xtvar postestimation — Postestimation tools for xtvar

Postestimation commands

The following postestimation commands are of special interest after `xtvar`:

Command	Description
`irf`	create and analyze IRFs
`vargranger`	Granger causality tests
`xtvarsoc`	model- and moment-selection criteria (MMSC)
`varstable`	check stability condition of estimates
`varwle`	Wald lag-exclusion statistics

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	fitted values, error components
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as fitted values and panel-level error components.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* equation(*eqno* | *eqname*)]

statistic	Description
Main	
xb	fitted values; the default
* xbu	fitted values including panel effect
* u	u_i, the fixed error component
* e	e_{it}, the overall error component

Unstarred statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when if e(sample) is not specified.

Options for predict

Main

xb calculates the fitted values, the linear prediction, for the specified equation. This is the default.

xbu calculates the fitted values including the panel-level fixed effect for the specified equation.

u calculates the panel-level fixed effect for the specified equation.

e calculates the overall error component, the residual, for the specified equation.

equation(*eqno* | *eqname*) specifies the equation for which the prediction is desired.

You need to specify either one equation number (*eqno*) or equation name (*eqname*) with equation(). For example, equation(#1) indicates that the calculation be made for the first equation, equation(#2) would refer to the second equation, and so on. You could also refer to the equation by its name, which is the same as the corresponding variable name; thus, equation(lnwage) would refer to the equation for the dependent variable named lnwage.

If you do not specify equation(), the results are the same as if you specified equation(#1).

For more information on using predict after multiple-equation estimation commands, see [R] **predict**.

xtvarsoc

Description for xtvarsoc

xtvarsoc calculates MMSC for panel-data vector autoregressive (VAR) models previously fit by xtvar. xtvarsoc can be used to select the appropriate lag conditional on a specification for the instrumental variables, and it can be used to select the appropriate specification for the instrumental-variables conditional on the lag order chosen for the model.

Menu for xtvarsoc

Statistics > Postestimation

Syntax for xtvarsoc

Lag-order selection syntax

xtvarsoc [, *lag_options*]

Instrument selection syntax

xtvarsoc *namelist* [, separator(#)]

namelist specifies the names of previously stored sets of xtvar estimates.

lag_options	Description
minlag(#)	minimum lag order to consider; default is minlag(1)
maxlag(#)	maximum lag order to consider; default is based on previous model
estimates(*estname*)	use previously stored results *estname*; default is to use active results
separator(#)	draw a separator line after every # rows; default is separator(0)

collect is allowed; see [U] **11.1.10 Prefix commands**.

Options for xtvarsoc

minlag(#) specifies the minimum lag order to consider when computing the MMSCs. The default is minlag(1).

maxlag(#) specifies the maximum lag order to consider when computing the MMSCs. By default, xtvarsoc uses the lag order of the currently active estimation results or the lag order of the model specified in the estimates() option as the maximum.

estimates(*estname*) specifies the name of a previously stored set of xtvar estimates. By default, xtvarsoc uses the currently active estimation results. See [R] **estimates** for information on manipulating estimation results.

separator(#) specifies how often separator lines should be drawn between rows. By default, separator lines do not appear. For example, separator(1) would draw a line between each row, separator(2) would draw a line between every other row, and so on.

Remarks and examples

Remarks are presented under the following headings:

Model stability and hypothesis testing

The following commands are available after `xtvar` to check the model's stability and perform certain hypothesis tests commonly used after fitting a VAR model:

`vargranger` (see [TS] **vargranger**);

`varstable` (see [TS] **varstable**); and

`varwle` (see [TS] **varwle**).

See example 6, example 7, and example 8 of [XT] **xtvar** for illustrations and discussions of each of these commands following `xtvar`.

IRFs

The full suite of commands listed in [TS] **irf** is available after `xtvar`. `irf create` estimates the following IRFs after `xtvar`:

1. Simple IRFs
2. Orthogonalized IRFs
3. Cumulative IRFs
4. Cumulative orthogonalized IRFs

For each of these IRFs, standard errors are by default computed using asymptotic formulas based on the delta method and the sampling variances of the estimated parameters. Optionally, you may specify the `bs` option to request that the standard errors instead be computed using the bootstrap. After `xtvar`, `irf create` uses a cluster bootstrap, sampling with replacement from panels included in the estimation sample. At each bootstrap replication, `irf create` refits your model using the same instrument specification that you used at estimation time. By default, if `xtvar` reports that it encountered a singular matrix, the bootstrap replication is ignored.

We show how to create and analyze IRFs after `xtvar` in example 9 and example 10 of [XT] **xtvar**.

MMSC

When working with time-series VAR models, analysts often use various information criteria (IC) to determine the optimal lag length of a VAR model. These include the IC of the Akaike (1973) information criterion (AIC), Schwarz's (1978) Bayesian information criterion (BIC), and the Hannan and Quinn (1979) information criterion (HQIC). Each of those ICs is a function of the maximized log likelihood of a model and includes a penalty term that is increasing in the number of parameters. A model with a lower, say, HQIC is to be preferred to one with a higher HQIC.

`xtvar` fits VARs with panel data using the generalized method of moments (GMM), which does not require us to place any strong distributional assumptions on the error terms, such as normality. There is no likelihood function that we can compute, so we cannot obtain the traditional ICs used with time-series VARs.

Andrews and Lu (2001) developed what they coined "model- and moment-selection criteria" for use with GMM models that are akin to the AIC, BIC, and HQIC for maximum likelihood models. Rather than use the maximized log likelihood, these MMSC are based on Hansen's (1982) *J* statistic, which is normally used to test the validity of the overidentifying restrictions of a GMM model. Like the traditional ICs, Andrews and Lu's MMSC apply a penalty term that favors more parsimonious models. Also like traditional ICs, a model with a lower MMSC is preferable to one with a higher MMSC. There are three variants of Andrews and Lu's (2001) MMSC, corresponding to the AIC, BIC, and HQIC. We therefore refer to these variants as the MMSC-AIC, MMSC-BIC, and MMSC-HQIC.

Much like with `varsoc` after fitting a time-series VAR model, you use `xtvarsoc` after fitting one or more models with `xtvar`; `xtvarsoc` reports MMSC rather than ICs after `xtvar`. `xtvarsoc`, in contrast to `varsoc`, does not provide results before fitting a model because panel-data VAR models require you to provide an appropriate specification of the instrumental variables, and you do that when you fit the model with `xtvar`.

Andrews and Lu's (2001) MMSC can be used for multiple purposes, including selecting a lag length as well as selecting an instrument specification. `xtvarsoc` therefore has two syntaxes, one for selecting a lag length and one for selecting instruments. For the lag-order selection syntax, you call `xtvarsoc` and optionally specify the maximum and minimum number of lags to consider. `xtvarsoc` will use the instrument specification you specified when fitting your candidate `xtvar` model, and it is careful to maintain the same estimation sample across all candidate models, with varying numbers of lags, that it fits. The key is that `xtvarsoc` will fit variants of your model with different lag lengths.

We illustrated how to use `xtvarsoc` for lag-order selection in example 4 of [XT] **xtvar**. We reiterate a few key points to bear in mind when using `xtvar` for lag-order selection. For model comparisons to be valid, the number of observations should remain constant across all candidate models. Moreover, when we are choosing the lag length, the number of moment conditions in each model should remain constant as well. If the number of moment conditions changes from model to model, then we are comparing not only models with different lag lengths but also models with different instrument specifications. If our goal is to choose an optimal lag length, we should compare only similar models. Finally, before comparing MMSC across models, you should first examine Hansen's *J* statistic so that you are choosing only between the set of models with valid moment conditions.

To use `xtvarsoc` for instrument selection, you need to specify a list of previously stored estimation results, and `xtvarsoc` will report Hansen's *J* statistic and the MMSC for the models you specified.

▷ Example 1: Choosing an instrument specification

In [XT] **xtvar**, we fit a three-variable panel-data VAR model using Dahlberg and Johansson's (2000) Swedish municipality data. In example 4, we used `xtvarsoc` to decide upon a model with four lags of the left-hand-side variables, and we specified the `maxldep(2)` option with `xtvar` so that two lags of those variables were used as instruments. Here we reexamine the `maxldep(2)` option. Should we have used three lags as instruments? Four? We can answer that question with `xtvarsoc`.

We first load the dataset and fit three variants of the model, changing the argument in `maxldep()` each time. We fit these models with the `quietly` prefix to save space, but you are welcome to omit that prefix.

```
. use https://www.stata-press.com/data/r19/swedishgov
(1979-1987 Swedish municipality data)

. quietly xtvar expenditures revenues grants, lags(4) maxldep(2)

. estimates title: "lags(4), maxldep(2)"

. estimates store ldep_2
```

```
. quietly xtvar expenditures revenues grants, lags(4) maxldep(3)
. estimates title: "lags(4), maxldep(3)"
. estimates store ldep_3
. quietly xtvar expenditures revenues grants, lags(4) maxldep(4)
. estimates title: "lags(4), maxldep(4)"
. estimates store ldep_4
```

We used `estimates title` to add a label to each of our estimation results before storing them with `estimates store`. This will help avoid ambiguity when we call `xtvarsoc`, which we do now:

```
. xtvarsoc ldep_2 ldep_3 ldep_4
Model- and moment-selection criteria
```

Est. #	N	MC	Hansen's J	df	p	MMSC-AIC	MMSC-BIC	MMSC-HQIC
1	1060	72	38.80	36	0.345	-33.199	-211.98	-100.95
2	1060	108	101.02	72	0.014	-42.976	-400.53	-178.49
3	1060	144	147.46	108	0.007	-68.544*	-604.87*	-271.81*

```
* indicates minimum value within column.
```

Est. #	Name	Title
1	ldep_2	lags(4), maxldep(2)
2	ldep_3	lags(4), maxldep(3)
3	ldep_4	lags(4), maxldep(4)

In our call to `xtvarsoc`, we simply specified the names of our stored results. Looking at the output, we see that all three estimates have the same sample size, so we are comparing models with the same estimation sample. Unlike when we use `xtvarsoc` to select the lag length, here we are not concerned that the number of moment conditions in each model—listed in the column marked `MC`—differs. After all, we are looking at different specifications for the instruments, and the instrument specification determines the number of moment conditions.

What caught our eye is the fact that we reject Hansen's J test of overidentifying restrictions for the models where we used more than two lags of the dependent variables as instruments. All three of the MMSC would lead us to pick the model with four lags as instruments, but because we reject the validity of the overidentifying restrictions for that model, we are inclined to stick with our original model that uses two lags.

◁

The previous example illustrates a common theme we have noticed when fitting panel-data VARs: in many cases, it is not so much a matter of selecting the "optimal" lag length or instrument specification as it is a matter of finding a model where we can find a valid set of instruments.

Stored results

`xtvarsoc` stores the following in `r()`:

Matrices

`r(results)`	sample size, J statistics, and MMSCs

Methods and formulas

Methods and formulas are presented under the following headings:

predict
irf create
xtvarsoc

predict

The complete panel-data VAR model is

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{D}\mathbf{v}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$$

The fitted values, obtained by specifying the `xb` option with `predict`, are

$$\widehat{\mathbf{y}}_{it} = \widehat{\mathbf{A}}_1\mathbf{y}_{i,t-1} + \cdots + \widehat{\mathbf{A}}_p\mathbf{y}_{i,t-p} + \widehat{\mathbf{B}}\mathbf{x}_{it} + \widehat{\mathbf{C}}\mathbf{w}_{it} + \widehat{\mathbf{D}}\mathbf{v}_{it}$$

The overall error component, $\boldsymbol{\nu}_{it} = \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$, consists of the fixed-effect term $\mathbf{u}_i$ and the residuals $\boldsymbol{\epsilon}_{it}$. Let $\widehat{\boldsymbol{\nu}}_{it} = \mathbf{y}_{it} - \widehat{\mathbf{y}}_{it}$. We estimate the fixed-effect term as $\widehat{\mathbf{u}}_i = \sum_t \widehat{\boldsymbol{\nu}}_{it}/T$, with the obvious adaptation for panels with missing time periods. We estimate the sample residuals as $\widehat{\boldsymbol{\epsilon}}_{it} = \widehat{\boldsymbol{\nu}}_{it} - \widehat{\mathbf{u}}_i$. Finally, we estimate the fitted values including the fixed-effect term as $\widehat{\mathbf{y}}_{it} + \widehat{\mathbf{u}}_i$.

irf create

The formulas for the IRFs, orthogonalized IRFs, their cumulative variants, and their standard errors after `xtvar` are identical to those used after `var`. See *Impulse–response function formulas for VAR models* in [TS] **irf create**.

xtvarsoc

Let N denote sample size, J denote Hansen's J test statistic of overidentifying restrictions, and d represent the corresponding degrees of freedom. The MMSC are computed as follows:

$$\begin{aligned} \text{MMSC}_{\text{AIC}} &= J - 2d \\ \text{MMSC}_{\text{BIC}} &= J - d\ln N \\ \text{MMSC}_{\text{HQIC}} &= J - 2d\ln\ln N \end{aligned}$$

The MMSC will not be available if the final weight matrix of the underlying model is not of full rank or if the model is just identified. When the weight matrix is not full rank is true, we report each MMSC as `NFR` and set the value to `.r` in the `r(results)` matrix result. When the model is just identified, we report `JI` for each MMSC and set the value to `.j` in `r(results)`.

References

Akaike, H. 1973. "Information theory and an extension of the maximum likelihood principle". In *Second International Symposium on Information Theory*, edited by B. N. Petrov and F. Csaki, 267–281. Budapest: Akailseoniai–Kiudo.

Andrews, D. W. K., and B. Lu. 2001. Consistent model and moment selection procedures for GMM estimation with application to dynamic panel data models. *Journal of Econometrics* 101: 123–164. https://doi.org/10.1016/S0304-4076(00)00077-4.

Dahlberg, M., and E. Johansson. 2000. An examination of the dynamic behaviour of local governments using GMM bootstrapping methods. *Journal of Applied Econometrics* 15: 401–416. https://doi.org/10.1002/1099-1255(200007/08)15:4<401::AID-JAE564>3.0.CO;2-G.

Hannan, E. J., and B. G. Quinn. 1979. The determination of the order of an autoregression. *Journal of the Royal Statistical Society,* B ser., 41: 190–195. https://doi.org/10.1111/j.2517-6161.1979.tb01072.x.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464. https://doi.org/10.1214/aos/1176344136.

Also see

[XT] **xtvar** — Panel-data vector autoregressive models

[U] 20 Estimation and postestimation commands

For suggested citations, see the FAQ on citing Stata documentation.