A Visual Guide to Stata Graphics

Fourth Edition

MICHAEL N. MITCHELL

A Stata Press Publication
StataCorp LLC
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2$_\varepsilon$
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Print ISBN-10: 1-59718-365-2
Print ISBN-13: 978-1-59718-365-9
ePub ISBN-10: 1-59718-366-0
ePub ISBN-13: 978-1-59718-366-6

Library of Congress Control Number: 2021951939

Dedication

I dedicate this book to the teachers of the world. I have been fortunate to have been touched by many special teachers, and I will always be grateful for what they kindly gave to me. I thank (in order of appearance) Larry Grossman, Fred Perske, Rosemary Sheridan, Donald Butler, Jim Torcivia, Richard O'Connell, Linda Fidell, and Jim Sidanius. These teachers all left me gifts of knowledge and life lessons that help me every day. Even if they do not all remember me, I will always remember them.

(Pages omitted)

Contents

(Pages omitted)

Preface to the Fourth Edition

When I was writing the first edition of this book, we all pictured a book printed in black and white. All the other books in the Stata Press catalog were in black and white. As the book was nearing completion, Stata Press found a printer who could print the book in full color. The book was nearly done—nearly done in black and white. I took a hard swallow, and we agreed that even though it would take extra time and rethinking parts of the book from scratch, the book should be in color. Seeing all the features that Stata has added for supporting colors, I find it hard to imagine this book any other way. This new edition goes *all in* on the features that Stata offers for displaying colors. In the third edition, the section on color styles had five examples—that section in this new edition includes over 50 examples [see Styles : Colors (401)]. Instead of trying to explain the look of colors at different intensities and opacities, I show you commands and graphs that illustrate different colors shown at differing intensities and using differing opacities. Further, I illustrate how these options interact when regions with different colors are overlaid atop each other. You can play with these examples to explore other combinations of colors/intensities/opacities, either alone or when overlapping one another.

In addition to the new coverage of colors, this new edition details the methods you can use for sizing objects, showing the three ways of sizing objects using *absolute units* (like points, inches, and centimeters) and the three ways you can size objects using *relative units* (such as using keywords like `large`, multipliers of the original size like `*2`, or sizes relative to the size of the graph, like `5rs`). Each of these units is illustrated in the context of sizing different elements, such as text [like titles, axis labels, marker labels, legends, and so on; see Styles : Textsize (447)]; markers [see Styles : Markersize (439)]; line widths [see Styles : Linewidth (434)]; and more. Each of those sections illustrates sizing of elements in isolation—additionally, Standard options : Sizing graphs (386) illustrates resizing the entire graph and the different results you obtain when individual elements are sized using relative units versus absolute units.

If you have used prior editions of the book, you may notice that this edition no longer includes a chapter on the Graph Editor and that the examples focus exclusively on the use of commands for creating graphs. This is not a commentary about the utility of the Graph Editor, but instead a reflection that this book was getting too large and that Stata has a growing library of video tutorials that *interactively show* how to create and modify graphs via the Stata interface. In section 1.7, I describe the utility of the interactive point-and-click interface for creating and modifying graphs and suggest videos I think illustrate key features.

The overall look of this book is dramatically different from the prior edition. The prior editions periodically changed the schemes to introduce novelty and pizzazz and to underscore how powerful schemes are for controlling the entire look of your graph. This new edition uses one common scheme and changes the scheme only when there is a rationale for choosing one scheme over another. With schemes in mind, the heart of section Standard

options : Schemes (366) shows three different kinds of graphs, one at a time, illustrating the look of that graph using selected schemes that ship with Stata, schemes included with this book, and several schemes from the worldwide Stata community.

Writing this fourth edition book was a great pleasure, especially for the respite it gave during such difficult and turbulent times. I deeply hope that this book finds you happy, healthy, and—most of all—safe.

Ventura, California
December 2021

(Pages omitted)

1 Introduction

This chapter begins by briefly telling you about how to access the datasets and schemes used in this book, so you can replicate and extend any of the examples for yourself. The next section gives you some tips about using this book, followed by short overview of the different kinds of Stata graphs that will be examined in this book. Next, I provide an overview of schemes and how they can be used to obtain different looks for graphs. The fourth section illustrates the structure of options in Stata graph commands. In a sense, the third, fourth, and fifth 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 next section is about the process of creating graphs, and the final section provides information about the interactive point and click interface for creating and editing graphs

1.1 Online supplements

I encourage you to download the data and schemes associated with this book. That will allow you to replicate, and extend, the examples shown in this book. You can quickly download all the datasets, schemes, and programs used in this book with the following `net` commands.

```
. net from https://www.stata-press.com/data/vgsg4/
. net get vgsg4
. net install vgsg4
```

The `net from` command connects you to the resources associated with this book. The `net get` command will download the datasets into your current working directory.[1] The `net install` command will install the schemes used in this book as well as the programs used in this book (for example, `vgcolormap`).[2]

You can visit the webpage for the book at

https://www.stata-press.com/books/visual-guide-to-stata-graphics/

This site will have additional information about the book, any updates on obtaining the latest scheme files illustrated in the book, and an *Errata* showing any errors that have been found.

1. You may want to store these datasets in a specific folder. In that case, prior to the `net get` command, you may want to make a folder for the datasets and then use the `cd` command to make that your current working directory.
2. If you have installed the schemes/programs previously, you will need to add the `replace` option—that is, `net install vgsg4, replace`.

Each graph shows the dataset used prior to creating the graph, for example,

Uses allstates.dta

This statement indicates that you will want to read the dataset `allstates.dta` into memory before issuing the graph command. The default scheme used for the book is `vg_s2cx`. If a different scheme is used, it will be specified via the `scheme()` option. For more information about schemes, see Intro : Schemes (15) and Standard options : Schemes (366).

1.2 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. 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 a legend across different types of graphs. It would be repetitive to go into detail about a legend for bar charts, box plots, and so on. Within each kind of graph, a legend is 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 (324)", 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 324.

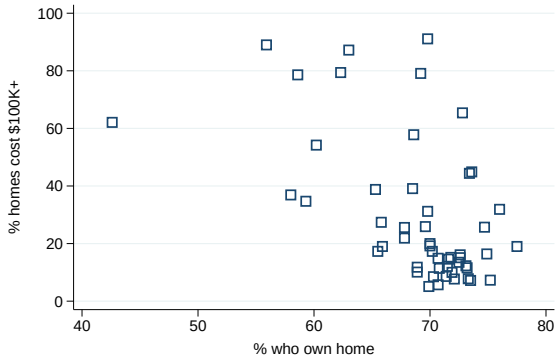
Sometimes it may take an extra cross-reference to get the information you need. Say that you want to make the *y*-axis title large for a bar chart by using the `ytitle()` option, so you first consult Bar : Y-axis (180). This gives you some information about using `ytitle()`, but then that section refers you to Options : Axis titles (290), where more details about axis titles are described. This section then refers you to Options : Textboxes (341) for more complete details about options to control the display of text. That section shows more details but then refers to Styles : Textsize (447), 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 (269) for more details. The `graph twoway` portion of the command is optional.
Uses allstates.dta



- The command itself is displayed in a `typewriter font`, and the salient part of the command (that is, `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 (for example, `graph twoway`), they are displayed in the `typewriter font`.
- Many of the descriptions contain cross-references, for example, Options : Markers (269), which means to flip to the *Options* chapter and then to the section *Markers*. Equivalently, go to page 269.
- 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 . . .".
- The descriptive text always concludes by telling you the name of the data file in memory when making the graph. Here the data file was `allstates.dta`.

If you want your graphs to look like the ones in the book, you can display them using the same schemes. See Introduction : Online supplements (1) 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_s2cx
. use allstates
. graph twoway scatter propval100 ownhome, msymbol(Sh)
```

After you issue the `set scheme vg_s2cx` command, subsequent graph commands will show graphs with the `vg_s2cx` scheme. You could also add the `scheme(vg_s2cx)` 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_s2cx)
```

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 `help graph`.

This book has been written based on the features available in Stata version 17. 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 17. For example, if you were running Stata version 18.0 but wanted a graph command to run as though you were running Stata 17, you could type

```
. version 17: graph twoway scatter propval100 ownhome
```

and the command would be executed as if you were running version 17. Or, perhaps you want a command to run as it did under Stata 16.1, you would then type

```
. version 16.1: graph twoway scatter propval100 ownhome
```

Finally, I would like to emphasize that the goal of this book is to help you learn and use the Stata graph commands 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. Additionally, if you are creating a graph as part of a manuscript to be submitted for publication, I recommend consulting the author guidelines from the publisher as well as looking at graphs that have been recently been published in the journal for guidelines for creating your graphs.

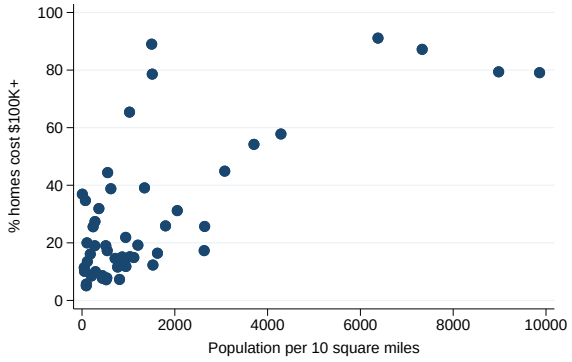
1.3 Types of Stata graphs

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.

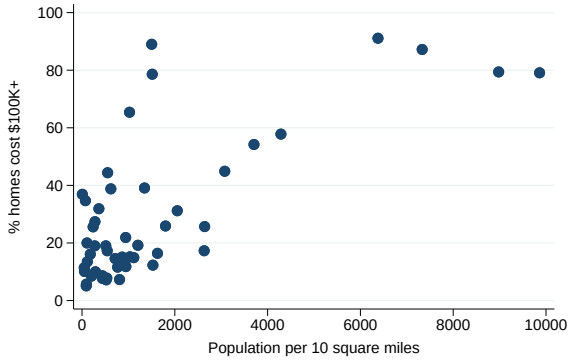
```
graph twoway scatter propval100 popden
```

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 (37) for more details about these kinds of plots.
Uses allstates.dta

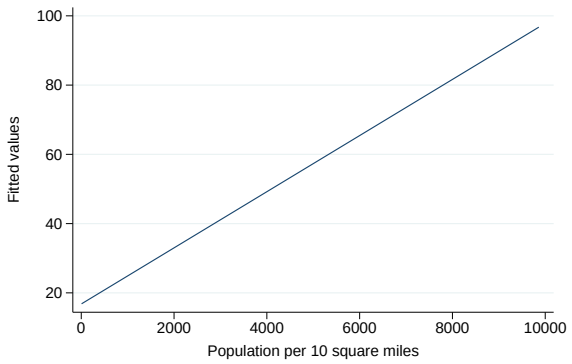



```
twoway scatter propval100 popden
```

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

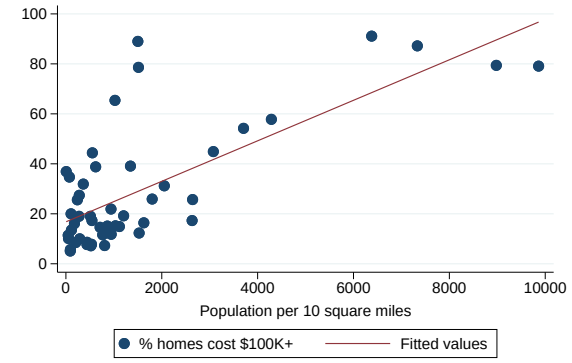


```
twoway lfit propval100 popden
```



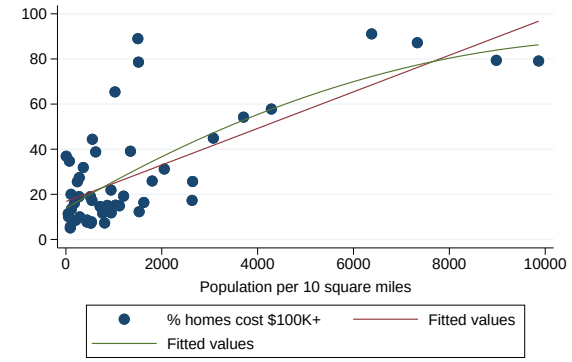
We now make a linear fit (`lfit`) line predicting `propval100` from `popden`. See Twoway : Fit (57) for more information about these kinds of plots.
Uses allstates.dta

```
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 (116).
Uses allstates.dta

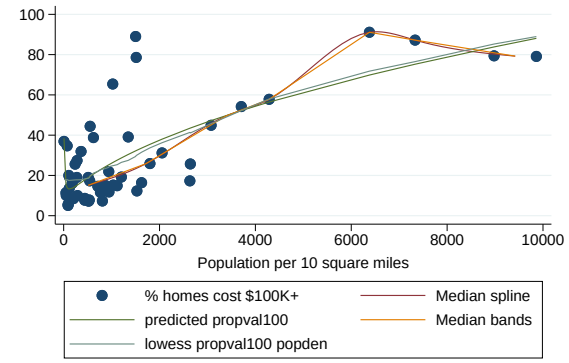
```
twoway (scatter propval100 popden) (lfit propval100 popden)
   (qfit propval100 popden)
```



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

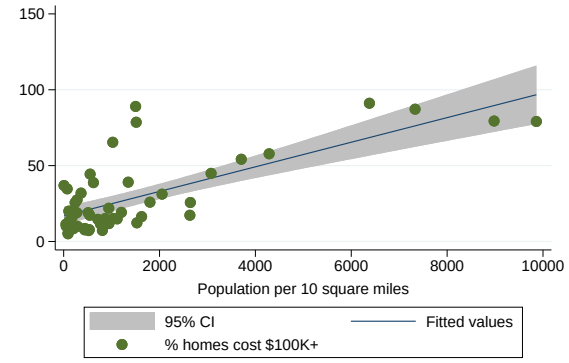
```
twoway (scatter propval100 popden) (mspline propval100 popden)
   (fpfit propval100 popden) (mband propval100 popden)
   (lowess propval100 popden)
```

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 (57).
Uses allstates.dta



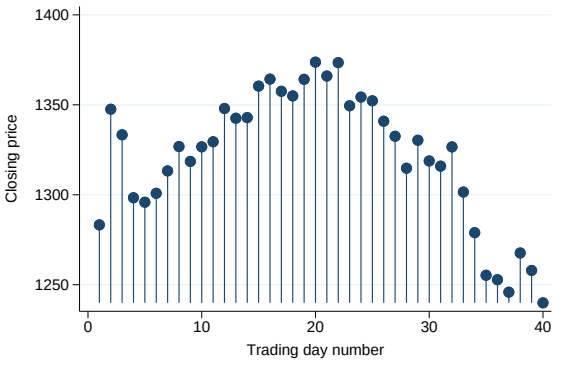

```
twoway (lfitci propval100 popden) (scatter propval100 popden)
```

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 (60) for more information about fit lines with confidence intervals.
Uses allstates.dta

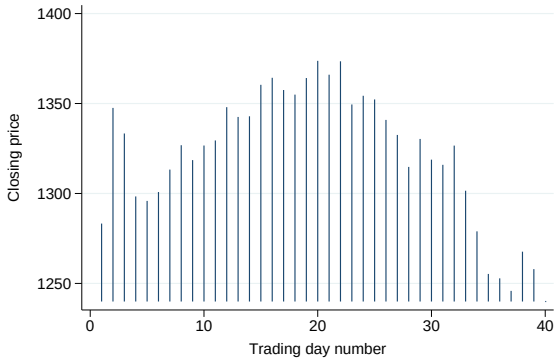



```
twoway dropline close tradeday
```

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 `scatter`plot 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 (37).
Uses spjanfeb2001.dta

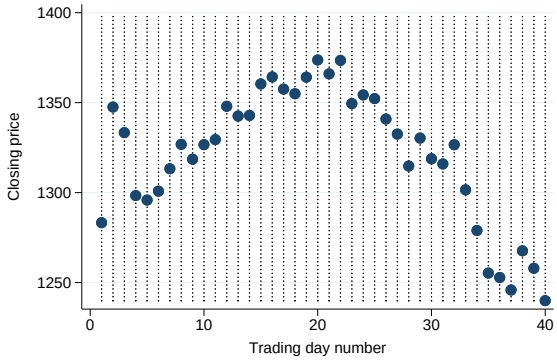


```
twoway spike close tradeday
```



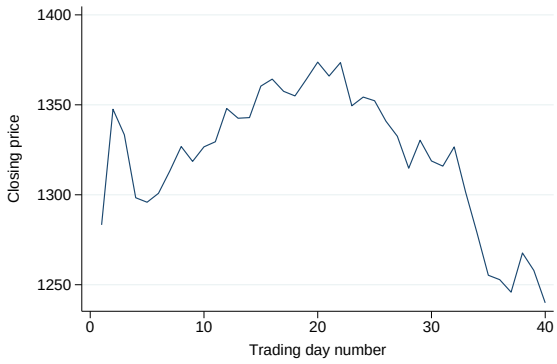
Here we use a `spike` plot to show the same graph as the previous one. It is like the `dropline` plot, but no markers are put on the top. For more details, see Twoway : Scatter (37).
Uses spjanfeb2001.dta

```
twoway dot close tradeday
```



The `dot` plot, like the `scatter`plot, shows markers for each data point but also adds a dotted line for each of the x values. For more details, see Twoway : Scatter (37).
Uses spjanfeb2001.dta

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



We use the `line` command in this example to make a simple line graph. See Twoway : Line (68) for more details about line graphs.
Uses spjanfeb2001.dta

(Pages omitted)

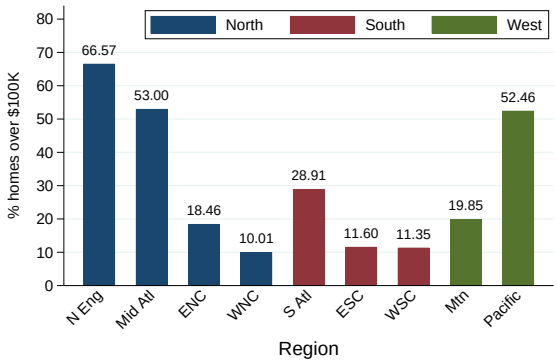
Although you can use an option like `legend()` with many, but not all, kinds of Stata graph commands, you can use other kinds of options with almost every kind of Stata graph. These are called *standard options*. To help you differentiate these kinds of options, they are discussed in their own chapter, Standard options (361). Because these options can be used with most types of graph commands, they are generally not discussed in the chapters about the different types of graphs, except when their usage interacts with the options illustrated. For example, `subtitle()` is a standard option, but its behavior takes on a special meaning when used with the `legend()` option, so the `subtitle()` option is discussed in the context of the legend. Consistent with what was previously shown, the syntax of standard options follows the same kinds of rules that have been illustrated, and their usage and behavior are uniform across the many types of Stata graph commands.

1.6 Building graphs

I have three agendas in writing this section. First, I wish to show the process of building complex graphs a little bit at a time. At the same time, I illustrate how to use the resources of this book to get the bits of information needed to build these graphs. Finally, I hope to show that, even though a complete Stata graph command might look complicated and overwhelming, the process of building the graph slowly is actually straightforward and logical.

Let's first build a bar chart that looks at property values broken down by region of the country. Then we will modify the legend and bar characteristics, add titles, and so forth.

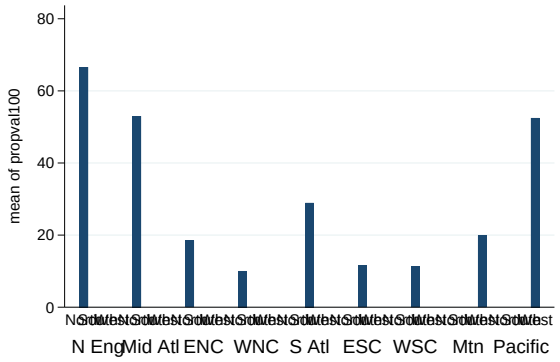
```
graph display
```



Say that we want to create this graph. For now, the syntax is concealed, just showing the `graph display` command to show the previously drawn graph. It might be overwhelming at first to determine all the options needed to make this graph. To ease our task, we will build it a bit at a time, refining the graph and fixing any problems we find. *Uses allstates.dta*

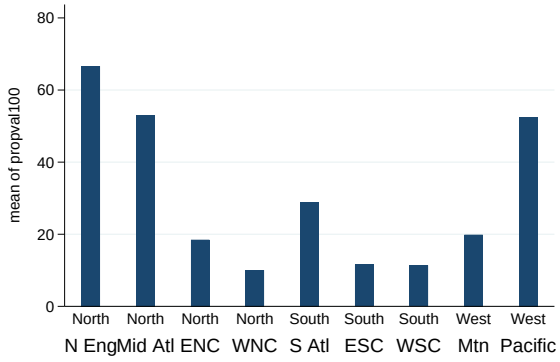
```
graph bar propval100, over(nsw) over(division)
```

We begin by seeing that this is a bar chart and look at Bar : Y-variables (139) and Bar : Over (144). We take our first step toward making this graph by making a bar chart showing `propval100` and adding `over(nsw)` and `over(division)` to break down the means by `nsw` and `division`.
Uses allstates.dta



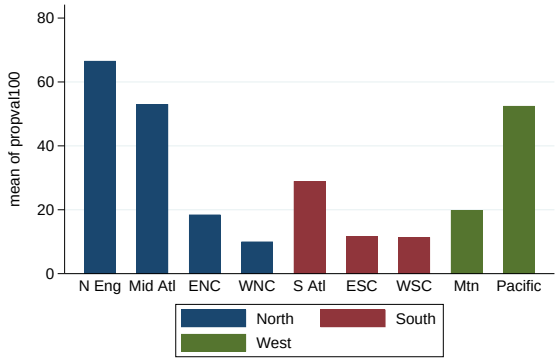

```
graph bar propval100, over(nsw) over(division) nofill
```

The previous graph is not quite what we want because we see every `division` shown with every `nsw`, but for example, the Pacific region only appears in the West. In Bar : Over (144), we see that we can add the `nofill` option to show only the combinations of `nsw` and `division` that exist in the data file. Next we will look at the colors of the bars.
Uses allstates.dta

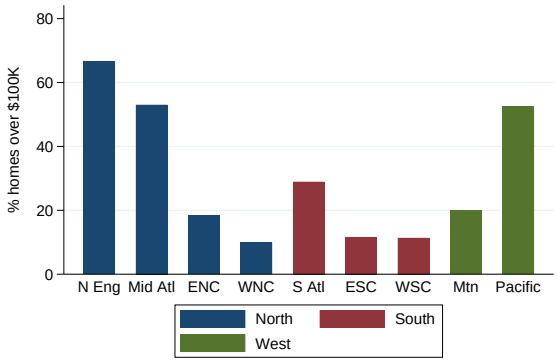



```
graph bar propval100, over(nsw) over(division) nofill asyvars
```

The last graph is getting closer, but we want the bars for North, South, and West displayed in different colors and labeled with a legend. In Bar : Y-variables (139), we see that the `asyvars` option will accomplish this. Next we will change the title for the y axis.
Uses allstates.dta

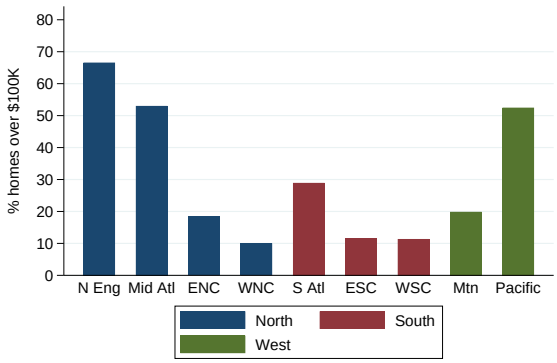


```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K")
```



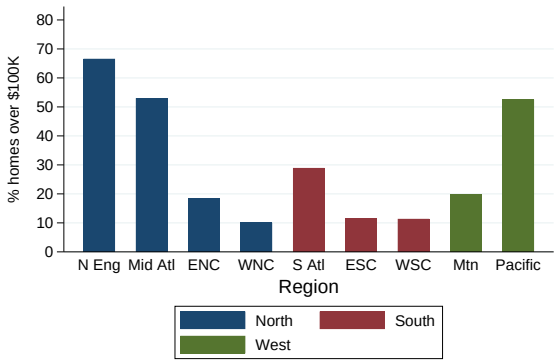
Now we want to put a title on the y axis. In Bar : Y-axis (180), we see examples illustrating the use of `ytitle()` for putting a title on the y axis. Here we put a title on the y axis, but now we want to change the labels for the y axis to go from 0 to 80, incrementing by 10.
Uses allstates.dta

```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K") ylabel(0(10)80)
```



The Bar : Y-axis (180) section also tells us about the `ylabel()` option. Now that we have the y axis labeled as we want, let's next look at the title for the x axis.
Uses allstates.dta

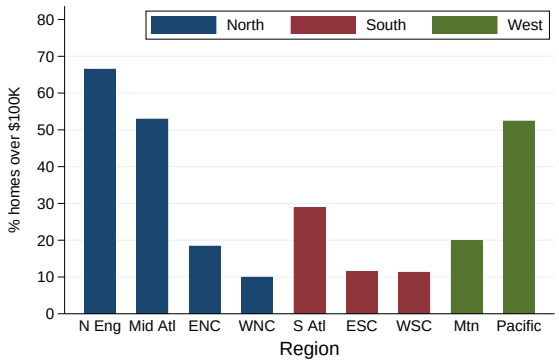
```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K") ylabel(0(10)80) b1title(Region)
```



After having used the `ytitle()` option to label the y axis, we might be tempted to use the `xtitle()` option to label the x axis, but this axis is a categorical variable. In Bar : Cat axis (162), we see that this axis is treated differently because of that. To put a title below the graph, we use the `b1title()` option. Now let's turn our attention to formatting the legend.
Uses allstates.dta

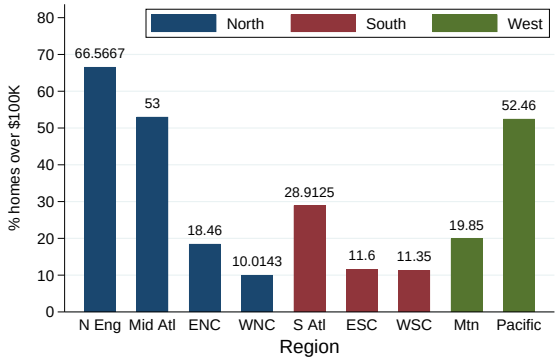
```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K") ylabel(0(10)80) b1title(Region)
   legend(rows(1) position(1) ring(0))
```

Here we want to use the `legend()` option to make the legend have one row in the top right corner within the plot area. In Bar : Legend (169), we see that the `rows(1)` option makes the legend appear in one row and that the `position(1)` option puts the legend in the one o'clock position. The `ring(0)` option puts the legend inside the plot region. Next let's label the bars.
Uses allstates.dta



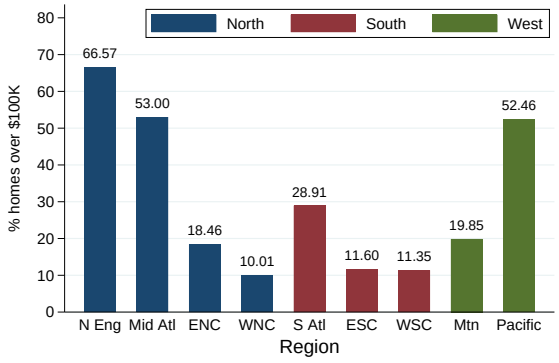

```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K") ylabel(0(10)80) b1title(Region)
   legend(rows(1) position(1) ring(0)) blabel(bar)
```

We want each bar labeled with the height of the bar. Bar : Legend (169) shows how we can do this by using the `blabel()` (bar label) option to label the bars in lieu of a legend. `blabel(bar)` labels the bars with their height.
Uses allstates.dta

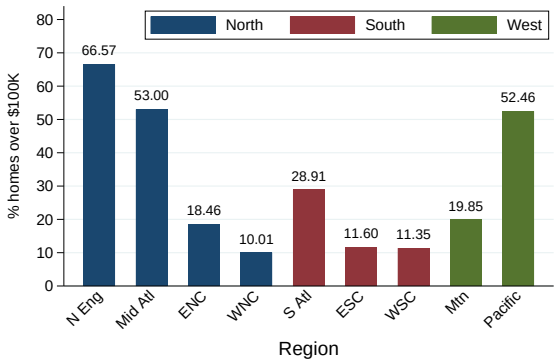



```
graph bar propval100, over(nsw) over(division) nofill asyvars
   ytitle("% homes over $100K") ylabel(0(10)80) b1title(Region)
   legend(rows(1) position(1) ring(0)) blabel(bar, format(%4.2f))
```

We want the label for each bar to end in two decimal places, and we see in Bar : Legend (169) that we can use the `format()` option to format these numbers as we want.
Uses allstates.dta



```
graph bar propval100, over(nsw) over(division, label(angle(45))) nofill
   ytitle("% homes over $100K") ylabel(0(10)80) b1title(Region)
   legend(rows(1) position(1) ring(0)) blabel(bar, format(%4.2f)) asyvars
```



Finally, in Bar : Cat axis (162), we see that we can add the `label(angle(45))` option to the `over()` option to specify that labels for that variable be shown at a 45-degree angle so they do not overlap each other.
Uses allstates.dta

I hope this section has shown that it is not that difficult to create complex graphs by building them one step at a time. You can use the resources in this book to seek out each piece of information you need and then put those pieces together the way you want to create your own graphs. For more information about how to integrate options to create complex Stata graphs, see Appendix : More examples (505).

1.7 Point-and-click interface

Many people have an aversion to the use of point-and-click methods for creating statistical results or statistical figures. A key part of this aversion is that such methods are frequently not repeatable, violating a key scientific principle of repeatability. However, the Stata point-and-click interface produces the commands that can be used to replicate each result/graph it creates. Because of this, the Stata point-and-click interface offers the advantages of an interactive point-and-click interface combined with repeatability. In this section, I would like to highlight some of the advantages of the point-and-click interface for the creation and customization of graphs.

Teaching about a dynamic point-and-click interface via a static medium like a book is very difficult. Instead, a point-and-click interface is taught more effectively via online videos, like the ones that you can find in the Stata Video Tutorials (also called the Stata YouTube channel). This library of videos covers a wide variety of topics, including Stata graphics. You can find these videos in a few different ways:

1. Visit the *Video Tutorials* webpage at https://www.stata.com/links/video-tutorials/, which lists all the video tutorials grouped by topics (including graphics).

2. Search the web for *StataCorp YouTube* to go to the *Stata YouTube Channel*; then click *PLAYLISTS* and then *Creating Graphs in Stata*.

3. You could search the Internet for *Stata YouTube Graphs*, which yields results from the *Stata YouTube Channel* as well as videos posted by others illustrating topics on Stata graphs.