

[Description](#)
[Options](#)[Quick start](#)
[Remarks and examples](#)[Menu](#)
[Also see](#)[Syntax](#)

Description

`twoway contourline` displays z as contour lines in (y, x) .

Quick start

Contour-line plot displaying the value of z for each (x, y) pair

```
twoway contourline z y x
```

Same as above, but draw contour lines for $z = -4$ to 1.5 in increments of 0.25

```
twoway contourline z y x, ccuts(-4(.25)1.5)
```

Specify 25 contour levels

```
twoway contourline z y x, levels(25)
```

Same as above, but with colored contour lines

```
twoway contourline z y x, levels(25) colorlines
```

Same as above, but with an alternative method of selecting line colors

```
twoway contourline z y x, levels(25) colorlines crule(chue)
```

Reverse the order cutpoints are listed in the legend

```
twoway contourline z y x, levels(25) colorlines reversekey
```

Add “My Title” as the title of the graph

```
twoway contourline z y x, title("My Title")
```

Add “My note” as a note on the graph

```
twoway contourline z y x, title("My Title") note("My note")
```

Menu

Graphics > Two-way graph (scatter, line, etc.)

Syntax

```
twoway contourline z y x [if] [in] [, options]
```

<i>options</i>	Description
<code>ccuts(<i>numlist</i>)</code>	list of values for contour lines or cuts
<code>levels(#)</code>	number of contour levels
<code>minmax</code>	include contour lines for minimum and maximum of <i>z</i>
<code>format(<i>%fmt</i>)</code>	display format for <code>ccuts()</code> or <code>levels()</code>
<code>colorlines</code>	display contour lines in different colors
<code>crule(<i>crule</i>)</code>	rule for creating contour-line colors
<code>scolor(<i>colorstyle</i>)</code>	starting color for contour rule
<code>ecolor(<i>colorstyle</i>)</code>	ending color for contour rule
<code>ccolors(<i>colorstylelist</i>)</code>	list of colors for contour lines
<code>clwidths(<i>linewidthstylelist</i>)</code>	list of widths for contour lines
<code>reversekey</code>	reverse the order of the keys in contour-line legend
<code>interp(<i>interpmethod</i>)</code>	interpolation method if (<i>z</i> , <i>y</i> , <i>x</i>) does not fill a regular grid
<code>twoway_options</code>	titles, legends, axes, added lines and text, by, regions, name, aspect ratio, etc.

<i>crule</i>	Description
<code>hue</code>	use equally spaced hues between <code>scolor()</code> and <code>ecolor()</code> ; the default
<code>chue</code>	use equally spaced hues between <code>scolor()</code> and <code>ecolor()</code> ; unlike <code>hue</code> , it uses $360 + \text{hue of the } \text{ecolor}()$ if the hue of the <code>ecolor()</code> is less than the hue of the <code>scolor()</code>
<code>intensity</code>	use equally spaced intensities with <code>ecolor()</code> as the base; <code>scolor()</code> is ignored
<code>linear</code>	use equally spaced interpolations of the RGB values between <code>scolor()</code> and <code>ecolor()</code>

<i>interpmethod</i>	Description
<code>thinplatespline</code>	thin-plate-spline interpolation; the default
<code>shepard</code>	Shepard interpolation
<code>none</code>	no interpolation; plot data as is

Options

`ccuts()`, `levels()`, `minmax`, and `format()` determine how many contours are created and the values of those contours.

`ccuts(numlist)` specifies the *z* values for the contour lines. Contour lines are drawn at each value of *numlist*.

`levels(#)` specifies the number of contour lines to create; # - 1 contour lines will be created.

`minmax` is a modifier of `levels()` and specifies that contour lines be drawn for the minimum and maximum values of z . By default, lines are drawn only for the cut values implied by `levels`, not the full range of z .

`format(%fmt)` specifies the display format used to create the labels in the [contour legend](#) for the contour lines.

`ccuts()` and `levels()` are different ways of specifying the contour cuts and may not be combined.

`colorlines`, `crule()`, `scolor()`, `ecolor()`, `ccolors()`, and `clwidths()` determine the colors and width that are used for each contour line.

`colorlines` specifies that the contour lines be drawn in different colors. Unless the `ccolors()` option is specified, the colors are determined by `crule()`.

`crule(crule)` specifies the rule used to set the colors for the contour lines. Valid *crules* are `hue`, `chue`, `intensity`, and `linear`. The default is `crule(hue)`.

`scolor(colorstyle)` specifies the starting color for the rule. See [\[G-4\] colorstyle](#).

`ecolor(colorstyle)` specifies the ending color for the rule. See [\[G-4\] colorstyle](#).

`ccolors(colorstylelist)` specifies a list of *colorstyles* for each contour line. If RGB, CMYK, HSV, hexadecimal, or an intensity-adjusted (for example, `red*.3`) *colorstyle* is specified, it should be placed in quotes. Examples of valid `ccolors()` options include `ccolors(red green magenta)` and `ccolors(red "55 132 22" ".3 .9 .3 hsv" blue)`. See [\[G-4\] colorstyle](#).

`clwidths(linewidthstylelist)` specifies a list of *linewidthstyles*, one for each contour line. See [\[G-4\] linewidthstyle](#).

`reversekey` specifies that the order of the keys in the contour-line legend be reversed. By default, the keys are ordered from top to bottom, starting with the key for the highest values of z . See *legend_option* in [\[G-3\] legend_options](#).

`interp(interpmethod)` specifies the interpolation method to use if z , y , and x do not fill a regular grid. Variables z , y , and x fill a regular grid if for every combination of nonmissing (y, x) , there is at least one nonmissing z corresponding to the pair in the dataset. For example, the following dataset forms a 2×2 grid.

```
. input z y x
      z y x
1.   1 1 1
2.   2 4 1
3.   3 4 1
4.   1 1 2
5.   1 4 2
6.   end
```

If there is more than one z value corresponding to a pair of (y, x) , the smallest z value is used in plotting. In the above example, there are two z values corresponding to pair $(4, 1)$, and the smallest value, 2, is used.

```
. input z y x
      z y x
1.   1 1 1
2.   2 2 1
3.   1 1 2
4.   end
```

does not fill a regular grid because there is no z value corresponding to the pair $(2, 2)$.

twoway_options are any of the options documented in [G-3] *twoway_options*. These include options for titling the graph (see [G-3] *title_options*); for saving the graph to disk (see [G-3] *saving_option*); for controlling the labeling and look of the axes (see [G-3] *axis_options*); for controlling the look, contents, position, and organization of the legend (see [G-3] *legend_options*); for adding lines (see [G-3] *added_line_options*) and text (see [G-3] *added_text_options*); and for controlling other aspects of the graph's appearance (see [G-3] *twoway_options*).

Remarks and examples

Remarks are presented under the following headings:

Controlling the number of contour lines and their values

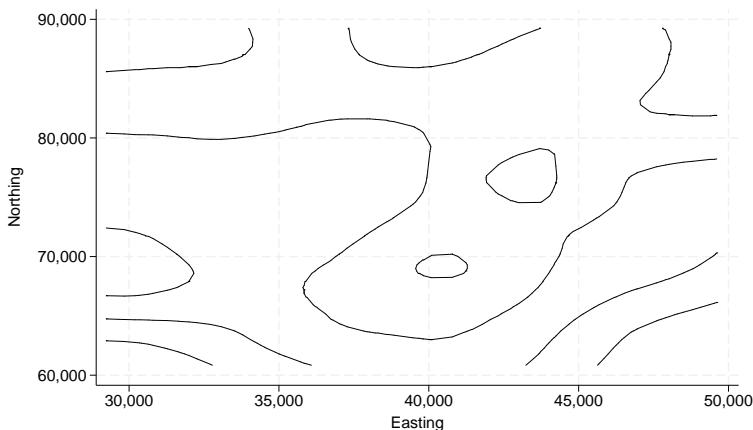
Controlling the colors of the contour lines

Choose the interpolation method

Controlling the number of contour lines and their values

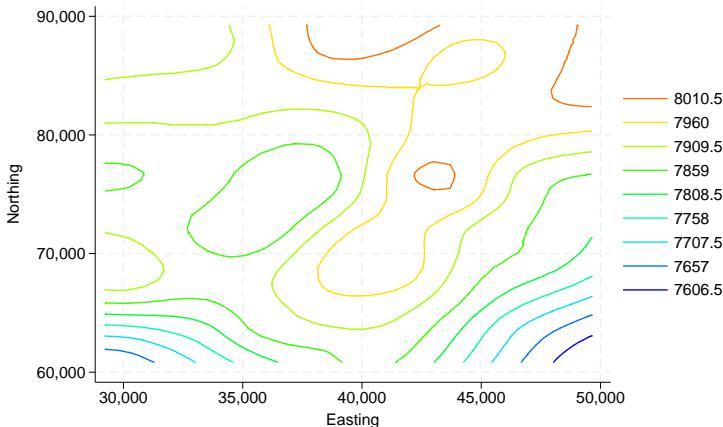
We could draw a contour-line plot with default values by typing

```
. use https://www.stata-press.com/data/r19/sandstone
(Subsea elevation of Lamont sandstone in an area of Ohio)
. twoway contourline depth northing easting
```



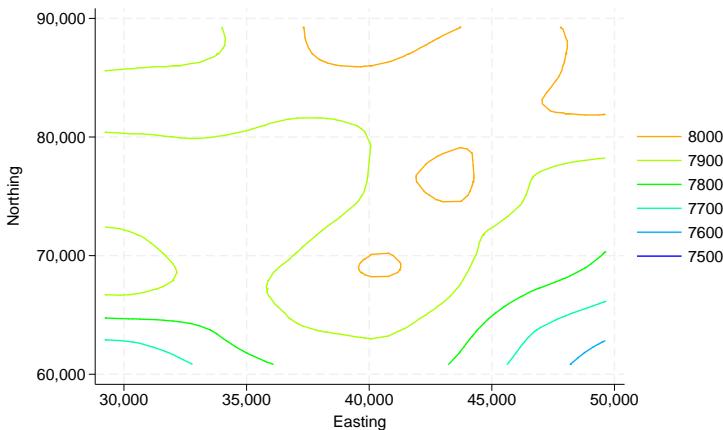
We add the `colorlines` option to display the values of cuts in the contour legend. We also include the `levels()` option to create `# - 1` contour lines equally spaced between `min(depth)` and `max(depth)`.

```
. twoway contourline depth northing easting, colorlines levels(10)
```



The `ccuts()` option gives you the finest control over creating contour lines. Here we use it to draw a contour-line plot with six cuts at 7500, 7600, 7700, 7800, 7900, and 8000.

```
. twoway contourline depth northing easting, colorlines ccuts(7500(100)8000)
```



Controlling the colors of the contour lines

`crule()`, `scolor()`, and `ecolor()` control the colors for each contour line.

```
. twoway contourline depth northing easting, level(10) format(%9.1f)
  colorlines scolor(green) ecolor(red)
```

draws a contour-line plot with lines of nine equally spaced z values between `min(depth)` and `max(depth)`. The starting color for lines is green and the ending color for lines is red. Also, the legend labels' display format is `%9.1f`.

`ccolors()` specifies a list of colors to be used for each contour line.

```
. twoway contourline depth northing easting, ccuts(7600(100)8000)  
  colorlines ccolors(red green magenta blue yellow)
```

Choose the interpolation method

See *Choose the interpolation method* in [G-2] [graph twoway contour](#).

Also see

[G-2] [graph twoway contour](#) — Two-way contour plot with area shading

[G-2] [graph twoway line](#) — Two-way line plots

[G-2] [graph twoway connected](#) — Two-way connected plots

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).