Title

stata.com

***region_options* — Options for shading and outlining regions and controlling graph size**

Description Quick start Syntax Options Remarks and examples Also see

Description

The *region_options* set the size, margins, and color of the area in which the graph appears.

Quick start

Change the graph size to $y = 7$ inches and $x = 5$ inches

graph_command ..., ... `ysize(7) xsize(5)`

Change the background color to white

graph_command ..., ... `graphregion(color(white))`

Change the plot region color to light blue

graph_command ..., ... `plotregion(color(ltblue))`

Make the plot region margin, the space between the axes and actual plot region, large

graph_command ..., ... `plotregion(margin(large))`

Draw a black box of medium thickness around the plot region

graph_command ..., ... `plotregion(lcolor(black) lwidth(medium))`

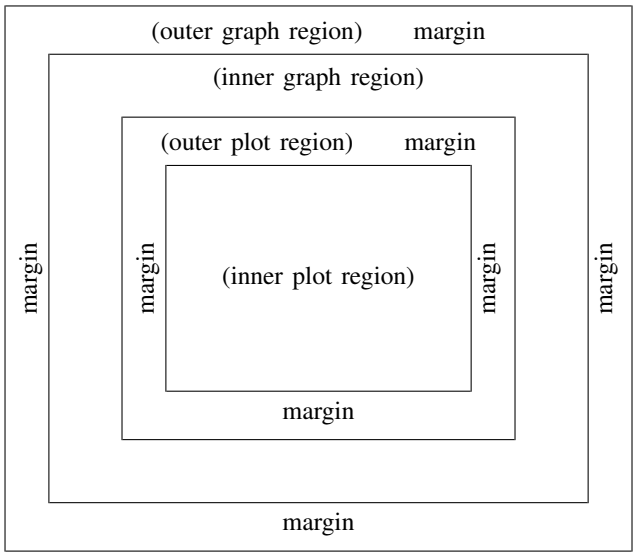
Syntax

region_options	Description
`ysize(`*graphsize*`)`	height of *available area*
`xsize(`*graphsize*`)`	width of *available area*
`graphregion(`*suboptions*`)`	attributes of *graph region*
`plotregion(`*suboptions*`)`	attributes of *plot region*

Options `ysize()` and `xsize()` are *unique*; options `graphregion()` and `plotregion()` are *merged-implicit*; see [G-4] **Concept: repeated options**.

suboptions	Description
`style(`*areastyle*`)`	overall style of outer region
`color(`*colorstyle*`)`	line and fill color and opacity of outer region
`fcolor(`*colorstyle*`)`	fill color and opacity of outer region
`lstyle(`*linestyle*`)`	overall style of outline
`lcolor(`*colorstyle*`)`	color and opacity of outline
`lwidth(`*linewidthstyle*`)`	thickness of outline
`lpattern(`*linepatternstyle*`)`	outline pattern (solid, dashed, etc.)
`lalign(`*linealignmentstyle*`)`	outline alignment (inside, outside, center)
`istyle(`*areastyle*`)`	overall style of inner region
`icolor(`*colorstyle*`)`	line and fill color and opacity of inner region
`ifcolor(`*colorstyle*`)`	fill color and opacity of inner region
`ilstyle(`*linestyle*`)`	overall style of outline
`ilcolor(`*colorstyle*`)`	color and opacity of outline
`ilwidth(`*linewidthstyle*`)`	thickness of outline
`ilpattern(`*linepatternstyle*`)`	outline pattern (solid, dashed, etc.)
`ilalign(`*linealignmentstyle*`)`	outline alignment (inside, outside, center)
`margin(`*marginstyle*`)`	margin between inner and outer regions

The *available area*, *graph region*, and *plot region* are defined



titles appear outside the borders of outer plot region

axes appear on the borders of the outer plot region

plot appears in inner plot region

Note: What are called the "graph region" and the "plot region" are sometimes the inner and sometimes the outer regions.

The available area and outer graph region are almost coincident; they differ only by the width of the border.

The borders of the outer plot or graph region are sometimes called the outer borders of the plot or graph region.

Options

`ysize(`*graphsize*`)` and `xsize(`*graphsize*`)` specify the height and width of the *available area*. *graphsize* is a numeric value followed by units `in`, `pt`, or `cm`. For example,

```
1in = 72pt = 2.54cm
```

When units are not specified, `in` is assumed. The defaults are usually `ysize(4)` and `xsize(5.5)`, but this, of course, is controlled by the scheme; see [G-4] **Schemes intro**. These two options can be used to control the overall aspect ratio of a graph. See *Controlling the aspect ratio* below.

The minimum *graphsize* is `1in`. The maximum *graphsize* is `100in`.

`graphregion(`*suboptions*`)` and `plotregion(`*suboptions*`)` specify attributes for the *graph region* and *plot region*.

Suboptions

`style(`*areastyle*`)` and `istyle(`*areastyle*`)` specify the overall style of the outer and inner regions. The other suboptions allow you to change the region's attributes individually, but `style()` and `istyle()` provide the starting points. See [G-4] ***areastyle*** for a list of choices.

`color(`*colorstyle*`)` and `icolor(`*colorstyle*`)` specify the color and opacity of the line used to outline the outer and inner regions; see [G-4] ***colorstyle*** for a list of choices.

`fcolor(`*colorstyle*`)` and `ifcolor(`*colorstyle*`)` specify the fill color and opacity for the outer and inner regions; see [G-4] ***colorstyle*** for a list of choices.

`lstyle(`*linestyle*`)` and `ilstyle(`*linestyle*`)` specify the overall style of the line used to outline the outer and inner regions, which includes its pattern (solid, dashed, etc.), thickness, and color. The other suboptions listed below allow you to change the line's attributes individually, but `lstyle()` and `ilstyle()` are the starting points. See [G-4] ***linestyle*** for a list of choices.

`lcolor(`*colorstyle*`)` and `ilcolor(`*colorstyle*`)` specify the color and opacity of the line used to outline the outer and inner regions; see [G-4] ***colorstyle*** for a list of choices.

`lwidth(`*linewidthstyle*`)` and `ilwidth(`*linewidthstyle*`)` specify the thickness of the line used to outline the outer and inner regions; see [G-4] ***linewidthstyle*** for a list of choices.

`lpattern(`*linepatternstyle*`)` and `ilpattern(`*linepatternstyle*`)` specify whether the line used to outline the outer and inner regions is solid, dashed, etc.; see [G-4] ***linepatternstyle*** for a list of choices. When `lpattern()` is specified, the line alignment is always `center`; thus, `lalign()` is ignored.

`lalign(`*linealignmentstyle*`)` and `ilalign(`*linealignmentstyle*`)` specify whether the line used to outline the outer and inner regions is drawn inside, is drawn outside, or is centered; see [G-4] ***linealignmentstyle*** for a list of choices.

`margin(`*marginstyle*`)` specifies the margin between the outer and inner regions; see [G-4] ***marginstyle***.

Remarks and examples

stata.com

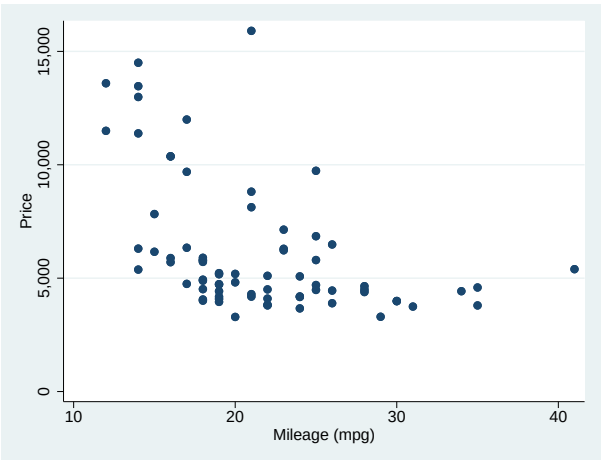
Remarks are presented under the following headings:

Setting the offset between the axes and the plot region

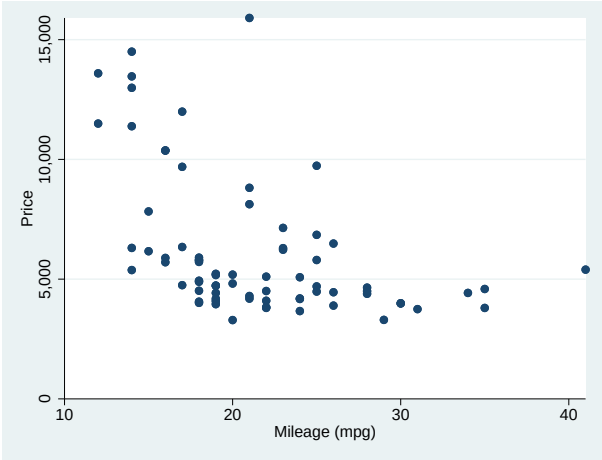
By default, most schemes (see [G-4] **Schemes intro**) offset the axes from the region in which the data are plotted. This offset is specified by `plotregion(margin(`*marginstyle*`))`; see [G-4] ***marginstyle***.

If you do not want the axes offset from the contents of the plot, specify `plotregion(margin(zero))`. Compare the next two graphs:

```
. use https://www.stata-press.com/data/r17/auto
(1978 automobile data)
. scatter price mpg
```



```
. scatter price mpg, plotr(m(zero))
```



Controlling the aspect ratio

Here we discuss controlling the overall aspect ratio of a graph. To control the aspect ratio of a plot region for `twoway`, `graph bar`, `graph box`, or `graph dot`, see [G-3] ***aspect_option***.

The way to control the aspect ratio of the overall graph is by specifying the `xsize()` or `ysize()` options. For instance, you draw a graph and find that the graph is too wide given its height. To address the problem, either increase `ysize()` or decrease `xsize()`. The usual defaults (which of course are determined by the scheme; see [G-4] **Schemes intro**) are `ysize(4)` and `xsize(5.5)`, so you might try

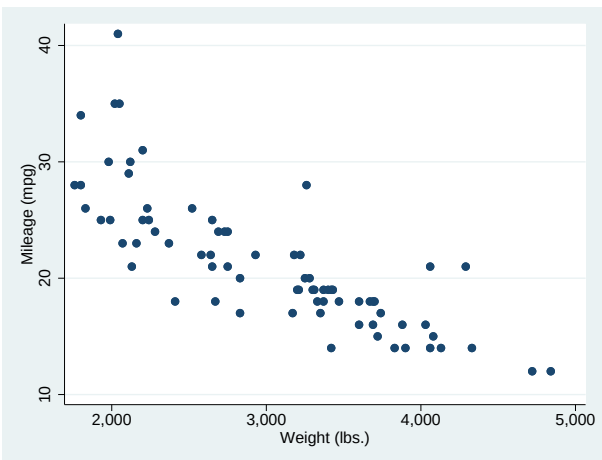
```
. graph ..., ... ysize(5)
```

or

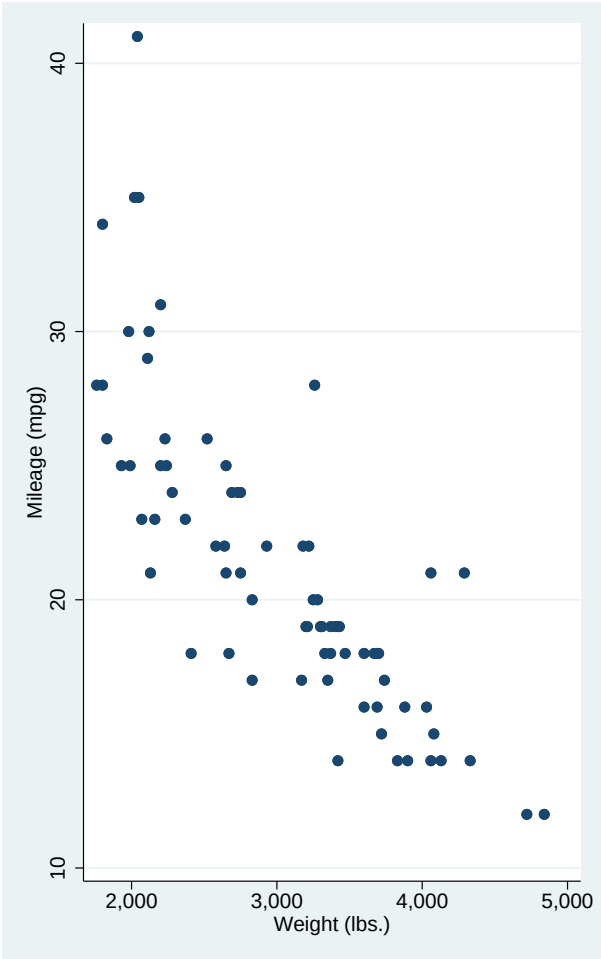
```
. graph ..., ... xsize(4.5)
```

For instance, compare

```
. scatter mpg weight
```

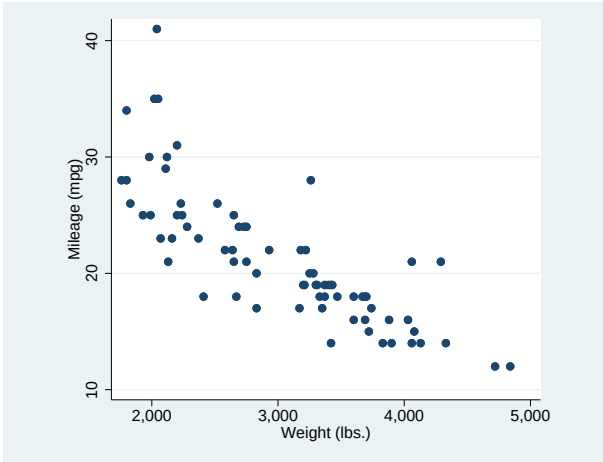


```
. scatter mpg weight, ysize(5)
```



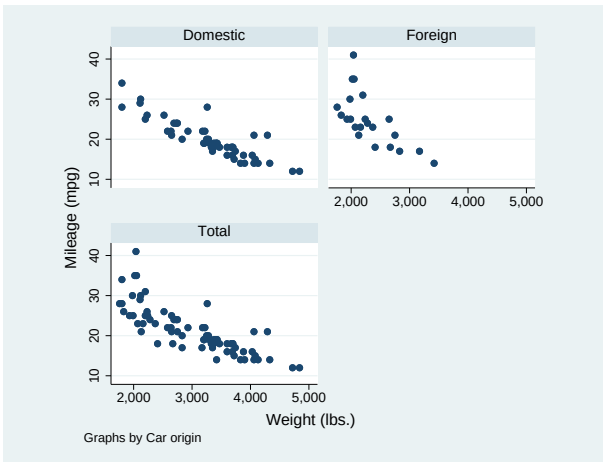
Another way to control the aspect ratio is to add to the outer margin of the *graph area*. This will keep the overall size of the graph the same while using less of the *available area*. For instance,

```
. scatter mpg weight, graphregion(margin(l+10 r+10))
```



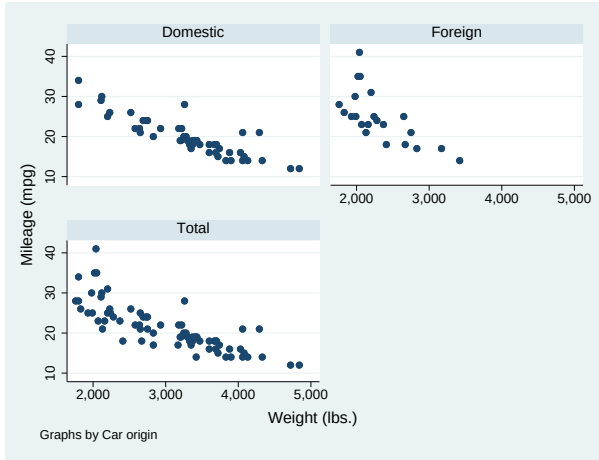
This method is especially useful when using `graph, by()`, but remember to specify the `graphregion(margin())` option inside the `by()` so that it affects the entire graph:

```
. scatter mpg weight, by(foreign, total graphr(m(l+10 r+10)))
```



Compare the above with

```
. scatter mpg weight, by(foreign, total)
```



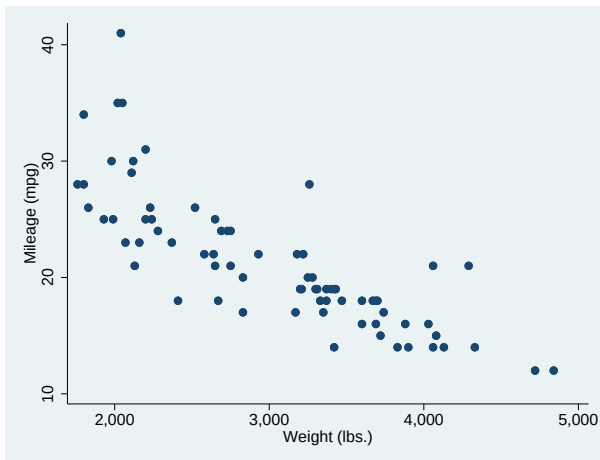
A similar, and often preferable, effect can be obtained by constraining the aspect ratio of the plot region itself; see [G-3] ***aspect_option***.

You do not have to get the aspect ratio or size right the first time you draw a graph; using `graph display`, you can change the aspect ratio of an already drawn graph—even a graph saved in a `.gph` file. See *Changing the size and aspect ratio* in [G-2] **graph display**.

Suppressing the border around the plot region

To eliminate the border around the plot region, specify `plotregion(style(none))`:

```
. use https://www.stata-press.com/data/r17/auto, clear
(1978 automobile data)
. scatter mpg weight, plotregion(style(none))
```

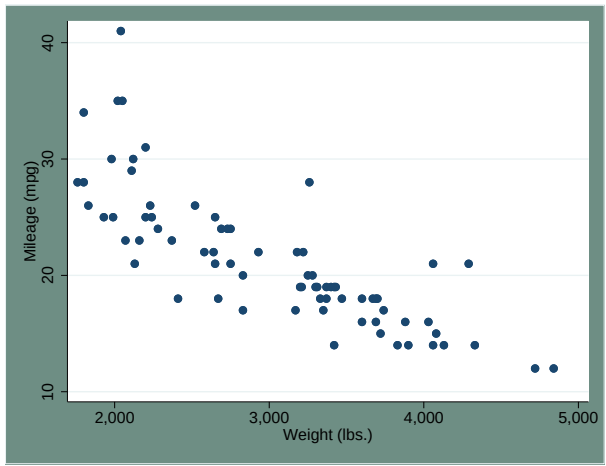


Setting background and fill colors

The background color of a graph is determined by default by the scheme you choose—see [G-4] **Schemes intro**—and is usually black or white, perhaps with a tint. Option `graphregion(fcolor(`*colorstyle*`))` allows you to override the scheme's selection. When doing this, choose a light background color for schemes that are naturally white and a dark background color for schemes that are naturally black, or you will have to type many options to make your graph look good.

Below we draw a graph, using a teal background:

```
. use https://www.stata-press.com/data/r17/auto, clear
(1978 automobile data)
. scatter mpg weight, graphregion(fcolor(teal))
```



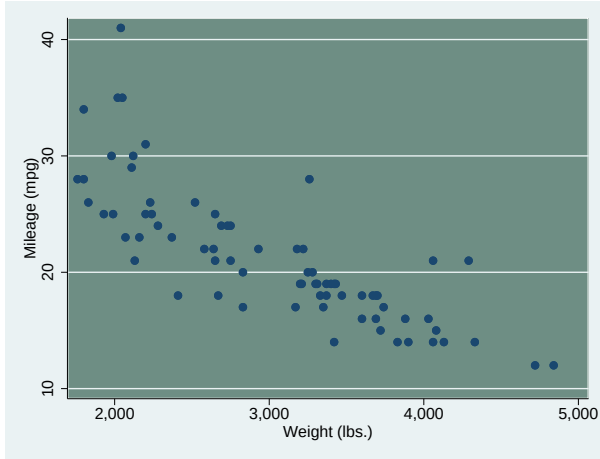
See [G-4] ***colorstyle*** for information on what you may specify inside the `graphregion(fcolor())` option.

In addition to `graphregion(fcolor())`, there are three other fill-color options:

`graphregion(ifcolor())`	fills *inner graph region*	←*of little use*
`plotregion(fcolor())`	fills *outer plot region*	←*useful*
`plotregion(ifcolor())`	fills *inner plot region*	←*could be useful*

`plotregion(fcolor())` is worth remembering. Below we make the plot region teal:

```
. scatter mpg weight, plotr(fcolor(teal))
```



The other two options—`graphregion(ifcolor())` and `plotregion(ifcolor())`—fill the *inner graph region* and *inner plot region*. Filling the *inner graph region* serves little purpose. Filling the *inner plot region*—which is the same as the *outer plot region* except that it omits the margin between the *inner plot region* and the axes—generally makes graphs appear too busy.

How graphs are constructed

`graph` works from the outside in, with the result that the dimensions of the *plot region* are what are left over.

`graph` begins with the *available area*, the size of which is determined by the `xsize()` and `ysize()` options. `graph` indents on all four sides by `graphregion(margin())`, so it defines the outer border of the *graph region*, the interior of which is the *inner graph region*.

Overall titles (if any) are now placed on the graph, and on each of the four sides, those titles are allocated whatever space they require. Next are placed any axis titles and labels, and they too are allocated whatever space necessary. That then determines the outer border of the *plot region* (or, more properly, the border of the *outer plot region*).

The axis (if any) is placed right on top of that border. `graph` now indents on all four sides by `plotregion(margin())`, and that determines the inner border of the plot region, meaning the border of the (*inner*) *plot region*. It is inside this that the data are plotted.

An implication of the above is that, if `plotregion(margin(zero))`, the axes are not offset from the region in which the data are plotted.

Now consider the lines used to outline the regions and the fill colors used to shade their interiors.

Starting once again with the *available area*, `graph` outlines its borders by using `graphregion(lstyle())`—which is usually `graphregion(lstyle(none))`—and fills the area with the `graphregion(fcolor())`.

`graph` now moves to the inner border of the *graph region*, outlines it using `graphregion(ilstyle())`, and fills the *graph region* with `graphregion(ifcolor())`.

`graph` moves to the outer border of the *plot region*, outlines it using `plotregion(lstyle())`, and fills the *outer plot region* with `plotregion(fcolor())`.

Finally, `graph` moves to the inner border of the *plot region*, outlines it using `plotregion(ilstyle())`, and fills the (*inner*) *plot region* with `plotregion(ifcolor())`.

Also see

[G-4] ***areastyle*** — Choices for look of regions

[G-4] ***colorstyle*** — Choices for color

[G-4] ***linealignmentstyle*** — Choices for whether outlines are inside, outside, or centered

[G-4] ***linepatternstyle*** — Choices for whether lines are solid, dashed, etc.

[G-4] ***linestyle*** — Choices for overall look of lines

[G-4] ***linewidthstyle*** — Choices for thickness of lines

[G-4] ***marginstyle*** — Choices for size of margins