

1 Read this—it will help

Contents

- 1.1 Getting Started with Stata
- 1.2 The User's Guide and the Reference manuals
 - 1.2.1 PDF manuals
 - 1.2.1.1 Video example
 - 1.2.2 Example datasets
 - 1.2.2.1 Video example
 - 1.2.3 Cross-referencing
 - 1.2.4 The index
 - 1.2.5 The subject table of contents
 - 1.2.6 Typography
 - 1.2.7 Vignette
- 1.3 What's new
 - 1.3.1 What's new (highlights)
 - 1.3.2 What's new that you will want to know
 - 1.3.3 What's new in statistics (general)
 - 1.3.4 What's new in statistics (SEM)
 - 1.3.5 What's new in statistics (time series)
 - 1.3.6 What's new in statistics (longitudinal/panel data)
 - 1.3.7 What's new in statistics (survival analysis)
 - 1.3.8 What's new in data management
 - 1.3.9 What's new in Mata
 - 1.3.10 What's new in programming
 - 1.3.11 What's new, Mac only
 - 1.3.12 What's more
- 1.4 References

A Complete Stata Documentation Set contains more than 11,000 pages of information in the following manuals:

- [GS] *Getting Started with Stata (Mac, Unix, or Windows)*
- [U] *Stata User's Guide*
- [R] *Stata Base Reference Manual*
- [D] *Stata Data Management Reference Manual*
- [G] *Stata Graphics Reference Manual*
- [XT] *Stata Longitudinal-Data/Panel-Data Reference Manual*
- [ME] *Stata Multilevel Mixed-Effects Reference Manual*
- [MI] *Stata Multiple-Imputation Reference Manual*
- [MV] *Stata Multivariate Statistics Reference Manual*
- [PSS] *Stata Power and Sample-Size Reference Manual*
- [P] *Stata Programming Reference Manual*
- [SEM] *Stata Structural Equation Modeling Reference Manual*
- [SVY] *Stata Survey Data Reference Manual*
- [ST] *Stata Survival Analysis and Epidemiological Tables Reference Manual*
- [TS] *Stata Time-Series Reference Manual*
- [TE] *Stata Treatment-Effects Reference Manual:
Potential Outcomes/Counterfactual Outcomes*
- [I] *Stata Glossary and Index*
- [M] *Mata Reference Manual*

In addition, installation instructions may be found in the *Installation Guide*, which comes in the DVD case.

1.1 Getting Started with Stata

There are three *Getting Started* manuals:

- [GSM] *Getting Started with Stata for Mac*
- [GSU] *Getting Started with Stata for Unix*
- [GSW] *Getting Started with Stata for Windows*

1. Learn how to use Stata—read the *Getting Started* (GSM, GSU, or GSW) manual.
2. Now turn to the other manuals; see [U] 1.2 **The User's Guide and the Reference manuals**.

1.2 The User's Guide and the Reference manuals

The *User's Guide* is divided into three sections: *Stata basics*, *Elements of Stata*, and *Advice*. The table of contents lists the chapters within each of these sections. Click on the chapter titles to see the detailed contents of each chapter.

The *Guide* is full of a lot of useful information about Stata; we recommend that you read it. If you only have time, however, to read one or two chapters, then read [U] 11 **Language syntax** and [U] 12 **Data**.

The other manuals are the *Reference* manuals. The *Stata Reference* manuals are each arranged like an encyclopedia—alphabetically. Look at the *Base Reference Manual*. Look under the name of a command. If you do not find the command, look in the index. A few commands are so closely related that they are documented together, such as `ranksum` and `median`, which are both documented in [R] `ranksum`.

Not all the entries in the *Base Reference Manual* are Stata commands; some contain technical information, such as [R] `maximize`, which details Stata's iterative maximization process, or [R] `error messages`, which provides information on error messages and return codes.

Like an encyclopedia, the *Reference* manuals are not designed to be read from cover to cover. When you want to know what a command does, complete with all the details, qualifications, and pitfalls, or when a command produces an unexpected result, read its description. Each entry is written at the level of the command. The descriptions assume that you have little knowledge of Stata's features when they are explaining simple commands, such as those for using and saving data. For more complicated commands, they assume that you have a firm grasp of Stata's other features.

If a Stata command is not in the *Base Reference Manual*, you can find it in one of the other *Reference* manuals. The titles of the manuals indicate the types of commands that they contain. The *Programming Reference Manual*, however, contains commands not only for programming Stata but also for manipulating matrices (not to be confused with the matrix programming language described in the *Mata Reference Manual*).

1.2.1 PDF manuals

Every copy of Stata comes with Stata's complete PDF documentation.

The PDF documentation may be accessed from within Stata by selecting **Help > PDF Documentation**. Even more convenient, every help file in Stata links to the equivalent manual entry. If you are reading `help regress`, simply click on [R] `regress` in the `Title` section of the help file to go directly to the [R] `regress` manual entry.

We provide recommended settings for your PDF viewer to optimize it for Stata's documentation at <http://www.stata.com/support/faqs/res/documentation.html>.

1.2.1.1 Video example

[PDF documentation in Stata](#)

1.2.2 Example datasets

Various examples in this manual use what is referred to as the automobile dataset, `auto.dta`. We have created a dataset on the prices, mileages, weights, and other characteristics of 74 automobiles and have saved it in a file called `auto.dta`. (These data originally came from the April 1979 issue of *Consumer Reports* and from the United States Government EPA statistics on fuel consumption; they were compiled and published by Chambers et al. [1983].)

In our examples, you will often see us type

```
. use http://www.stata-press.com/data/r13/auto
```

We include the `auto.dta` file with Stata. If you want to use it from your own computer rather than via the Internet, you can type

```
. sysuse auto
```

See [D] [sysuse](#).

You can also access `auto.dta` by selecting **File > Example Datasets...**, clicking on *Example datasets installed with Stata*, and clicking on `use` beside the `auto.dta` filename.

There are many other example datasets that ship with Stata or are available over the web. Here is a partial list of the example datasets included with Stata:

<code>auto.dta</code>	1978 Automobile Data
<code>auto2.dta</code>	1978 Automobile Data
<code>autornd.dta</code>	Subset of 1978 Automobile Data
<code>bplong.dta</code>	fictional blood pressure data
<code>bpwide.dta</code>	fictional blood pressure data
<code>cancer.dta</code>	Patient Survival in Drug Trial
<code>census.dta</code>	1980 Census data by state
<code>citytemp.dta</code>	City Temperature Data
<code>citytemp4.dta</code>	City Temperature Data
<code>educ99gdp.dta</code>	Education and GDP
<code>gnp96.dta</code>	U.S. GNP, 1967–2002
<code>lifeexp.dta</code>	Life expectancy, 1998
<code>network1.dta</code>	fictional network diagram data
<code>network1a.dta</code>	fictional network diagram data
<code>nlsw88.dta</code>	U.S. National Longitudinal Study of Young Women (NLSW, 1988 extract)
<code>nlswide1.dta</code>	U.S. National Longitudinal Study of Young Women (NLSW, 1988 extract)
<code>pop2000.dta</code>	U.S. Census, 2000, extract
<code>sandstone.dta</code>	Subsea elevation of Lamont sandstone in an area of Ohio
<code>sp500.dta</code>	S&P 500
<code>surface.dta</code>	NOAA Sea Surface Temperature
<code>tsline1.dta</code>	simulated time-series data
<code>tsline2.dta</code>	fictional data on calories consumed
<code>uslifeexp.dta</code>	U.S. life expectancy, 1900–1999
<code>uslifeexp2.dta</code>	U.S. life expectancy, 1900–1940
<code>voter.dta</code>	1992 presidential voter data
<code>xtline1.dta</code>	fictional data on calories consumed

All of these datasets may be used or described from the **Example Datasets...** menu listing.

Even more example datasets, including most of the datasets used in the reference manuals, are available at the Stata Press website (<http://www.stata-press.com/data/>). You can download the datasets with your browser, or you can use them directly from the Stata command line:

```
. use http://www.stata-press.com/data/r13/nlswork
```

An alternative to the `use` command for these example datasets is `webuse`. For example, typing

```
. webuse nlswork
```

is equivalent to the above `use` command. For more information, see [D] [webuse](#).

1.2.2.1 Video example

Example data included with Stata

1.2.3 Cross-referencing

The *Getting Started* manual, the *User's Guide*, and the *Reference* manuals cross-reference each other.

[R] **regress**

[D] **reshape**

[XT] **xtreg**

The first is a reference to the **regress** entry in the *Base Reference Manual*, the second is a reference to the **reshape** entry in the *Data Management Reference Manual*, and the third is a reference to the **xtreg** entry in the *Longitudinal-Data/Panel-Data Reference Manual*.

[GSW] **B Advanced Stata usage**

[GSM] **B Advanced Stata usage**

[GSU] **B Advanced Stata usage**

are instructions to see the appropriate section of the *Getting Started with Stata for Windows*, *Getting Started with Stata for Mac*, or *Getting Started with Stata for Unix* manual.

1.2.4 The index

At the end of each manual is an index for that manual. The *Glossary and Index* contains a **combined index** for all the manuals.

To find information and commands quickly, you can use Stata's **search** command; see [R] **search**. At the Stata command prompt, type **search geometric mean**. **search** searches Stata's keyword database and the Internet to find more commands and extensions for Stata written by Stata users.

1.2.5 The subject table of contents

A **subject table of contents** for the *User's Guide* and all the *Reference* manuals except the *Mata Reference Manual* is located in the *Glossary and Index*. This subject table of contents may also be accessed by clicking on **Contents** in the PDF bookmarks.

If you look under "Functions and expressions", you will see

[U]	Chapter 13	Functions and expressions
[D]	datetime	Date and time (%t) values and variables
[D]	egen	Extensions to generate
[D]	functions	Functions

1.2.6 Typography

We mix the ordinary typeface that you are reading now with a typewriter-style typeface that looks like this. When something is printed in the typewriter-style typeface, it means that something is a command or an option—it is something that Stata understands and something that you might actually type into your computer. Differences in typeface are important. If a sentence reads, "You could list the result ...", it is just an English sentence—you *could* list the result, but the sentence provides no clue as to how you might actually do that. On the other hand, if the sentence reads, "You could **list** the result ...", it is telling you much more—you could list the result, and you could do that by using the **list** command.

We will occasionally lapse into periods of inordinate cuteness and write, “We described the data and then listed the data.” You get the idea. `describe` and `list` are Stata commands. We purposely began the previous sentence with a lowercase letter. Because `describe` is a Stata command, it must be typed in lowercase letters. The ordinary rules of capitalization are temporarily suspended in favor of preciseness.

We also mix in words printed in italic type, such as “To perform the rank-sum test, type `ranksum varname`, by(*groupvar*)”. Italicized words are not supposed to be typed; instead, you are to substitute another word for them.

We would also like users to note our rule for punctuation of quotes. We follow a rule that is often used in mathematics books and British literature. The punctuation mark at the end of the quote is included in the quote only if it is a part of the quote. For instance, the pleased Stata user said she thought that Stata was a “very powerful program”. Another user simply said, “I love Stata.”

In this manual, however, there is little dialogue, and we follow this rule to precisely clarify what you are to type, as in, type “`cd c:`”. The period is outside the quotation mark because you should not type the period. If we had wanted you to type the period, we would have included two periods at the end of the sentence: one inside the quotation and one outside, as in, type “the orthogonal polynomial operator, *p*.”.

We have tried not to violate the other rules of English. If you find such violations, they were unintentional and resulted from our own ignorance or carelessness. We would appreciate hearing about them.

We have heard from Nicholas J. Cox of the Department of Geography at Durham University, UK, and express our appreciation. His efforts have gone far beyond dropping us a note, and there is no way with words that we can fully express our gratitude.

1.2.7 Vignette

If you look, for example, at the entry [\[R\] brier](#), you will see a brief biographical vignette of [Glenn Wilson Brier](#) (1913–1998), who did pioneering work on the measures described in that entry. A few such vignettes were added without fanfare in the Stata 8 manuals, just for interest, and many more were added in Stata 9, and even more have been added in each subsequent release. Ten new vignettes were added in Stata 13. A vignette could often appropriately go in several entries. For example, [George E. P. Box](#) deserves to be mentioned in entries other than [\[TS\] arima](#), such as [\[R\] boxcox](#). However, to save space, each vignette is given once only, and an [index](#) of all vignettes is given in the *Glossary and Index*.

Most of the vignettes were written by Nicholas J. Cox, Durham University, and were compiled using a wide range of reference books, articles in the literature, Internet sources, and information from individuals. Especially useful were the dictionaries of [Upton and Cook \(2014\)](#) and [Everitt and Skrondal \(2010\)](#) and the compilations of statistical biographies edited by [Heyde and Seneta \(2001\)](#) and [Johnson and Kotz \(1997\)](#). Of these, only the first provides information on people living at the time of publication.

1.3 What's new

This section is intended for users of the previous version of Stata. If you are new to Stata, you may as well skip to [\[U\] 1.3.12 What's more](#).

As always, Stata 13 is 100% compatible with the previous releases, but we remind programmers that it is important to put `version 12.1`, `version 12`, or `version 11`, etc., at the top of old do-

and ado-files so that they continue to work as you expect. You were supposed to do that when you wrote them, but if you did not, go back and do it now.

We will list all the changes, item by item, but first, here are the highlights.

1.3.1 What's new (highlights)

Here are the highlights. There are more, and do not assume that because we mention a category, we have mentioned everything new in the category. Detailed sections follow the highlights.

1. Long strings/BLOBs.

The maximum length of string variables increases from 244 to 2,000,000,000 characters. The standard string storage types `str1`, `str2`, ..., `str244` now continue to `str2045`, and after that comes `strL`, pronounced *sturl*. All of Stata's string functions work with two-billion-character-long strings, as do the rest of Stata's features, including importing, exporting, and ODBC. `strL` variables can contain binary strings. New functions, `fileread()` and `filewrite()`, make it easy to read and write entire files to and from `strL`s.

See [U] 12.4 Strings.

(BLOB stands for binary large object, jargon used by database programmers.)

2. Treatment effects.

A new suite of features allows you to estimate average treatment effects (ATE), average treatment effects on the treated (ATET), and potential-outcome means (POMs). Binary, multilevel, and multi-valued treatments are supported. You can model outcomes that are continuous, binary, count, or nonnegative.

Treatment-effects estimators measure the causal effect of treatment on an outcome in observational data.

Different treatment-effects estimators are provided for different situations.

When you know the determinants of participation (but not the determinants of outcome), inverse-probability weights (IPW) and propensity-score matching are provided.

When you know the determinants of outcome (but not the determinants of participation), regression adjustment and covariate matching are provided.

When you know the determinants of both, the doubly robust methods augmented IPW and IPW with regression adjustment are provided. These methods are doubly robust because you need to be right about only the specification of outcome, or of participation.

Also provided are two estimators that do not require conditional independence. Conditional independence means that the treatment and observed outcome are uncorrelated conditional on observed covariates. Put another way, conditional independence implies selection on observables. New estimation commands `etregress` and `etpoisson` relax the assumption. (`etregress` is an updated form of old command `treatreg`; `etpoisson` is new.)

See the all-new *Stata Treatment-Effects Reference Manual*, and in particular, see [TE] [teffects intro](#).

By the way, if treatment effects interest you, also see [SEM] [example 46g](#), where we use `gsem`—another new feature of Stata 13—to fit an endogenous treatment-effects model that can be modified to allow for generalized linear outcomes and multilevel effects.

3. Multilevel mixed effects and generalized linear structural equation modeling (SEM).

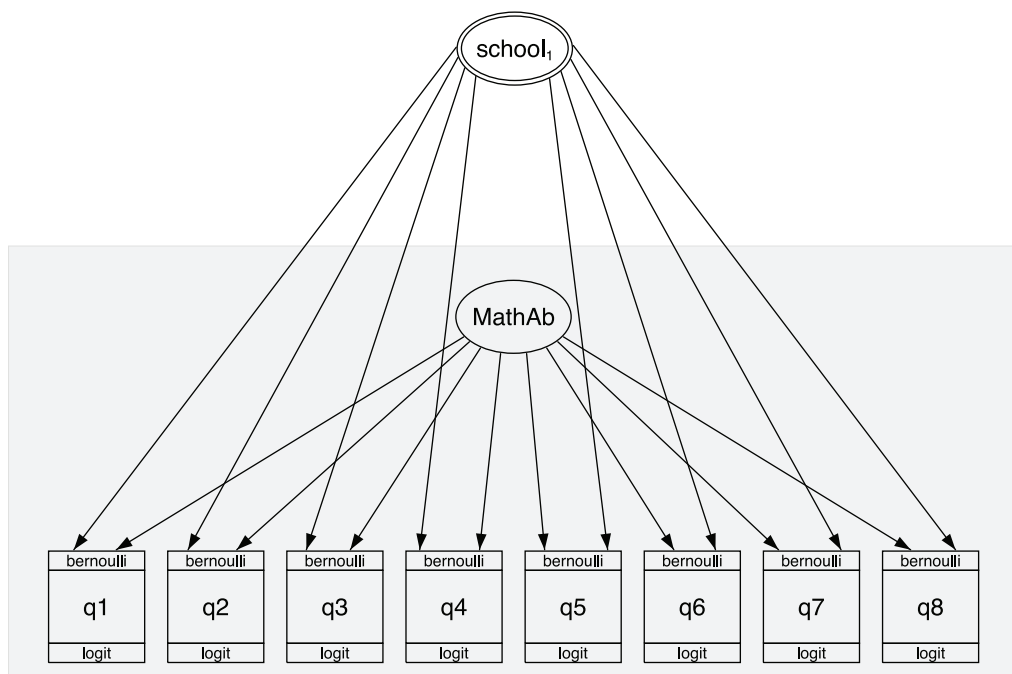
In addition to standard linear SEMs, Stata now provides what we are calling generalized SEMs for

short. Generalized SEMs allow for generalized linear response functions and allow for multilevel mixed effects.

Generalized linear response functions include binary outcomes (probit, logit, cloglog), count outcomes (Poisson, negative binomial), categorical outcomes (multinomial logit), ordered outcomes (ordered probit, ordered logit, ordered cloglog), and more, which is to say, generalized linear models (GLMs).

Multilevel mixed effects include nested random effects such as effects within patient within doctor within hospital and crossed random effects. Multilevel mixed effects also include random intercepts and random slopes.

In the language of SEM, “multilevel mixed effects” means latent variables at different levels of the data. This means Stata 13 can fit multilevel measurement models and multilevel structural equation models.



See [\[SEM\] intro 1](#).

Economists: See [\[SEM\] example 45g](#), where we show how to use Stata 13’s new SEM features to fit the Heckman selection model, which can be extended to generalized linear outcomes and random effects and random slopes.

4. New multilevel mixed-effects models.

Multilevel mixed-effects estimation has been improved and expanded and is now the subject of its own manual. Stata had 3 multilevel estimation commands; now it has 11.

The eight new multilevel mixed-effects estimation commands are

<code>melogit</code>	logistic regression
<code>meprobit</code>	probit regression
<code>mecloglog</code>	complementary log-log regression
<code>meologit</code>	ordered logistic regression
<code>meoprobit</code>	ordered probit regression
<code>mepoisson</code>	Poisson regression
<code>menbreg</code>	negative binomial regression
<code>meglm</code>	generalized linear models

These new estimation commands allow for constraints on variance components, provide robust and cluster-robust standard errors, and are fast.

The three existing multilevel estimation commands have been renamed: `xtmixed` is now `mixed`, `xtmelogit` is now `meqrlogit`, and `xtmepoisson` is now `meqrpoisson`. All three now present results by default in the variance metric rather than the standard deviation metric.

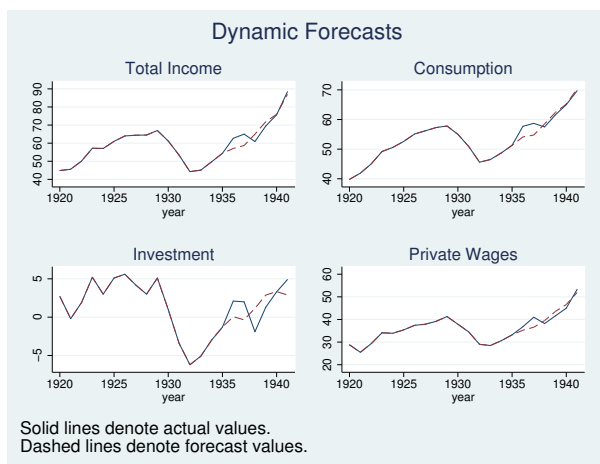
As we said, multilevel mixed-effects modeling is now the subject of its own manual. See [Stata Multilevel Mixed-Effects Reference Manual](#), and in particular, see [ME] [me](#).

5. Forecasts based on systems of equations.

Stata's new `forecast` command allows you to combine estimation results from multiple Stata commands or other sources to produce dynamic or static forecasts and produce forecast intervals.

You begin by fitting the equations of your model using Stata's estimation commands, or you can enter results that you obtained elsewhere. Then you use `forecast` to specify identities and exogenous variables to obtain a baseline forecast. Once you produce the baseline forecast, you can specify alternative paths for some variables and obtain forecasts based on those alternative paths. Thus you can produce forecasts under alternative scenarios and explore impacts of differing policies.

You can use `forecast`, for example, to produce macroeconomic forecasts.



In addition, `forecast` is particularly easy to use because `forecast` also provides an intuitive, interactive control panel to guide you and, if you do something wrong, `forecast` itself offers advice on how to fix the problem.

See [TS] `forecast`.

6. Power and sample size.

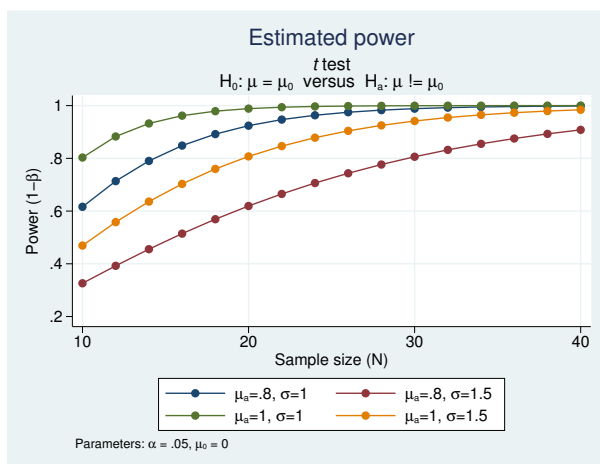
The new `power` command performs power and sample-size analysis. Included are

- Comparison of a mean to a reference value
- Comparison of a proportion to a reference value
- Comparison of a variance to a reference value
- Comparison of a correlation to a reference value

- Comparison of two independent means
- Comparison of two independent proportions
- Comparison of two independent variances
- Comparison of two independent correlations

- Comparison of two paired means
- Comparison of two paired proportions

Results can be displayed in customizable tables and graphs.



An integrated GUI lets you select your analysis type, input assumptions, and obtain desired results. Power and sample size is the subject of its own manual. See [Stata Power and Sample-Size Reference Manual](#); start by seeing [PSS] [intro](#).

7. New and extended panel-data estimators.

Two new random-effects panel-data estimation commands are added:

<code>xtoprobit</code>	ordered probit regression
<code>xtologit</code>	ordered logistic regression

These new commands allow for cluster-robust standard errors.

The following previously existing random-effects panel-data estimation commands now allow for cluster-robust standard errors:

<code>xtprobit</code>	probit regression
<code>xtlogit</code>	logistic regression
<code>xtcloglog</code>	complementary log-log regression
<code>xtpoisson</code>	Poisson regression

See [XT] `xt` for a complete list of all of Stata's panel-data estimators.

8. New commands are provided for calculating effect sizes after estimation in the way behavioral scientists, and especially psychologists, want to see them. Cohen's d , Hedges's g , Glass's Δ , η^2 , and ω^2 , with confidence intervals, are now provided:

- a. New commands `esize` and `esizei` calculate effect sizes comparing the difference between the means of a continuous variable for two groups. See [R] `esize`.
- b. New postestimation command `estat esize` computes effect sizes for linear models after `anova` and `regress`. See [R] `regress postestimation`.

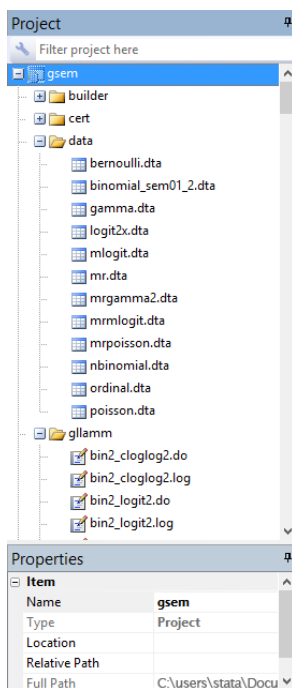
9. Project Manager.

The new Project Manager lets you organize your analysis files—your do-files, ado-files, datasets, raw files, etc. You can have multiple projects, and each can contain hundreds of files, or just a few.

You can see all the files in a project at a glance, filter on filenames, and click to open, edit, or run.

Projects are portable, meaning that you can pick the whole collection up at once and move it across computers or share it with colleagues.

Take a look:



Try it. Get started from the Do-file Editor by selecting **File > New > Project ...**

See [P] [Project Manager](#).

10. Java plugins.

You can now call Java methods directly from Stata. You can take advantage of the plethora of existing Java libraries or write your own Java code. You call Java using Stata's new `javacall` command. See [P] [java](#) and see the Java-Stata API specification at <http://www.stata.com/java/api/>.

Java recently encountered some negative publicity regarding security concerns. That publicity was about Java and web browsers *automatically* loading and running Java code from untrusted websites. It does not apply to Stata's implementation of Java. Stata's implementation is about running Java code already installed on your computer from known and trusted sources.

1.3.2 What's new that you will want to know

11. You can clear the Results window.

Use the new `cls` command. See [R] [cls](#).

12. Value labels of factor variables used to label output.

You use variable `i.sex`, and output now shows `male` and `female` in your model rather than 0 and 1 if variable `sex` has a value label. You can control how output looks. See more details below in [U] [1.3.3 What's new in statistics \(general\)](#).

13. Programmers can create Word and Excel files from Stata.

You can add paragraphs, insert images, insert tables, poke into individual cells, and more.

See [M-5] `_docx*()` to create Word documents.

See [P] `putexcel` and [M-5] `xl()` to interact with Excel files.

By the way, Stata could already import and export Excel files; see [D] `import excel`.

14. Searching is better.

Here's why:

- a. `Help > Search...` and the `search` command now default to searching the Internet as well as Stata's local keyword database. If you do not want that, type `set searchdefault local, permanently` to set Stata 13 to the old default.
- b. `search` without options now displays its results in the Viewer rather than in the Results window. (If any options are specified, however, results appear in the Results window.)
- c. Existing command `findit` is no longer documented but continues to work. Changes to `search` make `search` into the equivalent of `findit`.

See [R] `search`.

15. help now searches when no help is found.

`help xyz` now invokes `search xyz` if `xyz` is not found. See [R] `help`.

16. Stata now supports secure HTTP (HTTPS) and FTP. You can, for instance, use datasets from sites using either of the protocols. See [U] 3.6 Updating and adding features from the web.

17. Concerning the Data Editor,

- a. noncontiguous column selections are now allowed.
- b. `encode`, `decode`, `destring`, and `tostring` have been added as operations that can be performed on selected variables.
- c. the *Delete* key can now be used to drop data.

See [GS] 6 Using the Data Editor (*GSM*, *GSU*, or *GSW*).

18. Concerning the Do-file Editor,

- a. matching braces are highlighted.
- b. an adjustable column guide has been added.
- c. you can now zoom in and out.
- d. you can convert between the different types of end-of-line characters used by Windows and by Mac and Unix.

See [GS] 13 Using the Do-file Editor (*GSM*, *GSU*, or *GSW*).

19. Concerning Stata's GUI,

- a. the Properties window now displays the sorted-by variables.
- b. the **Jump To** menu in the Viewer now allows you to jump to the top of the page.
- c. Stata for Windows now supports Windows high-contrast themes.

20. .dta file format has changed.

The file format has changed because of the new `strL` variables. Stata 13 can, of course, read old-format datasets. If you need to create datasets in the previous format—used by Stata 11 and Stata 12—use the `saveold` command. See [D] `save`. If you want to know the details of the new `.dta` format, type `help dta`.

21. Official directory ado/updates no longer used.

Official ado-file updates are no longer stored in directory *installation-directory/ado/updates/*. Updates are now applied to *ado/base* directly. Modern operating systems do not approve of applications such as Stata having multiple files of the same name. The updates process remains the same.

22. Videos.

Type `help videos` to list and link to the videos on Stata's YouTube channel. We provide dozens of tutorials on Stata's features.

23. Fast PDF-manual navigation.

There are now links at the top of each manual entry to jump directly to section headings, and on each page's header, there is a link to take you to the beginning of the entry.

If you did not know already, clicking on the blue manual reference in the title of a help file jumps to the PDF documentation.

24. Manuals have color graphs.

If you want to use the same color graph scheme we use in the manuals, type `set scheme s2gcolor`. See [G-4] [scheme s2](#).

25. Ten new vignettes.

Scientific history buffs will want to read about the following:

- a. [Florence Nightingale](#)
- b. [Florence Nightingale David](#), a different person from Florence Nightingale
- c. [Charles William Dunnett](#)
- d. [Andrew Charles Harvey](#)
- e. [William Lee Hays](#)
- f. [Fred Nichols Kerlinger](#)
- g. [Janet Elizabeth Lane-Claypon](#)
- h. [martingale](#)
- i. [Elizabeth L. "Betty" Scott](#)
- j. [John Snow](#)

The following two items were added during the Stata 12 release:

26. New command `icc` computes intraclass correlation coefficients for one-way random-effects models, two-way random-effects models, and two-way mixed-effects models for both individual and average measurements. Intraclass correlations measure consistency of agreement or absolute agreement. See [R] [icc](#).
27. New postestimation command `estat icc` computes intraclass correlations at each nesting level for nested random-effects models fit by `mixed` and `melogit`. See [ME] [mixed postestimation](#) and [ME] [melogit postestimation](#).

1.3.3 What's new in statistics (general)

Already mentioned as highlights of the release were [treatment effects](#), [generalized SEMs](#), [multilevel mixed-effects models](#), [power and sample size](#), and [panel-data estimators](#). The following are also new:

28. Concerning sample-selection estimation commands,
 - a. new estimation command `heckoprobit` fits the parameters of an ordered probit model with sample selection. See [\[R\] heckoprobit](#).
 - b. existing estimation command `heckprob` is renamed `heckprobit`. See [\[R\] heckprobit](#).
29. Existing estimation command `hetprob` is renamed `hetprobit`. See [\[R\] hetprobit](#).
30. New estimation command `ivpoisson` fits the parameters of a Poisson regression model with endogenous regressors. Estimates can be obtained using the GMM or control-function estimators. See [\[R\] ivpoisson](#).
31. New command `mlexp` allows you to specify maximum likelihood models without writing an evaluator program. You can instead specify an expression representing the log-likelihood function in much the same way you would with `nl`, `nlsur`, or `gmm`. See [\[R\] mlexp](#).
32. Concerning fractional polynomials,

- a. new prefix command `fp`: replaces `fracpoly` for fitting models with fractional polynomial regressors. You type

```
. fp ...: estimation command
```

Results are the same. The new `fp` command supports more estimation commands, it is easier to use, and it is more flexible. You can substitute the same fractional polynomial into multiple places of the estimation command, which is especially useful in multiple-equation models. You may now use factor-variable notation in the estimation command.

- b. `fp generate` replaces `fracgen`.
- c. `fp plot` replaces `fracplot`.
- d. `fp predict` replaces `fracpred`.
- e. commands `fracpoly` and `fracgen` are no longer documented but continue to work. Commands `fracplot` and `fracpred` are still documented for use after [mfp](#).

See [\[R\] fp](#).

33. Concerning quantile-regression estimation commands,
 - a. existing estimation command `qreg` now accepts option `vce(robust)`.
 - b. existing estimation commands `qreg`, `iqreg`, `sqreg`, and `bsqreg` now allow factor variables to be used.

See [\[R\] qreg](#).

34. Syntax and methodology for `predict` after `boxcox` have changed. Predicted values are now calculated using Duan's smearing method by default. The previous back-transformed predicted-values estimates are provided if `predict`'s `btransform` option is specified and under version control. See [\[R\] boxcox postestimation](#).
35. Value labels of factor variables are now used by default to label estimation output. The numeric values (levels) were previously used and continue to be used if the factor variables are unlabeled. There are three new display options that may be used with estimation commands affecting how this works:

- a. Option `novlabel` displays factor-variable level values, just as Stata 12 did previously. (You can set `fvlabel off` to make `novlabel` the default.)
- b. Option `fvwrap(#)` specifies the number of lines to allow when long value labels must be wrapped. Labels requiring more than `#` lines are truncated. `fvwrap(1)` is the default. You can change the default by using `set fvwrap #`.
- c. Option `fvwraon()` specifies whether value labels that wrap will break at word boundaries. `fvwraon(word)` is the default, meaning to break at word boundaries.
`fvwraon(width)` specifies that line breaks may occur arbitrarily so as to maximize use of available space.

You can change the defaults by using `set fvwraon width` or `set fvwraon word`.

Current default settings are shown by `query` and also stored in `c(fvlabel)`, `c(fvwrap)`, and `c(fvwraon)`.

See [R] [set showbaselevels](#) and [P] [creturn](#).

36. Existing estimation command `proportion` now uses the logit transform when computing the limits of the confidence interval. The original behavior of using the normal approximation is preserved under version control or when the new `citype(normal)` option is specified. See [R] [proportion](#).
37. Concerning existing command `margins`,
 - a. option `at()` has new suboption `generate()`, which allows you to specify an expression to replace the values for any continuous variable in the model. For example, you can compute the predictive margins at `x+1` by typing

```
. margins, at(x = generate(x+1))
```

`at(generate())` can be combined with contrasts to estimate the effect of giving each subject an additional amount of `x`,

```
. margins, at((asobserved) _all) at(x= generate(x+1)) contrast(at(r._at))
```

See *Estimating treatment effects with margins* in [R] [margins, contrast](#).

- b. `margins` automatically uses the *t* distribution for computing *p*-values and confidence intervals when appropriate, which is after linear regression and ANOVA and whenever degrees of freedom are posted to `e(df_r)`.

The previous default behavior of always using the standard normal distribution for all *p*-values and confidence intervals is preserved under version control.

- c. new option `df(#)` specifies that `margins` is to use the *t* distribution when it otherwise would not.

See [R] [margins](#).

38. `nlcom` and `predictnl` now use the standard normal distribution for computing *p*-values and confidence intervals. Original behavior was to compute the *p*-values and CIs based on the *t* distribution in some cases. Original behavior is preserved under version control. In addition, if you want *p*-values and confidence intervals calculated using the *t* distribution, use new option `df(#)` to specify the degrees of freedom.

`testnl`'s calculated test statistic is now χ^2 rather than *F* unless you specify the `df()` option.

See [R] [nlcom](#), [R] [predictnl](#), and [R] [testnl](#).

39. `contrast`, `pwcompare`, and `lincom` have new option `df(#)` to use the t distribution in computing p -values and confidence intervals. For `contrast`, this option also causes the Wald table to use the F distribution.
- See [R] [contrast](#), [R] [pwcompare](#), and [R] [lincom](#).
40. `estimates table`'s option `label` is renamed `varlabel`. Original option `label` is allowed under version control. See [R] [estimates table](#).
41. The previously existing `sampsi` command is no longer documented because it is replaced by the new `power` command—a highlight of the release. See [PSS] [power](#).
42. Existing functions `normalden(x, μ, σ)` and `lnnormalden(x, μ, σ)` now allow you to omit argument μ or arguments μ and σ . $\mu = 0$ and $\sigma = 1$ is assumed. See `help normalden()`, `help lnnormalden()`, and [D] [functions](#).
43. The following new functions are added:

<code>t(df, t)</code>	cumulative Student's t distribution
<code>invt(df, p)</code>	inverse cumulative Student's t distribution
<code>ntden(df, np, t)</code>	density of noncentral Student's t distribution
<code>nt(df, np, t)</code>	cumulative noncentral Student's t distribution
<code>npnt(df, t, p)</code>	noncentrality parameter of noncentral Student's t distribution
<code>nttail(df, np, t)</code>	right-tailed noncentral Student's t distribution
<code>invnttail(df, np, p)</code>	inverse of right-tailed noncentral Student's t distribution
<code>nF(df₁, df₂, np, f)</code>	cumulative noncentral F distribution
<code>nFnF(df₁, df₂, f, p)</code>	noncentrality parameter of noncentral F distribution
<code>chi2den(df, x)</code>	density of χ^2 distribution
<code>fileread(f)</code>	return the contents of a file as a string
<code>filewrite(f, s[, r])</code>	create or overwrite file with the contents of a string
<code>fileexists(f)</code>	check whether a file exists
<code>filereaderror(s)</code>	use results returned by <code>fileread()</code> to determine whether an I/O error occurred

See `help functionname()` and [D] [functions](#).

1.3.4 What's new in statistics (SEM)

We have already mentioned [a highlight of the release](#), the new `gsem` command, for fitting generalized SEMs. The following are also new:

44. Existing estimation command `sem` has new option `noestimate`, which is useful when you are having convergence problems; you can use it to get the starting values into a Stata matrix (vector) that you can then modify to use as alternative starting values. See [SEM] [intro 12](#).
45. `sem` now supports time-series operators on all observed variables. See [SEM] [sem](#).
46. You can now use postestimation command `margins` after `sem`. See [SEM] [intro 7](#).
47. `sem` no longer reports in the estimation output any zero-valued constraints on covariances between exogenous variables; absence of the covariance indicates the presence of the constraint. Original behavior is preserved under version control.

48. The new options for controlling display of factor variables with value labels mentioned in [U] 1.3.3 **What's new in statistics (general)**—`nofvlabel`, `fvwrap(#)`, and `fvwrapon(word|width)`—work with *varname* of `sem`, `group(varname)`. `sem` itself does not allow factor variables, but the factor-variable display options nonetheless work with `group(varname)`.

Thus old options `wrap()` and `nolabel` are now officially `fvwrap()` and `fvnolabel`, although the old option names continue to work as synonyms. See [SEM] **sem reporting options**.

49. We now show how to construct path diagrams at the end of each estimation example in the manual. See [SEM] **example 1**, [SEM] **example 3**, . . .

1.3.5 What's new in statistics (time series)

We have already mentioned a [highlight of the release](#), the new `forecast` command. The following are also new:

50. New command `import haver` (available with Stata for Windows only) replaces old command `haver`. `import haver` imports economic and financial data from Haver Analytics databases. See [D] **import haver**.
51. Existing command `tsreport` now provides better information about gaps in time-series and panel datasets, including the length of each gap.

In addition, `tsreport` will provide information about missing values in variables even where there are no gaps.

See [TS] **tsreport**.

Also see item 55 in [U] 1.3.8 **What's new in data management** for information on the new command `bcal create`.

1.3.6 What's new in statistics (longitudinal/panel data)

We have already mentioned a [highlight of the release](#), new and extended panel-data estimators.

1.3.7 What's new in statistics (survival analysis)

52. Shared frailty survival models can no longer be fit when there is delayed entry or there are gaps in time under observation. Said differently, `stcox` and `streg` no longer allow option `shared()` when there are delayed entry or gaps. The use of shared frailty models to fit truncated survival data leads to inconsistent results unless the frailty distribution is independent of the covariates and the truncation point, which rarely happens in practice. If you have such data and can make the independence assumption—which is unlikely—estimation can be forced by specifying undocumented option `forceshared`. See [ST] **stcox** and [ST] **streg**. See `help st_forceshared` for information on the `forceshared` option.
53. Output produced by existing commands `stset`, `streset`, and `cttost` more accurately labels time at risk. What was labeled “total time at risk” is now labeled “total time at risk and under observation”. See [ST] **stset** and [ST] **cttost**.

1.3.8 What's new in data management

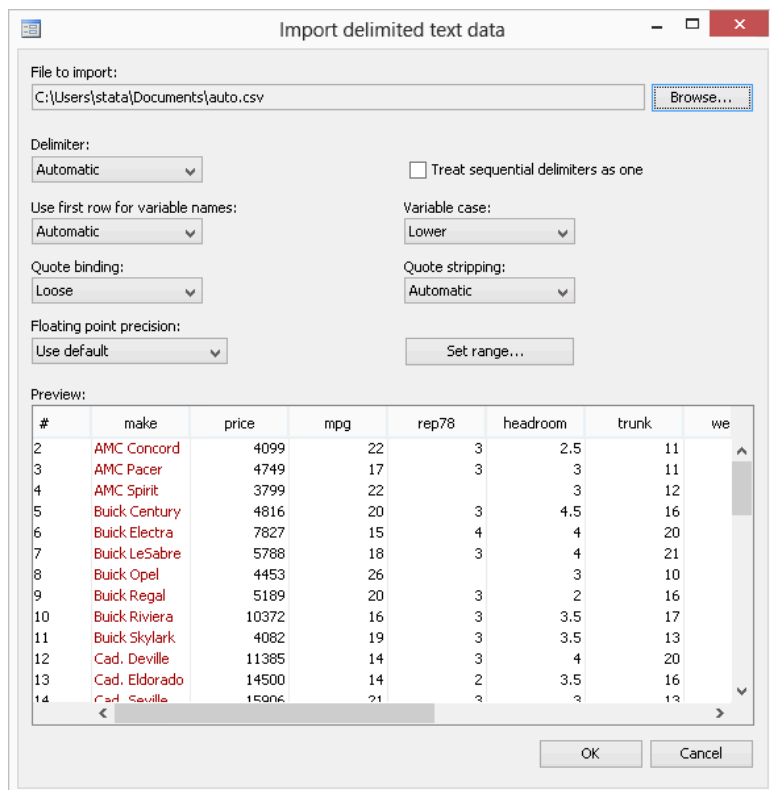
We have already mentioned a [highlight of the release](#), long strings/BLOBs.

54. New commands `import delimited` and `export delimited` supersede old commands `insheet` and `outsheet`. This is not just a renaming.

`import delimited` supports several different quoting methods. Some packages, for instance, use `"` in the middle of a string to represent an embedded double quote. Others do not.

`import delimited` now allows column and row ranges (subsets).

Use `import delimited`'s GUI to see a preview of the data and how they will be read. You can also customize the GUI.



Of course, `import delimited` and `export delimited` support Stata 13's new `strLs`.

See [\[D\] import delimited](#).

55. existing command `bcal` has new subcommand `create` to create a business calendar from the current dataset automatically. `bcal create` infers business holidays and closures from gaps in the data. See [\[D\] bcal](#).

56. String expressions now support string duplication via multiplication. For example, `3*"abc"` evaluates to `"abcabcabc"`. See `help strdup()` or [\[D\] functions](#).

57. Concerning long strings, that is, `strLs`,
- existing command `compress` has new option `nocoalesce` in support of the new `strL` string storage type. By default, `compress` coalesces the storage used to store duplicated `strL` values. `nocoalesce` prevents this.
In addition, `compress` always considers demoting `strL` variables to `str#` variables if that would save memory.
See [D] `compress`.
 - the output of existing command `memory` has changed to include information on new string storage type `strL`. See [D] `memory`.
 - the options of existing command `ds`, such as `has()` and `not()`, now understand `string` to mean both `strL` and `str#`, `strL` to mean `strL`, and `str#` to mean `str1`, `str2`, ..., `str2045`. See [D] `ds`.
 - existing command `type` has new option `lines(#)` to list the first `#` lines of the file. See [D] `type`.

Also see item 50 in [U] 1.3.5 **What's new in statistics (time series)** for information on the new command `import haver`.

1.3.9 What's new in Mata

58. **Programmers can create Word and Excel files from Stata.**
You can add paragraphs, insert images, insert tables, poke into individual cells, and more.
See [M-5] `_docx*()` to create Word documents.
See [P] `putexcel` and [M-5] `xl()` to interact with Excel files.
By the way, Stata could already import and export Excel files; see [D] `import excel`.
59. New functions in `solvenl()` allow you to solve arbitrary systems of nonlinear equations. Gauss–Seidel, damped Gauss–Seidel, Broyden–Powell, and Newton–Raphson techniques are provided. See [M-5] `solvenl()`.
60. The same statistical functions added to Stata have been added to Mata, namely,

Noncentral Student's t

$$\begin{aligned} p &= \text{nt}(df, np, t) \\ d &= \text{ntden}(df, np, t) \\ q &= \text{nttail}(df, np, t) \\ t &= \text{invnttail}(df, np, q) \\ np &= \text{npnt}(df, t, p) \end{aligned}$$

Student's t

$$\begin{aligned} p &= \text{t}(df, t) \\ t &= \text{invt}(df, p) \end{aligned}$$

Noncentral F

$$\begin{aligned} p &= \text{nF}(df_1, df_2, np, f) \\ np &= \text{npnF}(df_1, df_2, f, p) \end{aligned}$$

χ^2

$$d = \text{chi2den}(df, x)$$

See [M-5] `normal()`.

61. New function `selectindex()` returns a vector of indices for which $v[j] \neq 0$. For instance, if $v = (6, 0, 7, 0, 8)$, then `selectindex(v) = (1, 3, 5)`. `selectindex()` is useful with logical expressions, such as `x[selectindex(x:>1000)]`. See [M-5] `select()`.

1.3.10 What's new in programming

We have already mentioned the [Project Manager](#) and [Java plugins](#) as highlights of the release. The following are also new:

62. New command `putexcel` writes Stata expressions, matrices, and stored results to an Excel file. Excel 1997/2003 (.xls) files and Excel 2007/2010 (.xlsx) files are supported. See [P] `putexcel`. Mata programmers will also be interested in [M-5] `xl()`, a class to interact with Excel files.
63. A new set of Mata functions provide the ability to create Word documents. See [M-5] `_docx*()`.
64. Concerning `strL`s,
- `strL` is now a reserved word.
 - the maximum length of a string in string expressions increases from 244 to 2-billion characters. See [R] `limits`.
 - new `c(maxstrlvarlen)` returns the maximum possible length for `strL` variables.
 - `confirm ... variable` now understands `str#` to mean any `str1`, `str2`, ..., `str2045` variable; `strL` to mean `strL`; and `string` to mean `str#` or `strL`. See [P] `confirm`.
 - new function `fileread(filename [, startpos [, length]])` returns the contents of `filename`. See `help fileread()` and [D] `functions`.
 - new function `filewrite(filename, s [, 1|2])` writes `s` to the specified `filename`, optionally overwriting 1 or appending 2. See `help filewrite()` and [D] `functions`.
 - new function `fileexists(filename)` returns 1 if the specified `filename` exists, and returns 0 otherwise.
 - new function `filereaderror(s)` returns 0 or a positive integer, said value having the interpretation of a return code. It is used like this

```
. generate strL s = fileread(filename) if fileexists(filename)
. assert filereaderror(s)==0
```

or this

```
. generate strL s = fileread(filename) if fileexists(filename)
. generate rc = filereaderror(s)
```

That is, `filereaderror(s)` is used on the result returned by `fileread(filename)` to determine whether an I/O error occurred.

In the example, we only `fileread()` files that `fileexist()`. That is not required. If the file does not exist, that will be detected by `filereaderror()` as an error. The way we showed the example, we did not want to read missing files as errors. If we wanted to treat missing files as errors, we would have coded

```
. generate strL s = fileread(filename)
. assert filereaderror(s)==0
```

or

```
. generate strL s = fileread(filename)
. generate rc = filereaderror(s)
```

65. New command `expr_query exp` returns in `r()` the variables used in expression `exp`. See `help undocumented` and see `help expr_query`.
66. The maximum number of elements in a numlist increases from 1,600 to 2,500. See [U] 11.1.8 [numlist](#).
67. Existing command `ereturn post` now allows posting of noninteger as well as integer `dof()` values.
68. New `c(hostname)` returns the computer's hostname. See [P] [creturn](#).
69. New `c(maxvlabellen)` returns the maximum possible length for a value label.

1.3.11 What's new, Mac only

In addition to all the above What's New items, which apply to all platforms, Stata for Mac has several of its own new features:

70. The Do-file Editor in Stata for Mac has been completely rewritten. It now includes
 - code folding
 - more robust syntax highlighting that is consistent with highlighting in Windows and Unix
 - more color options for customizing its appearance
 - the ability to save the syntax-highlighting colors as separate themes
 - line ending preservation and normalization, which is useful for working in a mixed platform environment where do-files are exchanged between Windows and Macs
 - text-size zooming without having to change the font or font size
 - more drag-and-drop options
 - more control over the appearance of printed files
71. The Command window now has the same syntax highlighting as the Do-file Editor.
72. There is a new path control that not only shows the current working directory but also can change the current working directory and open Stata files without having to use the Open dialog.
73. Mac OS X 10.7 GUI enhancements such as full-screen support and textured backgrounds for spring-back scrolling are now supported.
74. There is a new interface for saving and managing saved preferences.
75. Applescript is better supported and enables users to directly access Stata macros, scalars, stored results, and datasets.
76. Stata for Mac is now 64-bit only and allows the application's file size to be roughly 67% smaller.

1.3.12 What's more

We have not listed all the changes, but we have listed the important ones.

Stata is continually being updated. Those between-release updates are available for free over the Internet.

Type update query and follow the instructions.

We hope that you enjoy Stata 13.

1.4 References

- Chambers, J. M., W. S. Cleveland, B. Kleiner, and P. A. Tukey. 1983. *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth.
- Everitt, B. S., and A. Skrondal. 2010. *The Cambridge Dictionary of Statistics*. 4th ed. Cambridge: Cambridge University Press.
- Heyde, C. C., and E. Seneta, ed. 2001. *Statisticians of the Centuries*. New York: Springer.
- Johnson, N. L., and S. Kotz, ed. 1997. *Leading Personalities in Statistical Sciences: From the Seventeenth Century to the Present*. New York: Wiley.
- Upton, G. J. G., and I. T. Cook. 2014. *A Dictionary of Statistics*. 3rd ed. Oxford: Oxford University Press.