

How can Group Data by Week in R? Can you provide an example?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can Group Data by Week in R? Can you provide an example?*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155790>

Grouping data by week in R is a process of organizing a dataset into weekly segments for better analysis and visualization. This can be achieved by using the "group_by()" function from the dplyr package in R. By specifying the week as the grouping variable, the data can be grouped into weekly categories, making it easier to compare and analyze trends within a specific time frame. For example, if we have a dataset of sales data with daily timestamps, we can group the data by week to see the overall sales trend for each week. This can be done by using the "group_by(week)" function followed by the "summarize()" function to calculate the total sales for each week. This will result in a new dataset with weekly data points, allowing for further analysis and visualization of the data.

Group Data by Week in R (With Example)

You can use the `strftime()` function in base R with the "%V" argument to group data by week in R.

This function uses the following basic syntax:

```
df$week_num <- strftime(df$date, format = "%V")
```

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

Example: Group Data by Week in R

Suppose we have the following data frame in R that shows the total sales of some item on various dates:

```
#create data frame
```

```
df <- data.frame(date=as.Date(c('1/8/2022', '1/9/2022'),
```

```
'2/10/2022', '2/15/2022',  
'3/5/2022', '3/22/2022', '3/27/2022'), '%m/%d/%Y'),  
sales=c(8, 14, 22, 23, 16, 17, 23))
```

```
#view data frame
```

```
df
```

```
date sales
```

```
1 2022-01-08 8
```

```
2 2022-01-09 14
```

```
3 2022-02-10 22
```

```
4 2022-02-15 23
```

```
5 2022-03-05 16
```

```
6 2022-03-22 17
```

```
7 2022-03-27 23
```

We can use the following code to add a column that shows the week number of each date:

```
#add column to show week number
```

```
df$week_num <- strftime(df$date, format = "%V")
```

```
#view updated data frame
```

```
df
```

```
date sales week_num
```

```
1 2022-01-08 8 01
```

```
2 2022-01-09 14 01
```

```
3 2022-02-10 22 06
```

```
4 2022-02-15 23 07
```

```
5 2022-03-05 16 09
```

```
6 2022-03-22 17 12
```

```
7 2022-03-27 23 12
```

Note: From the documentation, here is how `%V%` calculates date numbers: *"the week number of the year (Monday as the first day of the week) as a decimal number . If the week containing 1 January has four or more days in the new year, then it is considered week 1. Otherwise, it is the last week of the previous year, and the next week is week 1."*

Once we've created this new column, we can aggregate values based on week number.

For example, we can use the following code to calculate the sum of sales, grouped by week:

```
library(dplyr)
```

```
#calculate sum of sales, grouped by week
df %>%
  group_by(week_num) %>%
  summarize(total_sales = sum(sales))
```

```
# A tibble: 6 x 2
```

```
week_num total_sales
```

```
1 01 22
```

```
2 06 22
```

```
3 07 23
```

```
4 09 16
```

```
5 12 40
```

From the output we can see:

The sum of sales during week 1 was 22. The sum of sales during week 6 was 22. The sum of sales during week 7 was 23.

And so on.

We can also use another metric to aggregate the data.

```
library(dplyr)
```

```
#calculate mean of sales, grouped by week
df %>%
  group_by(week_num) %>%
  summarize(mean_sales = mean(sales))
```

```
# A tibble: 5 x 2
```

```
week_num mean_sales
```

```
1 01 11
```

```
2 06 22
```

```
3 07 23
```

```
4 09 16
```

```
5 12 20
```

From the output we can see:

The mean of sales during week 1 was 11. The mean of sales during week 6 was 22. The mean of sales during week 7 was 23.

And so on.

Additional Resources

The following tutorials explain how to perform other common tasks in R: