sspace postestimation — Postestimation tools for sspace

Postestimation commands predict Remarks and examples Methods and formulas
References Also see

Postestimation commands

The following standard postestimation commands are available after `sspace`:

Command	Description
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`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
`lrtest`	likelihood-ratio test
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions, latent states, residuals, etc.
`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 expected values. The root mean squared error is available for all predictions. All predictions are also available as static one-step-ahead predictions or as dynamic multistep predictions, and you can control when dynamic predictions begin.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] { *stub*`*` | *newvarlist* } [*if*] [*in*] [, *statistic options*]

statistic	Description
Main	
`xb`	observable variables
`states`	latent state variables
`residuals`	residuals
`rstandard`	standardized residuals

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

options	Description
Options	
`equation(`*eqnames*`)`	name(s) of equation(s) for which predictions are to be made
`rmse(`*stub*`*` | *newvarlist*`)`	put estimated root mean squared errors of predicted statistics in new variables
`dynamic(`*time_constant*`)`	begin dynamic forecast at specified time
Advanced	
`smethod(`*method*`)`	method for predicting unobserved states

method	Description
`onestep`	predict using past information
`smooth`	predict using all sample information
`filter`	predict using past and contemporaneous information

Options for predict

Main

`xb`, `states`, `residuals`, and `rstandard` specify the statistic to be predicted.

`xb`, the default, calculates the linear predictions of the observed variables.

`states` calculates the linear predictions of the latent state variables.

`residuals` calculates the residuals in the equations for observable variables. `residuals` may not be specified with `dynamic()`.

`rstandard` calculates the standardized residuals, which are the residuals normalized to be uncorrelated and to have unit variances. `rstandard` may not be specified with `smethod(filter)`, `smethod(smooth)`, or `dynamic()`.

Options

`equation(`*eqnames*`)` specifies the equation(s) for which the predictions are to be calculated. If you do not specify `equation()` or *stub*`*`, the results are the same as if you had specified the name of the first equation for the predicted statistic.

You specify a list of equation names, such as `equation(income consumption)` or `equation(factor1 factor2)`, to identify the equations. Specify names of state equations when predicting `states` and names of observable equations in all other cases.

`equation()` may not be specified with *stub*`*`.

`rmse(`*stub*`*` | *newvarlist*`)` puts the root mean squared errors of the predicted statistics into the specified new variables. The root mean squared errors measure the variances due to the disturbances but do not account for estimation error.

`dynamic(`*time_constant*`)` specifies when `predict` starts producing dynamic forecasts. The specified *time_constant* must be in the scale of the time variable specified in `tsset`, and the *time_constant* must be inside a sample for which observations on the dependent variables are available. For example, `dynamic(tq(2008q4))` causes dynamic predictions to begin in the fourth quarter of 2008, assuming that your time variable is quarterly; see [D] **Datetime**. If the model contains exogenous variables, they must be present for the whole predicted sample. `dynamic()` may not be specified with `rstandard`, `residuals`, or `smethod(smooth)`.

Advanced

`smethod(`*method*`)` specifies the method for predicting the unobserved states; `smethod(onestep)`, `smethod(filter)`, and `smethod(smooth)` cause different amounts of information on the dependent variables to be used in predicting the states at each time period.

`smethod(onestep)`, the default, causes `predict` to estimate the states at each time period using previous information on the dependent variables. The Kalman filter is performed on previous periods, but only the one-step predictions are made for the current period.

`smethod(smooth)` causes `predict` to estimate the states at each time period using all the sample data by the Kalman smoother. `smethod(smooth)` may not be specified with `rstandard`.

`smethod(filter)` causes `predict` to estimate the states at each time period using previous and contemporaneous data by the Kalman filter. The Kalman filter is performed on previous periods and the current period. `smethod(filter)` may be specified only with `states`.

Remarks and examples

We assume that you have already read [TS] **sspace**. In this entry, we illustrate some of the features of `predict` after using `sspace` to estimate the parameters of a state-space model.

All the predictions after `sspace` depend on the unobserved states, which are estimated recursively. Changing the sample can alter the state estimates, which can change all other predictions.

▷ Example 1: One-step predictions

In example 5 of [TS] **sspace**, we estimated the parameters of the dynamic-factor model

$$\begin{pmatrix} f_t \\ f_{t-1} \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{t-1} \\ f_{t-2} \end{pmatrix} + \begin{pmatrix} \nu_t \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} \Delta\texttt{ipman}_t \\ \Delta\texttt{income}_t \\ \Delta\texttt{hours}_t \\ \Delta\texttt{unemp}_t \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} f_t + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix}$$

where

$$\text{Var}\begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}$$

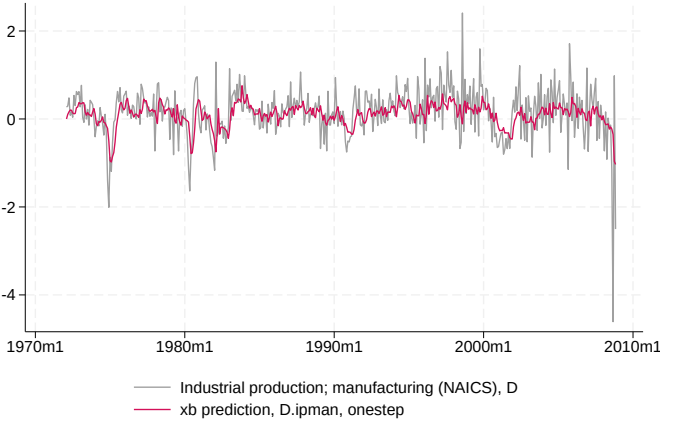
by typing

```
. use https://www.stata-press.com/data/r19/dfex
(St. Louis Fed (FRED) macro data)
. constraint 1 [lf]L.f = 1
. sspace (f L.f L.lf, state noconstant)
>        (lf L.f, state noconstant noerror)
>        (D.ipman f, noconstant)
>        (D.income f, noconstant)
>        (D.hours f, noconstant)
>        (D.unemp f, noconstant),
>        covstate(identity) constraints(1)
```

(*output omitted*)

Below we obtain the one-step predictions for each of the four dependent variables in the model, and then we graph the actual and predicted `ipman`:

```
. predict dep*
(option xb assumed; fitted values)
. tsline D.ipman dep1, lcolor(gs10) xtitle("") legend(rows(2))
```

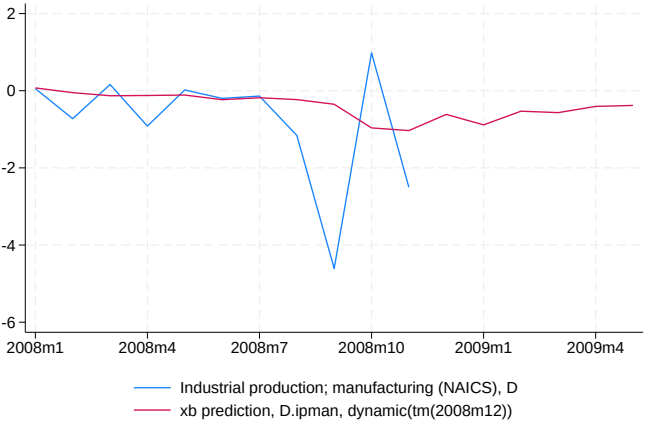


The graph shows that the one-step predictions account for only a small part of the swings in the realized `ipman`.

Example 2: Out-of-sample, dynamic predictions

We use the estimates from the previous example to make out-of-sample predictions. After using `tsappend` to extend the dataset by six periods, we use `predict` with the `dynamic()` option and graph the result.

```
. tsappend, add(6)
. predict Dipman_f, dynamic(tm(2008m12)) equation(D.ipman)
. tsline D.ipman Dipman_f if month>=tm(2008m1), xtitle("") legend(rows(2))
```



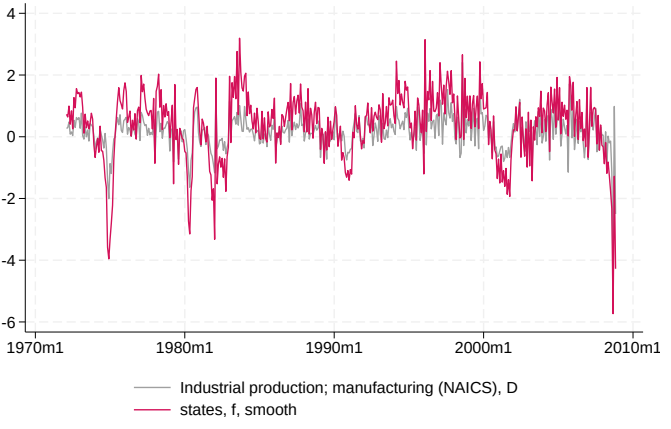
The model predicts that the changes in industrial production will remain negative for the forecast horizon, although they increase toward zero.

Example 3: Estimating an unobserved factor

In this example, we want to estimate the unobserved factor instead of predicting a dependent variable. Specifying `smethod(smooth)` causes `predict` to use all sample information in estimating the states by the Kalman smoother.

Below we estimate the unobserved factor by using the estimation sample, and we graph `ipman` and the estimated factor:

```
. predict fac if e(sample), states smethod(smooth) equation(f)
. tsline D.ipman fac, lcolor(gs10) xtitle("") legend(rows(2))
```

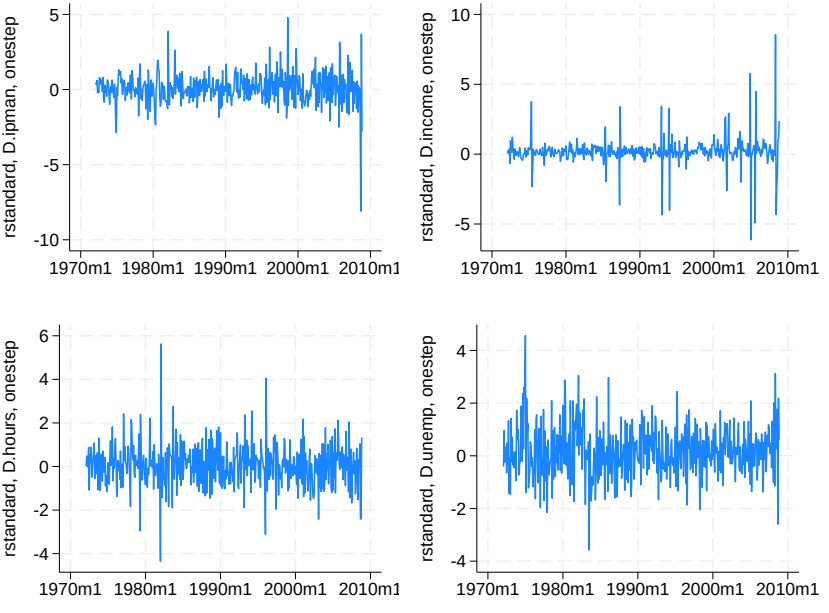


▷ Example 4: Calculating residuals

The residuals and the standardized residuals are frequently used to review the specification of the model.

Below we calculate the standardized residuals for each of the series and display them in a combined graph:

```
. predict sres1-sres4 if e(sample), rstandard
. tsline sres1, xtitle("") name(sres1)
. tsline sres2, xtitle("") name(sres2)
. tsline sres3, xtitle("") name(sres3)
. tsline sres4, xtitle("") name(sres4)
. graph combine sres1 sres2 sres3 sres4, name(combined)
```



◁

Methods and formulas

Estimating the unobserved states is key to predicting the dependent variables.

By default and with the `smethod(onestep)` option, `predict` estimates the states in each period by applying the Kalman filter to all previous periods and only making the one-step predictions to the current period. (See *Methods and formulas* of [TS] **sspace** for the Kalman filter equations.)

With the `smethod(filter)` option, `predict` estimates the states in each period by applying the Kalman filter on all previous periods and the current period. The computational difference between `smethod(onestep)` and `smethod(filter)` is that `smethod(filter)` performs the update step on the current period while `smethod(onestep)` does not. The statistical difference between `smethod(onestep)` and `smethod(filter)` is that `smethod(filter)` uses contemporaneous information on the dependent variables while `smethod(onestep)` does not.

As noted in [TS] **sspace**, `sspace` has both a stationary and a diffuse Kalman filter. `predict` uses the same Kalman filter used for estimation.

With the `smethod(smooth)` option, `predict` estimates the states in each period using all the sample information by applying the Kalman smoother. `predict` uses the Harvey (1989, sec. 3.6.2) fixed-interval smoother with model-based initial values to estimate the states when the estimated parameters imply a stationary model. De Jong (1989) provides a computationally efficient method. Hamilton (1994) discusses the model-based initial values for stationary state-space models. When the model is nonstationary, the De Jong (1989) diffuse Kalman smoother is used to predict the states. The smoothed estimates of the states are subsequently used to predict the dependent variables.

The dependent variables are predicted by plugging in the estimated states. The residuals are calculated as the differences between the predicted and the realized dependent variables. The root mean squared errors are the square roots of the diagonal elements of the mean squared error matrices that are computed by the Kalman filter. The standardized residuals are the residuals normalized by the Cholesky factor of their mean squared error produced by the Kalman filter.

`predict` uses the Harvey (1989, sec. 3.5) methods to compute the dynamic forecasts and the root mean squared errors. Let τ be the period at which the dynamic forecasts begin; τ must either be in the specified sample or be in the period immediately following the specified sample.

The dynamic forecasts depend on the predicted states in the period $\tau - 1$, which `predict` obtains by running the Kalman filter or the diffuse Kalman filter on the previous sample observations. The states in the periods prior to starting the dynamic predictions may be estimated using `smethod(onestep)` or `smethod(smooth)`.

Using an `if` or `in` qualifier to alter the prediction sample can change the estimate of the unobserved states in the period prior to beginning the dynamic predictions and hence alter the dynamic predictions. The initial states are estimated using `e(b)` and the prediction sample.

References

De Jong, P. 1988. The likelihood for a state space model. *Biometrika* 75: 165–169. https://doi.org/10.2307/2336450.

———. 1989. Smoothing and interpolation with the state-space model. *Journal of the American Statistical Association* 84: 1085–1088. https://doi.org/10.2307/2290087.

———. 1991. The diffuse Kalman filter. *Annals of Statistics* 19: 1073–1083. https://doi.org/10.1214/aos/1176348139.

Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctv14jx6sm.

Harvey, A. C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.

Lütkepohl, H. 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.

Also see

[TS] **sspace** — State-space models

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

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

[U] 20 Estimation and postestimation commands