

A Visual Guide to Stata Graphics

Third Edition

MICHAEL N. MITCHELL



A Stata Press Publication
StataCorp LP
College Station, Texas



© Copyright © 2004, 2008, 2012 by StataCorp LP
All rights reserved. First edition 2004
Second edition 2008
Third edition 2012

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845

Typeset in L^AT_EX 2_ε

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-106-4

ISBN-13: 978-1-59718-106-8

Library of Congress Control Number: 2011942526

No part of this book may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means—electronic, mechanical, photocopy, recording, or otherwise—without the prior written permission of StataCorp LP.

Stata, **STATA**, Stata Press, Mata, **MATA**, and NetCourse are registered trademarks of StataCorp LP.

Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations.

L^AT_EX 2_ε is a trademark of the American Mathematical Society.

(Pages omitted)

Contents

| | | |
|----------|---|-----------|
| | Dedication | v |
| | Acknowledgments | vii |
| | Preface to the Third Edition | xiii |
| | Preface to the Second Edition | xv |
| | Preface to the First Edition | xvii |
| 1 | Introduction | 1 |
| | 1.1 Using this book | 1 |
| | 1.2 Types of Stata graphs | 5 |
| | 1.3 Schemes | 15 |
| | 1.4 Options | 20 |
| | 1.5 Building graphs | 29 |
| 2 | Editor | 35 |
| | 2.1 Overview of the Graph Editor | 36 |
| | 2.2 Object Browser | 43 |
| | 2.3 Modifying objects | 47 |
| | 2.4 Adding objects | 54 |
| | 2.5 Moving objects | 62 |
| | 2.6 Hiding and showing objects | 73 |
| | 2.7 Locking and unlocking objects | 76 |
| | 2.8 Using the Graph Recorder | 78 |
| | 2.9 Graph Editor versus Stata commands | 82 |
| 3 | Twoway graphs | 89 |
| | 3.1 Scatterplots | 89 |
| | 3.2 Regression fits and splines | 106 |
| | 3.3 Regression confidence interval fits | 108 |
| | 3.4 Line plots | 112 |
| | 3.5 Area plots | 119 |

| | | |
|----------|--|------------|
| 3.6 | Bar plots | 121 |
| 3.7 | Range plots | 123 |
| 3.8 | Distribution plots | 133 |
| 3.9 | Contour plots | 141 |
| 3.10 | Options | 147 |
| 3.11 | Overlaying plots | 152 |
| 4 | Scatterplot matrix graphs | 161 |
| 4.1 | Marker options | 161 |
| 4.2 | Controlling axes | 164 |
| 4.3 | Matrix options | 169 |
| 4.4 | Graphing by groups | 170 |
| 5 | Bar graphs | 173 |
| 5.1 | Y variables | 173 |
| 5.2 | Graphing bars over groups | 177 |
| 5.3 | Options for controlling gaps between bars | 183 |
| 5.4 | Options for sorting bars | 188 |
| 5.5 | Controlling the categorical axis | 193 |
| 5.6 | Legends and labeling bars | 201 |
| 5.7 | Controlling the y axis | 213 |
| 5.8 | Changing the look of bars | 218 |
| 5.9 | Graphing by groups | 221 |
| 6 | Box plots | 227 |
| 6.1 | Specifying variables and groups | 227 |
| 6.2 | Options for controlling gaps between boxes | 233 |
| 6.3 | Options for sorting boxes | 235 |
| 6.4 | Controlling the categorical axis | 238 |
| 6.5 | Controlling legends | 244 |
| 6.6 | Controlling the y axis | 250 |
| 6.7 | Changing the look of boxes | 254 |
| 6.8 | Graphing by groups | 258 |
| 7 | Dot plots | 263 |
| 7.1 | Specifying variables and groups | 263 |
| 7.2 | Options for controlling gaps between dots | 268 |

| | | |
|-----------|--|------------|
| 7.3 | Options for sorting dots | 271 |
| 7.4 | Controlling the categorical axis | 274 |
| 7.5 | Controlling legends | 277 |
| 7.6 | Controlling the y axis | 279 |
| 7.7 | Changing the look of dot rulers | 282 |
| 7.8 | Graphing by groups | 285 |
| 8 | Pie charts | 289 |
| 8.1 | Types of pie charts | 289 |
| 8.2 | Sorting pie slices | 291 |
| 8.3 | Changing the look and color and exploding pie slices | 293 |
| 8.4 | Slice labels | 295 |
| 8.5 | Controlling legends | 300 |
| 8.6 | Graphing by groups | 303 |
| 9 | Options available for most graphs | 307 |
| 9.1 | Changing the look of markers | 307 |
| 9.2 | Creating and controlling marker labels | 320 |
| 9.3 | Connecting points and markers | 323 |
| 9.4 | Setting and controlling axis titles | 327 |
| 9.5 | Setting and controlling axis labels | 330 |
| 9.6 | Controlling axis scales | 339 |
| 9.7 | Selecting an axis | 343 |
| 9.8 | Graphing by groups | 346 |
| 9.9 | Controlling legends | 361 |
| 9.10 | Adding text to markers and positions | 374 |
| 9.11 | Options for text and textboxes | 379 |
| 9.12 | More options controlling the display of text | 388 |
| 10 | Standard options available for all graphs | 395 |
| 10.1 | Creating and controlling titles | 395 |
| 10.2 | Using schemes to control the look of graphs | 401 |
| 10.3 | Sizing graphs and their elements | 404 |
| 10.4 | Changing the look of graph regions | 406 |
| 11 | Styles for changing the look of graphs | 411 |
| 11.1 | Angles | 411 |

| | | |
|-----------|--|------------|
| 11.2 | Colors | 412 |
| 11.3 | Clock position | 414 |
| 11.4 | Compass direction | 415 |
| 11.5 | Connecting points | 416 |
| 11.6 | Line patterns | 420 |
| 11.7 | Line width | 422 |
| 11.8 | Margins | 423 |
| 11.9 | Marker size | 425 |
| 11.10 | Orientation | 426 |
| 11.11 | Marker symbols | 427 |
| 11.12 | Text size | 428 |
| 12 | Appendix | 431 |
| 12.1 | Overview of statistical graph commands | 431 |
| 12.2 | Common options for statistical graphs | 438 |
| 12.3 | The marginsplot command | 444 |
| 12.4 | Saving, redisplaying, and combining graphs | 456 |
| 12.5 | More examples: Putting it all together | 466 |
| 12.6 | Common mistakes | 477 |
| 12.7 | Customizing schemes | 479 |
| 12.8 | Online supplements | 482 |
| | Subject index | 485 |

Preface to the Third Edition

This third edition updates the second edition of this book, reflecting new features available in Stata version 12. Since version 10, Stata has added several new graphical features, including a command for creating contour plots, options that give you greater control over the display of text, and the ability to create graphs from the results of the `margins` command. Additional sections have been added to this third edition that illustrate these new features.

A new section has been added that illustrates the use of the `twoway contour` command; see [Twoway: Contour](#) (141). You can see [Options: Text Display](#) (388) for information about how to specify symbols, subscripts, and superscripts, as well as how to display text in bold or italics; this section also describes how you can display text using different fonts. A new section has also been added that describes how you can customize graphs created using the `marginsplot` command; see [Appendix: Marginsplot](#) (444).

This third edition also includes minor updates here and there to bring the text up to date for use with Stata version 12.

Simi Valley, California
December 2011

(Pages omitted)

1 Introduction

This chapter begins by briefly telling you about the organization of this book and giving you tips to help you use it most effectively. The next section gives a short overview of the different kinds of Stata graphs that will be examined in this book, and that section is followed by an overview of the different kinds of schemes that will be used for showing the graphs in this book. The fourth section illustrates the structure of options in Stata graph commands. In a sense, the second, third, and fourth sections of this chapter are a thumbnail preview of the entire book, showing the types of graphs covered, how you can control their overall look, and the general structure of options used within those graphs. The final section is about the process of creating graphs.

1.1 Using this book

I hope that you are eager to start reading this book but will take just a couple of minutes to read this section to get some suggestions that will make the book more useful to you. First, there are many ways you might read this book, but perhaps I can suggest some tips:

- Read this chapter before reading the other chapters, as it provides key information that will make the rest of the book more understandable.
- Although you might read a traditional book cover to cover, this book has been written so that the chapters stand on their own. You should feel free to dive into any chapter or section of any chapter.
- Sometimes you might find it useful to visually scan the graphs rather than to read. I think this is a good way to familiarize yourself with the kinds of features available in Stata graphs. If a certain feature catches your eye, you can stop and see the command that made the graph and even read the text explaining the command.
- Likewise, you might scan a chapter just by looking at the graphs and the part of the command in red, which is the part of the command highlighted in that graph. For example, scanning the chapter on bar charts in this way would quickly familiarize you with the kinds of features available for bar graphs and would show you how to obtain those features.

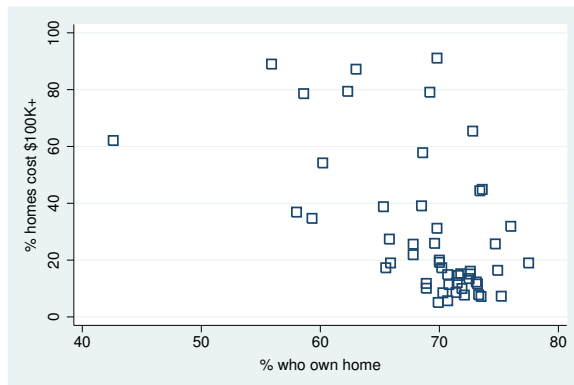
The right margin contains what I call the *Visual Table of Contents*. It is a useful tool for quickly finding the information you seek. I frequently use the *Visual Table of Contents* to cross-reference information within the book. By design, Stata graphs share many common features. For example, you use the same kinds of options to control legends across different types of graphs. It would be repetitive to go into detail about legends for bar charts, box

plots, and so on. Within each kind of graph, legends are briefly described and illustrated, but the details are described in the *Options* chapter in the section titled *Legend*. This is cross-referenced in the book by saying something like “for more details, see `Options: Legend (361)`”, indicating that you should look to the *Visual Table of Contents* and thumb to the *Options* chapter and then to the *Legend* section, which begins on page 361.


Sometimes it may take an extra cross-reference to get the information you need. Say that you want to make the y -axis title large for a bar chart by using the `ytitle()` option, so you first consult `Bar: Y-axis (213)`. This gives you some information about using `ytitle()`, but then that section refers you to `Options: Axis titles (327)`, where more details about axis titles are described. This section then refers you to `Options: Textboxes (379)` for more complete details about options to control the display of text. That section shows more details but then refers to `Styles: Textsize (428)`, where all the possible text sizes are described. I know this sounds like a lot of jumping around, but I hope that it feels more like drilling down for more detail, that you feel you are in control of the level of detail that you want, and that the *Visual Table of Contents* eases the process of getting the additional details.

Most pages of this book have three graphs per page, with each graph being composed of the graph itself, the command that produced it, and some descriptive text. An example is shown below, followed by some points to note.

```
graph twoway scatter propval100 ownhome, msymbol(Sh)
```







Here we use the `msymbol()` (marker symbol) option to make the symbols large hollow squares; see `Options: Markers (307)` for more details. The `graph twoway` portion of the command is optional.

 Double-click on any of the markers and change the **Symbol** to **Hollow square**.

Uses allstates.dta & scheme vg_s2c

- The command itself is displayed in a **typewriter font**, and the salient part of the command (i.e., `msymbol(Sh)`) is in **this color**—both in the command and when referenced in the descriptive text.
- When commands or parts of commands are given in the descriptive text (e.g., `graph twoway`), they are displayed in the **typewriter font**.
- Many of the descriptions contain cross-references, for example, `Options: Markers (307)`, which means to flip to the *Options* chapter and then to the section *Markers*. Equivalently, go to page 307.
- The names of some options are shorthand for two or more words that are sometimes explained; for instance, “we use the `msymbol()` (marker symbol) option to make ...”.

- Many examples include more instructions describing how the Stata Graph Editor can be used to accomplish the same customization as illustrated in the command (in this example, how the Graph Editor can be used to obtain the equivalent of `msymbol(Sh)`). The  icon indicates that the instructions that follow apply to the use of the Graph Editor. The instructions assume that you have run the command *omitting* the highlighted portion of the command (e.g., omitting `msymbol(Sh)`) and that you have started the Graph Editor. The Graph Editor can be started in one of three ways: 1) by selecting **File** and then **Start Graph Editor** from the Graph window menu, 2) by clicking the Start Graph Editor  icon in the Graph window toolbar, or 3) by right-clicking on the graph and selecting **Start Graph Editor**. Once the Graph Editor is started, you can follow the instructions given (e.g., you can double-click on any of the markers, and in the dialog box that appears, you can then change the setting for the *Symbol* option to *Hollow square*). See [Editor \(35\)](#) for more details about using the Graph Editor.
- The descriptive text always concludes by telling you the name of the data file and scheme used for making the graph. Here the data file was `allstates.dta`, and the scheme was `vg_s2c.scheme`. You can read the data file over the Internet by using the `vguse` command, which is added to Stata when you install the online supplements; see [Appendix: Online supplements \(482\)](#). If you are connected to the Internet and your Stata is fully up to date, you can simply type `vguse allstates` to use that file over the Internet, and you can run the graph command shown to create the graph.
- Sometimes there is not enough space to describe the command as well as describe how to use the Graph Editor to accomplish the customization illustrated. In such cases, the description will conclude with “ See the next graph”. The descriptive text for the next example will begin with the  icon and will be dedicated to illustrating how to use the Graph Editor for that particular customization.

If you want your graphs to look like the ones in the book, you can display them using the same schemes. See [Appendix: Online supplements \(482\)](#) for information about how to download the schemes used in this book. Once you have downloaded the schemes, you can then type the following commands in the Stata Command window:

```
. set scheme vg_s2c
. vguse allstates
. graph twoway scatter propval100 ownhome, msymbol(Sh)
```

After you issue the `set scheme vg_s2c` command, subsequent graph commands will show graphs with the `vg_s2c` scheme. You could also add the `scheme(vg_sc2)` option to the graph command to specify that the scheme be used just for that graph; for example,

```
. graph twoway scatter propval100 ownhome, msymbol(Sh) scheme(vg_s2c)
```

Generally, all commands and options are provided in their complete form. Commands and options are usually not abbreviated. However, for purposes of typing, you may want to use abbreviations. The previous example could have been abbreviated to

```
. gr tw sc propval100 ownhome, m(Sh)
```

The `gr` could have been omitted, leaving

```
. tw sc propval100 ownhome, m(Sh)
```

The `tw` also could have been omitted, leaving

```
. sc propval100 ownhome, m(Sh)
```

For guidance on appropriate abbreviations, consult [G-2] **graph**.

This book has been written based on the features available in Stata version 12.0. In the future, Stata may evolve to make the behavior of some of these commands change. If this happens, you can use the `version` command to make Stata run the graph commands as though they were run under version 12.0. For example, if you were running Stata version 13.0 but wanted a graph command to run as though you were running Stata 12.0, you could type

```
. version 12.0: graph twoway scatter propval100 ownhome
```

and the command would be executed as if you were running version 12.0. Or, perhaps you want a command to run as it did under Stata 11.2, you would then type

```
. version 11.2: graph twoway scatter propval100 ownhome
```

This book has a number of associated online resources to complement the book. **Appendix: Online supplements (482)** has more information about these online resources and how to access them. I strongly suggest that you install the online supplements, which make it easier to run the examples from the book. To install the supplemental programs, schemes, and help files, type from within Stata

```
. net from http://www.stata-press.com/data/vgsg  
. net install vgsg
```

For an overview of what you have installed, type `help vgsg` within Stata. Then, with the `vguse` command, you can use any dataset from the book. Likewise, all the custom schemes used in the book will be installed into your copy of Stata, and you can use them to display the graphs, as described earlier in this section.

Finally, I would like to emphasize that the goal of this book is to help you learn and use the Stata graph commands and the Graph Editor for the purposes of creating graphs in Stata. I assume that you know the kind of graph you want to create and that you are turning to this book for advice on how to make that graph. I don't provide guidance on how to select the right kind of graph for visualizing your data or the merits of one graphical method over another. For such guidance, I would refer readers to books such as *The Visual Display of Quantitative Information, Second Edition* by Edward R. Tufte and *Visualizing Data* by William S. Cleveland, as well as your favorite statistical book.

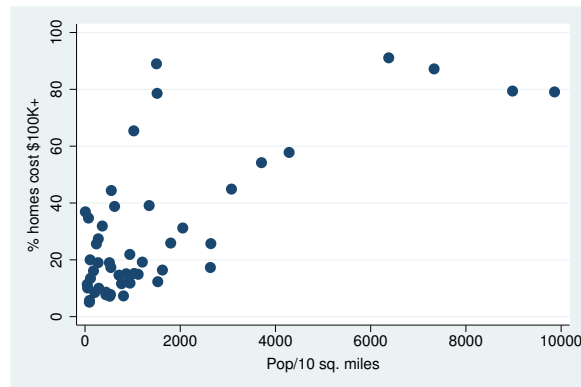
1.2 Types of Stata graphs

Stata has a wide variety of graph types. This section introduces the types of graphs Stata produces, and it covers twoway plots (including scatterplots, line plots, fit plots, fit plots with confidence intervals, area plots, bar plots, range plots, and distribution plots), scatterplot matrices, bar charts, box plots, dot plots, and pie charts. Let's begin by exploring the variety of twoway plots that can be created with `graph twoway`. For this introduction, they are combined into six families of related plots: scatterplots and fit plots, line plots, area plots, bar plots, range plots, and distribution plots. Now let's turn to scatterplots and fit plots.

```
graph twoway scatter propval100 popden
```

Here is a basic scatterplot. The variable `propval100` is placed on the y axis, and `popden` is placed on the x axis. See [Twoway: Scatter \(89\)](#) for more details about these kinds of plots.

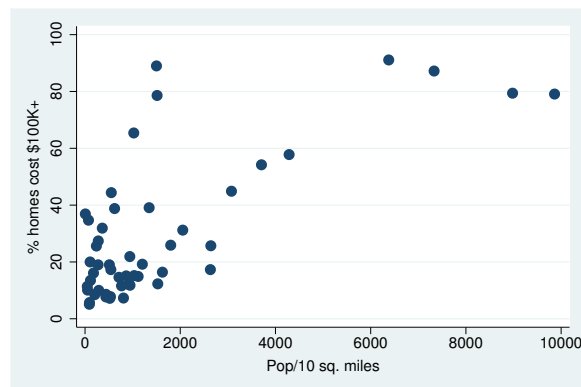
Uses `allstates.dta` & `scheme vg_s2c`



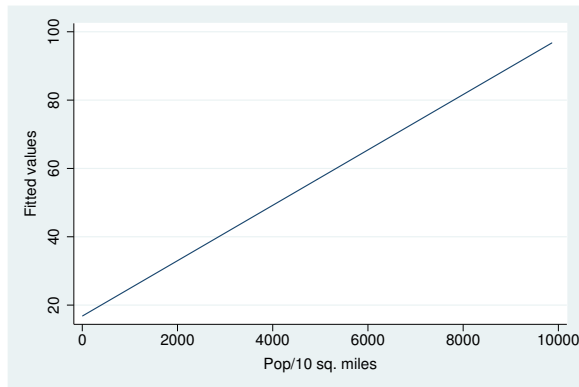
```
twoway scatter propval100 popden
```

We can start the previous command with just `twoway`, and Stata understands that this is shorthand for `graph twoway`.

Uses `allstates.dta` & `scheme vg_s2c`



```
twoway lfit propval100 popden
```



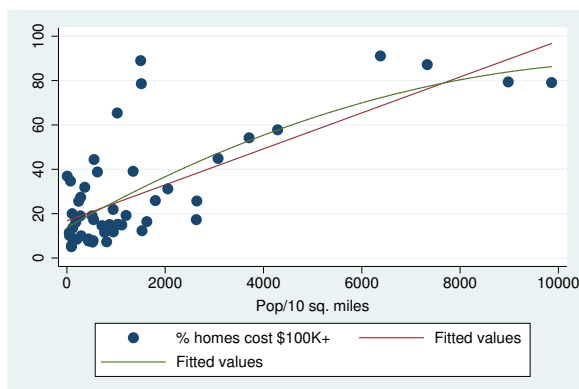
We now make a linear fit (`lfit`) line predicting `propval100` from `popden`. See [Twoway: Fit \(106\)](#) for more information about these kinds of plots. Uses `allstates.dta` & `scheme vg_s2c`

```
twoway (scatter propval100 popden) (lfit propval100 popden)
```



Stata allows us to overlay `twoway` graphs. In this example, we make a classic plot showing a scatterplot overlaid with a fit line by using the `scatter` and `lfit` commands. For more details about overlaying graphs, see [Twoway: Overlaying \(152\)](#). Uses `allstates.dta` & `scheme vg_s2c`

```
twoway (scatter propval100 popden) (lfit propval100 popden)
      (qfit propval100 popden)
```

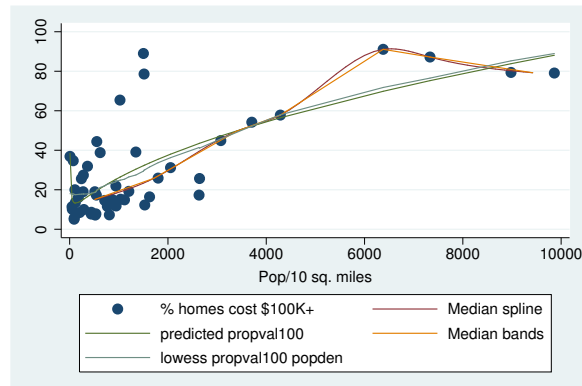


The ability to combine `twoway` plots is not limited to overlaying just two plots; we can overlay multiple plots. Here we overlay a scatterplot (`scatter`) with a linear fit (`lfit`) line and a quadratic fit (`qfit`) line. Uses `allstates.dta` & `scheme vg_s2c`

```
twoway (scatter propval100 popden) (mspline propval100 popden)
      (fpfit propval100 popden) (mband propval100 popden)
      (lowess propval100 popden)
```

Stata has other kinds of fit methods in addition to linear and quadratic fits. This example includes a median spline (`mspline`), fractional polynomial fit (`fpfit`), median band (`mband`), and lowess (`lowess`). For more details, see [Twoway: Fit \(106\)](#).

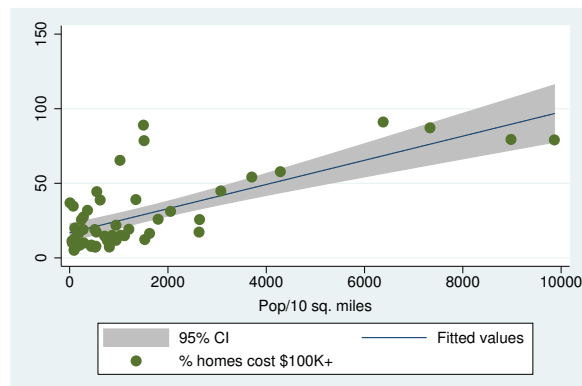
Uses `allstates.dta` & `scheme vg_s2c`



```
twoway (lfitci propval100 popden) (scatter propval100 popden)
```

In addition to being able to plot a fit line, we can plot a linear fit line with a confidence interval by using the `lfitci` command. We also overlay the linear fit and confidence interval with a scatterplot. See [Twoway: CI fit \(108\)](#) for more information about fit lines with confidence intervals.

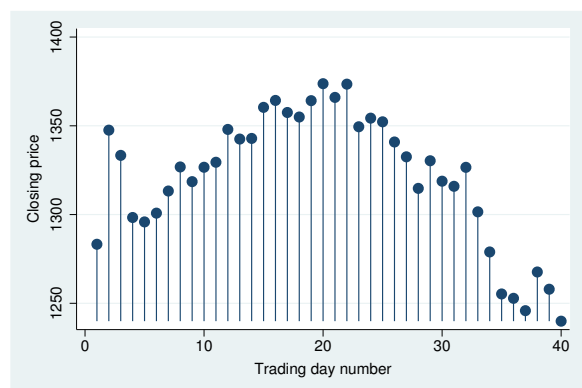
Uses `allstates.dta` & `scheme vg_s2c`



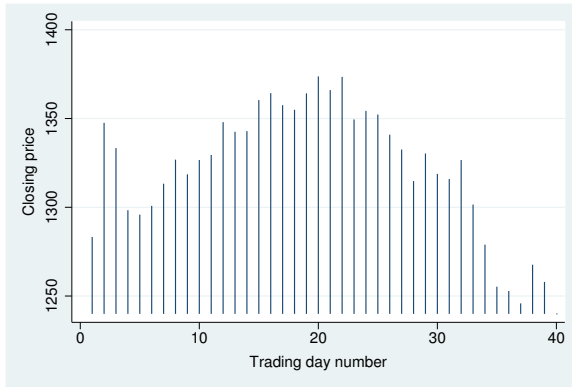
```
twoway dropline close tradeday
```

This `dropline` graph shows the closing prices of the S&P 500 by trading day for the first 40 days of 2001. A `dropline` graph is like a scatterplot because each data point is shown with a marker, but a dropline for each marker is shown as well. For more details, see [Twoway: Scatter \(89\)](#).

Uses `spjanfeb2001.dta` & `scheme vg_s2c`



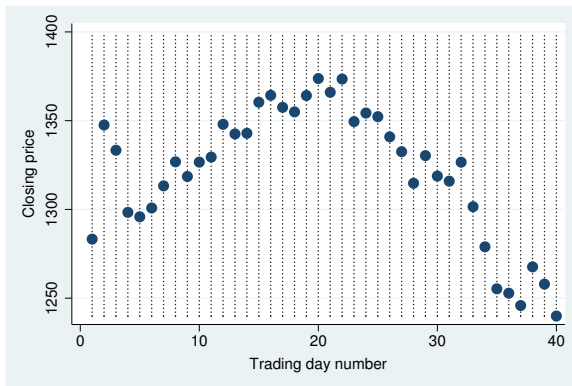

```
twoway spike close tradeday
```



Here we use a **spike** plot to show the same graph as the previous one. It is like the **dropline** plot, but no markers are put on the top. For more details, see [Twoway : Scatter \(89\)](#).

Uses spjanfeb2001.dta & scheme vg_s2c

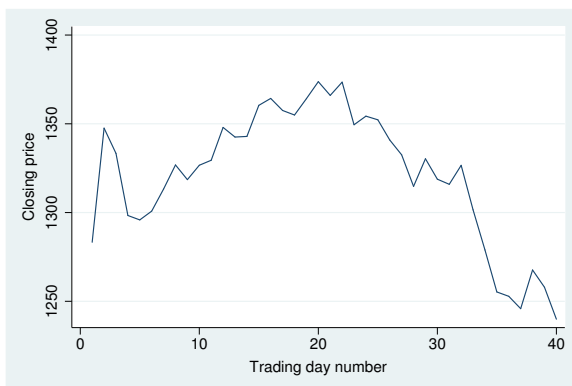
```
twoway dot close tradeday
```



The **dot** plot, like the **scatterplot**, shows markers for each data point but also adds a dotted line for each of the x values. For more details, see [Twoway : Scatter \(89\)](#).

Uses spjanfeb2001.dta & scheme vg_s2c

```
twoway line close tradeday, sort
```



We use the **line** command in this example to make a simple line graph. See [Twoway : Line \(112\)](#) for more details about line graphs.

Uses spjanfeb2001.dta & scheme vg_s2c

(Pages omitted)

1.3 Schemes

Whereas the previous section was about the different types of graphs Stata can make, this section is about the different kinds of looks that you can have for Stata graphs. The basic starting point for the look of a graph is a scheme, which controls just about every aspect of the look of the graph. A scheme sets the stage for the graph, but you can use options to override the settings in a scheme. As you might surmise, if you choose (or develop) a scheme that produces graphs similar to the final graph you want to make, you can reduce the need to customize your graphs using options. This section gives you a basic idea of what schemes can do and introduces you to the schemes used throughout the book. See [Intro:Using this book \(1\)](#) for more details about how to select and use schemes and [Appendix:Online supplements \(482\)](#) for more information about how to download them.

```
twoway scatter propval100 rent700 ownhome, scheme(vg_s1c)
```

This scatterplot illustrates the `vg_s1c` scheme. It is based on the `s1color` scheme but increases the sizes of elements in the graph to make them more readable. This scheme is in color and has a white background, both inside the plot region and in the surrounding area.

Uses `allstates.dta` & `scheme vg_s1c`



```
twoway scatter propval100 rent700 ownhome, scheme(vg_s1m)
```

This scatterplot is similar to the last one but uses the `vg_s1m` scheme, the monochrome equivalent of the `vg_s1c` scheme. It is based on the `s1mono` scheme but increases the sizes of elements in the graph to make them more readable. This scheme is in black and white and has a white background, both inside the plot region and in the surrounding area.

Uses `allstates.dta` & `scheme vg_s1m`



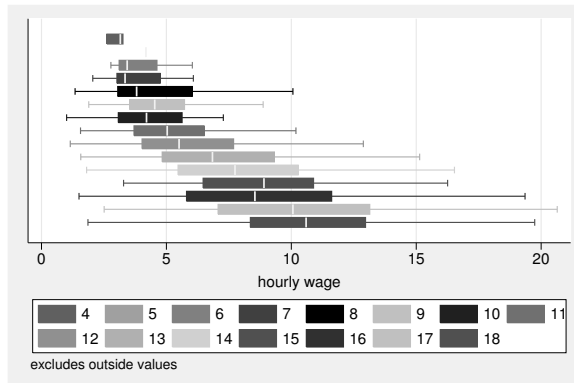
```
graph hbox wage, over(grade) asyvar nooutsides legend(rows(2))
  scheme(vg_s2c)
```



This box plot shows an example of the `vg_s2c` scheme. It is based on the `s2color` scheme but increases the sizes of elements in the graph to make them more readable. In this scheme, the plot region has a white background, but the surrounding area (the graph region) is light blue.

Uses `nsw.dta` & `scheme vg_s2c`

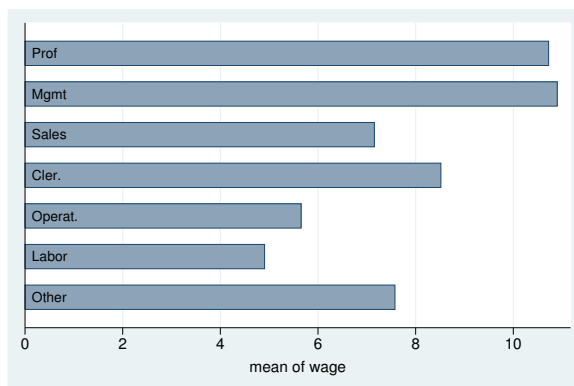
```
graph hbox wage, over(grade) asyvar nooutsides legend(rows(2))
  scheme(vg_s2m)
```



This box plot is similar to the previous one but uses the `vg_s2m` scheme, the monochrome equivalent of the `vg_s2c` scheme. This scheme is based on the `s2mono` scheme but increases the sizes of elements in the graph to make them more readable. This scheme is in black and white, and it has a white background in the plot region but is light gray in the surrounding graph region.

Uses `nsw.dta` & `scheme vg_s2m`

```
graph hbar wage, over(occ7, label(nolabels)) blabel(group, position(base))
  scheme(vg_palec)
```

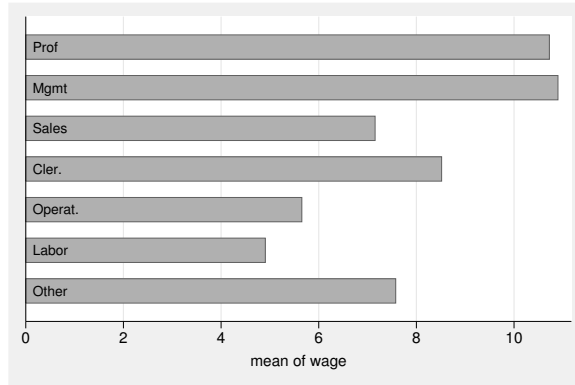


This horizontal bar chart shows an example of the `vg_palec` scheme. It is based on the `s2color` scheme but makes the colors of the bars/boxes/markers paler by decreasing the intensity of the colors. As shown in this example, one use of this scheme is to make the colors of the bars pale enough to include text labels inside bars.

Uses `nsw.dta` & `scheme vg_palec`

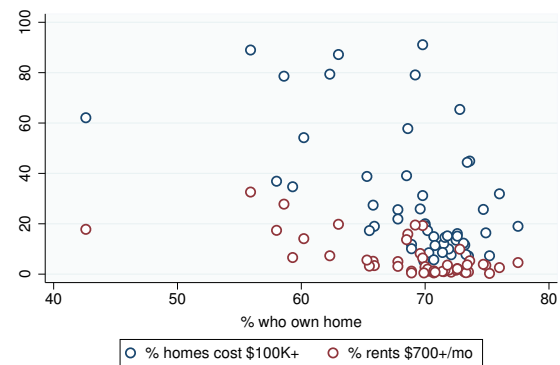
```
graph hbar wage, over(occ7, label(nolabels)) blabel(group, position(base))
  scheme(vg_palem)
```

This example is the same as the last one but uses the `vg_palem` scheme, the monochrome equivalent of the `vg_palec` scheme. This scheme is based on the `s2mono` scheme but makes the colors of the bars/boxes/markers paler by decreasing the intensity of the colors. Uses `nls.w.dta` & `scheme vg_palem`



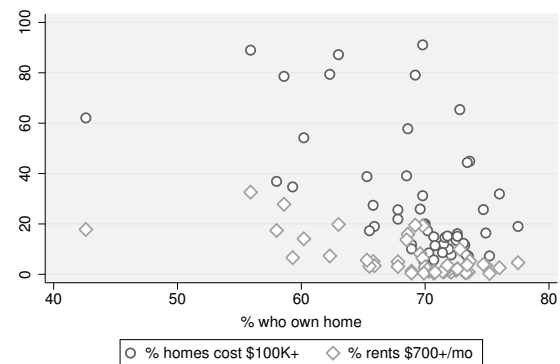
```
scatter propval100 rent700 ownhome, scheme(vg_outc)
```

This scatterplot illustrates the `vg_outc` scheme. It is based on the `s2color` scheme but makes the fill color of the bars/boxes/markers white, so they appear hollow. The plot region is a light blue to contrast with the white fill color. This scheme is useful to see the number of markers present where numerous markers are close or partially overlapping. Uses `allstates.dta` & `scheme vg_outc`

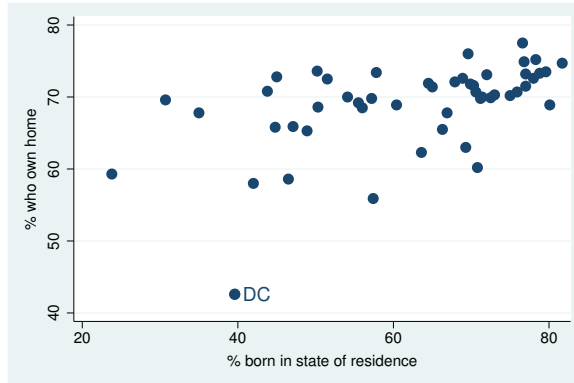


```
scatter propval100 rent700 ownhome, scheme(vg_outm)
```

This example is similar to the previous one but illustrates the `vg_outm` scheme, the monochrome equivalent of the `vg_outc` scheme. It is based on the `s2mono` scheme but makes the fill color of the bars/boxes/markers white, so they appear hollow. Uses `allstates.dta` & `scheme vg_outm`



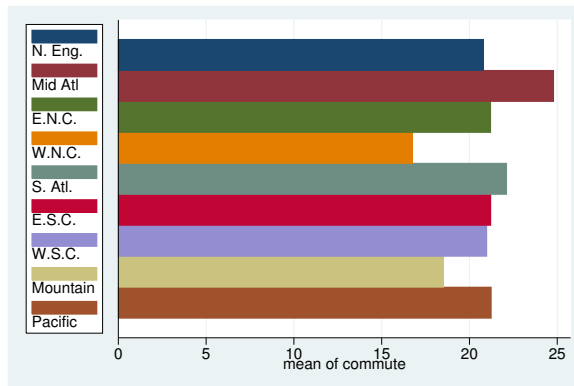
```
twoway (scatter ownhome borninstate if stateab=="DC", mlabel(stateab))
       (scatter ownhome borninstate), legend(off) scheme(vg_samec)
```



This is an example of the `vg_samec` scheme, which is based on the `s2color` scheme and makes all the markers, lines, bars, etc., the same color, shape, and pattern. Here the second `scatter` command labels Washington DC, which normally would be shown in a different color; with this scheme, the marker is the same. This scheme has a monochrome equivalent called `vg_samem`, which is not illustrated.

Uses allstates.dta & scheme vg_samec

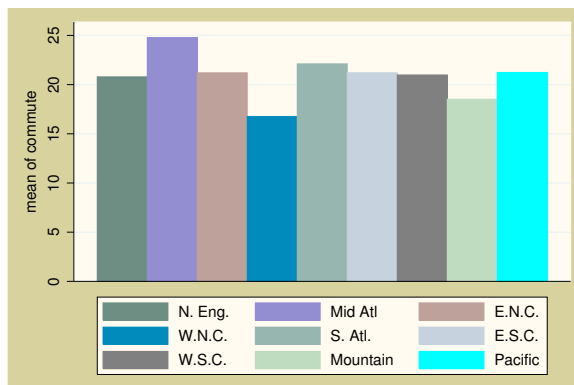
```
graph hbar commute, over(division) asyvar scheme(vg_lgndc)
```



This horizontal bar chart shows an example of the `vg_lgndc` scheme. It is based on the `s2color` scheme but changes the default attributes of the legend, namely, showing the legend in one column to the left of the plot region, with the key and symbols placed atop each other. It can be efficient to place the legend to the left of the graph. This scheme has a monochrome equivalent called `vg_lgndm`, which is not illustrated here.

Uses allstates.dta & scheme vg_lgndc

```
graph bar commute, over(division) asyvar legend(rows(3)) scheme(vg_past)
```



This bar chart shows an example of the `vg_past` scheme. It is based on the `s2color` scheme but selects subdued pastel colors and provides a sand background for the surrounding graph region and an eggshell color for the inner plot region and legend area.

Uses allstates.dta & scheme vg_past