

What is the process for changing the number of bins in a histogram?

Authored by
stats writer

May 4, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the process for changing the number of bins in a histogram?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=142700>

The process for changing the number of bins in a histogram involves adjusting the number of categories or intervals that the data is divided into on the x-axis. This can be done by either increasing or decreasing the number of bins, which in turn affects the width and height of each bar on the histogram. To change the number of bins, one must first determine the range of the data and then decide on an appropriate number of bins to accurately represent the data. This can be done manually by calculating the number of bins using a mathematical formula or by using software tools that automatically determine the optimal number of bins based on the data. Once the desired number of bins is determined, the histogram can be updated accordingly, providing a more detailed or simplified visual representation of the data.

R: Change Number of Bins in Histogram

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

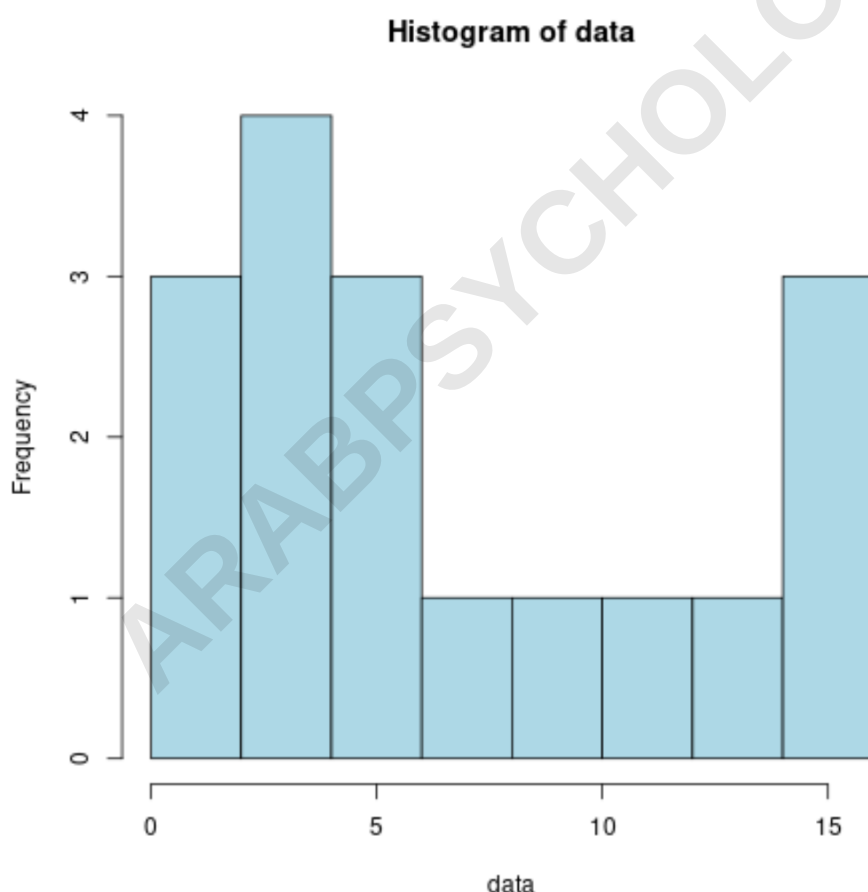
The following code shows how to create a basic histogram in R without specifying the number of bins:

```
#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

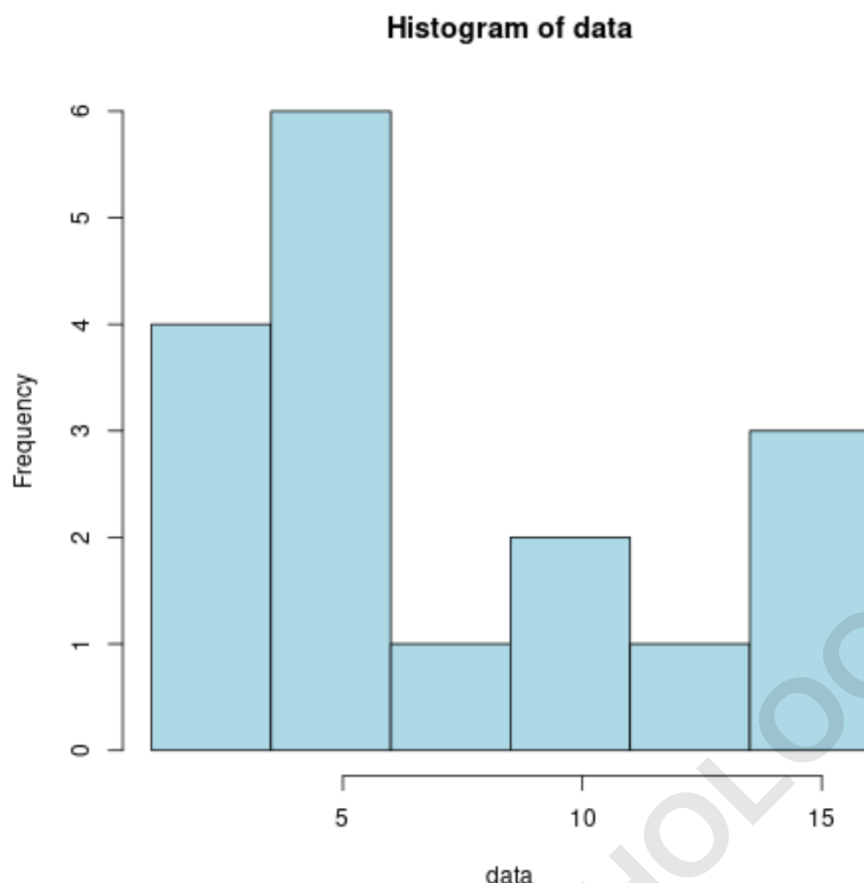
The following code shows how to create a histogram for the same vector of data and use exactly 6 bins:

```
#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:

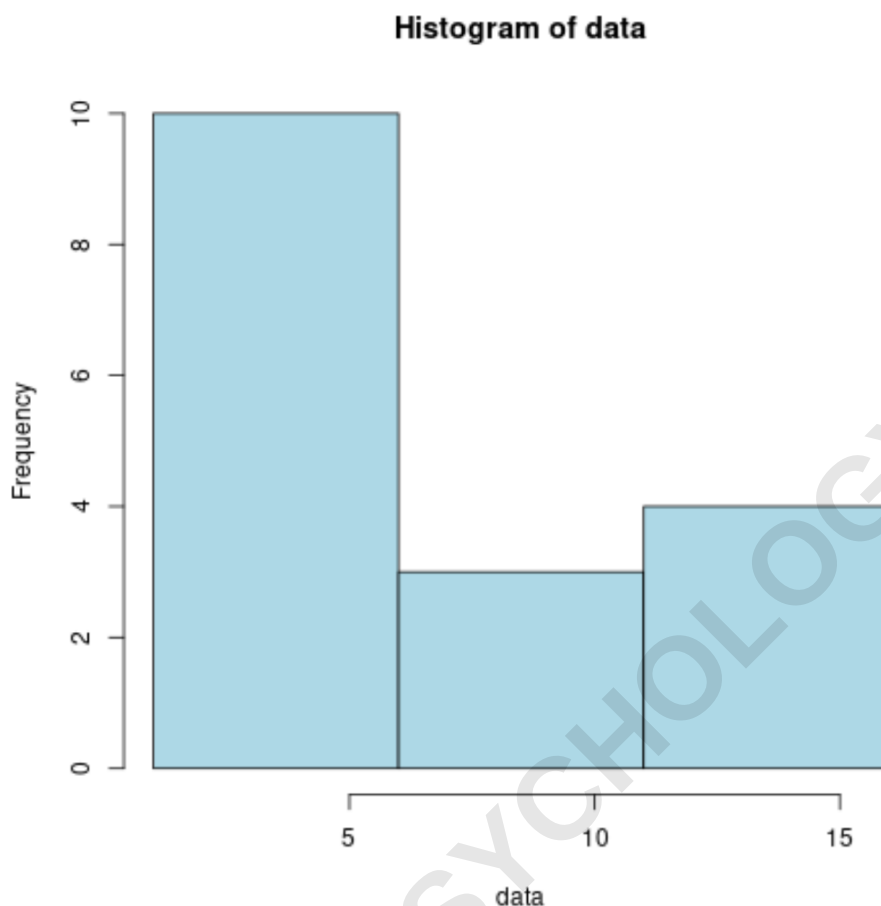
#define vector of data

```
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))



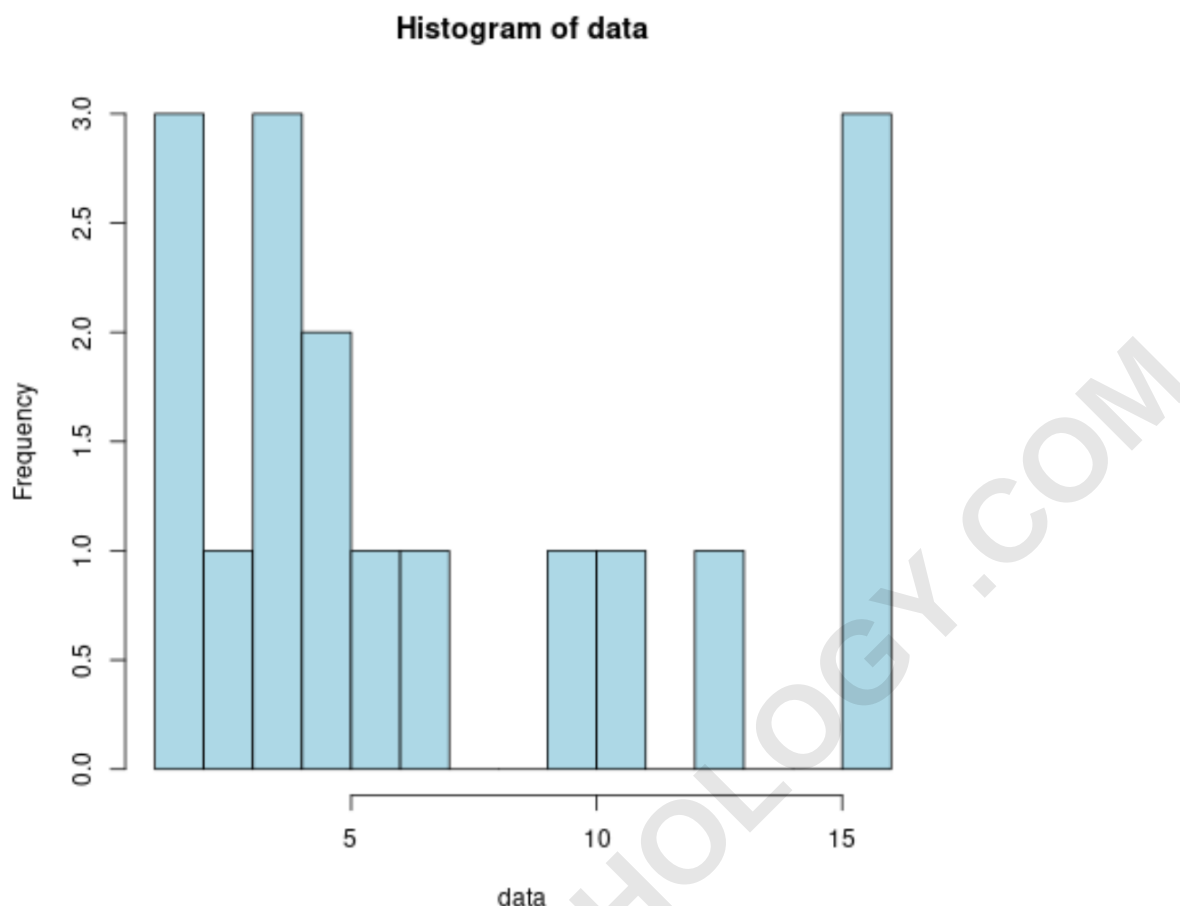
Conversely, if we use too many bins then we may just be visualizing the noise in a dataset:

#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: