

Description

The *Time-Series Reference Manual* organizes the commands alphabetically, making it easy to find individual command entries if you know the name of the command. This overview organizes and presents the commands conceptually, that is, according to the similarities in the functions that they perform. The table below lists the manual entries that you should see for additional information.

Data management tools and time-series operators.

These commands help you prepare your data for further analysis.

Univariate time series.

These commands are grouped together because they are either estimators or filters designed for univariate time series or preestimation or postestimation commands that are conceptually related to one or more univariate time-series estimators.

Multivariate time series.

These commands are similarly grouped together because they are either estimators designed for use with multivariate time series or preestimation or postestimation commands conceptually related to one or more multivariate time-series estimators.

Forecasting models.

These commands work as a group to provide the tools you need to create models by combining estimation results, identities, and other objects and to solve those models to obtain forecasts.

Within these three broad categories, similar commands have been grouped together.

Data management tools and time-series operators

[TS] tsset	Declare data to be time-series data
[TS] tsfill	Fill in gaps in time variable
[TS] tsappend	Add observations to a time-series dataset
[TS] tsreport	Report time-series aspects of a dataset or estimation sample
[TS] tsrevar	Time-series operator programming command
[TS] rolling	Rolling-window and recursive estimation
[D] Datetime business calendars	User-definable business calendars
[D] import fred	Import data from Federal Reserve Economic Data

Univariate time series**Estimators**

[TS] arfima	Autoregressive fractionally integrated moving-average models
[TS] arfima postestimation	Postestimation tools for arfima
[TS] arima	ARIMA, ARMAX, and other dynamic regression models
[TS] arima postestimation	Postestimation tools for arima
[TS] arch	Autoregressive conditional heteroskedasticity (ARCH) family of estimators
[TS] arch postestimation	Postestimation tools for arch
[TS] mswitch	Markov-switching regression models
[TS] mswitch postestimation	Postestimation tools for mswitch
[TS] newey	Regression with Newey–West standard errors
[TS] newey postestimation	Postestimation tools for newey
[TS] prais	Prais–Winsten and Cochrane–Orcutt regression
[TS] prais postestimation	Postestimation tools for prais
[TS] ucm	Unobserved-components model
[TS] ucm postestimation	Postestimation tools for ucm
[TS] threshold	Threshold regression
[TS] threshold postestimation	Postestimation tools for threshold

Time-series smoothers and filters

[TS] tsfilter bk	Baxter–King time-series filter
[TS] tsfilter bw	Butterworth time-series filter
[TS] tsfilter cf	Christiano–Fitzgerald time-series filter
[TS] tsfilter hp	Hodrick–Prescott time-series filter
[TS] tssmooth ma	Moving-average filter
[TS] tssmooth dexponential	Double-exponential smoothing
[TS] tssmooth exponential	Single-exponential smoothing
[TS] tssmooth hwinters	Holt–Winters nonseasonal smoothing
[TS] tssmooth shwinters	Holt–Winters seasonal smoothing
[TS] tssmooth nl	Nonlinear filter

Diagnostic tools

[TS] arfimasoc	Obtain lag-order selection statistics for ARFIMAs
[TS] arimasoc	Obtain lag-order selection statistics for ARMA
[TS] corrgram	Tabulate and graph autocorrelations
[TS] xcorr	Cross-correlogram for bivariate time series
[TS] cumsp	Graph cumulative spectral distribution
[TS] pergram	Periodogram
[TS] psdensity	Parametric spectral density estimation
[TS] estat acplot	Plot parametric autocorrelation and autocovariance functions
[TS] estat aroots	Check the stability condition of ARIMA estimates
[TS] dfgls	DF-GLS unit-root test
[TS] dfuller	Augmented Dickey–Fuller unit-root test
[TS] pperron	Phillips–Perron unit-root test
[TS] estat sbknown	Test for a structural break with a known break date
[TS] estat sbsingle	Test for a structural break with an unknown break date
[TS] estat sbcusum	Cumulative sum test for parameter stability
[R] regress postestimation time series	Postestimation tools for regress with time series
[TS] mswitch postestimation	Postestimation tools for mswitch
[TS] wntestb	Bartlett’s periodogram-based test for white noise
[TS] wntestq	Portmanteau (Q) test for white noise

Multivariate time series**Estimators**

[TS] dfactor	Dynamic-factor models
[TS] dfactor postestimation	Postestimation tools for dfactor
[TS] lpirf	Local-projection impulse–response functions
[TS] lpirf postestimation	Postestimation tools for lpirf
[TS] ivlpirf	Instrumental-variables local-projection impulse–response functions
[TS] ivlpirf postestimation	Postestimation tools for ivlpirf
[TS] mgarch ccc	Constant conditional correlation multivariate GARCH model
[TS] mgarch ccc postestimation	Postestimation tools for mgarch ccc
[TS] mgarch dcc	Dynamic conditional correlation multivariate GARCH model
[TS] mgarch dcc postestimation	Postestimation tools for mgarch dcc
[TS] mgarch dvec	Diagonal vech multivariate GARCH model
[TS] mgarch dvec postestimation	Postestimation tools for mgarch dvec
[TS] mgarch vcc	Varying conditional correlation multivariate GARCH model
[TS] mgarch vcc postestimation	Postestimation tools for mgarch vcc
[TS] sspace	State-space models
[TS] sspace postestimation	Postestimation tools for sspace
[TS] var	Vector autoregressive models
[TS] var postestimation	Postestimation tools for var
[TS] var svar	Structural vector autoregressive models
[TS] var svar postestimation	Postestimation tools for svar
[TS] var ivsvar	Instrumental-variables structural vector autoregressive models
[TS] var ivsvar postestimation	Postestimation tools for ivsvar
[TS] varbasic	Fit a simple VAR and graph IRFs or FEVDs
[TS] varbasic postestimation	Postestimation tools for varbasic
[TS] vec	Vector error-correction models
[TS] vec postestimation	Postestimation tools for vec

Also see *Stata Dynamic Stochastic General Equilibrium Models Reference Manual*.

Diagnostic tools

[TS] varlmar	Perform LM test for residual autocorrelation
[TS] varnorm	Test for normally distributed disturbances
[TS] varsoc	Obtain lag-order selection statistics for VAR and VEC models
[TS] varstable	Check eigenvalue stability condition
[TS] varwle	Obtain Wald lag-exclusion statistics
[TS] veclmar	Perform LM test for residual autocorrelation
[TS] vecnorm	Test for normally distributed disturbances
[TS] vecrank	Estimate the cointegrating rank of a VEC model
[TS] vecstable	Check the stability condition of VEC model estimates

Forecasting, inference, and interpretation

[TS] irf create	Obtain IRFs, dynamic-multiplier functions, and FEVDs
[TS] fcast compute	Compute dynamic forecasts
[TS] vargranger	Perform pairwise Granger causality tests

Graphs and tables

[TS] corrgram	Tabulate and graph autocorrelations
[TS] xcorr	Cross-correlogram for bivariate time series
[TS] pergram	Periodogram
[TS] irf graph	Graphs of IRFs, dynamic-multiplier functions, and FEVDs
[TS] irf cgraph	Combined graphs of IRFs, dynamic-multiplier functions, and FEVDs
[TS] irf ograph	Overlaid graphs of IRFs, dynamic-multiplier functions, and FEVDs
[TS] irf table	Tables of IRFs, dynamic-multiplier functions, and FEVDs
[TS] irf ctable	Combined tables of IRFs, dynamic-multiplier functions, and FEVDs
[TS] fcast graph	Graph forecasts after fcast compute
[TS] tsline	Plot time-series data
[TS] varstable	Check eigenvalue stability condition
[TS] vecstable	Check the stability condition of VEC model estimates
[TS] wntestb	Bartlett's periodogram-based test for white noise

Results management tools

[TS] irf add	Add results from an IRF file to the active IRF file
[TS] irf describe	Describe an IRF file
[TS] irf drop	Drop IRF results from the active IRF file
[TS] irf rename	Rename an IRF result in an IRF file
[TS] irf set	Set the active IRF file

Forecasting models

[TS] forecast	Econometric model forecasting
[TS] forecast adjust	Adjust variables to produce alternative forecasts
[TS] forecast clear	Clear current model from memory
[TS] forecast coefvector	Specify an equation via a coefficient vector
[TS] forecast create	Create a new forecast model
[TS] forecast describe	Describe features of the forecast model
[TS] forecast drop	Drop forecast variables
[TS] forecast estimates	Add estimation results to a forecast model
[TS] forecast exogenous	Declare exogenous variables
[TS] forecast identity	Add an identity to a forecast model
[TS] forecast list	List forecast commands composing current model
[TS] forecast query	Check whether a forecast model has been started
[TS] forecast solve	Obtain static and dynamic forecasts

Remarks and examples

Remarks are presented under the following headings:

Data management tools and time-series operators
Univariate time series: Estimators
Univariate time series: Time-series smoothers and filters
Univariate time series: Diagnostic tools
Multivariate time series: Estimators
Multivariate time series: Diagnostic tools
Forecasting models
Additional resources

Data management tools and time-series operators

Because time-series estimators are, by definition, a function of the temporal ordering of the observations in the estimation sample, Stata's time-series commands require the data to be sorted and indexed by time, using the `tsset` command, before they can be used. `tsset` is simply a way for you to tell Stata which variable in your dataset represents time; `tsset` then sorts and indexes the data appropriately for use with the time-series commands. Once your dataset has been `tsset`, you can use Stata's time-series operators in data manipulation or programming using that dataset and when specifying the syntax for most time-series commands. Stata has time-series operators for representing the lags, leads, differences, and seasonal differences of a variable. The time-series operators are documented in [TS] [tsset](#).

You can also define a business-day calendar so that Stata's time-series operators respect the structure of missing observations in your data. The most common example is having Monday come after Friday in market data. [D] [Datetime business calendars](#) provides a discussion and examples.

`tsset` can also be used to declare that your dataset contains cross-sectional time-series data, often referred to as panel data. When you use `tsset` to declare your dataset to contain panel data, you specify a variable that identifies the panels and a variable that identifies the time periods. Once your dataset has been `tsset` as panel data, the time-series operators work appropriately for the data.

`tsfill`, which is documented in [TS] [tsfill](#), can be used after `tsset` to fill in missing times with missing observations. `tsset` will report any gaps in your data, and `tsreport` will provide more details about the gaps. `tsappend` adds observations to a time-series dataset by using the information set by `tsset`. This function can be particularly useful when you wish to predict out of sample after fitting

a model with a time-series estimator. `tsrevar` is a programmer's command that provides a way to use *varlists* that contain time-series operators with commands that do not otherwise support time-series operators.

`rolling` performs rolling regressions, recursive regressions, and reverse recursive regressions. Any command that stores results in `e()` or `r()` can be used with `rolling`.

`import fred` can be used to import economic and financial time-series data from the Federal Reserve Economic Data (FRED) into Stata. See [D] [import fred](#) for details on the `import fred` command and the FRED interface, which allows you to easily search available series and select which ones you want to import.

Univariate time series: Estimators

The seven univariate time-series estimators currently available in Stata are `arfima`, `arima`, `arch`, `mswitch`, `newey`, `prais`, and `ucm`. `newey` and `prais` are really just extensions to ordinary linear regression. When you fit a linear regression on time-series data via ordinary least squares (OLS), if the disturbances are autocorrelated, the parameter estimates are usually consistent, but the estimated standard errors tend to be underestimated. Several estimators have been developed to deal with this problem. One strategy is to use OLS for estimating the regression parameters and use a different estimator for the variances, one that is consistent in the presence of autocorrelated disturbances, such as the Newey–West estimator implemented in `newey`. Another strategy is to model the dynamics of the disturbances. The estimators found in `prais`, `arima`, `arch`, `arfima`, and `ucm` are based on such a strategy.

`prais` implements two such estimators: the Prais–Winsten and the Cochrane–Orcutt generalized least-squares (GLS) estimators. These estimators are GLS estimators, but they are fairly restrictive in that they permit only first-order autocorrelation in the disturbances. Although they have certain pedagogical and historical value, they are somewhat obsolete. Faster computers with more memory have made it possible to implement full information maximum likelihood (FIML) estimators, such as Stata's `arima` command. These estimators permit much greater flexibility when modeling the disturbances and are more efficient estimators.

`arima` provides the means to fit linear models with autoregressive moving-average (ARMA) disturbances, or in the absence of linear predictors, autoregressive integrated moving-average (ARIMA) models. This means that, whether you think that your data are best represented as a distributed-lag model, a transfer-function model, or a stochastic difference equation, or you simply wish to apply a Box–Jenkins filter to your data, the model can be fit using `arima`. `arch`, a conditional maximum likelihood estimator, has similar modeling capabilities for the mean of the time series but can also model autoregressive conditional heteroskedasticity in the disturbances with a wide variety of specifications for the variance equation.

`arfima` estimates the parameters of autoregressive fractionally integrated moving-average (ARFIMA) models, which handle higher degrees of dependence than ARIMA models. ARFIMA models allow the autocorrelations to decay at the slower hyperbolic rate, whereas ARIMA models handle processes whose autocorrelations decay at an exponential rate.

Markov-switching models are used for series that are believed to transition over a finite set of unobserved states, allowing the process to evolve differently in each state. The transitions occur according to a Markov process. `mswitch` estimates the state-dependent parameters of Markov-switching dynamic regression models and Markov-switching autoregression models.

Unobserved-components models (UCMs) decompose a time series into trend, seasonal, cyclical, and idiosyncratic components and allow for exogenous variables. `ucm` estimates the parameters of UCMs by maximum likelihood. UCMs can also model the stationary cyclical component using the stochastic-cycle parameterization that has an intuitive frequency-domain interpretation.

`threshold` fits a linear model that allows coefficients to differ across regions determined by threshold values. Threshold models can be used, for instance, to capture breaks or asymmetries that are observed in many time series. The number of thresholds may be known, or `threshold` can choose the optimal number of thresholds. See [TS] [threshold](#).

Univariate time series: Time-series smoothers and filters

In addition to the estimators mentioned above, Stata also provides time-series filters and smoothers. The Baxter–King and Christiano–Fitzgerald band-pass filters and the Butterworth and Hodrick–Prescott high-pass filters are implemented in `tsfilter`; see [TS] [tsfilter](#) for an overview.

Also included are a simple, uniformly weighted, moving-average filter with unit weights; a weighted moving-average filter in which you can specify the weights; single- and double-exponential smoothers; Holt–Winters seasonal and nonseasonal smoothers; and a nonlinear smoother. Most of these smoothers were originally developed as ad hoc procedures and are used for reducing the noise in a time series (smoothing) or forecasting. Although they have limited application for signal extraction, these smoothers have all been found to be optimal for some underlying modern time-series models; see [TS] [tssmooth](#).

Univariate time series: Diagnostic tools

Stata’s time-series commands also include several preestimation and postestimation diagnostic and interpretation commands. `corrgram` estimates the autocorrelation function and partial autocorrelation function of a univariate time series, as well as Q statistics. These functions and statistics are often used to determine the appropriate model specification before fitting ARIMA models. `corrgram` can also be used with `wntestb` and `wntestq` to examine the residuals after fitting a model for evidence of model misspecification. Stata’s time-series commands also include the commands `pergram` and `cumsp`, which provide the log-standardized periodogram and the cumulative-sample spectral distribution, respectively, for time-series analysts who prefer to estimate in the frequency domain rather than the time domain.

Before fitting an ARIMA or ARFIMA model, you can compute information criteria to determine the correct number of autoregressive (AR) and moving average (MA) terms to include in the model. `arimasoc` computes Akaike’s information criterion, Schwarz’s Bayesian information criterion, and the Hannan and Quinn information criterion for a range of AR and MA terms to help select the correct AR and MA orders for your ARIMA model. Similarly, `arfimasoc` computes these statistics for a series of ARFIMA models.

`psdensity` computes the spectral density implied by the parameters estimated by `arfima`, `arima`, or `ucm`. The estimated spectral density shows the relative importance of components at different frequencies. `estat acplot` computes the autocorrelation and autocovariance functions implied by the parameters estimated by `arima`. These functions provide a measure of the dependence structure in the time domain.

`xcorr` estimates the cross-correlogram for bivariate time series and can similarly be used for both preestimation and postestimation. For example, the cross-correlogram can be used before fitting a transfer-function model to produce initial estimates of the IRF. This estimate can then be used to determine the optimal lag length of the input series to include in the model specification. It can also be used as a postestimation tool after fitting a transfer function. The cross-correlogram between the residual from a transfer-function model and the prewhitened input series of the model can be examined for evidence of model misspecification.

When you fit ARMA or ARIMA models, the dependent variable being modeled must be covariance stationary (ARMA models), or the order of integration must be known (ARIMA models). Stata has three commands that can test for the presence of a unit root in a time-series variable: `dfuller` performs the augmented Dickey–Fuller test, `pperron` performs the Phillips–Perron test, and `dfgls` performs a modified Dickey–Fuller test. `arfirm` can also be used to investigate the order of integration. After estimation, you can use `estat aroots` to check the stationarity of an ARMA process.

After using `mswitch` to fit a Markov-switching model, two postestimation commands help interpret the results. `estat transition` reports the transition probabilities and the corresponding standard errors in a tabular form. `estat duration` computes the expected duration of being in a given state and displays. See [TS] [mswitch postestimation](#).

After fitting a model with `regress` or `ivregress`, `estat sbknown` and `estat sbsingle` test for structural breaks. `estat sbknown` tests for breaks at known break dates, and `estat sbsingle` tests for a break at an unknown break date; see [TS] [estat sbknown](#) and [TS] [estat sbsingle](#). After `regress`, `estat sbcsum` tests for stability of regression coefficients. This can be a test for structural breaks due to changes in regression coefficients over time; see [TS] [estat sbcsum](#).

The remaining diagnostic tools for univariate time series are for use after fitting a linear model via OLS with Stata’s `regress` command. They are documented collectively in [R] [regress postestimation time series](#). They include `estat dwatson`, `estat durbinalt`, `estat bgodfrey`, and `estat archlm`. `estat dwatson` computes the Durbin–Watson d statistic to test for the presence of first-order autocorrelation in the OLS residuals. `estat durbinalt` likewise tests for the presence of autocorrelation in the residuals. By comparison, however, Durbin’s alternative test is more general and easier to use than the Durbin–Watson test. With `estat durbinalt`, you can test for higher orders of autocorrelation, the assumption that the covariates in the model are strictly exogenous is relaxed, and there is no need to consult tables to compute rejection regions, as you must with the Durbin–Watson test. `estat bgodfrey` computes the Breusch–Godfrey test for autocorrelation in the residuals, and although the computations are different, the test in `estat bgodfrey` is asymptotically equivalent to the test in `estat durbinalt`. Finally, `estat archlm` performs Engle’s LM test for the presence of autoregressive conditional heteroskedasticity.

Multivariate time series: Estimators

Stata provides commands for fitting the most widely applied multivariate time-series models. `var`, `svar`, and `ivsvar` fit vector autoregressive, structural vector autoregressive, and instrumental-variables structural vector autoregressive models to stationary data. `lpirf` fits a local-projection model to estimate IRFs by least squares. `ivlpirf` fits a local-projection model to estimate IRFs and accounts for endogeneity of an impulse variable by using instrumental variables. `vec` fits cointegrating vector error-correction models. `dfactor` fits dynamic-factor models. `mgarch ccc`, `mgarch dcc`, `mgarch dvecch`, and `mgarch vcc` fit multivariate GARCH models. `sspace` fits state-space models. Many linear time-series models, including vector autoregressive moving-average (VARMA) models and structural time-series models, can be cast as state-space models and fit by `sspace`. `dsge` fits linear dynamic stochastic general equilibrium (DSGE) models, and `dsgenl` fits nonlinear DSGE models. DSGE models are unique in that equations in the system allow current values of variables to depend not only on past values but also on expectations of future values.

Multivariate time series: Diagnostic tools

Before fitting a multivariate time-series model, you must specify the number of lags of the dependent variable to include. `varsoc` produces statistics for determining the order of a vector autoregressive (VAR) model or vector error-correction (VEC) model.

Several postestimation commands perform the most common specification analysis on a previously fitted VAR, structural VAR (SVAR), or instrumental-variables SVAR model. You can use `varlmar` to check for serial correlation in the residuals, `varnorm` to test the null hypothesis that the disturbances come from a multivariate normal distribution, and `varstable` to see if the fitted model is stable. Two common types of inference about VAR models are whether one variable Granger-causes another and whether a set of lags can be excluded from the model. `vargranger` reports Wald tests of Granger causation, and `varwle` reports Wald lag exclusion tests.

Similarly, several postestimation commands perform the most common specification analysis on a previously fitted VEC model. You can use `vecldmar` to check for serial correlation in the residuals, `vecnorm` to test the null hypothesis that the disturbances come from a multivariate normal distribution, and `vecstable` to analyze the stability of the previously fitted VEC model.

VAR models and VEC models are often fit to produce baseline forecasts. `fcast` produces dynamic forecasts from previously fitted VAR models and VEC models. See [TS] [fcast compute](#) and [TS] [fcast graph](#).

Many researchers fit VAR models, SVAR models, VEC models, and DSGE models because they want to analyze how unexpected shocks affect the dynamic paths of the variables. Stata has a suite of `irf` commands for estimating IRFs and interpreting, presenting, and managing these estimates; see [TS] [irf](#). The `irf` commands can also be used to further explore results after obtaining IRF estimates from `lpirf` and `ivlpirf`.

Forecasting models

Stata provides a set of commands for obtaining forecasts by solving models, collections of equations that jointly determine the outcomes of one or more variables. You use Stata estimation commands such as `regress`, `reg3`, `var`, and `vec` to fit stochastic equations and store the results using `estimates store`. Then you create a forecast model using `forecast create` and use commands, including `forecast estimates` and `forecast identity`, to build models consisting of estimation results, nonstochastic relationships (identities), and other model features. Models can be as simple as a single linear regression for which you want to obtain dynamic forecasts, or they can be complicated systems consisting of dozens of estimation results and identities representing a complete macroeconomic model.

The `forecast solve` command allows you to obtain both stochastic and dynamic forecasts. Confidence intervals for forecasts can be obtained via stochastic simulation incorporating both parameter uncertainty and additive random shocks. By using `forecast adjust`, you can incorporate outside information and specify different paths for some of the model's variables to obtain forecasts under alternative scenarios. See [TS] [forecast](#).

Additional resources

In addition to the manual, there are several other resources available for learning about Stata's time-series features.

- On the web: You can watch the *Time series in Stata* playlist on our YouTube Channel. You can read our posts about time-series analysis at *The Stata Blog: Not Elsewhere Classified*. You can search for blog posts from within Stata by typing

```
. search time series blog
```

You can also connect with other Stata users interested in discussing time series and other Stata topics on Statalist, the Stata forum. Visit <https://www.statalist.org/>.

- Training: Stata's NetCourse 461 provides a comprehensive introduction to univariate time-series analysis. See <https://www.stata.com/netcourse/univariate-time-series-intro-nc461> for more information. We also offer a classroom and web-based training course, *Time-Series Analysis Using Stata*.
- Books: We recommend the Stata Press book *Introduction to Time Series Using Stata*, Rev. ed., by Sean Beckett (2020). See <https://www.stata-press.com/books/introduction-to-time-series-using-stata> for full comments from our technical group.

A good reference for financial applications using Stata is the Stata Press book *Financial Econometrics Using Stata* by Simona Boffelli and Giovanni Urga (2016). See <https://www.stata-press.com/books/financial-econometrics-stata/> for comments from our technical group.

References

- Baum, C. F. 2005. *Stata: The language of choice for time-series analysis?* *Stata Journal* 5: 46–63.
- Baum, C. F., S. Hurn, and K. Lindsay. 2021. *The BDS test of independence*. *Stata Journal* 21: 279–294.
- Beckett, S. 2020. *Introduction to Time Series Using Stata*. Rev. ed. College Station, TX: Stata Press.
- Boffelli, S., and G. Urga. 2016. *Financial Econometrics Using Stata*. College Station, TX: Stata Press.
- Box-Steffensmeier, J. M., J. R. Freeman, M. P. Hitt, and J. C. W. Pevehouse. 2014. *Time Series Analysis for the Social Sciences*. New York: Cambridge University Press.
- Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press. <https://doi.org/10.2307/j.ctv14jx6sm>.
- Lütkepohl, H. 1993. *Introduction to Multiple Time Series Analysis*. 2nd ed. New York: Springer.
- . 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.
- Pickup, M. 2015. *Introduction to Time Series Analysis*. Thousand Oaks, CA: Sage.
- Stock, J. H., and M. W. Watson. 2001. Vector autoregressions. *Journal of Economic Perspectives* 15: 101–115. <https://doi.org/10.1257/jep.15.4.101>.

Also see

[U] 1.3 What's new

[R] Intro — Introduction to base reference manual

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).