

How can I compare three columns in R using an example?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I compare three columns in R using an example?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151928>

To compare three columns in R, you can use the "ifelse" function to create a new column that displays the results of the comparison. For example, if you have three columns named "A", "B", and "C", you can use the following code to create a new column "Comparison" that displays "TRUE" if the values in column "A" are greater than both "B" and "C", otherwise it displays "FALSE":

```
df$Comparison = ifelse(df$B & df$A > df$C, TRUE, FALSE)
```

This function allows for easy comparison of multiple columns in a dataset, providing a clear and concise way to analyze data and identify patterns.

Compare Three Columns in R (With Example)

You can use the following basic syntax to compare the values in three columns in R:

```
df$all_matching <- df$A == df$B & df$B == df$C
```

This syntax creates a new column called `all_matching` that returns a value of `TRUE` if all of the columns have matching values, otherwise it returns `FALSE`.

The following example shows how to use this syntax in practice.

Example: Compare Three Columns in R

Suppose we have the following data frame in R with three columns:

```
#create data frame
```

```
df <- data.frame(A=c(4, 0, 3, 3, 6, 8, 7, 9, 12),  
B=c(4, 2, 3, 5, 6, 4, 7, 7, 12),  
C=c(4, 0, 3, 5, 5, 10, 7, 9, 12))
```

```
#view data frame
```

```
df
```

```
A B C  
1 4 4 4  
2 0 2 0  
3 3 3 3  
4 3 5 5  
5 6 6 5  
6 8 4 10  
7 7 7 7  
8 9 7 9  
9 12 12 12
```

We can use the following code to create a new column called `all_matching` that returns `TRUE` if all three columns match in a given row and `FALSE` if they do not:

```
#create new column that checks if values in all three
```

columns match

```
df$all_matching <- df$A == df$B & df$B == df$C
```

```
#view updated data frame
```

```
df
```

```
A B C all_matching
```

```
1 4 4 4 TRUE
```

```
2 0 2 0 FALSE
```

```
3 3 3 3 TRUE
```

```
4 3 5 5 FALSE
```

```
5 6 6 5 FALSE
```

```
6 8 4 10 FALSE
```

```
7 7 7 7 TRUE
```

```
8 9 7 9 FALSE
```

```
9 12 12 12 TRUE
```

The new column called `all_matching` shows whether or not the values in all three columns match in a given row.

For example:

All three values match in the first row, so `TRUE` is returned. Not every value matches in the second row, so

FALSE is returned. All three values match in the third row, so **TRUE** is returned.

And so on.

If you would like to return values other than **TRUE** and **FALSE**, you can specify those values in an **ifelse()** function.

For example, we can use the following code to return 'Yes' if the values in all three columns match or 'No' otherwise:

```
#create new column that checks if values in all three  
columns match  
df$all_matching <- ifelse(df$A == df$B & df$B == df$C,  
'Yes', 'No')
```

```
#view updated data frame
```

```
df
```

```
A B C all_matching
```

```
1 4 4 4 Yes
```

```
2 0 2 0 No
```

```
3 3 3 3 Yes
```

```
4 3 5 5 No
```

5 6 6 5 No

6 8 4 10 No

7 7 7 7 Yes

8 9 7 9 No

9 12 12 12 Yes

The new column now returns 'Yes' or 'No' instead of TRUE or FALSE.

ARABPSYCHOLOGY.COM