var — Vector autoregressive models

Syntax

`var` *depvarlist* [*if*] [*in*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`lags(`*numlist*`)`	use lags *numlist* in the VAR
`exog(`*varlist*`)`	use exogenous variables *varlist*
Model 2	
`constraints(`*numlist*`)`	apply specified linear constraints
`nolog`	suppress SURE iteration log
`iterate(`*#*`)`	set maximum number of iterations for SURE; default is `iterate(1600)`
`tolerance(`*#*`)`	set convergence tolerance of SURE
`noisure`	use one-step SURE
`dfk`	make small-sample degrees-of-freedom adjustment
`small`	report small-sample t and F statistics
`nobigf`	do not compute parameter vector for coefficients implicitly set to zero
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
`lutstats`	report Lütkepohl lag-order selection statistics
`nocnsreport`	do not display constraints
display_options	control column formats, row spacing, and line width
`coeflegend`	display legend instead of statistics

You must `tsset` your data before using `var`; see [TS] **tsset**.
depvarlist and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
`by`, `fp`, `rolling`, `statsby`, and `xi` are allowed; see [U] **11.1.10 Prefix commands**.
`coeflegend` does not appear in the dialog box.
See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Menu

Statistics > Multivariate time series > Vector autoregression (VAR)

Description

var fits a multivariate time-series regression of each dependent variable on lags of itself and on lags of all the other dependent variables. var also fits a variant of vector autoregressive (VAR) models known as the VARX model, which also includes exogenous variables. See [TS] **var intro** for a list of commands that are used in conjunction with var.

Options

Model

noconstant; see [R] **estimation options**.

lags(*numlist*) specifies the lags to be included in the model. The default is lags(1 2). This option takes a *numlist* and not simply an integer for the maximum lag. For example, lags(2) would include only the second lag in the model, whereas lags(1/2) would include both the first and second lags in the model. See [U] **11.1.8 numlist** and [U] **11.4.4 Time-series varlists** for more discussion of numlists and lags.

exog(*varlist*) specifies a list of exogenous variables to be included in the VAR.

Model 2

constraints(*numlist*); see [R] **estimation options**.

nolog suppresses the log from the iterated seemingly unrelated regression algorithm. By default, the iteration log is displayed when the coefficients are estimated through iterated seemingly unrelated regression. When the constraints() option is not specified, the estimates are obtained via OLS, and nolog has no effect. For this reason, nolog can be specified only when constraints() is specified. Similarly, nolog cannot be combined with noisure.

iterate(*#*) specifies an integer that sets the maximum number of iterations when the estimates are obtained through iterated seemingly unrelated regression. By default, the limit is 1,600. When constraints() is not specified, the estimates are obtained using OLS, and iterate() has no effect. For this reason, iterate() can be specified only when constraints() is specified. Similarly, iterate() cannot be combined with noisure.

tolerance(*#*) specifies a number greater than zero and less than 1 for the convergence tolerance of the iterated seemingly unrelated regression algorithm. By default, the tolerance is 1e-6. When the constraints() option is not specified, the estimates are obtained using OLS, and tolerance() has no effect. For this reason, tolerance() can be specified only when constraints() is specified. Similarly, tolerance() cannot be combined with noisure.

noisure specifies that the estimates in the presence of constraints be obtained through one-step seemingly unrelated regression. By default, var obtains estimates in the presence of constraints through iterated seemingly unrelated regression. When constraints() is not specified, the estimates are obtained using OLS, and noisure has no effect. For this reason, noisure can be specified only when constraints() is specified.

dfk specifies that a small-sample degrees-of-freedom adjustment be used when estimating $\boldsymbol{\Sigma}$, the error variance–covariance matrix. Specifically, $1/(T-\overline{m})$ is used instead of the large-sample divisor $1/T$, where $\overline{m}$ is the average number of parameters in the functional form for $\mathbf{y}_t$ over the K equations.

small causes var to report small-sample t and F statistics instead of the large-sample normal and chi-squared statistics.

`nobigf` requests that `var` not save the estimated parameter vector that incorporates coefficients that have been implicitly constrained to be zero, such as when some lags have been omitted from a model. `e(bf)` is used for computing asymptotic standard errors in the postestimation commands `irf create` and `fcast compute`; see [TS] **irf create** and [TS] **fcast compute**. Therefore, specifying `nobigf` implies that the asymptotic standard errors will not be available from `irf create` and `fcast compute`. See *Fitting models with some lags excluded*.

Reporting

`level(#)`; see [R] **estimation options**.

`lutstats` specifies that the Lütkepohl (2005) versions of the lag-order selection statistics be reported. See *Methods and formulas* in [TS] **varsoc** for a discussion of these statistics.

`nocnsreport`; see [R] **estimation options**.

display_options: `vsquish`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **estimation options**.

The following option is available with `var` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

A VAR is a model in which K variables are specified as linear functions of p of their own lags, p lags of the other $K-1$ variables, and possibly exogenous variables. A VAR with p lags is usually denoted a VAR(p). For more information, see [TS] **var intro**.

▷ Example 1: VAR model

To illustrate the basic usage of `var`, we replicate the example in Lütkepohl (2005, 77–78). The data consists of three variables: the first difference of the natural log of investment, `dln_inv`; the first difference of the natural log of income, `dln_inc`; and the first difference of the natural log of consumption, `dln_consump`. The dataset contains data through the fourth quarter of 1982, though Lütkepohl uses only the observations through the fourth quarter of 1978.

```
. use http://www.stata-press.com/data/r13/lutkepohl2
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1)
. tsset
        time variable:  qtr, 1960q1 to 1982q4
                delta:  1 quarter
```

```
. var dln_inv dln_inc dln_consump if qtr<=tq(1978q4), lutstats dfk
Vector autoregression
Sample:  1960q4 - 1978q4                        No. of obs      =        73
Log likelihood =   606.307           (lutstats) AIC             = -24.63163
FPE            =  2.18e-11                      HQIC            = -24.40656
Det(Sigma_ml)  =  1.23e-11                      SBIC            = -24.06686
Equation           Parms      RMSE     R-sq      chi2     P>chi2
----------------------------------------------------------------
dln_inv               7     .046148   0.1286   9.736909   0.1362
dln_inc               7     .011719   0.1142   8.508289   0.2032
dln_consump           7     .009445   0.2513   22.15096   0.0011
----------------------------------------------------------------

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
dln_inv      |
     dln_inv |
         L1. |  -.3196318   .1254564    -2.55   0.011    -.5655218   -.0737419
         L2. |  -.1605508   .1249066    -1.29   0.199    -.4053633    .0842616
             |
     dln_inc |
         L1. |   .1459851   .5456664     0.27   0.789    -.9235013    1.215472
         L2. |   .1146009   .5345709     0.21   0.830    -.9331388    1.162341
             |
 dln_consump |
         L1. |   .9612288   .6643086     1.45   0.148    -.3407922     2.26325
         L2. |   .9344001   .6650949     1.40   0.160     -.369162    2.237962
             |
       _cons |  -.0167221   .0172264    -0.97   0.332    -.0504852    .0170409
-------------+----------------------------------------------------------------
dln_inc      |
     dln_inv |
         L1. |   .0439309   .0318592     1.38   0.168     -.018512    .1063739
         L2. |   .0500302   .0317196     1.58   0.115    -.0121391    .1121995
             |
     dln_inc |
         L1. |  -.1527311   .1385702    -1.10   0.270    -.4243237    .1188615
         L2. |   .0191634   .1357525     0.14   0.888    -.2469067    .2852334
             |
 dln_consump |
         L1. |   .2884992    .168699     1.71   0.087    -.0421448    .6191431
         L2. |     -.0102   .1688987    -0.06   0.952    -.3412354    .3208353
             |
       _cons |   .0157672   .0043746     3.60   0.000     .0071932    .0243412
-------------+----------------------------------------------------------------
dln_consump  |
     dln_inv |
         L1. |   -.002423   .0256763    -0.09   0.925    -.0527476    .0479016
         L2. |   .0338806   .0255638     1.33   0.185    -.0162235    .0839847
             |
     dln_inc |
         L1. |   .2248134   .1116778     2.01   0.044      .005929    .4436978
         L2. |   .3549135   .1094069     3.24   0.001     .1404798    .5693471
             |
 dln_consump |
         L1. |  -.2639695   .1359595    -1.94   0.052    -.5304451    .0025062
         L2. |  -.0222264   .1361204    -0.16   0.870    -.2890175    .2445646
             |
       _cons |   .0129258   .0035256     3.67   0.000     .0060157    .0198358
------------------------------------------------------------------------------
```

The output has two parts: a header and the standard Stata output table for the coefficients, standard errors, and confidence intervals. The header contains summary statistics for each equation in the VAR and statistics used in selecting the lag order of the VAR. Although there are standard formulas for all the lag-order statistics, Lütkepohl (2005) gives different versions of the three information criteria that drop the constant term from the likelihood. To obtain the Lütkepohl (2005) versions, we specified the `lutstats` option. The formulas for the standard and Lütkepohl versions of these statistics are given in *Methods and formulas* of [TS] **varsoc**.

The `dfk` option specifies that the small-sample divisor $1/(T-\overline{m})$ be used in estimating $\mathbf{\Sigma}$ instead of the maximum likelihood (ML) divisor $1/T$, where $\overline{m}$ is the average number of parameters included in each of the K equations. All the lag-order statistics are computed using the ML estimator of $\mathbf{\Sigma}$. Thus, specifying `dfk` will not change the computed lag-order statistics, but it will change the estimated variance–covariance matrix. Also, when `dfk` is specified, a `dfk`-adjusted log likelihood is computed and stored in `e(ll_dfk)`.

◁

The `lag()` option takes a *numlist* of lags. To specify a model that includes the first and second lags, type

```
. var y1 y2 y3, lags(1/2)
```

not

```
. var y1 y2 y3, lags(2)
```

because the latter specification would fit a model that included only the second lag.

Fitting models with some lags excluded

To fit a model that has only a fourth lag, that is,

$$\mathbf{y}_t = \mathbf{v} + \mathbf{A}_4\mathbf{y}_{t-4} + \mathbf{u}_t$$

you would specify the `lags(4)` option. Doing so is equivalent to fitting the more general model

$$\mathbf{y}_t = \mathbf{v} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \mathbf{A}_3\mathbf{y}_{t-3} + \mathbf{A}_4\mathbf{y}_{t-4} + \mathbf{u}_t$$

with $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ constrained to be $\mathbf{0}$. When you fit a model with some lags excluded, `var` estimates the coefficients included in the specification ($\mathbf{A}_4$ here) and stores these estimates in `e(b)`. To obtain the asymptotic standard errors for impulse–response functions and other postestimation statistics, Stata needs the complete set of parameter estimates, including those that are constrained to be zero; `var` stores them in `e(bf)`. Because you can specify models for which the full set of parameter estimates exceeds Stata's limit on the size of matrices, the `nobigf` option specifies that `var` not compute and store `e(bf)`. This means that the asymptotic standard errors of the postestimation functions cannot be obtained, although bootstrap standard errors are still available. Building `e(bf)` can be time consuming, so if you do not need this full matrix, and speed is an issue, use `nobigf`.

Fitting models with exogenous variables

▷ Example 2: VAR model with exogenous variables

We use the `exog()` option to include exogenous variables in a VAR.

```
. var dln_inc dln_consump if qtr<=tq(1978q4), dfk exog(dln_inv)

Vector autoregression

Sample:  1960q4 - 1978q4                        No. of obs      =        73
Log likelihood =  478.5663                      AIC             = -12.78264
FPE            =  9.64e-09                      HQIC            = -12.63259
Det(Sigma_ml)  =  6.93e-09                      SBIC            = -12.40612

Equation           Parms      RMSE     R-sq      chi2     P>chi2
----------------------------------------------------------------
dln_inc               6     .011917   0.0702   5.059587   0.4087
dln_consump           6     .009197   0.2794   25.97262   0.0001
----------------------------------------------------------------

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
dln_inc      |
     dln_inc |
         L1. |  -.1343345   .1391074    -0.97   0.334    -.4069801    .1383111
         L2. |   .0120331   .1380346     0.09   0.931    -.2585097    .2825759
             |
 dln_consump |
         L1. |   .3235342   .1652769     1.96   0.050    -.0004027     .647471
         L2. |   .0754177   .1648624     0.46   0.647    -.2477066     .398542
             |
     dln_inv |   .0151546   .0302319     0.50   0.616    -.0440987     .074408
       _cons |   .0145136   .0043815     3.31   0.001     .0059259    .0231012
-------------+----------------------------------------------------------------
dln_consump  |
     dln_inc |
         L1. |   .2425719   .1073561     2.26   0.024     .0321578     .452986
         L2. |   .3487949   .1065281     3.27   0.001     .1400036    .5575862
             |
 dln_consump |
         L1. |  -.3119629   .1275524    -2.45   0.014    -.5619611   -.0619648
         L2. |  -.0128502   .1272325    -0.10   0.920    -.2622213    .2365209
             |
     dln_inv |   .0503616   .0233314     2.16   0.031     .0046329    .0960904
       _cons |   .0131013   .0033814     3.87   0.000     .0064738    .0197288
------------------------------------------------------------------------------
```

All the postestimation commands for analyzing VARs work when exogenous variables are included in a model, but the asymptotic standard errors for the h-step-ahead forecasts are not available.

◁

Fitting models with constraints on the coefficients

`var` permits model specifications that include constraints on the coefficient, though `var` does not allow for constraints on Σ. See [TS] **var intro** and [TS] **var svar** for ways to constrain Σ.

▷ Example 3: VAR model with constraints

In the first example, we fit a full VAR(2) to a three-equation model. The coefficients in the equation for `dln_inv` were jointly insignificant, as were the coefficients in the equation for `dln_inc`; and many individual coefficients were not significantly different from zero. In this example, we constrain the coefficient on `L2.dln_inc` in the equation for `dln_inv` and the coefficient on `L2.dln_consump` in the equation for `dln_inc` to be zero.

```
. constraint 1 [dln_inv]L2.dln_inc = 0

. constraint 2 [dln_inc]L2.dln_consump = 0

. var dln_inv dln_inc dln_consump if qtr<=tq(1978q4), lutstats dfk
> constraints(1 2)
Estimating VAR coefficients

Iteration 1:   tolerance =  .00737681
Iteration 2:   tolerance =  3.998e-06
Iteration 3:   tolerance =  2.730e-09

Vector autoregression

Sample:  1960q4 - 1978q4                        No. of obs      =        73
Log likelihood =  606.2804            (lutstats) AIC             = -31.69254
FPE            =  1.77e-14                      HQIC            = -31.46747
Det(Sigma_ml)  =  1.05e-14                      SBIC            = -31.12777

Equation           Parms      RMSE     R-sq      chi2      P>chi2
-----------------------------------------------------------------

dln_inv               6     .043895   0.1280   9.842338   0.0798
dln_inc               6     .011143   0.1141   8.584446   0.1268
dln_consump           7     .008981   0.2512   22.86958   0.0008
-----------------------------------------------------------------

 ( 1)  [dln_inv]L2.dln_inc = 0
 ( 2)  [dln_inc]L2.dln_consump = 0
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
dln_inv						
dln_inv						
L1.	-.320713	.1247512	-2.57	0.010	-.5652208	-.0762051
L2.	-.1607084	.124261	-1.29	0.196	-.4042555	.0828386
dln_inc						
L1.	.1195448	.5295669	0.23	0.821	-.9183873	1.157477
L2.	-2.55e-17	1.18e-16	-0.22	0.829	-2.57e-16	2.06e-16
dln_consump						
L1.	1.009281	.623501	1.62	0.106	-.2127586	2.231321
L2.	1.008079	.5713486	1.76	0.078	-.1117438	2.127902
_cons	-.0162102	.016893	-0.96	0.337	-.0493199	.0168995
dln_inc						
dln_inv						
L1.	.0435712	.0309078	1.41	0.159	-.017007	.1041495
L2.	.0496788	.0306455	1.62	0.105	-.0103852	.1097428
dln_inc						
L1.	-.1555119	.1315854	-1.18	0.237	-.4134146	.1023908
L2.	.0122353	.1165811	0.10	0.916	-.2162595	.2407301
dln_consump						
L1.	.29286	.1568345	1.87	0.062	-.01453	.6002501
L2.	1.78e-19	8.28e-19	0.22	0.829	-1.45e-18	1.80e-18
_cons	.015689	.003819	4.11	0.000	.0082039	.0231741
dln_consump						
dln_inv						
L1.	-.0026229	.0253538	-0.10	0.918	-.0523154	.0470696
L2.	.0337245	.0252113	1.34	0.181	-.0156888	.0831378
dln_inc						
L1.	.2224798	.1094349	2.03	0.042	.0079912	.4369683
L2.	.3469758	.1006026	3.45	0.001	.1497984	.5441532
dln_consump						
L1.	-.2600227	.1321622	-1.97	0.049	-.519056	-.0009895
L2.	-.0146825	.1117618	-0.13	0.895	-.2337315	.2043666
_cons	.0129149	.003376	3.83	0.000	.0062981	.0195317

None of the free parameter estimates changed by much. Whereas the coefficients in the equation `dln_inv` are now significant at the 10% level, the coefficients in the equation for `dln_inc` remain jointly insignificant.

◁

Stored results

`var` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_gaps)`	number of gaps in sample
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_dv)`	number of dependent variables
`e(df_eq)`	average number of parameters in an equation
`e(df_m)`	model degrees of freedom
`e(df_r)`	residual degrees of freedom (`small` only)
`e(ll)`	log likelihood
`e(ll_dfk)`	`dfk` adjusted log likelihood (`dfk` only)
`e(obs_#)`	number of observations on equation #
`e(k_#)`	number of parameters in equation #
`e(df_m#)`	model degrees of freedom for equation #
`e(df_r#)`	residual degrees of freedom for equation # (`small` only)
`e(r2_#)`	R-squared for equation #
`e(ll_#)`	log likelihood for equation #
`e(chi2_#)`	x^2 for equation #
`e(F_#)`	F statistic for equation # (`small` only)
`e(rmse_#)`	root mean squared error for equation #
`e(aic)`	Akaike information criterion
`e(hqic)`	Hannan–Quinn information criterion
`e(sbic)`	Schwarz–Bayesian information criterion
`e(fpe)`	final prediction error
`e(mlag)`	highest lag in VAR
`e(tmin)`	first time period in sample
`e(tmax)`	maximum time
`e(detsig)`	determinant of `e(Sigma)`
`e(detsig_ml)`	determinant of $\widehat{\Sigma}_{\rm ml}$
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`var`
`e(cmdline)`	command as typed
`e(depvar)`	names of dependent variables
`e(endog)`	names of endogenous variables, if specified
`e(exog)`	names of exogenous variables, and their lags, if specified
`e(exogvars)`	names of exogenous variables, if specified
`e(eqnames)`	names of equations
`e(lags)`	lags in model
`e(exlags)`	lags of exogenous variables in model, if specified
`e(title)`	title in estimation output
`e(nocons)`	`nocons`, if `noconstant` is specified
`e(constraints)`	`constraints`, if specified
`e(cnslist_var)`	list of specified constraints
`e(small)`	`small`, if specified
`e(lutstats)`	`lutstats`, if specified
`e(timevar)`	time variable specified in `tsset`
`e(tsfmt)`	format for the current time variable
`e(dfk)`	`dfk`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(Sigma)`	$\widehat{\boldsymbol{\Sigma}}$ matrix
`e(V)`	variance–covariance matrix of the estimators
`e(bf)`	constrained coefficient vector
`e(exlagsm)`	matrix mapping lags to exogenous variables
`e(G)`	Gamma matrix; see *Methods and formulas*

Functions

`e(sample)`	marks estimation sample

Methods and formulas

When there are no constraints placed on the coefficients, the VAR(p) is a seemingly unrelated regression model with the same explanatory variables in each equation. As discussed in Lütkepohl (2005) and Greene (2008, 696), performing linear regression on each equation produces the maximum likelihood estimates of the coefficients. The estimated coefficients can then be used to calculate the residuals, which in turn are used to estimate the cross-equation error variance–covariance matrix $\boldsymbol{\Sigma}$.

Per Lütkepohl (2005), we write the VAR(p) with exogenous variables as

$$\mathbf{y}_t = \mathbf{A}\mathbf{Y}_{t-1} + \mathbf{B}_0\mathbf{x}_t + \mathbf{u}_t \tag{5}$$

where

$\mathbf{y}_t$ is the $K \times 1$ vector of endogenous variables,

$\mathbf{A}$ is a $K \times Kp$ matrix of coefficients,

$\mathbf{B}_0$ is a $K \times M$ matrix of coefficients,

$\mathbf{x}_t$ is the $M \times 1$ vector of exogenous variables,

$\mathbf{u}_t$ is the $K \times 1$ vector of white noise innovations, and

$\mathbf{Y}_t$ is the $Kp \times 1$ matrix given by $\mathbf{Y}_t = \begin{pmatrix} \mathbf{y}_t \\ \vdots \\ \mathbf{y}_{t-p+1} \end{pmatrix}$

Although (5) is easier to read, the formulas are much easier to manipulate if it is instead written as

$$\mathbf{Y} = \mathbf{BZ} + \mathbf{U}$$

where

$$\begin{array}{ll} \mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T) & \mathbf{Y} \text{ is } K \times T \\ \mathbf{B} = (\mathbf{A}, \mathbf{B}_0) & \mathbf{B} \text{ is } K \times (Kp + M) \\ \mathbf{Z} = \begin{pmatrix} \mathbf{Y}_0 \ldots, \mathbf{Y}_{T-1} \\ \mathbf{x}_1 \ldots, \mathbf{x}_T \end{pmatrix} & \mathbf{Z} \text{ is } (Kp + M) \times T \\ \mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_T) & \mathbf{U} \text{ is } K \times T \end{array}$$

Intercept terms in the model are included in $\mathbf{x}_t$. If there are no exogenous variables and no intercept terms in the model, $\mathbf{x}_t$ is empty.

The coefficients are estimated by iterated seemingly unrelated regression. Because the estimation is actually performed by `reg3`, the methods are documented in [R] **reg3**. See [P] **makecns** for more on estimation with constraints.

Let $\widehat{\mathbf{U}}$ be the matrix of residuals that are obtained via $\mathbf{Y} - \widehat{\mathbf{B}}\mathbf{Z}$, where $\widehat{\mathbf{B}}$ is the matrix of estimated coefficients. Then the estimator of $\mathbf{\Sigma}$ is

$$\widehat{\mathbf{\Sigma}} = \frac{1}{\widetilde{T}} \widehat{\mathbf{U}}'\widehat{\mathbf{U}}$$

By default, the maximum likelihood divisor of $\widetilde{T} = T$ is used. When `dfk` is specified, a small-sample degrees-of-freedom adjustment is used; then, $\widetilde{T} = T - \overline{m}$ where $\overline{m}$ is the average number of parameters per equation in the functional form for $\mathbf{y}_t$ over the K equations.

`small` specifies that Wald tests after `var` be assumed to have F or t distributions instead of chi-squared or standard normal distributions. The standard errors from each equation are computed using the degrees of freedom for the equation.

The "gamma" matrix stored in `e(G)` referred to in *Stored results* is the $(Kp + 1) \times (Kp + 1)$ matrix given by

$$\frac{1}{T} \sum_{t=1}^{T} (1, \mathbf{Y}_t')(1, \mathbf{Y}_t')'$$

The formulas for the lag-order selection criteria and the log likelihood are discussed in [TS] **varsoc**.

Acknowledgment

We thank Christopher F. Baum of the Department of Economics at Boston College and author of the Stata Press books *An Introduction to Modern Econometrics Using Stata* and *An Introduction to Stata Programming* for his helpful comments.

References

Greene, W. H. 2008. *Econometric Analysis*. 6th ed. Upper Saddle River, NJ: Prentice Hall.

Hamilton, J. D. 1994. *Time Series Analysis*. Princeton: Princeton University Press.

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.

Stock, J. H., and M. W. Watson. 2001. Vector autoregressions. *Journal of Economic Perspectives* 15: 101–115.

Watson, M. W. 1994. Vector autoregressions and cointegration. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden. Amsterdam: Elsevier.

Also see

[TS] **var postestimation** — Postestimation tools for var

[TS] **tsset** — Declare data to be time-series data

[TS] **dfactor** — Dynamic-factor models

[TS] **forecast** — Econometric model forecasting

[TS] **mgarch** — Multivariate GARCH models

[TS] **sspace** — State-space models

[TS] **var svar** — Structural vector autoregressive models

[TS] **varbasic** — Fit a simple VAR and graph IRFs or FEVDs

[TS] **vec** — Vector error-correction models

[U] 20 Estimation and postestimation commands

[TS] **var intro** — Introduction to vector autoregressive models