

## Description

`fcast compute` produces dynamic forecasts of the dependent variables in a model previously fit by `var`, `svar`, or `vec`. `fcast compute` creates new variables and, if necessary, extends the time frame of the dataset to contain the prediction horizon.

## Quick start

Dynamic forecasts stored in `f_y1`, `f_y2`, and `f_y3` after fitting a model with `var` for dependent variables `y1`, `y2`, and `y3`

```
fcast compute f_
```

Same as above, but begin forecast on the first quarter of 1979

```
fcast compute f_, dynamic(q(1979q1))
```

Same as above, but specify that 10 periods should be forecasted

```
fcast compute f_, dynamic(q(1979q1)) step(10)
```

## Menu

Statistics > Multivariate time series > VEC/VAR forecasts > Compute forecasts (required for graph)

## Syntax

After `var` and `svar`

```
fcast_compute prefix [ , options1 ]
```

After `vec`

```
fcast_compute prefix [ , options2 ]
```

*prefix* is the prefix appended to the names of the dependent variables to create the names of the variables holding the dynamic forecasts.

<i>options<sub>1</sub></i>	Description
Main	
<code>step(#)</code>	set # periods to forecast; default is <code>step(1)</code>
<code>dynamic(time_constant)</code>	begin dynamic forecasts at <i>time_constant</i>
<code>estimate(estname)</code>	use previously stored results <i>estname</i> ; default is to use active results
<code>replace</code>	replace existing forecast variables that have the same prefix
Std. errors	
<code>nose</code>	suppress asymptotic standard errors
<code>bs</code>	obtain standard errors from bootstrapped residuals
<code>bsp</code>	obtain standard errors from parametric bootstrap
<code>bscentile</code>	estimate bounds by using centiles of bootstrapped dataset
<code>reps(#)</code>	perform # bootstrap replications; default is <code>reps(200)</code>
<code>nodots</code>	suppress the usual dot after each bootstrap replication
<code>saving(filename[, replace])</code>	save bootstrap results as <i>filename</i> ; use <code>replace</code> to overwrite existing <i>filename</i>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>

<i>options<sub>2</sub></i>	Description
Main	
<code>step(#)</code>	set # periods to forecast; default is <code>step(1)</code>
<code>dynamic(time_constant)</code>	begin dynamic forecasts at <i>time_constant</i>
<code>estimate(estname)</code>	use previously stored results <i>estname</i> ; default is to use active results
<code>replace</code>	replace existing forecast variables that have the same prefix
<code>differences</code>	save dynamic predictions of the first-differenced variables
Std. errors	
<code>nose</code>	suppress asymptotic standard errors
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>

Default is to use asymptotic standard errors if no options are specified.

`fcast compute` can be used only after `var`, `svar`, and `vec`; see [TS] [var](#), [TS] [var svar](#), and [TS] [vec](#).

You must `tsset` your data before using `fcast compute`; see [TS] [tsset](#).

## Options

### Main

`step(#)` specifies the number of periods to be forecast. The default is `step(1)`.

`dynamic(time_constant)` specifies the period to begin the dynamic forecasts. The default is the period after the last observation in the estimation sample. The `dynamic()` option accepts either a Stata date function that returns an integer or an integer that corresponds to a date using the current `tsset` format. `dynamic()` must specify a date in the range of two or more periods into the estimation sample to one period after the estimation sample.

`estimate(estname)` specifies that `fcast compute` use the estimation results stored as *estname*. By default, `fcast compute` uses the active estimation results. See [R] [estimates](#) for more information on manipulating estimation results.

`replace` causes `fcast compute` to replace the variables in memory with the specified predictions.

`differences` specifies that `fcast compute` also save dynamic predictions of the first-differenced variables. `differences` can be specified only with `vec` estimation results.

### Std. errors

`nose` specifies that the asymptotic standard errors of the forecasted levels, and thus the asymptotic confidence intervals for the levels, not be calculated. By default, the asymptotic standard errors and the asymptotic confidence intervals of the forecasted levels are calculated.

`bs` specifies that `fcast compute` use confidence bounds estimated by a simulation method based on bootstrapping the residuals.

`bsp` specifies that `fcast compute` use confidence bounds estimated via simulation in which the innovations are drawn from a multivariate normal distribution.

`bscentile` specifies that `fcast compute` use centiles of the bootstrapped dataset to estimate the bounds of the confidence intervals. By default, `fcast compute` uses the estimated standard errors and the quantiles of the standard normal distribution determined by `level()`.

`reps(#)` gives the number of repetitions used in the simulations. The default is `reps(200)`.

`nodots` specifies that no dots be displayed while obtaining the simulation-based standard errors. By default, for each replication, a dot is displayed.

`saving(filename[, replace])` specifies the name of the file to hold the dataset that contains the bootstrap replications. The `replace` option overwrites any file with this name.

`replace` specifies that *filename* be overwritten if it exists. This option is not shown in the dialog box.

#### Reporting

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [\[U\] 20.8 Specifying the width of confidence intervals](#).

## Remarks and examples

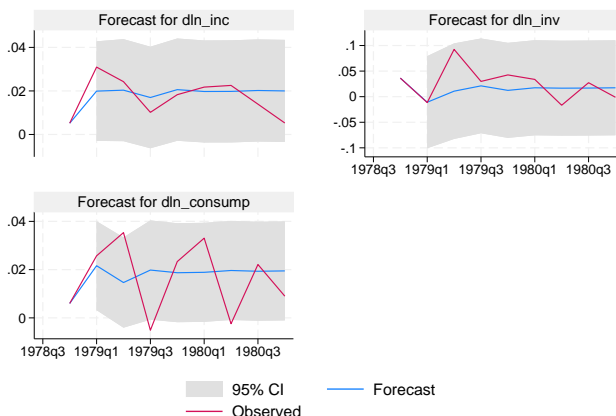
Researchers often use vector autoregressive (VAR) models and vector error-correction (VEC) models to construct dynamic forecasts. `fcast compute` computes dynamic forecasts of the dependent variables in a VAR model or a VEC model previously fit by `var`, `svar`, or `vec`. If you are interested in conditional, one-step-ahead predictions, use `predict` (see [\[TS\] var](#), [\[TS\] var svar](#), and [\[TS\] vec](#)).

To obtain and analyze dynamic forecasts, you fit a model, use `fcast compute` to compute the dynamic forecasts, and use `fcast graph` to graph the results.

### ► Example 1

Typing the following commands fits a VAR model with two lags, computes eight-step dynamic predictions for each endogenous variable, and produces the graph:

```
. use https://www.stata-press.com/data/r19/lutkepohl2
. var dln_inc dln_consump dln_inv if qtr<tq(1979q1)
. fcast compute m2_, step(8)
. fcast graph m2_dln_inc m2_dln_inv m2_dln_consump, observed
```



The graph shows that the model is better at predicting changes in income and investment than in consumption. The graph also shows how quickly the predictions from the two-lag model settle down to their mean values.

fcast compute creates new variables in the dataset. If there are  $K$  dependent variables in the previously fitted model, fcast compute generates  $4K$  new variables:

$K$  new variables that hold the forecasted levels, named by appending the specified prefix to the name of the original variable

$K$  estimated lower bounds for the forecast interval, named by appending the specified prefix and the suffix “\_LB” to the name of the original variable

$K$  estimated upper bounds for the forecast interval, named by appending the specified prefix and the suffix “\_UB” to the name of the original variable

$K$  estimated standard errors of the forecast, named by appending the specified prefix and the suffix “\_SE” to the name of the original variable

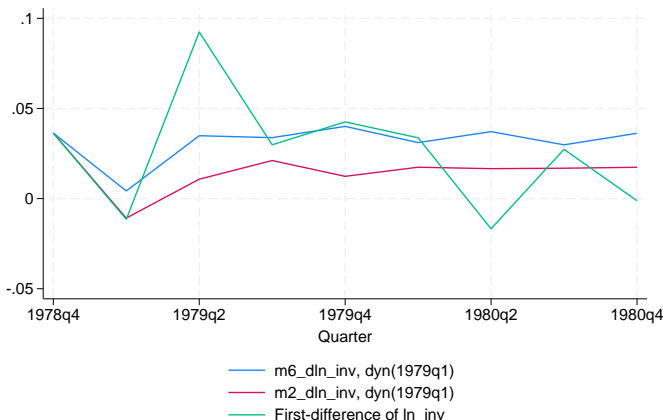
If you specify options so that fcast compute does not calculate standard errors, the  $3K$  variables that hold them and the bounds of the confidence intervals are not generated.

If the model previously fit is a VEC model, specifying differences generates another  $K$  variables that hold the forecasts of the first differences of the dependent variables, named by appending the prefix “prefixD\_” to the name of the original variable.

## ► Example 2

Plots of the forecasts from different models along with the observations from a holdout sample can provide insights to their relative forecasting performance. Continuing the previous example,

```
. var dln_inc dln_consump dln_inv if qtr<tq(1979q1), lags(1/6)
  (output omitted)
. fcast compute m6_, step(8)
. tsline m6_dln_inv m2_dln_inv dln_inv
> if m6_dln_inv < ., legend(cols(1))
```



The model with six lags predicts changes in investment better than the two-lag model in some periods but markedly worse in other periods.

## Methods and formulas

Methods and formulas are presented under the following headings:

*Predictions after var and svar*

*Dynamic forecasts after vec*

### Predictions after var and svar

A VAR model with endogenous variables  $\mathbf{y}_t$  and exogenous variables  $\mathbf{x}_t$  can be written as

$$\mathbf{y}_t = \mathbf{v} + \mathbf{A}_1 \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{B} \mathbf{x}_t + \mathbf{u}_t$$

where

$$t = 1, \dots, T$$

$\mathbf{y}_t = (y_{1t}, \dots, y_{Kt})'$  is a  $K \times 1$  random vector,

the  $\mathbf{A}_i$  are fixed  $(K \times K)$  matrices of parameters,

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

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

$\mathbf{v}$  is a  $(K \times 1)$  vector of fixed parameters, and

$\mathbf{u}_t$  is assumed to be white noise; that is,

$$E(\mathbf{u}_t) = \mathbf{0}_K$$

$$E(\mathbf{u}_t \mathbf{u}_t') = \Sigma$$

$$E(\mathbf{u}_t \mathbf{u}_s') = \mathbf{0}_K \text{ for } t \neq s$$

fcast compute will dynamically predict the variables in the vector  $\mathbf{y}_t$  conditional on  $p$  initial values of the endogenous variables and any exogenous  $\mathbf{x}_t$ . Adopting the notation from Lütkepohl (2005, 402) to fit the case at hand, the optimal  $h$ -step-ahead forecast of  $\mathbf{y}_{t+h}$  conditional on  $\mathbf{x}_t$  is

$$\mathbf{y}_t(h) = \hat{\mathbf{v}} + \hat{\mathbf{A}}_1 \mathbf{y}_t(h-1) + \cdots + \hat{\mathbf{A}}_p \mathbf{y}_t(h-p) + \hat{\mathbf{B}} \mathbf{x}_t \quad (1)$$

If there are no exogenous variables, (1) becomes

$$\mathbf{y}_t(h) = \hat{\mathbf{v}} + \hat{\mathbf{A}}_1 \mathbf{y}_t(h-1) + \cdots + \hat{\mathbf{A}}_p \mathbf{y}_t(h-p)$$

When there are no exogenous variables, fcast compute can compute the asymptotic confidence bounds.

As shown by Lütkepohl (2005, 204–205), the asymptotic estimator of the covariance matrix of the prediction error is given by

$$\widehat{\Sigma}_{\hat{\mathbf{y}}}(h) = \widehat{\Sigma}_y(h) + \frac{1}{T} \widehat{\Omega}(h) \quad (2)$$

where

$$\begin{aligned} \widehat{\Sigma}_y(h) &= \sum_{i=0}^{h-1} \widehat{\Phi}_i \widehat{\Sigma} \widehat{\Phi}_i' \\ \widehat{\Omega}(h) &= \frac{1}{T} \sum_{t=0}^T \left\{ \sum_{i=0}^{h-1} \mathbf{z}_t' (\widehat{\mathbf{B}}')^{h-1-i} \otimes \widehat{\Phi}_i \right\} \widehat{\Sigma}_{\beta} \left\{ \sum_{i=0}^{h-1} \mathbf{z}_t' (\widehat{\mathbf{B}}')^{h-1-i} \otimes \widehat{\Phi}_i \right\}' \\ \widehat{\mathbf{B}} &= \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{v}} & \hat{\mathbf{A}}_1 & \hat{\mathbf{A}}_2 & \cdots & \hat{\mathbf{A}}_{p-1} & \hat{\mathbf{A}}_p \\ \mathbf{0} & \mathbf{I}_K & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_K & & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_K & \mathbf{0} \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
\mathbf{Z}_t &= (1, \mathbf{y}'_t, \dots, \mathbf{y}'_{t-p-1})' \\
\widehat{\boldsymbol{\Phi}}_0 &= \mathbf{I}_K \\
\widehat{\boldsymbol{\Phi}}_i &= \sum_{j=1}^i \widehat{\boldsymbol{\Phi}}_{i-j} \widehat{\mathbf{A}}_j \quad i = 1, 2, \dots \\
\widehat{\mathbf{A}}_j &= \mathbf{0} \quad \text{for } j > p
\end{aligned}$$

$\widehat{\boldsymbol{\Sigma}}$  is the estimate of the covariance matrix of the innovations, and  $\widehat{\boldsymbol{\Sigma}}_\beta$  is the estimated VCE of the coefficients in the VAR model. The formula in (2) is general enough to handle the case in which constraints are placed on the coefficients in the VAR( $p$ ) model.

Equation (2) is made up of two terms.  $\widehat{\boldsymbol{\Sigma}}_y(h)$  is the estimated mean squared error (MSE) of the forecast.  $\widehat{\boldsymbol{\Sigma}}_y(h)$  estimates the error in the forecast arising from the unseen innovations.  $T^{-1}\widehat{\boldsymbol{\Omega}}(h)$  estimates the error in the forecast that is due to using estimated coefficients instead of the true coefficients. As the sample size grows, uncertainty with respect to the coefficient estimates decreases, and  $T^{-1}\widehat{\boldsymbol{\Omega}}(h)$  goes to zero.

If  $\mathbf{y}_t$  is normally distributed, the bounds for the asymptotic  $(1 - \alpha)100\%$  interval around the forecast for the  $k$ th component of  $\mathbf{y}_t$ ,  $h$  periods ahead, are

$$\hat{\mathbf{y}}_{k,t}(h) \pm z_{(\frac{\alpha}{2})} \hat{\sigma}_k(h) \quad (3)$$

where  $\hat{\sigma}_k(h)$  is the  $k$ th diagonal element of  $\widehat{\boldsymbol{\Sigma}}_y(h)$ .

Specifying the `bs` option causes the standard errors to be computed via simulation, using bootstrapped residuals. Both `var` and `svar` contain estimators for the coefficients of a VAR model that are conditional on the first  $p$  observations on the endogenous variables in the data. Similarly, these algorithms are conditional on the first  $p$  observations of the endogenous variables in the data. However, the simulation-based estimates of the standard errors are also conditional on the estimated coefficients. The asymptotic standard errors are not conditional on the coefficient estimates because the second term on the right-hand side of (2) accounts for the uncertainty arising from using estimated parameters.

For a simulation with  $R$  repetitions, this method uses the following algorithm:

1. Fit the model and save the estimated coefficients.
2. Use the estimated coefficients to calculate the residuals.

3. Repeat steps 3a–3c  $R$  times.
  - 3a. Draw a simple random sample with replacement of size  $T + h$  from the residuals. When the  $t$ th observation is drawn, all  $K$  residuals are selected, preserving any contemporaneous correlation among the residuals.
  - 3b. Use the sampled residuals,  $p$  initial values of the endogenous variables, any exogenous variables, and the estimated coefficients to construct a new sample dataset.
  - 3c. Save the simulated endogenous variables for the  $h$  forecast periods in the bootstrapped dataset.
4. For each endogenous variable and each forecast period, the simulated standard error is the estimated standard error of the  $R$  simulated forecasts. By default, the upper and lower bounds of the  $(1 - \alpha)100\%$  are estimated using the simulation-based estimates of the standard errors and the normality assumption, as in (3). If the `bscentile` option is specified, the sample centiles for the upper and lower bounds of the  $R$  simulated forecasts are used for the upper and lower bounds of the confidence intervals.

If the `bsp` option is specified, a parametric simulation algorithm is used. Specifically, everything is as above except that 3a is replaced by 3a(bsp) as follows:

- 3a(bsp). Draw  $T + h$  observations from a multivariate normal distribution with covariance matrix  $\widehat{\Sigma}$ .

The algorithm above assumes that  $h$  forecast periods come after the original sample of  $T$  observations. If the  $h$  forecast periods lie within the original sample, smaller simulated datasets are sufficient.

## Dynamic forecasts after vec

*Methods and formulas* of [TS] `vec` discusses how to obtain the one-step predicted differences and levels. `fcast compute` uses the previous dynamic predictions as inputs for later dynamic predictions.

Per Lütkepohl (2005, sec. 6.5), `fcast compute` uses

$$\widehat{\Sigma}_{\hat{y}}(h) = \left( \frac{T}{T-d} \right) \sum_{i=0}^{h-1} \widehat{\Phi}_i \widehat{\Omega} \widehat{\Phi}_i$$

where the  $\widehat{\Phi}_i$  are the estimated matrices of impulse–response functions,  $T$  is the number of observations in the sample,  $d$  is the number of degrees of freedom, and  $\widehat{\Omega}$  is the estimated cross-equation variance matrix. The formulas for  $d$  and  $\widehat{\Omega}$  are given in *Methods and formulas* of [TS] `vec`.

The estimated standard errors at step  $h$  are the square roots of the diagonal elements of  $\widehat{\Sigma}_{\hat{y}}(h)$ .

Per Lütkepohl (2005), the estimated forecast-error variance does not consider parameter uncertainty. As the sample size gets infinitely large, the importance of parameter uncertainty diminishes to zero.

## References

- Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press. <https://doi.org/10.2307/j.ctv14jx6sm>.
- Lütkepohl, H. 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.



## Also see

[TS] [fcast graph](#) — Graph forecasts after fcast compute

[TS] [var intro](#) — Introduction to vector autoregressive models

[TS] [vec intro](#) — Introduction to vector error-correction models

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](#).

