Title

stata.com

forecast — Econometric model forecasting

Description Quick start Syntax Remarks and examples References
Also see

Description

`forecast` is a suite of commands for obtaining forecasts by solving models, collections of equations that jointly determine the outcomes of one or more variables. Equations can be stochastic relationships fit using estimation commands such as `regress`, `ivregress`, `var`, or `reg3`; or they can be nonstochastic relationships, called identities, that express one variable as a deterministic function of other variables. Forecasting models may also include exogenous variables whose values are already known or determined by factors outside the purview of the system being examined. The `forecast` commands can also be used to obtain dynamic forecasts in single-equation models.

The `forecast` suite lets you incorporate outside information into your forecasts through the use of add factors and similar devices, and you can specify the future path for some model variables and obtain forecasts for other variables conditional on that path. Each set of forecast variables has its own name prefix or suffix, so you can compare forecasts based on alternative scenarios. Confidence intervals for forecasts can be obtained via stochastic simulation and can incorporate both parameter uncertainty and additive error terms.

`forecast` works with both time-series and panel datasets. Time-series datasets may not contain any gaps, and panel datasets must be strongly balanced.

This manual entry provides an overview of forecasting models and several examples showing how the `forecast` commands are used together. See the individual subcommands' manual entries for detailed discussions of the various options available and specific remarks about those subcommands.

Quick start

Estimate a linear and an ARIMA regression, and store their results as `myreg` and `myarima`, respectively

```
regress y1 x1 x2
estimates store myreg
arima y2 x3 y1, ar(1) ma(1)
estimates store myarima
```

Create a forecast model with the name `mymodel`

```
forecast create mymodel
```

Add stored estimates `myreg` and `myarima` to forecast model `mymodel`

```
forecast estimates myreg
forecast estimates myarima
```

Compute dynamic forecasts from 2012 through 2020 of `y1` and `y2` using `mymodel` with nonmissing values of `x1`, `x2`, and `x3` for the entire forecast horizon

```
forecast solve, begin(2012) end(2020)
```

See [TS] **forecast adjust**, [TS] **forecast coefvector**, [TS] **forecast create**, [TS] **forecast describe**, [TS] **forecast drop**, [TS] **forecast estimates**, [TS] **forecast exogenous**, [TS] **forecast identity**, [TS] **forecast list**, and [TS] **forecast solve** for additional *Quick starts*.

Syntax

```
forecast subcommand ... [ , options ]
```

subcommand	Description
`create`	create a new model
`estimates`	add estimation result to current model
`identity`	specify an identity (nonstochastic equation)
`coefvector`	specify an equation via a coefficient vector
`exogenous`	declare exogenous variables
`solve`	obtain one-step-ahead or dynamic forecasts
`adjust`	adjust a variable by add factoring, replacing, etc.
`describe`	describe a model
`list`	list all `forecast` commands composing current model
`clear`	clear current model from memory
`drop`	drop forecast variables
`query`	check whether a forecast model has been started

Remarks and examples

stata.com

A forecasting model is a system of equations that jointly determine the outcomes of one or more endogenous variables, whereby the term *endogenous* variables contrasts with *exogenous* variables, whose values are not determined by the interplay of the system's equations. A model, in the context of the `forecast` commands, consists of

1. zero or more stochastic equations fit using Stata estimation commands and added to the current model using `forecast estimates`. These stochastic equations describe the behavior of endogenous variables.
2. zero or more nonstochastic equations (identities) defined using `forecast identity`. These equations often describe the behavior of endogenous variables that are based on accounting identities or adding-up conditions.
3. zero or more equations stored as coefficient vectors and added to the current model using `forecast coefvector`. Typically, you will fit your equations in Stata and use `forecast estimates` to add them to the model. `forecast coefvector` is used to add equations obtained elsewhere.
4. zero or more exogenous variables declared using `forecast exogenous`.
5. at least one stochastic equation or identity.
6. optional adjustments to be made to the variables of the model declared using `forecast adjust`. One use of adjustments is to produce forecasts under alternative scenarios.

The `forecast` commands are designed to be easy to use, so without further ado, we dive headfirst into an example.

▷ Example 1: Klein's model

Example 3 of [R] **reg3** shows how to fit Klein's (1950) model of the U.S. economy using the three-stage least-squares estimator (3SLS). Here we focus on how to make forecasts from that model once the parameters have been estimated. In Klein's model, there are seven equations that describe the seven endogenous variables. Three of those equations are stochastic relationships, while the rest are identities:

$$\mathtt{c}_t = \beta_0 + \beta_1 \mathtt{p}_t + \beta_2 \mathtt{p}_{t-1} + \beta_3 \mathtt{w}_t + \epsilon_{1t} \tag{1}$$
$$\mathtt{i}_t = \beta_4 + \beta_5 \mathtt{p}_t + \beta_6 \mathtt{p}_{t-1} + \beta_7 \mathtt{k}_{t-1} + \epsilon_{2t} \tag{2}$$
$$\mathtt{wp}_t = \beta_8 + \beta_9 \mathtt{y}_t + \beta_{10} \mathtt{y}_{t-1} + \beta_{11} \mathtt{yr}_t + \epsilon_{3t} \tag{3}$$
$$\mathtt{y}_t = \mathtt{c}_t + \mathtt{i}_t + \mathtt{g}_t \tag{4}$$
$$\mathtt{p}_t = \mathtt{y}_t - \mathtt{t}_t - \mathtt{wp}_t \tag{5}$$
$$\mathtt{k}_t = \mathtt{k}_{t-1} + \mathtt{i}_t \tag{6}$$
$$\mathtt{w}_t = \mathtt{wg}_t + \mathtt{wp}_t \tag{7}$$

The variables in the model are defined as follows:

Name	Description	Type
c	Consumption	endogenous
p	Private-sector profits	endogenous
wp	Private-sector wages	endogenous
wg	Government-sector wages	exogenous
w	Total wages	endogenous
i	Investment	endogenous
k	Capital stock	endogenous
y	National income	endogenous
g	Government spending	exogenous
t	Indirect bus. taxes + net exports	exogenous
yr	Time trend = Year − 1931	exogenous

Our model has four exogenous variables: government-sector wages (wg), government spending (g), a time-trend variable (yr), and, for simplicity, a variable that lumps indirect business taxes and net exports together (t). To make out-of-sample forecasts, we must populate those variables over the entire forecast horizon before solving our model. (We use the phrases "solve our model" and "obtain forecasts from our model" interchangeably.)

We will illustrate the entire process of fitting and forecasting our model, though our focus will be on the latter task. See [R] **reg3** for a more in-depth look at fitting models like this one. Before we solve our model, we first estimate the parameters of the stochastic equations by loading the dataset and calling reg3:

```
. use https://www.stata-press.com/data/r17/klein2
. reg3 (c p L.p w) (i p L.p L.k) (wp y L.y yr), endog(w p y) exog(t wg g)
Three-stage least-squares regression
```

Equation	Obs	Params	RMSE	"R-squared"	chi2	P>chi2
c	21	3	.9443305	0.9801	864.59	0.0000
i	21	3	1.446736	0.8258	162.98	0.0000
wp	21	3	.7211282	0.9863	1594.75	0.0000

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
c						
p						
--.	.1248904	.1081291	1.16	0.248	-.0870387	.3368194
L1.	.1631439	.1004382	1.62	0.104	-.0337113	.3599992
w	.790081	.0379379	20.83	0.000	.715724	.8644379
_cons	16.44079	1.304549	12.60	0.000	13.88392	18.99766
i						
p						
--.	-.0130791	.1618962	-0.08	0.936	-.3303898	.3042316
L1.	.7557238	.1529331	4.94	0.000	.4559805	1.055467
k						
L1.	-.1948482	.0325307	-5.99	0.000	-.2586072	-.1310893
_cons	28.17785	6.793768	4.15	0.000	14.86231	41.49339
wp						
y						
--.	.4004919	.0318134	12.59	0.000	.3381388	.462845
L1.	.181291	.0341588	5.31	0.000	.1143411	.2482409
yr	.149674	.0279352	5.36	0.000	.094922	.2044261
_cons	1.797216	1.115854	1.61	0.107	-.3898181	3.984251

```
Endogenous variables: c i wp w p y
 Exogenous variables: L.p L.k L.y yr t wg g
```

The output from `reg3` indicates that we have a total of six endogenous variables even though our model in fact has seven. The discrepancy stems from (6) of our model. The capital stock variable (`k`) is a function of the endogenous investment variable and is therefore itself endogenous. However, k_t does not appear in any of our model's stochastic equations, so we did not declare it in the `endog()` option of `reg3`; from a purely estimation perspective, the contemporaneous value of the capital stock variable is irrelevant, though it does play a role in terms of solving our model. We next store the estimation results using `estimates store`:

```
. estimates store klein
```

Now we are ready to define our model using the `forecast` commands. We first tell Stata to initialize a new model; we will call our model `kleinmodel`:

```
. forecast create kleinmodel
  Forecast model kleinmodel started.
```

The name you give the model mainly controls how output from forecast commands is labeled. More importantly, forecast create creates the internal data structures Stata uses to keep track of your model.

The next step is to add all the equations to the model. To add the three stochastic equations we fit using reg3, we use forecast estimates:

```
. forecast estimates klein
  Added estimation results from reg3.
  Forecast model kleinmodel now contains 3 endogenous variables.
```

That command tells Stata to find the estimates stored as klein and add them to our model. forecast estimates uses those estimation results to determine that there are three endogenous variables (c, i, and wp), and it will save the estimated parameters and other information that forecast solve will later need to obtain predictions for those variables. forecast estimates confirmed our request by reporting that the estimation results added were from reg3.

forecast estimates reports that our forecast model has three endogenous variables because our reg3 command included three left-hand-side variables. The fact that we specified three additional endogenous variables in the endog() option of reg3 so that reg3 reports a total of six endogenous variables is irrelevant to forecast. All that matters is the number of left-hand-side variables in the model.

We also need to specify the four identities, equations (4) through (7), that determine the other four endogenous variables in our model. To do that, we use forecast identity:

```
. forecast identity y = c + i + g
  Forecast model kleinmodel now contains 4 endogenous variables.
. forecast identity p = y - t - wp
  Forecast model kleinmodel now contains 5 endogenous variables.
. forecast identity k = L.k + i
  Forecast model kleinmodel now contains 6 endogenous variables.
. forecast identity w = wg + wp
  Forecast model kleinmodel now contains 7 endogenous variables.
```

You specify identities similarly to how you use the generate command, except that the left-hand-side variable is an endogenous variable in your model rather than a new variable you want to create in your dataset. Time-series operators often come in handy when specifying identities; here we expressed capital, a stock variable, as its previous value plus current-period investment, a flow variable. An identity defines an endogenous variable, so each time we use forecast identity, the number of endogenous variables in our forecast model increases by one.

Finally, we will tell Stata about the four exogenous variables. We do that with the forecast exogenous command:

```
. forecast exogenous wg
  Forecast model kleinmodel now contains 1 declared exogenous variable.
. forecast exogenous g
  Forecast model kleinmodel now contains 2 declared exogenous variables.
. forecast exogenous t
  Forecast model kleinmodel now contains 3 declared exogenous variables.
. forecast exogenous yr
  Forecast model kleinmodel now contains 4 declared exogenous variables.
```

`forecast` keeps track of the exogenous variables that you declare using the `forecast exogenous` command and reports the number currently in the model. When you later use `forecast solve`, `forecast` verifies that these variables contain nonmissing data over the forecast horizon. In fact, we could have instead typed

```
. forecast exogenous wg g t yr
```

but to avoid confusing ourselves, we prefer to issue one command for each variable in our model.

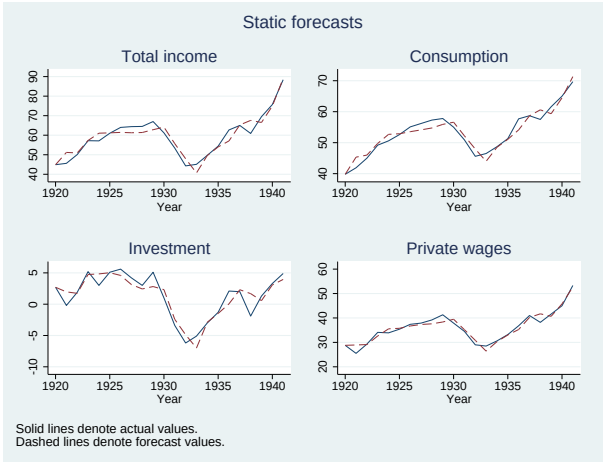
Now Stata knows everything it needs to know about the structure of our model. `klein2.dta` in memory contains annual observations from 1920 to 1941. Before we make out-of-sample forecasts, we should first see how well our model works by comparing its forecasts with actual data. There are a couple of ways to do that. The first is to produce static forecasts. In static forecasts, actual values of all lagged variables that appear in the model are used. Because actual values will be missing beyond the last historical time period in the dataset, static forecasts can only forecast one period into the future (assuming only first lags appear in the model); for that reason, they are often called *one-step-ahead* forecasts. To obtain these one-step-ahead forecasts, we type

```
. forecast solve, prefix(s_) begin(1921) static
Computing static forecasts for model kleinmodel.
------------------------------------------------------------
Starting period:  1921
Ending period:    1941
Forecast prefix:  s_
1921:  ........................................
1922:  ........................................
1923:  ........................................
  (output omitted)
1940:  ........................................
1941:  ........................................
Forecast 7 variables spanning 21 periods.
------------------------------------------------------------
```

We specified `begin(1921)` to request that the first year for which forecasts are produced be 1921. Our model includes variables that are lagged one period; because our data start in 1920, 1921 is the first year in which we can evaluate all the equations of the model. If we did not specify the `begin(1921)` option, `forecast solve` would have started forecasting in 1941. By default, `forecast solve` looks for the earliest time period in which any of the endogenous variables contains a missing value and begins forecasting in that period. In `klein2.dta`, `k` is missing in 1941.

The header of the output confirms that we requested static forecasts for our model, and it indicates that it will produce forecasts from 1921 through 1941, the last year in our dataset. By default, `forecast solve` produces a status report in which the time period being forecast is displayed along with a dot for each iteration the equation solver performs. The footer of the output confirms that we forecast seven endogenous variables for 21 years.

The command we just typed will create seven new variables in our dataset, one for each endogenous variable, containing the static forecasts. Because we specified `prefix(s_)`, the seven new variables will be named `s_c`, `s_i`, `s_wp`, `s_y`, `s_p`, `s_k`, and `s_w`. Here we graph a subset of the variables and their forecasts:



Our static forecasts appear to fit the data relatively well. Had they not fit well, we would have to go back and reexamine the specification of our model. If the static forecasts are poor, then the dynamic forecasts that use previous periods' forecast values are unlikely to work well either. On the other hand, even if the model produces good static forecasts, it may not produce accurate dynamic forecasts more than one or two periods into the future.

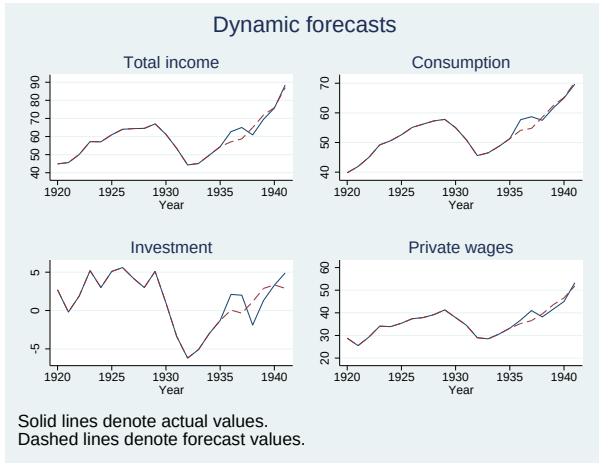
Another way to check how well a model forecasts is to produce dynamic forecasts for time periods in which observed values are available. Here we begin dynamic forecasts in 1936, giving us six years' data with which to compare actual and forecast values and then graph our results:

```
. forecast solve, prefix(d_) begin(1936)
Computing dynamic forecasts for model kleinmodel.
------------------------------------------------------------

Starting period:  1936
Ending period:    1941
Forecast prefix:  d_

1936:  ........................................
1937:  ........................................
1938:  ........................................
1939:  ........................................
1940:  ........................................
1941:  ........................................

Forecast 7 variables spanning 6 periods.
------------------------------------------------------------
```



Most of the in-sample forecasts look okay, though our model was unable to predict the outsized increase in investment in 1936 and the sharp drop in 1938.

Our first example was particularly easy because all the endogenous variables appeared in levels. However, oftentimes the endogenous variables are better modeled using mathematical transformations such as logarithms, first differences, or percentage changes; transformations of the endogenous variables may appear as explanatory variables in other equations. The next few examples illustrate these complications.

Example 2: Models with transformed endogenous variables

`hardware.dta` contains hypothetical quarterly sales data from the Hughes Hardware Company, a huge regional distributor of building products. Hughes Hardware has three main product lines: dimensional lumber (`dim`), sheet goods such as plywood and fiberboard (`sheet`), and miscellaneous hardware, including fasteners and hand tools (`misc`). Based on past experience, we know that dimensional lumber sales are closely tied to the level of new home construction and that other product lines' sales can be modeled in terms of the quantity of lumber sold. We are going to use the following set of equations to model sales of the three product lines:

$$
\begin{aligned}
\%\Delta\texttt{dim}_t &= \beta_{10} + \beta_{11}\ln(\texttt{starts}_t) + \beta_{12}\%\Delta\texttt{gdp}_t + \beta_{13}\texttt{unrate}_t + \epsilon_{1t} \\
\texttt{sheet}_t &= \beta_{20} + \beta_{21}\texttt{dim}_t + \beta_{22}\%\Delta\texttt{gdp}_t + \beta_{23}\texttt{unrate}_t + \epsilon_{2t} \\
\texttt{misc}_t &= \beta_{30} + \beta_{31}\texttt{dim}_t + \beta_{32}\%\Delta\texttt{gdp}_t + \beta_{33}\texttt{unrate}_t + \epsilon_{3t}
\end{aligned}
$$

Here $\texttt{starts}_t$ represents the number of new homes for which construction began in quarter t, $\texttt{gdp}_t$ denotes real (inflation-adjusted) gross domestic product (GDP), and $\texttt{unrate}_t$ represents the quarterly average unemployment rate. Our equation for $\texttt{dim}_t$ is written in terms of percentage changes from quarter to quarter rather than in levels, and the percentage change in GDP appears as a regressor in each equation rather than the level of GDP itself. In our model, these three macroeconomic factors are exogenous, and here we will reserve the last few years' data to make forecasts; in practice, we would need to make our own forecasts of these macroeconomic variables or else purchase a forecast.

We will approximate the percentage change variables by taking first differences of the natural logarithms of the respective underlying variables. In terms of estimation, this does not present any challenges. Here we load the dataset into memory, create the necessary log-transformed variables,

and fit the three equations using regress with the data through the end of 2009. We use quietly to suppress the output from regress to save space, and we store each set of estimation results as we go. In Stata, we type

```
. use https://www.stata-press.com/data/r17/hardware, clear
(Hughes Hardware sales data)
. generate lndim = ln(dim)
. generate lngdp = ln(gdp)
. generate lnstarts = ln(starts)
. quietly regress D.lndim lnstarts D.lngdp unrate if qdate <= tq(2009q4)
. estimates store dim
. quietly regress sheet dim D.lngdp unrate if qdate <= tq(2009q4)
. estimates store sheet
. quietly regress misc dim D.lngdp unrate if qdate <= tq(2009q4)
. estimates store misc
```

The equations for sheet goods and miscellaneous items do not present any challenges for forecast, so we proceed by creating a new forecast model named salesfcast and adding those two equations:

```
. forecast create salesfcast, replace
  (Forecast model kleinmodel ended.)
  Forecast model salesfcast started.
. forecast estimates sheet
  Added estimation results from regress.
  Forecast model salesfcast now contains 1 endogenous variable.
. forecast estimates misc
  Added estimation results from regress.
  Forecast model salesfcast now contains 2 endogenous variables.
```

The equation for dimensional lumber requires more finesse. First, because our dependent variable contains a time-series operator, we must use the names() option of forecast estimates to specify a valid name for the endogenous variable being added:

```
. forecast estimates dim, names(dlndim)
  Added estimation results from regress.
  Forecast model salesfcast now contains 3 endogenous variables.
```

We have entered the endogenous variable dlndim into our model, but it represents the left-hand-side variable of the regression equation we just added. That is, dlndim is the first difference of the logarithm of dim, the sales variable we ultimately want to forecast. We can specify an identity to reverse the first-differencing, providing us with a variable containing the logarithm of dim:

```
. forecast identity lndim = L.lndim + dlndim
  Forecast model salesfcast now contains 4 endogenous variables.
```

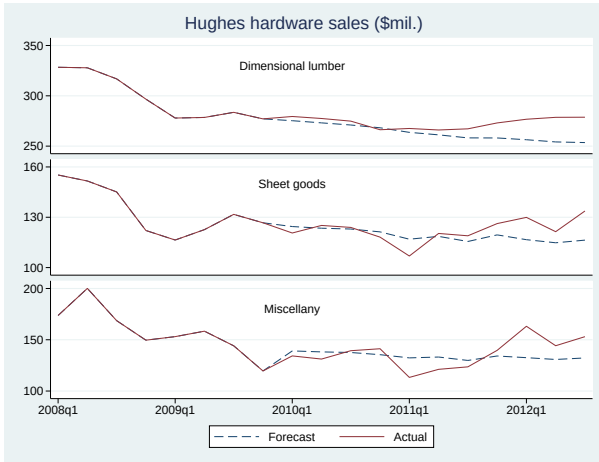
Finally, we can specify another identity to obtain dim from lndim:

```
. forecast identity dim = exp(lndim)
  Forecast model salesfcast now contains 5 endogenous variables.
```

Now we can solve the model. We will obtain dynamic forecasts starting in the first quarter of 2010, and we will use the `log(off)` option to suppress the iteration log:

```
. forecast solve, begin(tq(2010q1)) log(off)
Computing dynamic forecasts for model salesfcast.
------------------------------------------------------------
Starting period:  2010q1
Ending period:    2012q3
Forecast prefix:  f_
Forecast 5 variables spanning 11 periods.
------------------------------------------------------------
```

We did not specify the `prefix()` or `suffix()` option, so by default, `forecast` prefixed our forecast variables with `f_`. The following graph illustrates our forecasts:



Our model performed well in 2010, but it did not forecast the pickup in sales that occurred in 2011 and 2012.

◁

❑ Technical note

For more information about working with log-transformed variables, see the second technical note in [TS] **forecast estimates**.

❑

The `forecast` commands can also be used to make forecasts for strongly balanced panel datasets. A panel dataset is strongly balanced when all the panels have the same number of observations, and the observations for different panels were all made at the same times. Our next example illustrates how to produce a forecast with panel data and highlights a couple of key assumptions one must make.

▷ Example 3: Forecasting a panel dataset

In the previous example, we mentioned that Hughes Hardware was a regional distributor of building products. In fact, Hughes Hardware operates in five states across the southern United States: Texas, Oklahoma, Louisiana, Arkansas, and Mississippi. The company is in the process of deciding whether it should open additional distribution centers or move existing ones to new locations. As part of the process, we need to make sales forecasts for each of the states the company serves.

To make our state-level forecasts, we will use essentially the same model that we did for the company-wide forecast, though we will also include state-specific effects. The model we will use is

$$\begin{aligned}
\%\Delta\texttt{dim}_{it} &= \beta_{10} + \beta_{11}\ln(\texttt{starts}_{it}) + \beta_{12}\texttt{rgspgrowth}_{it} + \beta_{13}\texttt{unrate}_{it} + u_{1i} + \epsilon_{1it} \\
\texttt{sheet}_{it} &= \beta_{20} + \beta_{21}\texttt{dim}_{it} + \beta_{22}\texttt{rgspgrowth}_{it} + \beta_{23}\texttt{unrate}_{it} + u_{2i} + \epsilon_{2it} \\
\texttt{misc}_{it} &= \beta_{30} + \beta_{31}\texttt{dim}_{it} + \beta_{32}\texttt{rgspgrowth}_{it} + \beta_{33}\texttt{unrate}_{it} + u_{3i} + \epsilon_{3it}
\end{aligned}$$

The subscript i indexes states, and we have replaced the `gdp` variable that was in our previous model with `rgspgrowth`, which measures the annual growth rate in real gross state product (GSP), the state-level analogue to national GDP. The GSP data are released only annually, so we have replicated the annual growth rate for all four quarterly observations in a given year. For example, `rgspgrowth` is about 5.3 for the four observations for the state of Texas in the year 2007; in 2007, Texas's real GSP was 5.3% higher than in 2006.

The state-level error terms are u_{1i}, u_{2i}, and u_{3i}. Here we will use the fixed-effects estimator and fit the three equations via `xtreg, fe`, again using data only through the end of 2009 so that we can examine how well our model forecasts. Our first task is to fit the three equations and store the estimation results. At the same time, we will also use `predict` to obtain the predicted fixed-effects terms. You will see why in just a moment. Because the regression results are not our primary concern here, we will use `quietly` to suppress the output.

In Stata, we type

```
. use https://www.stata-press.com/data/r17/statehardware, clear
(Hughes' state-level sales data)
. generate lndim = ln(dim)
. generate lnstarts = ln(starts)
. quietly xtreg D.lndim lnstarts rgspgrowth unrate if qdate <= tq(2009q4), fe
. predict dlndim_u, u
(45 missing values generated)
. estimates store dim
. quietly xtreg sheet dim rgspgrowth unrate if qdate <= tq(2009q4), fe
. predict sheet_u, u
(40 missing values generated)
. estimates store sheet
. quietly xtreg misc dim rgspgrowth unrate if qdate <= tq(2009q4), fe
. predict misc_u, u
(40 missing values generated)
. estimates store misc
```

Having fit the model, we are almost ready to make forecasts. First, though, we need to consider how to handle the state-level error terms. If we simply created a forecast model, added our three estimation results, then called `forecast solve`, Stata would forecast $\texttt{misc}_{it}$, for example, as a function of $\texttt{dim}_{it}$, $\texttt{rgspgrowth}_{it}$, $\texttt{unrate}_{it}$, and the estimate of the constant term β_{30}. However, our model implies that $\texttt{misc}_{it}$ also depends on u_{3i} and the idiosyncratic error term ϵ_{3it}. We will ignore the idiosyncratic error for now (but see the discussion of simulations in [TS] **forecast solve**). By construction, u_{3i} has a mean of zero when averaged across all panels, but in general, u_{3i} is nonzero for any individual panel. Therefore, we should include it in our forecasts.

After you fit a model with `xtreg`, you can predict the panel-specific error component for the subset of observations in the estimation sample. Typically, `xtreg` is used in situations where the number of observations per panel T is modest. In those cases, the estimates of the panel-specific error components are likely to be "noisy" (analogous to estimating a sample mean with just a few observations). Often asymptotic analyses of panel-data estimators assume T is fixed, and in those cases, the estimators of the panel-specific errors are inconsistent.

However, in forecasting applications, the number of observations per panel is usually larger than in most other panel-data applications. With enough observations, we can have more confidence in the estimated panel-specific errors. If we are willing to assume that we have decent estimates of the panel-specific errors and that those panel-level effects will remain constant over the forecast horizon, then we can incorporate them into our forecasts. Because `predict` only provided us with estimates of the panel-level effects for the estimation sample, we need to extend them into the forecast horizon. An easy way to do that is to use `egen` to create a new set of variables:

```
. by state: egen dlndim_u2 = mean(dlndim_u)
. by state: egen sheet_u2 = mean(sheet_u)
. by state: egen misc_u2 = mean(misc_u)
```

We can use `forecast adjust` to incorporate these terms into our forecasts. The following commands define our forecast model, including the estimated panel-specific terms:

```
. forecast create statemodel, replace
  (Forecast model salesfcast ended.)
  Forecast model statemodel started.
. forecast estimates dim, name(dlndim)
  Added estimation results from xtreg.
  Forecast model statemodel now contains 1 endogenous variable.
. forecast adjust dlndim = dlndim + dlndim_u2
  Endogenous variable dlndim now has 1 adjustment.
. forecast identity lndim = L.lndim + dlndim
  Forecast model statemodel now contains 2 endogenous variables.
. forecast identity dim = exp(lndim)
  Forecast model statemodel now contains 3 endogenous variables.
. forecast estimates sheet
  Added estimation results from xtreg.
  Forecast model statemodel now contains 4 endogenous variables.
. forecast adjust sheet = sheet + sheet_u2
  Endogenous variable sheet now has 1 adjustment.
. forecast estimates misc
  Added estimation results from xtreg.
  Forecast model statemodel now contains 5 endogenous variables.
. forecast adjust misc = misc + misc_u2
  Endogenous variable misc now has 1 adjustment.
```

We used `forecast adjust` to perform our adjustment to `dlndim` immediately after we added those estimation results so that we would not forget to do so and before we used identities to obtain the actual `dim` variable. However, we could have specified the adjustment at any time. Regardless of when you specify an adjustment, `forecast solve` performs those adjustments immediately after the variable being adjusted is computed.

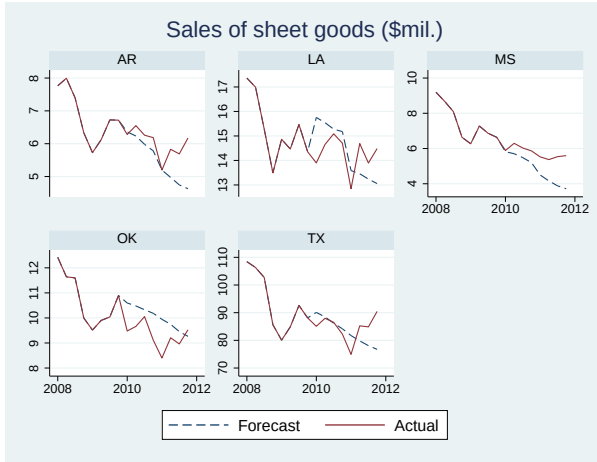
Now we can solve our model. Here we obtain dynamic forecasts beginning in the first quarter of 2010:

```
. forecast solve, begin(tq(2010q1))
Computing dynamic forecasts for model statemodel.
------------------------------------------------------------
Starting period:  2010q1
Ending period:    2011q4
Number of panels: 5
Forecast prefix:  f_

Solving panel 1
Solving panel 2
Solving panel 3
Solving panel 4
Solving panel 5

Forecast 5 variables spanning 8 periods for 5 panels.
------------------------------------------------------------
```

Here is our state-level forecast for sheet goods:



Similar to our company-wide forecast, our state-level forecast failed to call the bottom in sales that occurred in 2011. Because our model missed the shift in sales momentum in every one of the five states, we would be inclined to go back and try respecifying one or more of the equations in our model. On the other hand, if our model forecasted most of the states well but performed poorly in just a few states, then we would first want to investigate whether any events in those states could account for the unexpected results.

◁

❑ Technical note

Stata also provides the `areg` command for fitting a linear regression with a large dummy-variable set and is designed for situations where the number of groups (panels) is fixed, while the number of observations per panel increases with the sample size. When the goal is to create a forecast model for panel data, you should nevertheless use `xtreg` rather than `areg`. The `forecast` commands require knowledge of the panel-data settings declared using `xtset` as well as panel-related estimation information saved by the other panel-data commands in order to produce forecasts with panel datasets.

❑

In the previous example, none of our equations contained lagged dependent variables as regressors. If an equation did contain a lagged dependent variable, then one could use a dynamic panel-data (DPD) estimator such as `xtabond`, `xtdpd`, or `xtdpdsys`. DPD estimators are designed for cases where the number of observations per panel T is small. As shown by Nickell (1981), the bias of the standard fixed- and random-effects estimators in the presence of lagged dependent variables is of order $1/T$ and is thus particularly severe when each panel has relatively few observations. Judson and Owen (1999) perform Monte Carlo experiments to examine the relative performance of different panel-data estimators in the presence of lagged dependent variables when used with panel datasets having dimensions more commonly encountered in macroeconomic applications. Based on their results, while the bias of the standard fixed-effects estimator (LSDV in their notation) is not inconsequential even when $T = 20$, for $T = 30$, the fixed-effects estimator does work as well as most alternatives. The only estimator that appreciably outperformed the standard fixed-effects estimator when $T = 30$ is the least-squares dummy variable corrected estimator (LSDVC in their notation). Bruno (2005) provides a Stata implementation of that estimator. Many datasets used in forecasting situations contain even more observations per panel, so the "Nickell bias" is unlikely to be a major concern.

In this manual entry, we have provided an overview of the `forecast` commands and provided several examples to get you started. The command-specific entries fill in the details.

Video example

Tour of forecasting

References

Baum, C. F., and S. Hurn. 2021. *Environmental Econometrics Using Stata*. College Station, TX: Stata Press.

Box-Steffensmeier, J. M., J. R. Freeman, M. P. Hitt, and J. C. W. Pevehouse. 2014. *Time Series Analysis for the Social Sciences*. New York: Cambridge University Press.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

Judson, R. A., and A. L. Owen. 1999. Estimating dynamic panel data models: a guide for macroeconomists. *Economics Letters* 65: 9–15. https://doi.org/10.1016/S0165-1765(99)00130-5.

Klein, L. R. 1950. *Economic Fluctuations in the United States 1921–1941*. New York: Wiley.

Nickell, S. J. 1981. Biases in dynamic models with fixed effects. *Econometrica* 49: 1417–1426. https://doi.org/10.2307/1911408.

Rossi, B., and M. Soupre. 2017. Implementing tests for forecast evaluation in the presence of instabilities. *Stata Journal* 17: 850–865.

Also see

[TS] **tsset** — Declare data to be time-series data

[TS] **var** — Vector autoregressive models

[R] **ivregress** — Single-equation instrumental-variables regression

[R] **reg3** — Three-stage estimation for systems of simultaneous equations

[R] **regress** — Linear regression

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[XT] **xtset** — Declare data to be panel data