dfactor postestimation — Postestimation tools for dfactor

Postestimation commands

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

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`	expected values, unobserved factors, autocorrelated disturbances, innovations, 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, unobserved factors, autocorrelated disturbances, and innovations. 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	
`y`	dependent variable, which is `xbf` + `residuals`
`xb`	linear predictions using the observable independent variables
`xbf`	linear predictions using the observable independent variables plus the factor contributions
`factors`	unobserved factor variables
`residuals`	autocorrelated disturbances
`innovations`	innovations, the observed dependent variable minus the predicted `y`

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*`)`	specify name(s) of equation(s) for which predictions are to be made
`rmse(`*stub** | *newvarlist*`)`	put estimated root mean squared errors of predicted objects 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

The mathematical notation used in this section is defined in *Description* of [TS] **dfactor**.

Main

y, xb, xbf, factors, residuals, and innovations specify the statistic to be predicted.

y, the default, predicts the dependent variables. The predictions include the contributions of the unobserved factors, the linear predictions by using the observable independent variables, and any autocorrelation, $\widehat{\mathbf{P}}\widehat{\mathbf{f}}_t + \widehat{\mathbf{Q}}\mathbf{x}_t + \widehat{\mathbf{u}}_t$.

xb calculates the linear prediction by using the observable independent variables, $\widehat{\mathbf{Q}}\mathbf{x}_t$.

xbf calculates the contributions of the unobserved factors plus the linear prediction by using the observable independent variables, $\widehat{\mathbf{P}}\widehat{\mathbf{f}}_t + \widehat{\mathbf{Q}}\mathbf{x}_t$.

factors estimates the unobserved factors, $\widehat{\mathbf{f}}_t = \widehat{\mathbf{R}}\mathbf{w}_t + \widehat{\mathbf{A}}_1\widehat{\mathbf{f}}_{t-1} + \widehat{\mathbf{A}}_2\widehat{\mathbf{f}}_{t-2} + \cdots + \widehat{\mathbf{A}}_{t-p}\widehat{\mathbf{f}}_{t-p}$.

residuals calculates the autocorrelated residuals, $\widehat{\mathbf{u}}_t = \widehat{\mathbf{C}}_1\widehat{\mathbf{u}}_{t-1} + \widehat{\mathbf{C}}_2\widehat{\mathbf{u}}_{t-2} + \cdots + \widehat{\mathbf{C}}_{t-q}\widehat{\mathbf{u}}_{t-q}$.

innovations calculates the innovations, $\widehat{\epsilon}_t = \mathbf{y}_t - \widehat{\mathbf{P}}\widehat{\mathbf{f}}_t + \widehat{\mathbf{Q}}\mathbf{x}_t - \widehat{\mathbf{u}}_t$.

Options

equation(*eqnames*) specifies the equation(s) for which the predictions are to be calculated.

You specify equation names, such as equation(income consumption) or equation(factor1 factor2), to identify the equations. For the factors statistic, you must specify names of equations for factors; for all other statistics, you must specify names of equations for observable variables.

If you do not specify equation() and do not specify *stub**, the results are the same as if you had specified the name of the first equation for the predicted statistic.

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

rmse(*stub** | *newvarlist*) puts the root mean squared errors of the predicted objects 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 xb, xbf, innovations, smethod(filter), or smethod(smooth).

Advanced

smethod(*method*) specifies the method used to predict the unobserved states in the model. smethod() may not be specified with xb.

smethod(onestep), the default, causes predict to use 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(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 `factors` and `residuals`.

Remarks and examples

We assume that you have already read [TS] **dfactor**. In this entry, we illustrate some of the features of `predict` after using `dfactor`.

`dfactor` writes the specified model as a state-space model and estimates the parameters by maximum likelihood. The unobserved factors and the residuals are states in the state-space form of the model, and they are estimated by the Kalman filter or the Kalman smoother. The `smethod()` option controls how these states are estimated.

The Kalman filter or Kalman smoother is run over the specified sample. Changing the sample can alter the predicted value for a given observation, because the Kalman filter and Kalman smoother are recursive algorithms.

After estimating the parameters of a dynamic-factor model, there are many quantities of potential interest. Here we will discuss several of these statistics and illustrate how to use `predict` to compute them.

▷ Example 1: One-step, out-of-sample forecasts

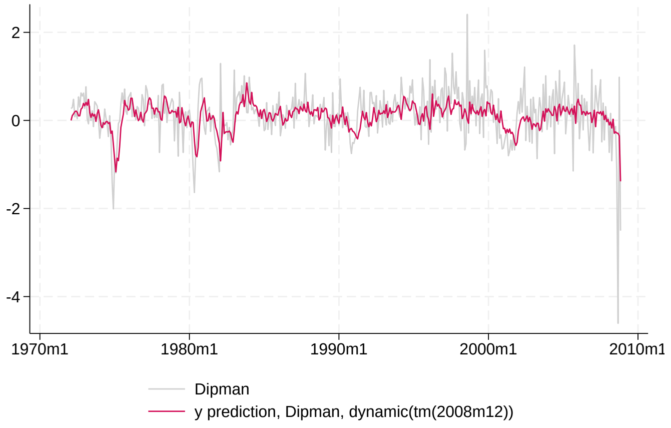
Let's begin by estimating the parameters of the dynamic-factor model considered in example 2 in [TS] **dfactor**.

```
. use https://www.stata-press.com/data/r19/dfex
(St. Louis Fed (FRED) macro data)

. dfactor (D.(ipman income hours unemp) = , noconstant ar(1)) (f = , ar(1/2))
  (output omitted)
```

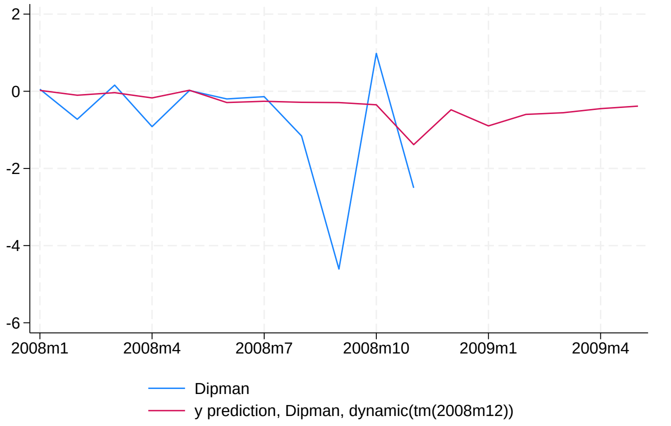
While several of the six statistics computed by `predict` might be of interest, we will look only at a few of these statistics for `D.ipman`. We begin by obtaining one-step predictions in the estimation sample and a six-month dynamic forecast for `D.ipman`. The graph of the in-sample predictions indicates that our model accounts only for a small fraction of the variability in `D.ipman`.

```
. tsappend, add(6)
. predict Dipman_f, dynamic(tm(2008m12)) equation(D.ipman)
(option y assumed; fitted values)
. tsline D.ipman Dipman_f if month<=tm(2008m11), lcolor(gs13) xtitle("")
> legend(rows(2))
```



Graphing the last year of the sample and the six-month out-of-sample forecast yields

```
. tsline D.ipman Dipman_f if month>=tm(2008m1), xtitle("") legend(rows(2))
```

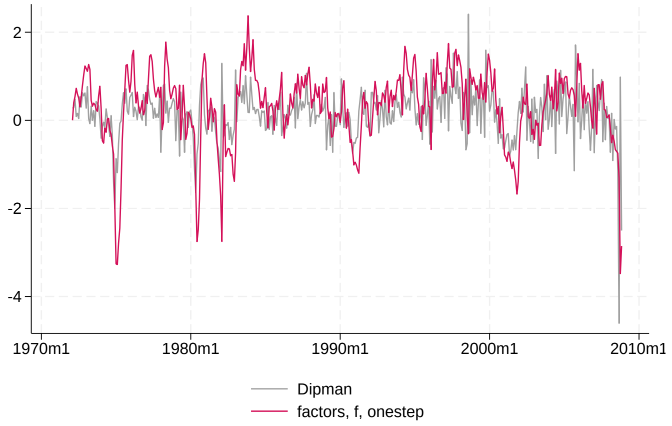


◁

▷ Example 2: Estimating an unobserved factor

Another common task is to estimate an unobserved factor. We can estimate the unobserved factor at each time period by using only previous information (the `smethod(onestep)` option), previous and contemporaneous information (the `smethod(filter)` option), or all the sample information (the `smethod(smooth)` option). We are interested in the one-step predictive power of the unobserved factor, so we use the default, `smethod(onestep)`.

```
. predict fac if e(sample), factor
. tsline D.ipman fac, lcolor(gs10) xtitle("") legend(rows(2))
```



◁

Methods and formulas

`dfactor` estimates the parameters by writing the model in state-space form and using `sspace`. Analogously, `predict` after `dfactor` uses the methods described in [TS] **sspace postestimation**. The unobserved factors and the residuals are states in the state-space form of the model.

See *Methods and formulas* of [TS] **sspace postestimation** for how predictions are made after estimating the parameters of a state-space model.

Also see

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

[TS] **sspace** — State-space models

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

[U] 20 Estimation and postestimation commands

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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.