

schemes intro — Introduction to schemes
[Description](#)[Syntax](#)[Remarks and examples](#)[Also see](#)

Description

A scheme specifies the overall look of the graph.

`set scheme` sets the default scheme; see [G-2] [set scheme](#) for more information about this command.

Option `scheme()` specifies the graphics scheme to be used with this particular `graph` command without changing the default.

Syntax

```
set scheme schemename [, permanently ]
```

```
graph ... [, ...scheme(schemename) ... ]
```

<i>schemename</i>	Foreground	Background	Description
s2color	color	white	factory setting
s2mono	monochrome	white	s2color in monochrome
s2gcolor	color	white	used in the Stata manuals
s2manual	monochrome	white	s2gcolor in monochrome
s2gmanual	monochrome	white	previously used in the [G] manual
s1rcolor	color	black	a plain look on black background
s1color	color	white	a plain look
s1mono	monochrome	white	a plain look in monochrome
s1manual	monochrome	white	s1mono but smaller
economist	color	white	<i>The Economist</i> magazine
sj	monochrome	white	<i>Stata Journal</i>

See [G-4] [scheme s1](#), [G-4] [scheme s2](#), [G-4] [scheme economist](#), and [G-4] [scheme sj](#).

Other *schemenames* may be available; type

```
. graph query, schemes
```

to obtain the complete list of schemes installed on your computer.

Remarks and examples

Remarks are presented under the following headings:

The role of schemes
Finding out about other schemes
Setting your default scheme
The scheme is applied at display time
Background color
Foreground color
Obtaining new schemes
Examples of schemes

See `help scheme files` for a discussion of how to create your own schemes.

The role of schemes

When you type, for instance,

```
. scatter yvar xvar
```

results are the same as if you typed

```
. scatter yvar xvar, scheme(your_default_scheme)
```

If you have not used the `set scheme` command to change your default scheme, *your_default_scheme* is `s2color`.

The scheme specifies the overall look for the graph, and by that we mean just about everything you can imagine. It determines such things as whether *y* axes are on the left or the right, how many values are by default labeled on the axes, and the colors that are used. In fact, almost every statement made in other parts of this manual stating how something appears, or the relationship between how things appear, must not be taken too literally. How things appear is in fact controlled by the scheme:

- In [G-4] *symbolstyle*, we state that markers—the ink that denotes the position of points on a plot—have a default size of `msize(medium)` and that small symbols have a size of `msize(small)`. That is generally true, but the size of the markers is in fact set by the scheme, and a scheme might specify different default sizes.
- In [G-3] *axis_choice_options*, we state that when there is one “*y* axis”, which appears on the left, and when there are two, the second appears on the right. What is in fact true is that where axes appear is controlled by the scheme and that most schemes work the way described. Another scheme named `economist`, however, displays things differently.
- In [G-3] *title_options*, we state where the titles, subtitles, etc., appear, and we provide a diagram so that there can be no confusion. But where the titles, subtitles, etc., appear is in fact controlled by the scheme, and what we have described is what is true for the scheme named `s2color`.

The list goes on and on. If it has to do with the look of the result, it is controlled by the scheme.

To understand just how much difference the scheme can make, you should type

```
. scatter yvar xvar, scheme(economist)
```

`scheme(economist)` specifies a look similar to that used by *The Economist* magazine (<http://www.economist.com>), whose graphs we believe to be worthy of emulation. By comparison with the `s2color` scheme, the `economist` scheme moves *y* axes to the right, makes titles left justified, defaults grid lines to be on, sets a background color, and moves the note to the top right and expects it to be a number.

Finding out about other schemes

A list of schemes is provided in the syntax diagram above, but do not rely on the list being up to date. Instead, type

```
. graph query, schemes
```

to obtain the complete list of schemes installed on your computer.

Try drawing a few graphs with each:

```
. graph ..., ... scheme(schemename)
```

Setting your default scheme

If you want to set your default scheme to, say, `economist`, type

```
. set scheme economist
```

The `economist` scheme will now be your default scheme for the rest of this session, but the next time you use Stata, you will be back to using your old default scheme. If you type

```
. set scheme economist, permanently
```

`economist` will become your default scheme both the rest of this session and in future sessions.

If you want to change your scheme back to `s2color`—the default scheme in Stata as originally shipped—type

```
. set scheme s2, permanently
```

See [\[G-2\] set scheme](#).

The scheme is applied at display time

Say that you type

```
. graph mpg weight, saving(mygraph)
```

to create and save the file `mygraph.gph` (see [\[G-3\] saving_option](#)). If later you redisplay the graph by typing

```
. graph use mygraph
```

the graph will reappear as you originally drew it. It will be displayed using the same scheme with which it was originally drawn, regardless of your current `set scheme` setting. If you type

```
. graph use mygraph, scheme(economist)
```

the graph will be displayed using the `economist` scheme. It will be the same graph but will look different. You can change the scheme with which a graph is drawn beforehand, on the original `graph` command, or later.

Background color

In the table at the beginning of the entry, we categorized the background color as being white or black, although actually what we mean is light or dark because some of the schemes set background tinting. We mean that “white” background schemes are suitable for printing. Printers (both the mechanical ones and the human ones) prefer that you avoid dark backgrounds because of the large amounts of ink required and the corresponding problems with bleed-through. On the other hand, dark backgrounds look good on monitors.

In any case, you may change the background color of a scheme by using the *region_options* `graphregion(fcolor())`, `graphregion(icolor())`, `plotregion(fcolor())`, and `plotregion(icolor())`; see [G-3] *region_options*. When overriding the background color, choose light colors for schemes that naturally have white backgrounds and dark colors for regions that naturally have black backgrounds.

Schemes that naturally have a black background are by default printed in monochrome. See [G-2] [set printcolor](#) if you wish to override this.

If you are producing graphs for printing on white paper, we suggest that you choose a scheme with a naturally white background.

Foreground color

In the table at the beginning of this entry, we categorized the foreground as being color or monochrome. This refers to whether lines, markers, fills, etc., are presented by default in color or monochrome. Regardless of the scheme you choose, you can specify options such as `mcolor()` and `lcolor()`, to control the color for each item on the graph.

Just because we categorized the foreground as monochrome, this does not mean you cannot specify colors in the options.

Obtaining new schemes

Your copy of Stata may already have schemes other than those documented in this manual. To find out, type

```
. graph query, schemes
```

Also, new schemes are added and existing schemes updated along with all the rest of Stata, so if you are connected to the Internet, type

```
. update query
```

and follow any instructions given; see [R] [update](#).

Finally, other users may have created schemes that could be of interest to you. To search the Internet, type

```
. search scheme
```

From there, you will be able to click to install any schemes that interest you; see [R] [search](#).

Once a scheme is installed, which can be determined by verifying that it appears in the list shown by

```
. graph query, schemes
```

you can use it with the `scheme()` option

```
. graph ... , ... scheme(newscheme)
```

or you can set it as your default, temporarily

```
. set scheme newscheme
```

or permanently

```
. set scheme newscheme, permanently
```

Examples of schemes

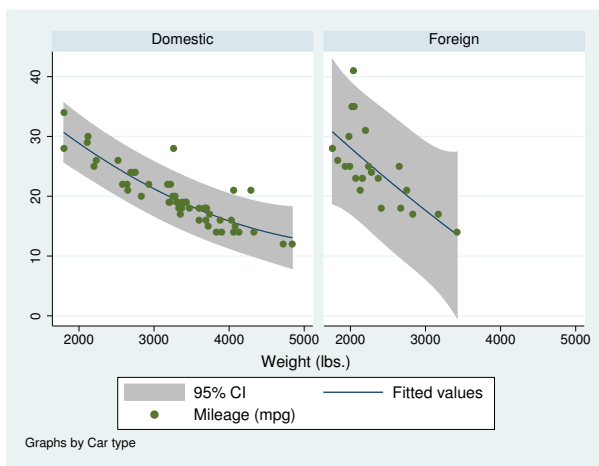


Figure 1. `s2color`

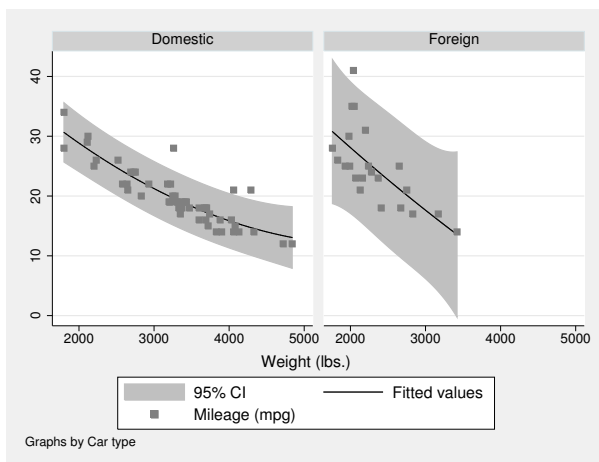


Figure 2. `s2mono`

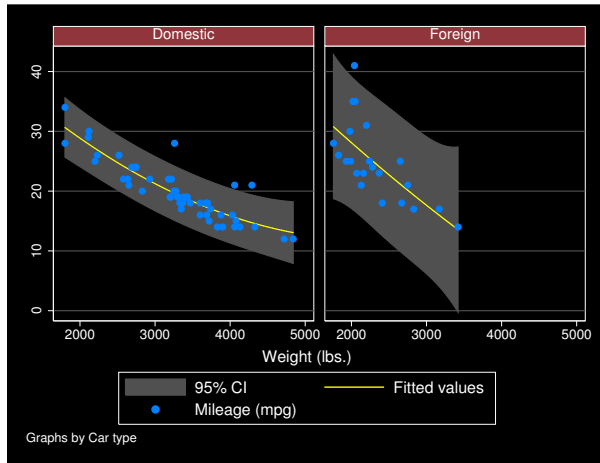


Figure 3. s1rcolor

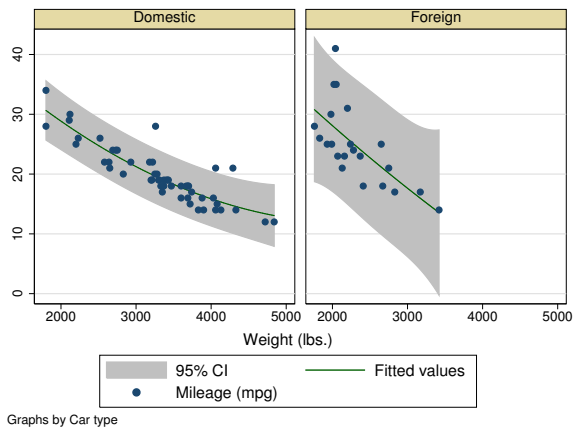
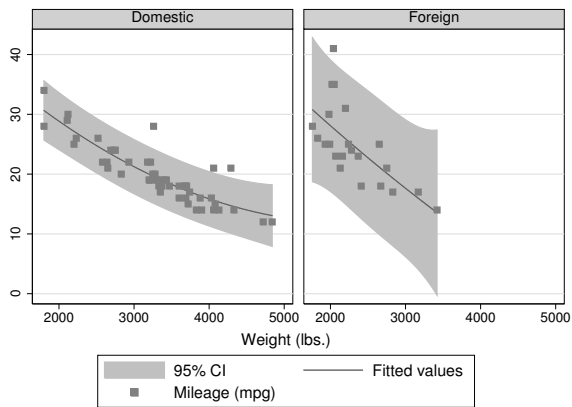


Figure 4. s1color



Graphs by Car type

Figure 5. s1mono

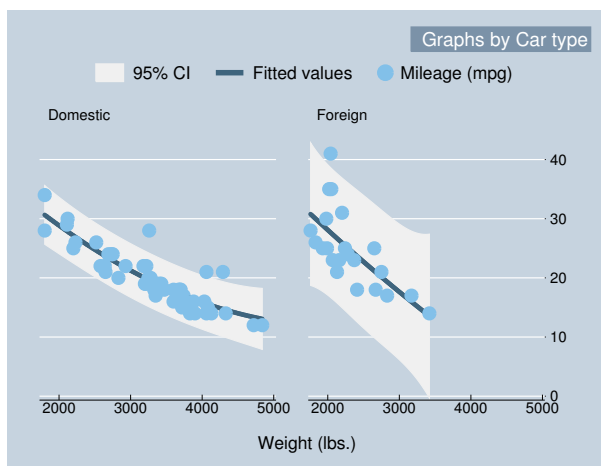


Figure 6. economist

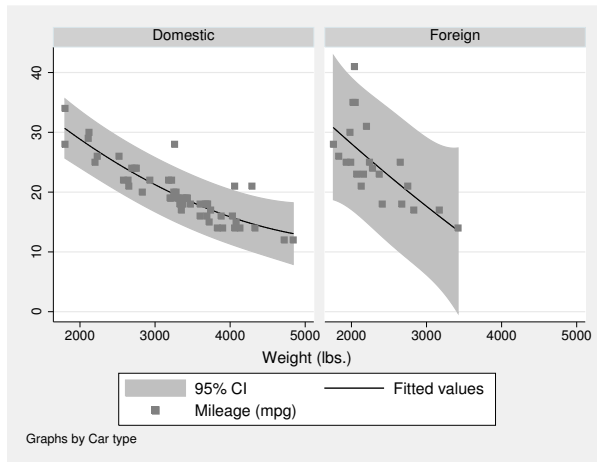


Figure 7. sj

Also see

[G-2] **set scheme** — Set default scheme

[G-3] **scheme_option** — Option for specifying scheme

[G-4] **scheme economist** — Scheme description: economist

[G-4] **scheme s1** — Scheme description: s1 family

[G-4] **scheme s2** — Scheme description: s2 family

[G-4] **scheme sj** — Scheme description: sj