values in the "ifelse" statement. This means that the code will still execute even if the condition is not fully met or if there are missing values. This feature makes R more flexible and robust for handling data with missing values.

Include NA in ifelse Statement in R

Often you may want to use an ifelse statement in R to create a new column in a data frame whose values are based on the values in an existing column.

To do so, you can use the following basic syntax:

```
df$new_column<- ifelse(df$col1=='A', 'val_if_true',  
'val_if_false')
```

However, if NA values are present in a column then the values in the new column will automatically be NA.

To avoid this, you can use the !is.na() function as follows:

```
df$new_column<- ifelse(df$col1=='A' &!is.na(df$col1),  
'val_if_true', 'val_if_false')
```

The following example shows how to use this syntax in

practice.

Example: Include NA in ifelse Statement in R

Suppose we have the following data frame in R that contains information about various basketball players:

```
#create data frame
```

```
df <- data.frame(player=c('A', 'B', 'C', 'D', 'E', 'F'),  
conf=c('West', NA, 'West', 'East', 'East', 'East'),  
points=c(30, 35, 11, 18, 14, NA))
```

```
#view data frame
```

```
df
```

```
player conf points
```

```
1 A West 30
```

```
2 B <NA> 35
```

```
3 C West 11
```

```
4 D East 18
```

```
5 E East 14
```

```
6 F East NA
```

Now suppose we attempt to create a new column called class that takes on the following values:

'West_Player' if conf is equal to 'West' 'Other' if conf is not equal to 'West'

The following code shows how to do so:

```
#create new column called 'class'  
df$class <- ifelse(df$conf=='West', 'West_Player',  
'Other')
```

```
#view updated data frame
```

```
df
```

```
player conf points class  
1 A West 30 West_Player  
2 B <NA> 35 <NA>  
3 C West 11 West_Player  
4 D East 18 Other  
5 E East 14 Other  
6 F East NA Other
```

Notice that the value for class in row 2 is equal to NA since the corresponding value in the conf column was equal to NA.

To avoid this, we can use the !is.na() function as

follows:

```
#create new column called 'class'
```

```
df$class <- ifelse(df$conf=='West' & !is.na(df$conf),  
'West_Player', 'Other')
```

```
#view updated data frame
```

```
df
```

```
player conf points class
```

```
1 A West 30 West_Player
```

```
2 B <NA> 35 Other
```

```
3 C West 11 West_Player
```

```
4 D East 18 Other
```

```
5 E East 14 Other
```

```
6 F East NA Other
```

By using the `!is.na()` function, we specified that the value in the `conf` column must be equal to 'West' *and* not equal to NA in order for the value in the new class column to be 'West_Player'.