

**forecast adjust** — Adjust variables to produce alternative forecasts

[Description](#)  
[Stored results](#)

[Quick start](#)  
[Reference](#)

[Menu](#)  
[Also see](#)

[Syntax](#)

[Remarks and examples](#)

## Description

`forecast adjust` specifies an adjustment to be applied to an endogenous variable in the model. Adjustments are typically used to produce alternative forecast scenarios or to incorporate outside information into a model. For example, you could use `forecast adjust` with a macroeconomic model to simulate the effect of an oil price shock whereby the price of oil spikes \$50 higher than your model otherwise predicts in a given quarter.

## Quick start

Adjust the endogenous variable `y` in `forecast` to account for the variable `shock` in 1990

```
forecast adjust y = y + shock if year==1990
```

Adjust the endogenous variable `y` in `forecast` to account for a structural change in its mean that occurred in year 2000

```
forecast adjust y = y + 400000 if year > 2000
```

## Menu

Statistics > Time series > Forecasting

## Syntax

```
forecast adjust varname = exp [if] [in]
```

*varname* is the name of an endogenous variable that has been previously added to the model using `forecast estimates` or `forecast coefvector`.

*exp* represents a Stata expression; see [\[U\] 13 Functions and expressions](#).

`collect` is allowed; see [\[U\] 11.1.10 Prefix commands](#).

## Remarks and examples

stata.com

When preparing a forecast, you often want to produce several different scenarios. The baseline scenario is the default forecast that your model produces. It reflects the interplay among the equations and exogenous variables without any outside forces acting on the model. Users of forecasts often want answers to questions like “What happens to the economy if housing prices decline 10% more than your baseline forecast suggests they will?” or “What happens to unemployment and interest rates if tax rates increase?” `forecast adjust` lets you explore such questions by specifying alternative paths for one or more endogenous variables in your model.

## ► Example 1: Revisiting the Klein model

In [example 1](#) of [\[TS\] forecast](#), we produced a baseline forecast for the classic [Klein \(1950\)](#) model. We noted that investment declined quite substantially in 1938. Suppose the government had a plan such as a one-year investment tax credit that it could enact in 1939 to stimulate investment. Based on discussions with accountants, tax experts, and business leaders, say this plan would encourage an additional \$1 billion in investment in 1939. How would this additional investment affect the economy?

To answer this question, we first refit the [Klein \(1950\)](#) model from [\[TS\] forecast](#) using the data through 1938 and then obtain dynamic forecasts starting in 1939. We will prefix these forecast variables with `bl_` to indicate they are the baseline forecasts. In Stata, we type

```
. use https://www.stata-press.com/data/r17/klein2
. quietly reg3 (c p L.p w) (i p L.p L.k) (wp y L.y yr) if year < 1939,
> endog(w p y) exog(t wg g)
. estimates store klein
. forecast create kleinmodel
Forecast model kleinmodel started.
. forecast estimates klein
Added estimation results from reg3.
Forecast model kleinmodel now contains 3 endogenous variables.
. 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.
. 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 solve, prefix(bl_) begin(1939)
Computing dynamic forecasts for model kleinmodel.
```

---

```
Starting period: 1939
Ending period: 1941
Forecast prefix: bl_
1939: .....
.....
1940: .....
.....
1941: .....
.....
Forecast 7 variables spanning 3 periods.
```

---

To model our \$1 billion increase in investment in 1939, we type

```
. forecast adjust i = i + 1 if year == 1939
Endogenous variable i now has 1 adjustment.
```

While computing the forecasts for 1939, whenever forecast evaluates the equation for *i*, it will set *i* to be higher than it would otherwise be by 1. Now we re-solve our model using the prefix *alt\_* to indicate this is an alternative forecast:

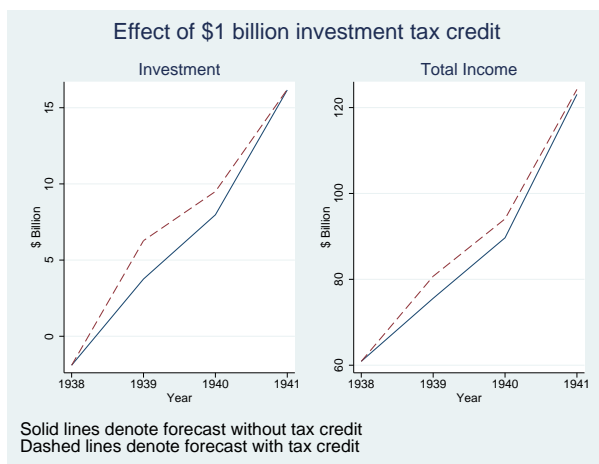
```
. forecast solve, prefix(alt_) begin(1939)
Computing dynamic forecasts for model kleinmodel.

Starting period: 1939
Ending period: 1941
Forecast prefix: alt_

1939: .....
      .....
1940: .....
      .....
1941: .....
      .....

Forecast 7 variables spanning 3 periods.
```

The following graph shows how investment and total income respond to this policy shock.



Both investment and total income would be higher not just in 1939 but also in 1940; the higher capital stock implied by the additional investment raises total output (and hence income) even after the tax credit expires. Let's look at these two variables in more detail:

```
. list year bl_i alt_i bl_y alt_y if year >= 1938, sep(0)
```

	year	bl_i	alt_i	bl_y	alt_y
19.	1938	-1.9	-1.9	60.9	60.9
20.	1939	3.757227	6.276423	75.57685	80.71709
21.	1940	7.971523	9.501909	89.67435	94.08473
22.	1941	16.16375	16.20362	123.0809	124.238

Although we simulated a policy that we thought would encourage \$1 billion in investment, investment in fact rises about \$2.5 billion in 1939 according to our model. That is because higher investment raises total income, which also affects private-sector profits, which beget further changes in investment, and so on.

The investment multiplier in this example might strike you as implausibly large, but it highlights an important attribute of forecasting models. Studying each equation's estimated coefficients in isolation can help to unveil some specification errors, but one must also consider how those equations interact. It is possible to construct models in which each equation appears to be well specified, but the model nevertheless forecasts poorly or suggests unlikely behavior in response to policy shocks.



In the [previous example](#), we applied a single adjustment to a single endogenous variable in a single time period. However, `forecast adjust` allows you to specify `forecast adjust` multiple times with each endogenous variable, and many real-world policy simulations require adjustments to multiple variables. You can also consider policies that affect variables for multiple periods.

For example, suppose we wanted to see what would happen if our investment tax credit lasted two years instead of one. One way would be to use `forecast adjust` twice:

```
. forecast adjust i = i + 1 if year == 1939
. forecast adjust i = i + 1 if year == 1940
```

A second way would be to make that adjustment using one command:

```
. forecast adjust i = i + 1 if year == 1939 | year == 1940
```

To make adjustments lasting more than one or two periods, you should create an adjustment variable, which makes more sense. A third way to simulate our two-year tax credit is

```
. generate i_adj = 0
. replace i_adj = 1 if year == 1939 | year == 1940
. forecast adjust i = i + i_adj
```

So far in our discussion of `forecast adjust`, we have always shown an endogenous variable being adjusted by adding a number or variable to it. However, any valid expression is allowed on the right-hand side of the equals sign. If you want to explore the effects of a policy that will increase investment by 10% in 1939, you could type

```
. forecast adjust i = 1.1*i if year == 1939
```

If you believe investment will be  $-2.0$  in 1939, you could type

```
. forecast adjust i = -2.0 if year == 1939
```

An alternative way to force forecasts of endogenous variables to take on prespecified values is discussed in [example 1](#) of [\[TS\] forecast solve](#).

## Stored results

`forecast adjust` stores the following in `r()`:

Macros

<code>r(lhs)</code>	left-hand-side (endogenous) variable
<code>r(rhs)</code>	right-hand side of identity
<code>r(basenames)</code>	base names of variables found on right-hand side
<code>r(fullnames)</code>	full names of variables found on right-hand side

## Reference

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

## Also see

[TS] [forecast](#) — Econometric model forecasting

[TS] [forecast solve](#) — Obtain static and dynamic forecasts