Title

graph twoway contour — Twoway contour plot with area shading

Description

`twoway contour` displays z as filled contours in (y, x).

Quick start

Contour plot displaying `z` for each (`x`, `y`) pair

```
twoway contour z y x
```

Same as above, but specify 5 contour levels

```
twoway contour z y x, levels(5)
```

Specify cutpoints for the contour lines

```
twoway contour z y x, ccuts(-2 -1 0 1)
```

Same as above

```
twoway contour z y x, ccuts(-2(1)1)
```

Specify contour colors of equally spaced intensities

```
twoway contour z y x, ccuts(-2(1)1) crule(intensity)
```

Same as above, but specify navy as the end, or darkest, color

```
twoway contour z y x, ccuts(-2(1)1) crule(intensity) ecolor(navy)
```

Specify cutpoints for the contour lines and label them

```
twoway contour z y x,                                    ///
     zlabel(-2 "Cut 1" -1 "Cut 2" 0 "Cut 3" 1 "Cut 4")
```

Draw contour plot as a heat map

```
twoway contour z y x, heatmap
```

Menu

Graphics > Twoway graph (scatter, line, etc.)

Syntax

twoway contour *z y x* [*if*] [*in*] [, *options*]

options	Description
ccuts(*numlist*)	list of values for contour lines or cuts
levels(#)	number of contour levels
minmax	include minimum and maximum of *z* in levels
crule(*crule*)	rule for creating contour-level colors
scolor(*colorstyle*)	starting color for contour rule
ecolor(*colorstyle*)	ending color for contour rule
ccolors(*colorstylelist*)	list of colors for contour levels
heatmap	draw the contour plot as a heat map
interp(*interpmethod*)	interpolation method if (*z*, *y*, *x*) does not fill a regular grid
twoway_options	titles, legends, axes, added lines and text, by, regions, name, aspect ratio, etc.

crule	Description
hue	use equally spaced hues between scolor() and ecolor(); the default
chue	use equally spaced hues between scolor() and ecolor(); unlike hue, it uses 360 + hue of the ecolor() if the hue of the ecolor() is less than the hue of the scolor()
intensity	use equally spaced intensities with ecolor() as the base; scolor() is ignored
linear	use equally spaced interpolations of the RGB values between scolor() and ecolor()

interpmethod	Description
thinplatespline	thin-plate-spline interpolation; the default
shepard	Shepard interpolation
none	no interpolation; plot data as is

Options

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

An alternative way of controlling the contour values is using the standard axis-label options available through the zlabel() option; see [G-3] ***axis_label_options***. Even when ccuts() or levels() are specified, you can further control the appearance of the contour labels using the zlabel() option.

`ccuts(`*numlist*`)` specifies the z values for the contour lines. Contour lines are drawn at each value of *numlist* and color- or shade-filled levels are created for each area between the lines and for the areas below the minimum and above the maximum.

`levels(#)` specifies the number of filled contour levels to create; $\# - 1$ contour cuts will be created.

`minmax` is a modifier of `levels()` and specifies that the minimum and maximum values of z be included in the cuts.

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

`crule()`, `scolor()`, `ecolor()`, and `ccolors()` determine the colors that are used for each filled contour level.

`crule(`*crule*`)` specifies the rule used to set the colors for the contour levels. Valid *crule*s 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 *colorstyle*s for the area of each contour level. If RGB, CMYK, HSV, or intensity-adjusted (for example, `red*.3`) colorstyle is specified, they 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***.

`heatmap` draws colored rectangles centered on each grid point. The color is determined by the z value of the grid point.

`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

stata.com

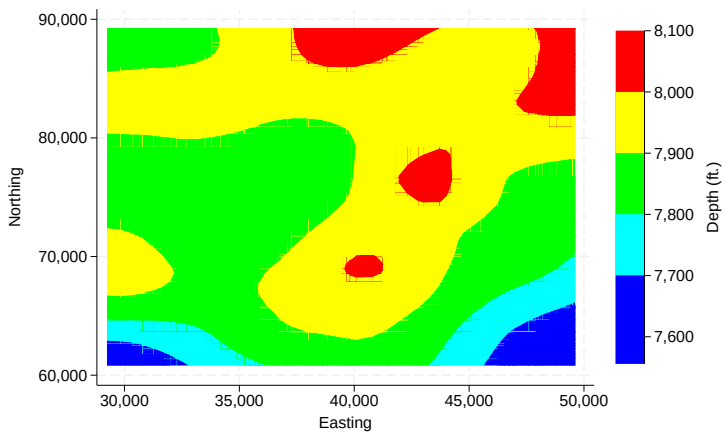
Remarks are presented under the following headings:

Controlling the number of contours and their values

We could draw a contour plot with default values by typing

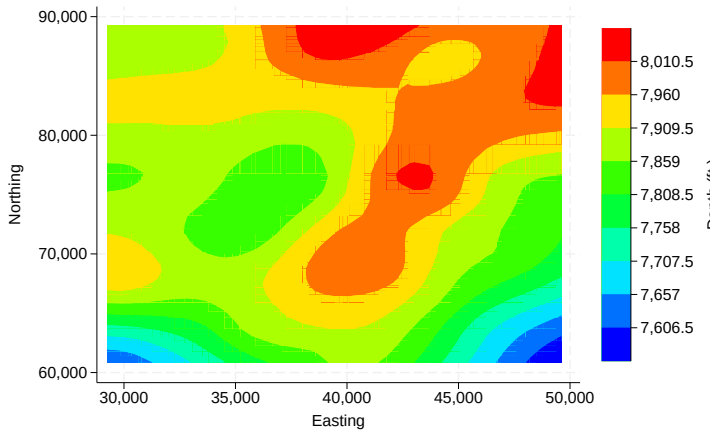
```
. use https://www.stata-press.com/data/r18/sandstone
(Subsea elevation of Lamont sandstone in an area of Ohio)

. twoway contour depth northing easting
```



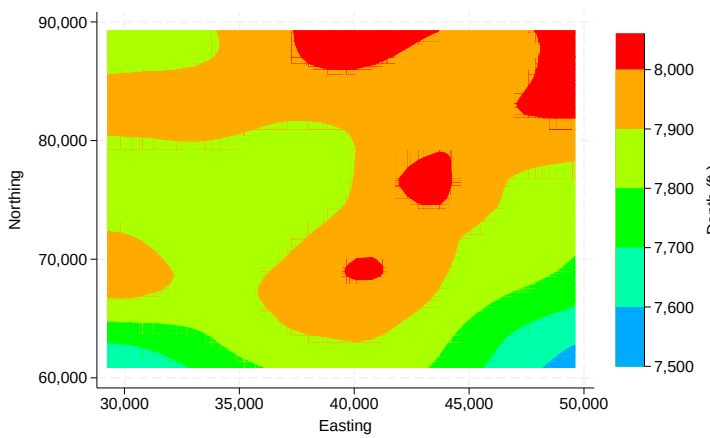
We could add the `levels()` option to the above command to create # − 1 equally spaced contours between `min(depth)` and `max(depth)`.

```
. twoway contour depth northing easting, levels(10)
```



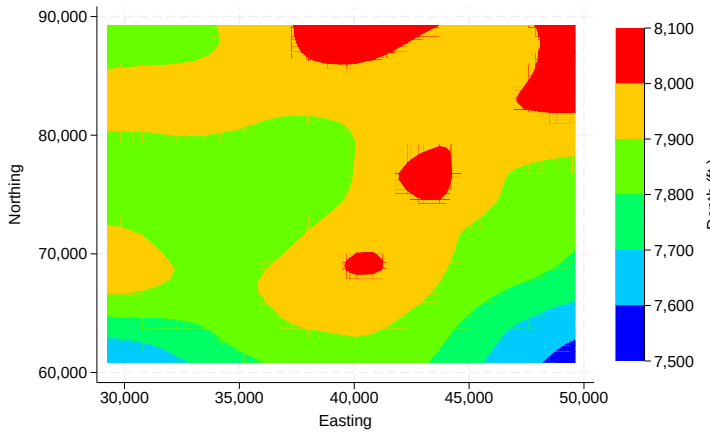
We could use the `ccuts()` option to draw a contour plot with 7 levels determined by 6 cuts at 7500, 7600, 7700, 7800, 7900, and 8000. `ccuts()` gives you the finest control over creating contour levels.

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



`zlabel()` controls the axis on the contour legend. When `ccuts()` and `levels()` are not specified, `zlabel()` also controls the number and value of contours. To obtain about 7 nicely spaced cuts, specify `zlabel(#7)`:

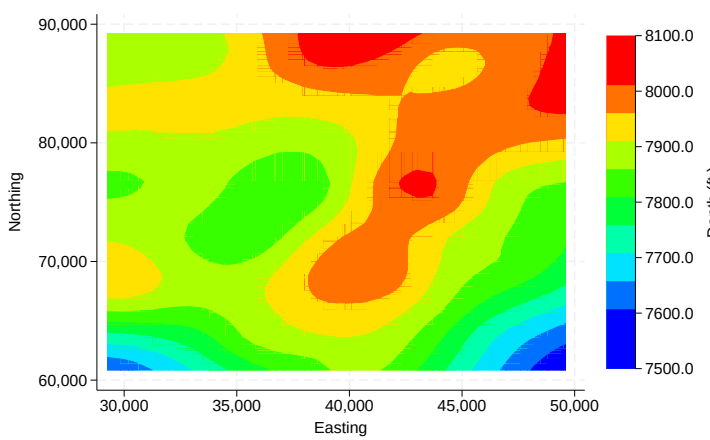
```
. twoway contour depth northing easting, zlabel(#7)
```



With either `levels()` or `ccuts()`, `zlabel()` becomes an option that only affects the labels of the contour legend. The contour legend can label different values than the actual contour cuts. The legend can have more (or fewer) ticks than the number of contour levels. See [G-3] *axis_label_options* for details.

We now specify the `twoway contour` command with the `levels()` and `zlabel()` options and the `format()` suboption to draw a 10-level contour plot with 7 labels on the contour legend. The labels' display format is `%9.1f`.

```
. twoway contour depth northing easting, levels(10) zlabel(#7, format(%9.1f))
```



Controlling the colors of the contour areas

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

```
. twoway contour depth northing easting, level(10) scolor(green) ecolor(red)
```

draws a 10-level contour plot with starting color green and ending color red. Because the hue of green is 120 and the hue of red is 0, the hues of levels are moving downward under the default `crule(hue)`. Hence you will see yellow, but not blue and purple.

For the above example, you can use `crule(chue)` if you want hues of the levels to move up:

```
. twoway contour depth northing easting, level(10) crule(chue) scolor(green)
    ecolor(red)
```

Now you will see blue and purple as the hue varies from 120 to 360 ($0 + 360$), but not yellow.

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

```
. twoway contour depth northing easting, levels(5)
    ccolors(red green magenta blue yellow)
```

Choose the interpolation method

If z, y, and x do not fill a regular grid, the missing z values on grid points (y, x) need to be interpolated.

Thin-plate-spline interpolation uses a weight vector $(\mathbf{w}_i)$ obtained from solving a dimension $n+3$ linear equation system, where n is the number of unique pairs (y, x) with nonmissing z values in the dataset. Then the z value on a pair (y, x) can be interpolated by

$$z = w_1 \times f(y - y_1, x - x_1) + \cdots + w_n \times f(y - y_n, x - x_n) + w_{n+1} + w_{n+2} \times x + w_{n+3} \times y$$

where $f(y, x) = \sqrt{y^2 + x^2}$. `interp(thinplatespline)` is the default.

Shepard interpolation obtains the z value on a pair (y, x) from

$$z = (z_1 \times f(y - y_1, x - x_1) + \cdots + z_n \times \frac{f(y - y_n, x - x_n)}{\sum}$$

where $\sum$ is

$$\sum = f(y - y_1, x - x_1) + \cdots + f(y - y_n, x - x_n)$$

and $f(y, x) = 1/(x^2 + y^2)$. You specify `interp(shepard)` to use this method.

For the detailed formulas of thin-plate-spline and Shepard interpolation, see Press et al. (2007, 140–144).

Thin-plate-spline interpolation needs to solve a dimension $n + 3$ linear system, where n is the number of unique pairs (y, x) with nonmissing z value in the dataset. It becomes expensive when n becomes large. A rule-of-thumb number for choosing the thin-plate-spline method is n 1000.

Shepard interpolation is usually not as good as thin-plate-spline interpolation but is faster.

Method `none` plots data as is without any interpolation. Any grid cell with edge points containing a missing z value will be displayed using background color. If the dataset (z, y, x) is dense (that is, there are few missing grid points), `interp(none)` may be adequate.

Video example

Contour plots in Stata

Reference

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 2007. *Numerical Recipes: The Art of Scientific Computing*. 3rd ed. New York: Cambridge University Press.

Also see

[G-2] **graph twoway contourline** — Twoway contour-line plot

[G-2] **graph twoway area** — Twoway line plot with area shading

[G-2] **graph twoway rarea** — Range plot with area shading