

How can histogram breaks be specified in R, and what are some examples of doing so?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can histogram breaks be specified in R, and what are some examples of doing so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165447>

Histogram breaks in R can be specified using the "breaks" parameter in the "hist()" function. This parameter allows the user to specify the number of bins or the specific values where the bins should be divided. For example, specifying "breaks=5" will create a histogram with 5 equally sized bins, while specifying "breaks=c(0,10,20,30)" will create bins with boundaries at 0, 10, and 20.

Another way to specify histogram breaks in R is by using the "seq()" function. This function allows the user to create a sequence of numbers, which can then be used as the breaks for the histogram. For instance, using "breaks=seq(0,100,by=10)" will create bins with boundaries at 0, 10, 20, 30, and so on until 100.

Histogram breaks can also be specified using the "pretty()" function, which calculates evenly spaced breaks based on the range of the data. For example, using "breaks=pretty(data)" will create bins with evenly spaced boundaries based on the range of the data.

In summary, R provides various methods for specifying histogram breaks, such as using the "breaks" parameter, the "seq()" function, and the "pretty()" function. These options allow the user to customize the histogram and better understand the distribution of their data.

Specify Histogram Breaks in R (With Examples)

By default, the hist() function in R uses to determine how many bins to use in a histogram.

Sturges' Rule uses the following formula to determine the optimal number of bins to use in a histogram:

$$\text{Optimal Bins} = \lceil \log_2 n \rceil + 1$$

where:

n: The total number of in the dataset. $\lceil \cdot \rceil$: Symbols that mean "ceiling" - i.e. round the answer up to the nearest integer.

For example, if there are 31 observations in a dataset, Sturge's Rule will use the following formula to determine the optimal number of bins to use in a histogram:

$$\text{Optimal Bins} = \lceil \log_2(31) + 1 \rceil = \lceil 4.954 + 1 \rceil = \lceil 5.954 \rceil = 6.$$

According to Sturges' Rule, we should use 6 bins in the histogram to visualize this dataset.

If you use the `hist()` function in R, Sturges' Rule will be used to automatically choose the number of bins to display in the histogram.

`hist(data)`

Even if you use the `breaks` argument to specify a different number of bins to use, R will only use this as a "suggestion" for how many bins to use.

`hist(data, breaks=7)`

However, you can use the following code to force R to use a specific number of bins in a histogram:

```
#create histogram with 7 bins
```

```
hist(data, breaks = seq(min(data), max(data), length.out  
= 8))
```

Note: You must use a length of $n+1$ for length.out where n is your desired number of bins.

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

Example: Specify Histogram Breaks in R

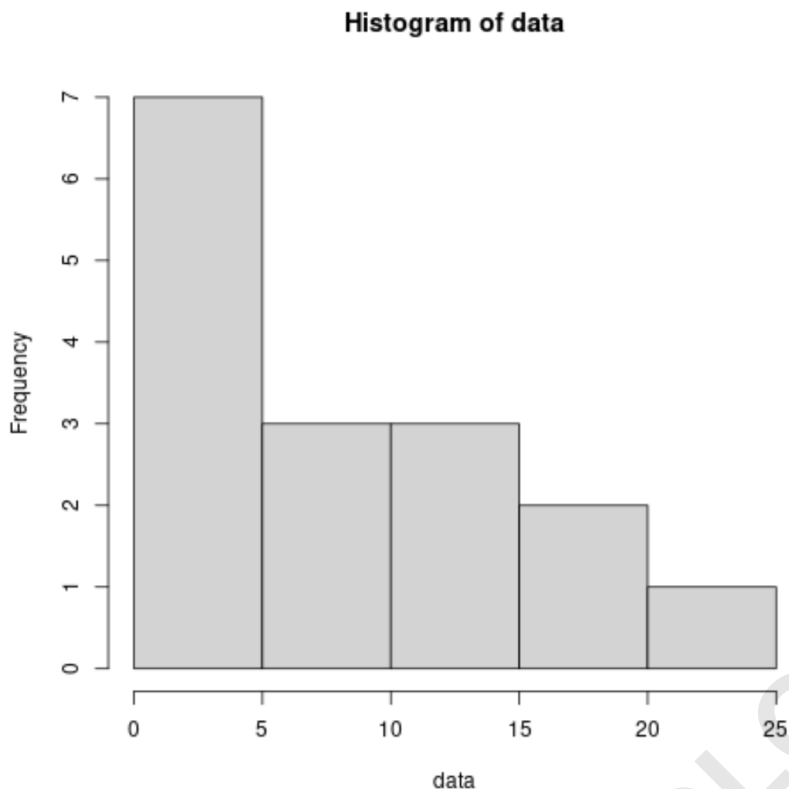
```
#create vector of 16 values
```

```
data <- c(2, 3, 3, 3, 4, 4, 5, 6, 8, 10, 12, 14, 15, 18, 20, 21)
```

If we use the hist() function, R will create the following histogram with 5 bins:

```
#create histogram
```

```
hist(data)
```

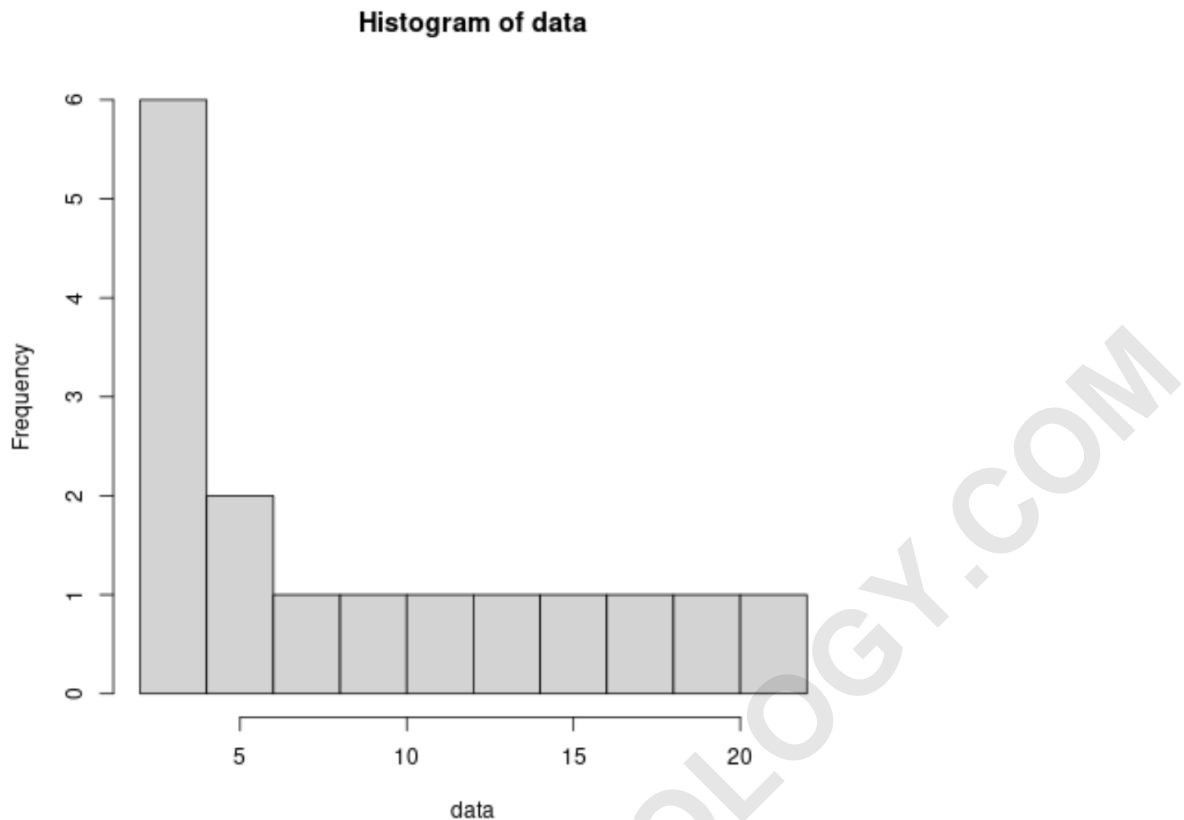


Note: R used to determine that 5 bins was the optimal number of bins to use to visualize a dataset with 16 observations.

If we attempt to use the breaks argument to specify 7 bins to use in the histogram, R will only take this as a "suggestion" and instead choose to use 10 bins:

#attempt to create histogram with 7 bins

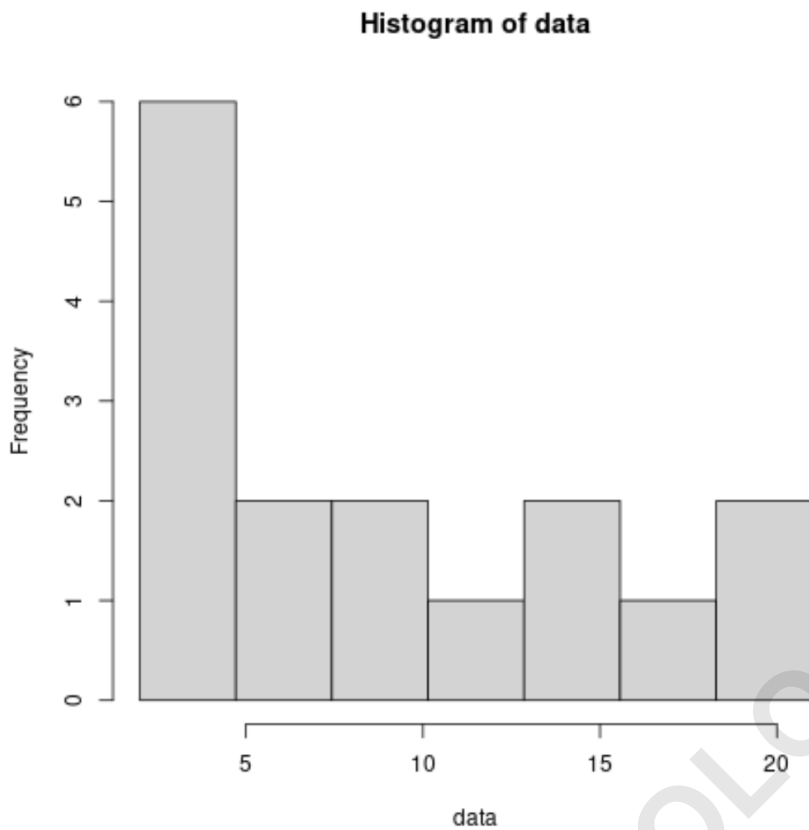
hist(data, breaks=7)



However, we can use the following code to force R to use 7 bins in the histogram:

```
#create histogram with 7 bins
```

```
hist(data, breaks = seq(min(data), max(data), length.out = 8))
```



Notice that the result is a histogram with 7 equally-spaced bins.

Additional Resources