

How can I group data by hour in R, with an example?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group data by hour in R, with an example?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152417>

In R, the "group_by" function from the dplyr package can be used to group data by hour. This function allows for easy manipulation and analysis of data based on specific time intervals. For example, if you have a dataset containing hourly sales data for a store, you can use the "group_by" function to group the data by hour and then calculate the total sales for each hour. This will provide valuable insights into the store's peak sales hours and can help with scheduling and inventory management. The syntax for grouping by hour in R would be: "group_by(data, hour = hour(datetime))" where "data" is the name of your dataset and "datetime" is the column containing the date and time information.

Group Data by Hour in R (With Example)

You can use the following syntax to group data by hour and perform some aggregation in R:

```
library(dplyr)
```

```
library(lubridate)
```

```
#group by hours in time column and calculate sum of sales
```

```
df %>%
```

```
group_by(time=floor_date(time, '1 hour')) %>%
```

```
summarize(sum_sales=sum(sales))
```

This particular example groups the values by hour in a column called time and then calculates the sum of values in the sales column for each hour.

The following example shows how to use this syntax in

practice.

Example: Group Data by Hour in R

Suppose we have the following data frame that shows the number of sales made at various times throughout the day for some store:

```
#create data frame
```

```
df <- data.frame(time=as.POSIXct(c('2022-01-01  
01:14:00', '2022-01-01 01:24:15',  
'2022-01-01 02:52:19', '2022-01-01 02:54:00',  
'2022-01-01 04:05:10', '2022-01-01 05:35:09'))),  
sales=c(18, 20, 15, 14, 10, 9))
```

```
#view data frame
```

```
df
```

```
time sales
```

```
1 2022-01-01 01:14:00 18  
2 2022-01-01 01:24:15 20  
3 2022-01-01 02:52:19 15  
4 2022-01-01 02:54:00 14  
5 2022-01-01 04:05:10 10  
6 2022-01-01 05:35:09 9
```

We can use the following syntax to group the time column by hours and calculate the sum of sales for each hour:

```
library(dplyr)
```

```
library(lubridate)
```

```
#group by hours in time column and calculate sum of sales
```

```
df %>%
```

```
group_by(time=floor_date(time, '1 hour')) %>%
```

```
summarize(sum_sales=sum(sales))
```

```
`summarise()` ungrouping output (override with  
`.groups` argument)
```

```
# A tibble: 4 x 2
```

```
time sum_sales
```

```
1 2022-01-01 01:00:00 38
```

```
2 2022-01-01 02:00:00 29
```

```
3 2022-01-01 04:00:00 10
```

```
4 2022-01-01 05:00:00 9
```

From the output we can see:

A total of 38 sales were made during the first hour. A total of 29 sales were made during the second hour. A total of 10 sales were made during the fourth hour. A total of 9 sales were made during the fifth hour.

Note that we can also perform some other aggregation.

For example, we could calculate the mean number of sales per hour:

```
library(dplyr)
```

```
library(lubridate)
```

```
#group by hours in time column and calculate mean of sales
```

```
df %>%
```

```
group_by(time=floor_date(time, '1 hour')) %>%
```

```
summarize(mean_sales=mean(sales))
```

```
`summarise()` ungrouping output (override with  
`.groups` argument)
```

```
# A tibble: 4 x 2
```

```
time mean_sales
```

```
1 2022-01-01 01:00:00 19
```

```
2 2022-01-01 02:00:00 14.5
```

```
3 2022-01-01 04:00:00 10
```

```
4 2022-01-01 05:00:00 9
```

From the output we can see:

The mean sales made in the first hour were 19. The mean sales made in the second hour were 14.5. The mean sales made in the fourth hour were 10. The mean sales made in the fifth hour were 9.

Feel free to group your own data frame by hour and calculate any specific metric you'd like by modifying the metric in the summarize() function.