

ucm postestimation — Postestimation tools for ucm

Postestimation commands
Methods and formulas

[predict](#)
Also see

[estat](#)

Remarks and examples

Postestimation commands

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

Command	Description
<code>estat period</code>	display cycle periods in time units
<code>psdensity</code>	estimate the spectral density

The following standard postestimation commands are also available:

Command	Description
<code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing and inference for linear combinations of coefficients
<code>lrtest</code>	likelihood-ratio test
<code>nlcom</code>	point estimates, standard errors, testing and inference for nonlinear combinations of coefficients
<code>predict</code>	linear predictions, trend components, seasonal components, cycle components, residuals, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, trend components, seasonal components, cyclical components, and standardized and unstandardized residuals. 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]
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>xb</code>	linear prediction using exogenous variables
<code>trend</code>	trend component
<code>seasonal</code>	seasonal component
<code>cycle</code>	cyclical component
<code>residuals</code>	residuals
<code>rstandard</code>	standardized residuals

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

<i>options</i>	Description
----------------	-------------

Options

<code>rmse(<i>stub*</i> <i>newvarlist</i>)</code>	put estimated root mean squared errors of predicted statistics in the new variable
<code>dynamic(<i>time_constant</i>)</code>	begin dynamic forecast at specified time

Advanced

<code>smethod(<i>method</i>)</code>	method for predicting unobserved components
-------------------------------------	---

<i>method</i>	Description
---------------	-------------

<code>onestep</code>	predict using past information
<code>smooth</code>	predict using all sample information
<code>filter</code>	predict using past and contemporaneous information

Options for predict

Main

`xb`, `trend`, `seasonal`, `cycle`, `residuals`, and `rstandard` specify the statistic to be predicted.

`xb`, the default, calculates the linear predictions using the exogenous variables. `xb` may not be used with the `smethod(filter)` option.

`trend` estimates the unobserved trend component.

`seasonal` estimates the unobserved seasonal component.

`cycle` estimates the unobserved cyclical component.

`residuals` calculates the residuals in the equation for the dependent variable. `residuals` may not be specified with `dynamic()`.

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

Options

`rmse` (`stub*` | `newvarlist`) puts the root mean squared errors of the predicted statistic into the specified new variable. Multiple variables are only required for predicting cycles of a model that has more than one cycle. The root mean squared errors measure the variances due to the disturbances but do not account for estimation error. The `stub*` syntax is for models with multiple cycles, where you provide the prefix and `predict` will add a numeric suffix for each predicted cycle.

`dynamic` (`time_constant`) specifies when `predict` should start 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 variable 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 the `rstandard`, `residuals`, or `smethod(smooth)` option.

Advanced

`smethod` (`method`) specifies the method for predicting the unobserved components. `smethod()` causes different amounts of information on the dependent variable to be used in predicting the components at each time period.

`smethod(onestep)`, the default, causes `predict` to estimate the components at each time period using previous information on the dependent variable. 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 components at each time period using all the sample data by the Kalman smoother. `smethod(smooth)` may not be specified with the `rstandard` option.

`smethod(filter)` causes `predict` to estimate the components 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 not be specified with the `xb` option.

estat

Description for estat

`estat period` transforms an estimated central frequency to an estimated period after `ucm`.

Menu for estat

Statistics > Postestimation

Syntax for estat

```
estat period [ , options ]
```

<i>options</i>	Description
<code>cformat(%<i>fmt</i>)</code>	specify numeric format
<code><u>l</u>evel(#)</code>	set confidence level; default is <code>level(95)</code>

Options for estat period

`cformat(%fmt)` sets the display format for the table numeric values. The default is `cformat(%9.0g)`.

`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

[stata.com](#)

We assume that you have already read [\[TS\] ucm](#). In this entry, we illustrate some features of `predict` after using `ucm` to estimate the parameters of an unobserved-components model.

All predictions after `ucm` depend on the unobserved components, which are estimated recursively using a Kalman filter. Changing the sample can alter the state estimates, which can change all other predictions.

▷ Example 1

We begin by modeling monthly data on the median duration of employment spells in the United States. We include a stochastic-seasonal component because the data have not been seasonally adjusted.

```
. use https://www.stata-press.com/data/r18/uduration2
(BLS data, not seasonally adjusted)

. ucm duration, seasonal(12) cycle(1) difficult
searching for initial values .....
(setting technique to bhhh)
Iteration 0: Log likelihood = -409.79649
Iteration 1: Log likelihood = -403.38335
Iteration 2: Log likelihood = -403.37399 (backed up)
Iteration 3: Log likelihood = -403.36926 (backed up)
Iteration 4: Log likelihood = -403.36689 (backed up)
(switching technique to nr)
Iteration 5: Log likelihood = -403.36674 (backed up)
Iteration 6: Log likelihood = -397.87755 (not concave)
Iteration 7: Log likelihood = -395.5778 (not concave)
Iteration 8: Log likelihood = -390.45932 (not concave)
Iteration 9: Log likelihood = -389.47811 (not concave)
Iteration 10: Log likelihood = -388.88244 (not concave)
Iteration 11: Log likelihood = -388.536
Iteration 12: Log likelihood = -388.30418
Iteration 13: Log likelihood = -388.25717
Iteration 14: Log likelihood = -388.25675
Iteration 15: Log likelihood = -388.25675
Refining estimates:
Iteration 0: Log likelihood = -388.25675
Iteration 1: Log likelihood = -388.25675

Unobserved-components model
Components: random walk, seasonal(12), order 1 cycle
Sample: 1967m7 thru 2008m12
Log likelihood = -388.25675
Number of obs = 498
Wald chi2(2) = 7.17
Prob > chi2 = 0.0277
```

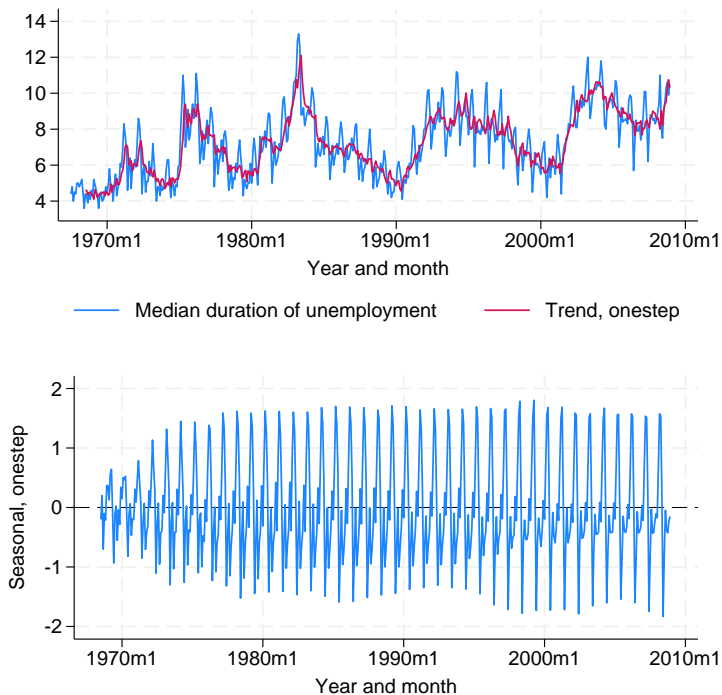
duration	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
frequency	1.641531	.7250323	2.26	0.024	.2204938	3.062568
damping	.2671232	.1050168	2.54	0.011	.0612939	.4729524
var(level)	.1262922	.0221428	5.70	0.000	.0828932	.1696912
var(seasonal)	.0017289	.0009647	1.79	0.037	0	.0036196
var(cycle1)	.0641496	.0211839	3.03	0.001	.0226299	.1056693

Note: Model is not stationary.

Note: Tests of variances against zero are one sided, and the two-sided confidence intervals are truncated at zero.

Below we predict the trend and the seasonal components to get a look at the model fit.

```
. predict strend, trend
. predict season, seasonal
. tsline duration strend, name(trend) nodraw legend(rows(1))
. tsline season, name(season) yline(0,lwidth(vthin)) nodraw
. graph combine trend season, rows(2)
```

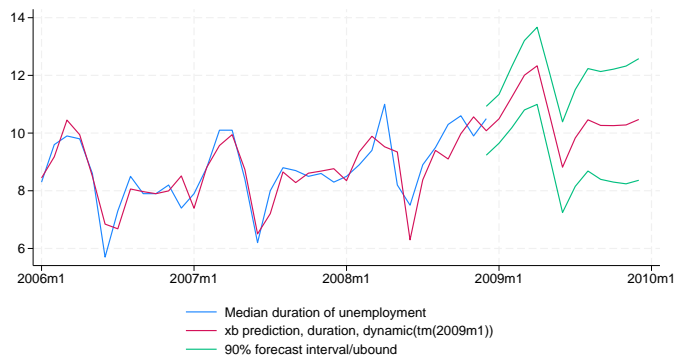


The trend tracks the data well. That the seasonal component appears to change over time indicates that the stochastic-seasonal component might fit better than a deterministic-seasonal component.

► Example 2

In this example, we use the model to forecast the median unemployment duration. We use the root mean squared error of the prediction to compute a confidence interval of our dynamic predictions. Recall that the root mean squared error accounts for variances due to the disturbances but not due to the estimation error.

```
. tsappend, add(12)
. predict duration_f, dynamic(tm(2009m1)) rmse(rmse)
. scalar z = invnormal(0.95)
. generate lbound = duration_f - z*rmse if tm>=tm(2008m12)
(497 missing values generated)
. generate ubound = duration_f + z*rmse if tm>=tm(2008m12)
(497 missing values generated)
. label variable lbound "90% forecast interval"
. tsline duration duration_f if tm>=tm(2006m1) ||
> tsline lbound ubound if tm>=tm(2008m12),
> ysize(2) xtitle("") legend(cols(1))
```



The model forecasts a large temporary increase in the median duration of unemployment.

◀

Methods and formulas

For details on the ucm postestimation methods, see [TS] [sspace postestimation](#).

See [TS] [psdensity](#) for the methods used to estimate the spectral density.

Also see

[TS] [ucm](#) — Unobserved-components model

[TS] [psdensity](#) — Parametric spectral density estimation after arima, arfima, and ucm

[TS] [sspace postestimation](#) — Postestimation tools for sspace

[U] 20 Estimation and postestimation commands

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 © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).