
histogram

syntax: `histogram(data)`

plots out a histogram of the single column of data. Relative frequencies are used.

`histogram(data1,data2, ...)`

plots out a series of histograms, one for each data set. All the histograms are shown on the same scale, and using the same bin divisions.

`histogram(data1, 'title1', data2, 'title2', ...)`

allows you to specify the title for each histogram.

There are two optional arguments that can be specified.

- A first argument of `'abs'` instructs the program to use absolute frequencies (i.e. the total number of points in each bin).
- A first or second argument of a single number instructs the program to use exactly that number of bins in constructing the histogram.

For example

```
>> histogram('abs', 100, data)
```

constructs a histogram with 100 bins, using absolute frequencies.

```
>> histogram(100, data)
```

constructs a histogram with 100 bins, using relative frequencies.

examples: To illustrate, we'll generate some simulated data, using `normal` which generates a matrix of random number from a normal distribution with mean=0 and standard deviation = 1.

```
data1 = normal(100,0,1); 100 points with mean 0, std 1
```

```
data2 = normal(1000,0,2); 1000 points mean 0, std 2
```

```
data3 = normal(200,3,1); 200 points with mean 3, std 1
```

Note that the data sets happen to be of different lengths.

```
>> histogram(data1, 'data1')
```

generates the histogram shown below to the left with title "data1". Relative frequencies are used as indicated on the axis label. This means that the area of the bars, calculated using the scales shown on the axis, is exactly 1.0.

The same histogram can be generated using absolute frequencies.

```
>> histogram('abs', data1, 'data1')
```

as shown on the right. In this histogram, the heights of all the bars added together gives the total number of points in the data set.

It is sometimes convenient to be able to compare many histograms, all on the same scale.

```
>> histogram(data1,'data1',data2,'data2',data3,'data3')
```

If you want the histograms to be on different scales, then use a separate call to `histogram` for each data set individually.

warning: The title for each histogram, if used, must be in single quotes as in `'this is the title'`.

Each data set must be a single column of numbers.

After plotting out multiple histograms, subsequent plots using other programs will be made in the place of the last histogram drawn. To avoid this, give either the command

```
>> figure
```

to create a new plotting window, or

```
>> subplot
```

to instruct MATLAB to use the whole window for plotting.

See also: The built-in MATLAB commands `hist`, `figure`, `subplot`.

<p>This document is an excerpt from <i>Resampling Stats in MATLAB</i> Daniel T. Kaplan Copyright (c) 1999 by Daniel T. Kaplan, All Rights Reserved This document differs from the published book in pagination and in the omission (unintentional, but unavoidable for technical reasons) of figures and cross-references from the book. It is provided as a courtesy to those who wish to examine the book, but not intended as a replacement for the published book, which is available from Resampling Stats, Inc. www.resample.com 703-522-2713</p>
