

# How can I aggregate daily data to monthly and yearly in R?

Authored by  
**stats writer**

April 20, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I aggregate daily data to monthly and yearly in R?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137516>

Aggregating daily data to monthly and yearly in R refers to the process of combining daily data points into larger time intervals such as monthly or yearly. This is often done to simplify and summarize the data for easier analysis and interpretation. In R, this can be achieved using various functions and packages such as the "aggregate" function and the "xts" package. By grouping the daily data and applying a chosen aggregation function, the resulting output will be a single data point for each specified time interval, such as the average or sum of the daily data. This allows for a more comprehensive understanding of the data over longer time periods and can be useful in identifying trends and patterns.

## Aggregate Daily Data to Monthly and Yearly in R

Occasionally you may want to aggregate daily data to weekly, monthly, or yearly data in R.

This tutorial explains how to easily do so using the lubridate and dplyr packages.

**Example: Aggregate Daily Data in R**

Suppose we have the following data frame in R that shows the daily sales of some item over the course of 100 consecutive days:

```
#make this example reproducible  
set.seed(1)
```

```
#create data frame
```

```
df <- data.frame(date = as.Date("2020-12-01") + 0:99,  
sales = runif(100, 20, 50))
```

```
#view first six rows
```

```
head(df)
```

```
date sales
```

```
1 2020-12-01 27.96526
```

```
2 2020-12-02 31.16372
```

```
3 2020-12-03 37.18560
```

```
4 2020-12-04 47.24623
```

```
5 2020-12-05 26.05046
```

```
6 2020-12-06 46.95169
```

To aggregate this data, we can use the `floor_date()` function from the `lubridate` package which uses the following syntax:

```
floor_date(x, unit)
```

where:

**x:** A vector of date objects.  
**unit:** A time unit to round to. Options include `second`, `minute`, `hour`, `day`, `week`, `month`, `bimonth`, `quarter`, `halfyear`, and `year`.

The following code snippets show how to use this function along with the `group_by()` and `summarize()`

functions from the dplyr package to find the mean sales by week, month, and year:

## Mean Sales by Week

```
library(lubridate)library(dplyr)
```

```
#round dates down to week
```

```
df$week <- floor_date(df$date, "week")
```

```
#find mean sales by week
```

```
df %>%
```

```
group_by(week) %>%
```

```
summarize(mean = mean(sales))# A tibble: 15 x 2
```

```
week mean
```

```
1 2020-11-29 33.9
```

```
2 2020-12-06 35.3
```

```
3 2020-12-13 39.0
```

```
4 2020-12-20 34.4
```

```
5 2020-12-27 33.6
```

```
6 2021-01-03 35.9
```

```
7 2021-01-10 37.8
```

```
8 2021-01-17 36.8
```

```
9 2021-01-24 32.8
```

```
10 2021-01-31 33.9
11 2021-02-07 34.1
12 2021-02-14 41.6
13 2021-02-21 31.8
14 2021-02-28 35.2
15 2021-03-07 37.1
```

## Mean Sales by Month

```
library(lubridate)library(dplyr)
```

```
#round dates down to week
df$month <- floor_date(df$date, "month")
```

```
#find mean sales by month
```

```
df %>%group_by(month) %>%summarize(mean
=mean(sales))# A tibble: 4 x 2
month mean
```

```
1 2020-12-01 35.3
2 2021-01-01 35.6
3 2021-02-01 35.2
4 2021-03-01 37.0
```

## Mean Sales by Year

```
library(lubridate)library(dplyr)
```

```
#round dates down to week
```

```
df$year <- floor_date(df$date, "year")
```

```
#find mean sales by month
```

```
df %>%group_by(year) %>%summarize(mean =  
mean(sales))
```

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

```
year mean
```

```
1 2020-01-01 35.3
```

```
2 2021-01-01 35.7
```

Note that we chose to aggregate by the mean, but we could use any summary statistic we'd like such as the median, mode, max, min, etc.

**[How to Calculate the Mean by Group in R](#)**

**[How to Calculate Cumulative Sums in R](#)**

**[How to Plot a Time Series in R](#)**