

How to Easily Adjust Histogram Bin Numbers in R

Authored by
stats writer

December 4, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Adjust Histogram Bin Numbers in R*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=104927>

To change the number of bins in a histogram using R, the user can specify the argument "breaks" in the `hist()` function. This argument takes the form of a vector of bin boundaries for the histogram, and the user can specify the number of bins by setting the length of the vector of bin boundaries. The default number of bins is 30, but this can be changed by the user as per their requirements.

When you create a histogram in R, a formula known as is used to determine the optimal number of bins to use.

However, you can use the following syntax to override this formula and specify an exact number of bins to use in the histogram:

```
hist(data, breaks = seq(min(data), max(data), length.out = 7))
```

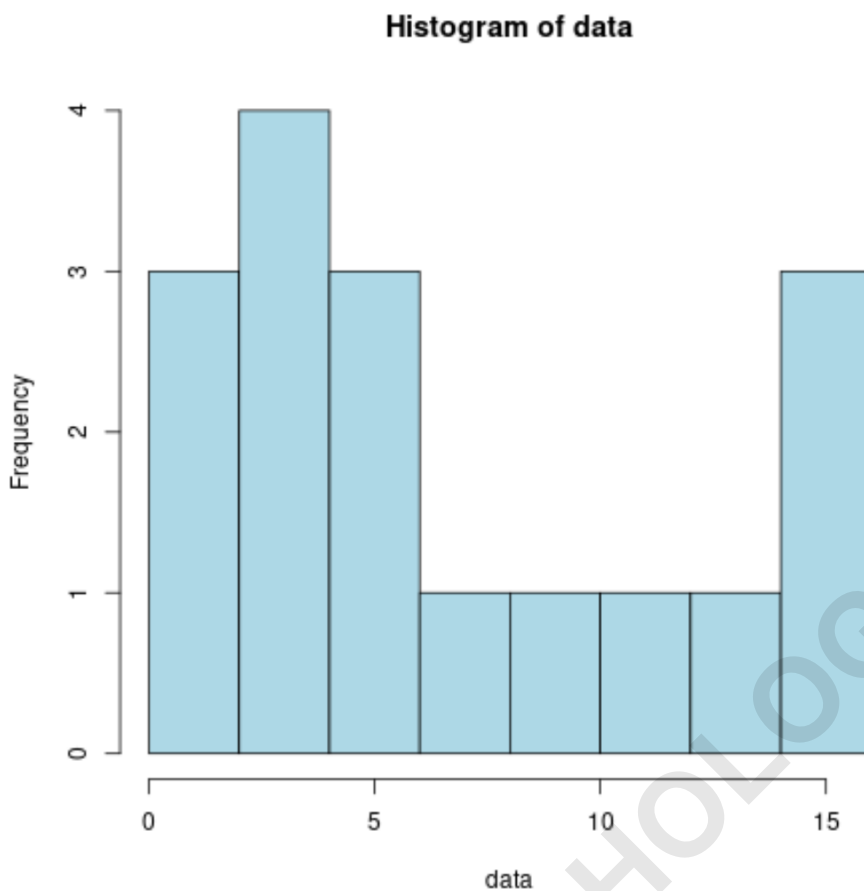
Note that the number of bins used in the histogram will be one less than the number specified in the **length.out** argument.

The following examples show how to use this syntax in practice.

Example 1: Create a Basic Histogram

The following code shows how to create a basic histogram in R without specifying the number of bins:

```
#define vector of data  
data <- c(1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 10, 11, 13, 16, 16, 16)  
  
#create histogram of data  
hist(data, col = 'lightblue')
```



Using Sturges' Rule, R decided to use **8** total bins in the histogram.

Example 2: Specify Number of Bins to Use in Histogram

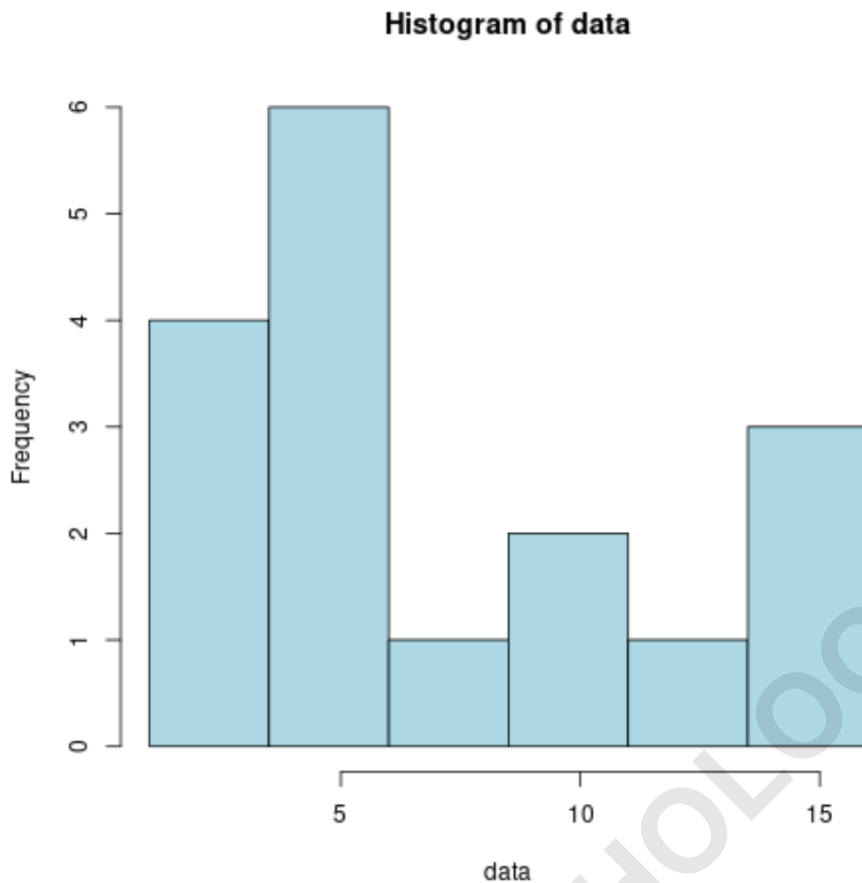
The following code shows how to create a histogram for the same vector of data and use exactly **6** bins:

```
#define vector of data
```

```
data <- c(1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 10, 11, 13, 16, 16, 16)
```

```
#create histogram with 6 bins
```

```
hist(data, col = 'lightblue', breaks = seq(min(data), max(data), length.out = 7))
```



Cautions on Choosing a Specific Number of Bins

The number of bins used in a histogram has a huge impact on how we interpret a dataset.

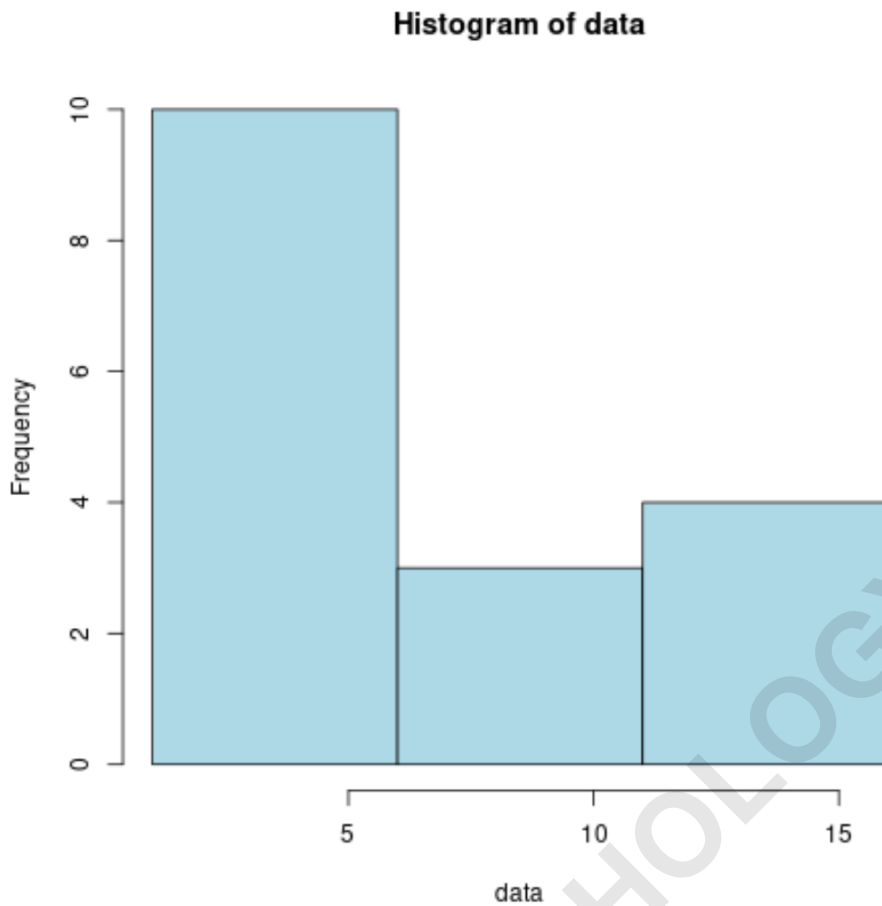
If we use too few bins, the true underlying pattern in the data can be hidden:

#define vector of data

```
data <- c(1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 10, 11, 13, 16, 16, 16)
```

#create histogram with 3 bins

```
hist(data, col = 'lightblue', breaks = seq(min(data), max(data), length.out = 4))
```



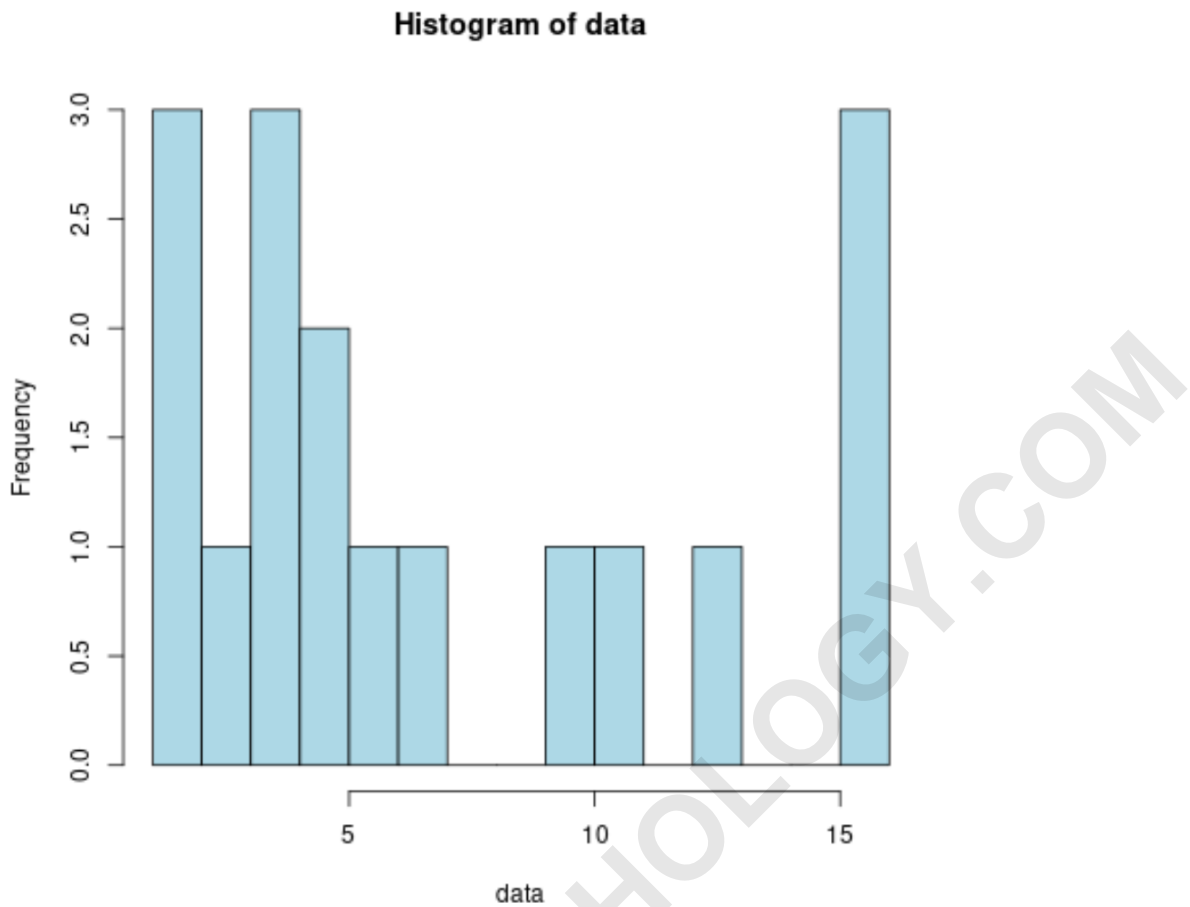
Conversely, if we use too many bins then we may just be visualizing the noise in a dataset:

#define vector of data

```
data <- c(1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 10, 11, 13, 16, 16, 16)
```

#create histogram with 15 bins

```
hist(data, col = 'lightblue', breaks = seq(min(data), max(data), length.out = 16))
```



In general, the default Sturges' Rule used in R tends to produce histograms that have an optimal number of bins.

Feel free to use the code provided here to create a histogram with an exact number of bins, but be careful not to choose too many or too few bins.

The following tutorials explain how to perform other common functions with histograms in R: