Title

stata.com

histogram — Histograms for continuous and categorical variables

Description

`histogram` draws histograms of *varname*, which is assumed to be the name of a continuous variable unless the `discrete` option is specified.

Quick start

Histogram of `v1`
```
histogram v1
```

Add a normal density curve to the graph
```
histogram v1, normal
```

Add a kernal density estimate to the graph
```
histogram v1, normal kdensity
```

Add "My Title" as the title of the graph
```
histogram v1, normal kdensity title("My Title")
```

Specify the number of bins as 10
```
histogram v1, bin(10)
```

Specify the width of the bins as 2
```
histogram v1, width(2)
```

Specify that `v2` should be treated as discrete
```
histogram v2, discrete
```

As above, but with narrower bars and space between the bars
```
histogram v2, discrete barwidth(.8)
```

Add labels to the bars on the x axis
```
histogram v2, discrete barwidth(.8) xlabel(1 "Category 1" ///
     2 "Category 2" 3 "Category 3" 4 "Category 4")
```

Show frequencies on the y axis
```
histogram v1, frequency
```

Show percentages on the y axis
```
histogram v1, percent
```

Produce histograms for each value of categorical variable `catvar`
```
histogram v1, by(catvar)
```

As above, but with histograms arranged in a single column
```
histogram v1, by(cvar, cols(1))
```

Menu

Graphics > Histogram

Syntax

```
histogram varname [if] [in] [weight] [, [continuous_opts | discrete_opts] options]
```

continuous_opts	Description
Main	
`bin(#)`	set number of bins to #
`width(#)`	set width of bins to #
`start(#)`	set lower limit of first bin to #

discrete_opts	Description
Main	
`discrete`	specify that data are discrete
`width(#)`	set width of bins to #
`start(#)`	set theoretical minimum value to #

options	Description
Main	
`density`	draw as density; the default
`fraction`	draw as fractions
`frequency`	draw as frequencies
`percent`	draw as percentages
bar_options	rendition of bars
`binrescale`	recalculate bin sizes when `by()` is specified
`addlabels`	add height labels to bars
`addlabopts(`*marker_label_options*`)`	affect rendition of labels
Density plots	
`normal`	add a normal density to the graph
`normopts(`*line_options*`)`	affect rendition of normal density
`kdensity`	add a kernel density estimate to the graph
`kdenopts(`*kdensity_options*`)`	affect rendition of kernel density
Add plots	
`addplot(`*plot*`)`	add other plots to the histogram
Y axis, X axis, Titles, Legend, Overall, By	
twoway_options	any options documented in [G-3] ***twoway_options***

`fweights` are allowed; see **[U] 11.1.6 weight**.

Options for use in the continuous case

Main

`bin(#)` and `width(#)` are alternatives. They specify how the data are to be aggregated into bins: `bin()` by specifying the number of bins (from which the width can be derived) and `width()` by specifying the bin width (from which the number of bins can be derived).

If neither option is specified, results are the same as if `bin(`*k*`)` had been specified, where

$$k = \min\left\{\text{sqrt}(N), 10\ln(N)/\ln(10)\right\}$$

and where N is the (weighted) number of observations.

`start(#)` specifies the theoretical minimum of *varname*. The default is `start(`*m*`)`, where *m* is the observed minimum value of *varname*.

Specify `start()` when you are concerned about sparse data, for instance, if you know that *varname* can have a value of 0, but you are concerned that 0 may not be observed.

`start(#)`, if specified, must be less than or equal to *m*, or else an error will be issued.

Options for use in the discrete case

Main

`discrete` specifies that *varname* is discrete and that you want each unique value of *varname* to have its own bin (bar of histogram).

`width(#)` is rarely specified in the discrete case; it specifies the width of the bins. The default is `width(`*d*`)`, where *d* is the observed minimum difference between the unique values of *varname*.

Specify `width()` if you are concerned that your data are sparse. For example, in theory *varname* could take on the values, say, 1, 2, 3, ..., 9, but because of the sparseness, perhaps only the values 2, 4, 7, and 8 are observed. Here the default width calculation would produce `width(2)`, and you would want to specify `width(1)`.

`start(#)` is also rarely specified in the discrete case; it specifies the theoretical minimum value of *varname*. The default is `start(`*m*`)`, where *m* is the observed minimum value.

As with `width()`, specify `start(#)` if you are concerned that your data are sparse. In the previous example, you might also want to specify `start(1)`. `start()` does nothing more than add white space to the left side of the graph.

The value of # in `start()` must be less than or equal to *m*, or an error will be issued.

Options for use in the continuous and discrete cases

Main

`density`, `fraction`, `frequency`, and `percent` specify whether you want the histogram scaled to density units, fractional units, frequencies, or percentages. `density` is the default.

`density` scales the height of the bars so that the sum of their areas equals 1.

`fraction` scales the height of the bars so that the sum of their heights equals 1.

`frequency` scales the height of the bars so that each bar's height is equal to the number of observations in the category. Thus the sum of the heights is equal to the total number of observations.

`percent` scales the height of the bars so that the sum of their heights equals 100.

bar_options are any of the options allowed by `graph twoway bar`; see [G-2] **graph twoway bar**.

One of the most useful *bar_options* is `barwidth(#)`, which specifies the width of the bars in *varname* units. By default, `histogram` draws the bars so that adjacent bars just touch. If you want gaps between the bars, do not specify `histogram`'s `width()` option—which would change how the histogram is calculated—but specify the *bar_option* `barwidth()` or the `histogram` option `gap`, both of which affect only how the bar is rendered.

The *bar_option* `horizontal` cannot be used with the `addlabels` option.

`binrescale` specifies that bin size and plot range be recalculated for each group when `by()` is specified. If `normal` is specified, the mean and standard deviation of each overlaid normal density plot are recalculated in each group. Similarly, if `kdensity` is specified, the scaling of the overlaid kernel density plot is recalculated in each group.

`addlabels` specifies that the top of each bar be labeled with the density, fraction, or frequency, as determined by the `density`, `fraction`, and `frequency` options.

`addlabopts(`*marker_label_options*`)` specifies how to render the labels atop the bars. See [G-3] ***marker_label_options***. Do not specify the *marker_label_option* `mlabel(`*varname*`)`, which specifies the variable to be used; this is specified for you by `histogram`.

`addlabopts()` will accept more options than those documented in [G-3] ***marker_label_options***. All options allowed by `twoway scatter` are also allowed by `addlabopts()`; see [G-2] **graph twoway scatter**. One particularly useful option is `yvarformat()`; see [G-3] ***advanced_options***.

Density plots

`normal` specifies that the histogram be overlaid with an appropriately scaled normal density. The normal will have the same mean and standard deviation as the data.

`normopts(`*line_options*`)` specifies details about the rendition of the normal curve, such as the color and style of line used. See [G-2] **graph twoway line**.

`kdensity` specifies that the histogram be overlaid with an appropriately scaled kernel density estimate of the density. By default, the estimate will be produced using the Epanechnikov kernel with an "optimal" half-width. This default corresponds to the default of `kdensity`; see [R] **kdensity**. How the estimate is produced can be controlled using the `kdenopts()` option described below.

`kdenopts(`*kdensity_options*`)` specifies details about how the kernel density estimate is to be produced along with details about the rendition of the resulting curve, such as the color and style of line used. The kernel density estimate is described in [G-2] **graph twoway kdensity**. As an example, if you wanted to produce kernel density estimates by using the Gaussian kernel with optimal half-width, you would specify `kdenopts(gauss)` and if you also wanted a half-width of 5, you would specify `kdenopts(gauss width(5))`.

Add plots

`addplot(`*plot*`)` allows adding more `graph twoway` plots to the graph; see [G-3] ***addplot_option***.

twoway_options are any of the options documented in [G-3] ***twoway_options***. This includes, most importantly, options for titling the graph (see [G-3] ***title_options***), options for saving the graph to disk (see [G-3] ***saving_option***), and the `by()` option, which will allow you to simultaneously graph histograms for different subsets of the data (see [G-3] ***by_option***).

Remarks and examples

stata.com

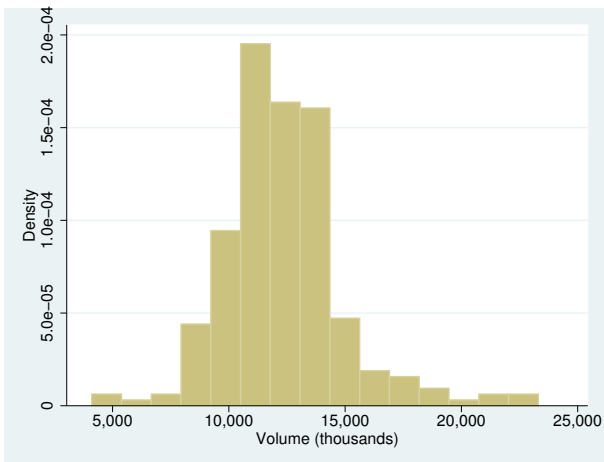
Remarks are presented under the following headings:

For an example of editing a histogram with the Graph Editor, see Pollock (2011, 29–31).

Histograms of continuous variables

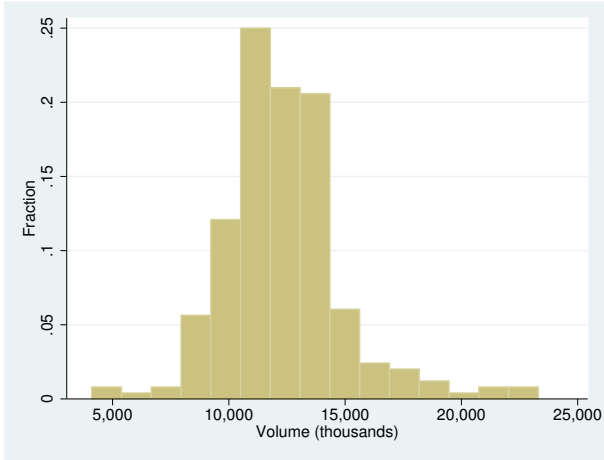
`histogram` assumes that the variable is continuous, so you need type only `histogram` followed by the variable name:

```
. use http://www.stata-press.com/data/r14/sp500
(S&P 500)
. histogram volume
(bin=15, start=4103, width=1280.3533)
```



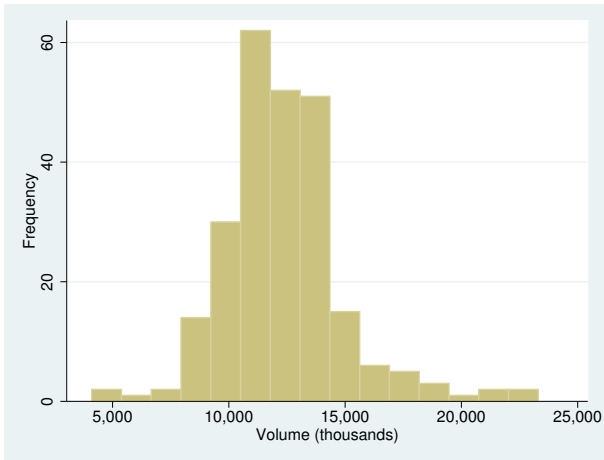
The small values reported for density on the y axis are correct; if you added up the area of the bars, you would get 1. Nevertheless, many people are used to seeing histograms scaled so that the bar heights sum to 1,

```
. histogram volume, fraction
(bin=15, start=4103, width=1280.3533)
```



and others are used to seeing histograms so that the bar height reflects the number of observations,

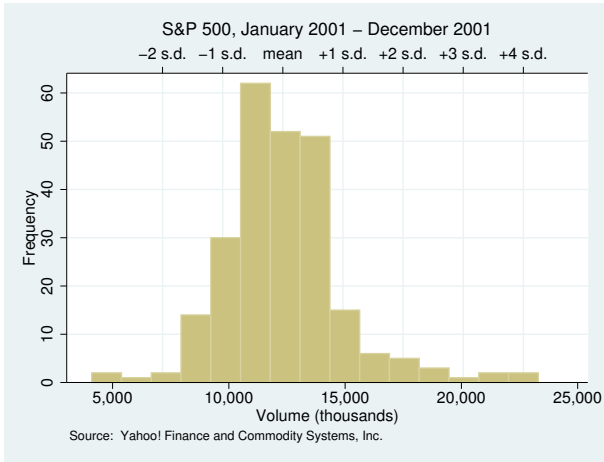
```
. histogram volume, frequency
(bin=15, start=4103, width=1280.3533)
```



Regardless of the scale you prefer, you can specify other options to make the graph look more impressive:

```
. summarize volume
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      volume |        248    12320.68    2585.929      4103    23308.3

. histogram volume, freq
>         xaxis(1 2)
>         ylabel(0(10)60, grid)
>         xlabel(12321 "mean"
>                9735 "-1 s.d."
>               14907 "+1 s.d."
>                7149 "-2 s.d."
>               17493 "+2 s.d."
>               20078 "+3 s.d."
>               22664 "+4 s.d."
>                                , axis(2) grid gmax)
>         xtitle("", axis(2))
>         subtitle("S&P 500, January 2001 - December 2001")
>         note("Source:  Yahoo! Finance and Commodity Systems, Inc.")
(bin=15, start=4103, width=1280.3533)
```



For an explanation of the `xaxis()` option—it created the upper and lower x axis—see [G-3] ***axis_choice_options***. For an explanation of the `ylabel()` and `xlabel()` options, see [G-3] ***axis_label_options***. For an explanation of the `subtitle()` and `note()` options, see [G-3] ***title_options***.

Overlaying normal and kernel density estimates

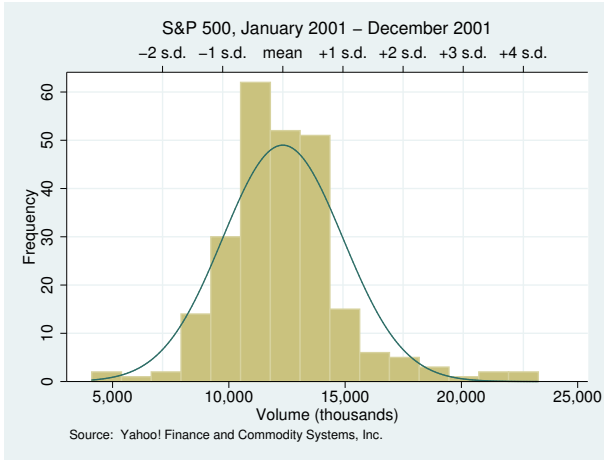
Specifying `normal` will overlay a normal density over the histogram. It would be enough to type

```
. histogram volume, normal
```

but we will add the option to our more impressive rendition:

```
. summarize volume
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      volume |        248    12320.68    2585.929      4103    23308.3
```

```
. histogram volume, freq normal
>         xaxis(1 2)
>         ylabel(0(10)60, grid)
>         xlabel(12321 "mean"
>                9735 "-1 s.d."
>               14907 "+1 s.d."
>                7149 "-2 s.d."
>               17493 "+2 s.d."
>               20078 "+3 s.d."
>               22664 "+4 s.d."
>                                , axis(2) grid gmax)
>         xtitle("", axis(2))
>         subtitle("S&P 500, January 2001 - December 2001")
>         note("Source:  Yahoo! Finance and Commodity Systems, Inc.")
(bin=15, start=4103, width=1280.3533)
```



If we instead wanted to overlay a kernel density estimate, we could specify `kdensity` in place of `normal`.

Histograms of discrete variables

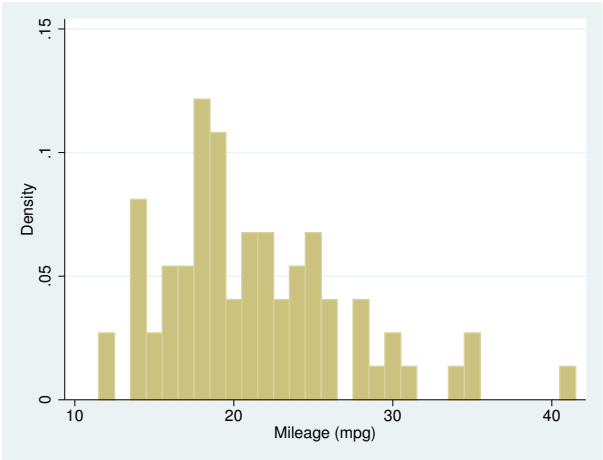
Specify `histogram`'s discrete option when you wish to treat the data as discrete—when you wish each unique value of the variable to be assigned its own bin. For instance, in the automobile data, `mpg` is a continuous variable, but the mileage ratings have been measured to integer precision. If we were to type

```
. use http://www.stata-press.com/data/r14/auto
(1978 Automobile Data)
. histogram mpg
(bin=8, start=12, width=3.625)
```

`mpg` would be treated as continuous and categorized into eight bins by the default number-of-bins calculation, which is based on the number of observations, 74.

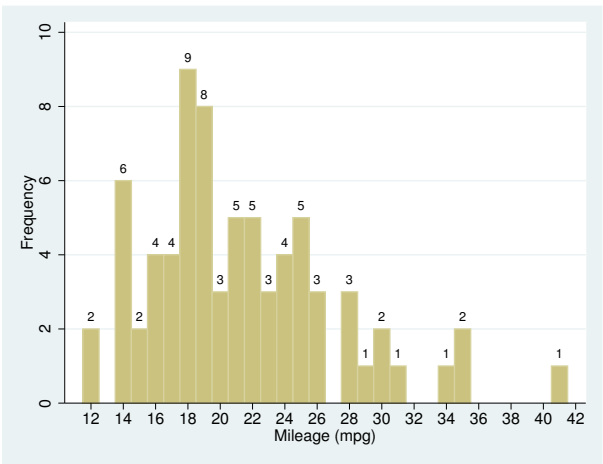
Adding the `discrete` option makes a histogram with a bin for each of the 21 unique values.

```
. histogram mpg, discrete
(start=12, width=1)
```



Just as in the continuous case, the y axis was reported in density, and we could specify the `fraction` or `frequency` options if we wanted it to be reported differently. Below we specify `frequency`, we specify `addlabels` to add a report of frequencies printed above the bars, we specify `ylabel(,grid)` to add horizontal grid lines, and we specify `xlabel(12(2)42)` to label the values 12, 14, ..., 42 on the x axis:

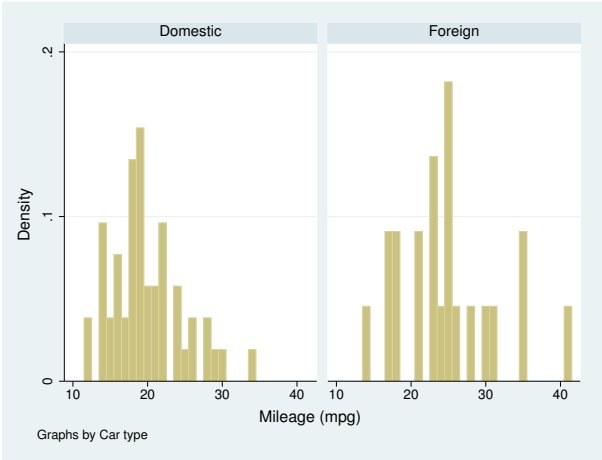
```
. histogram mpg, discrete freq addlabels ylabel(,grid) xlabel(12(2)42)
(start=12, width=1)
```



Use with by()

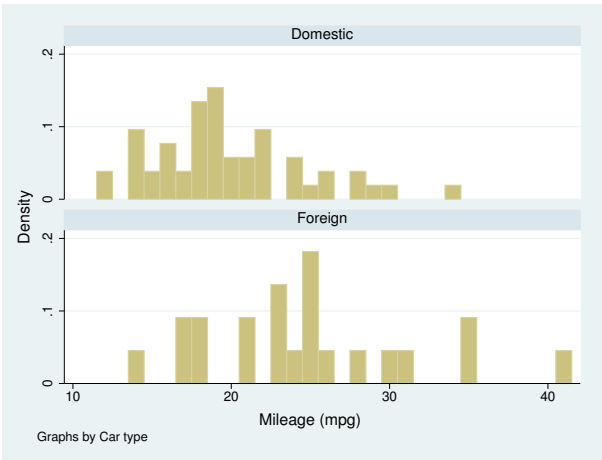
histogram may be used with graph twoway's by(); for example,

```
. use http://www.stata-press.com/data/r14/auto
(1978 Automobile Data)
. histogram mpg, discrete by(foreign)
```



Here results would be easier to compare if the graphs were presented in one column:

```
. histogram mpg, discrete by(foreign, col(1))
```



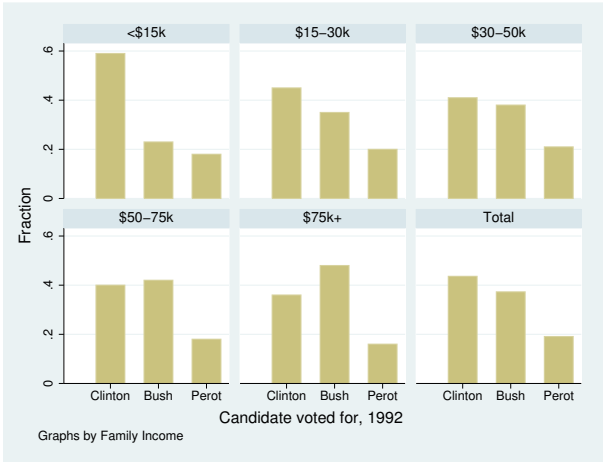
col(1) is a by() suboption—see [G-3] *by_option*—and there are other useful suboptions, such as total, which will add an overall total histogram. total is a suboption of by(), not an option of histogram, so you would type

```
. histogram mpg, discrete by(foreign, total)
```

and not histogram mpg, discrete by(foreign) total.

As another example, Lipset (1993) reprinted data from the *New York Times* (November 5, 1992) collected by the Voter Research and Surveys based on questionnaires completed by 15,490 U.S. presidential voters from 300 polling places on election day in 1992.

```
. use http://www.stata-press.com/data/r14/voter
. histogram candi [freq=pop], discrete fraction by(inc, total)
> gap(40) xlabel(2 3 4, valuelabel)
```



We specified `gap(40)` to reduce the width of the bars by 40%. We also used `xlabel()`'s `valuelabel` suboption, which caused our bars to be labeled "Clinton", "Bush", and "Perot", rather than 2, 3, and 4; see [G-3] ***axis_label_options***.

Video example

Histograms in Stata

References

Cox, N. J. 2004. Speaking Stata: Graphing distributions. *Stata Journal* 4: 66–88.

——. 2005. Speaking Stata: Density probability plots. *Stata Journal* 5: 259–273.

Harrison, D. A. 2005. Stata tip 20: Generating histogram bin variables. *Stata Journal* 5: 280–281.

Lipset, S. M. 1993. The significance of the 1992 election. *PS: Political Science and Politics* 26: 7–16.

Pollock, P. H., III. 2011. *A Stata Companion to Political Analysis*. 2nd ed. Washington, DC: CQ Press.

Also see

[R] **kdensity** — Univariate kernel density estimation

[R] **spikeplot** — Spike plots and rootograms

[G-2] **graph twoway histogram** — Histogram plots