Description

_estimates hold, _estimates unhold, _estimates dir, _estimates clear, and _estimates drop provide a low-level mechanism for setting aside and later restoring up to 300 estimation results.

_estimates hold moves, or copies if the copy option is specified, all information associated with the last estimation command into *holdname*. If *holdname* is a temporary name, it will automatically be deleted when you exit from the current program.

_estimates unhold restores the information from the estimation command previously moved into *holdname* and eliminates *holdname*.

_estimates dir lists the *holdnames* under which estimation results are currently held.

_estimates clear eliminates all set aside results. Also, if the restore option is specified when the estimates are held, those estimates will be automatically restored when the program concludes. It is not necessary to perform an _estimates unhold in that case.

_estimates drop eliminates the estimation results stored under the specified *holdnames*.

_estimates is a programmer's command designed to be used within programs. estimates is a user's command to manage multiple estimation results. estimates uses _estimates to hold and unhold results, and it adds features such as model-selection indices and looping over results. Postestimation commands, such as suest and lrtest, assume that estimation results are stored using estimates rather than _estimates.

Syntax

Move estimation results into holdname

 _estimates hold *holdname* [, copy restore nullok varname(*newvar*)]

Restore estimation results

 _estimates unhold *holdname* [, not]

List names holding estimation results

 _estimates dir

Eliminate estimation results

 _estimates clear

Eliminate specified estimation results

 _estimates drop { *holdnames* | _all }

where *holdname* is the name under which estimation results will be held.

Options

copy requests that all information associated with the last estimation command be copied into *holdname*.
  By default, it is moved, meaning that the estimation results temporarily disappear. The default action
  is faster and uses less memory.

restore requests that the information in *holdname* be automatically restored when the program ends,
  regardless of whether that occurred because the program exited normally, the user pressed *Break*, or
  there was an error.

nullok specifies that it is valid to store null results. After restoring a null result, no estimation results
  are active.

varname(*newvar*) specifies the variable name under which esample() will be held. If varname() is
  not specified, *holdname* is used. If the variable already exists in the data, an error message is shown.
  This variable is visible to users. If it is dropped, _estimates unhold will not be able to restore the
  estimation sample e(sample) and sets e(sample) to 1.

not specifies that the previous _estimates hold, restore request for automatic restoration be can-
  celed. The previously held estimation results are discarded from memory without restoration, now or
  later.

Remarks and examples

_estimates hold and _estimates unhold are typically used in programs and ado-files, although
they can be used interactively. After fitting, say, a regression by using regress, you can replay the
regression by typing regress without arguments, and you can obtain predicted values with predict, and
the like; see [U] 20 Estimation and postestimation commands. This is because Stata stored information
associated with the regression in what we will call the "last estimation results". The last estimation results
include the coefficient vector and the variance–covariance matrix, as well as the other e() stored results.

When you type _estimates hold myreg, Stata moves the last estimation results to a holding area
named myreg. After issuing this command, you can no longer replay the regression, calculate predicted
values, etc. From Stata's point of view, the estimates are gone. When you type _estimates unhold
myreg, however, Stata moves the estimates back. You can once again type regress without arguments,
calculate predicted values, and everything else just as if the last estimation results were never disturbed.

If you instead type _estimates hold myreg, copy, Stata copies, rather than moves, the results,
meaning that you can still redisplay results. Obviously, you hold estimates because you want to fit some
other model and then get these estimates back, so generally, holding by moving works as well as holding
by copying. Sometimes, however, you may want to hold by copy so that you can modify the estimates
in memory and still retrieve the original.

▷ Example 1

You could run a regression, `hold` the results, run another regression, and then `unhold` the original results. One method you could use is

```
regress y x1 x2 x3              (fit first model)
_estimates hold model1          (and hold on to it)
regress y x1 x2 x3 x4           (fit the second model)
_estimates hold model2          (and hold on to it, too)
use newdata                     (use another dataset)
_estimates unhold model1        (get the first model)
predict yhat1                   (predict using first regression)
_estimates unhold model2        (get the second model)
predict yhat2                   (predict using second regression)
```

You are not limited to doing this with regression; you can do this with any estimation command.

◁

❏ Technical note

Warning: Holding estimation results can tie up considerable amounts of memory, depending on the kind of model and the number of variables in it. This is why there is a limit of 300 held estimation results.

❏

`_estimates dir`, `_estimates drop`, and `_estimates clear` are utilities associated with `_estimates hold` and `_estimates unhold`. `_estimates dir` lists the names of held estimation results. `_estimates drop` drops held estimation results. `_estimates clear` is equivalent to `_estimates drop _all`.

❏ Technical note

Despite our interactive example, `_estimates hold` and `_estimates unhold` are typically used inside programs. For instance, `linktest` fits a model of the dependent variable, the prediction, and the prediction squared and shows the result. Yet when it is over, the user's original model remains as the last estimation result just as if no intervening model had been estimated. `linktest` does this by holding the original model, performing its task, and then restoring the original model.

In addition to moving Stata's last estimation result matrices, e(b) and e(V), `_estimates hold` and `_estimates unhold` also move the other e() results. When you `hold` the current estimates, e(b), e(V), e(cmd), e(depvar), and the other e() results disappear. When you `unhold` them, they are restored.

To avoid naming conflicts, we recommend that estimates be held under a name created by `tempvar` or `tempname`; see [P] **macro**. Thus the code fragment is

```
tempvar est
_estimates hold 'est'
(code including new estimation)
_estimates unhold 'est'
```

❏

Estimates held under a temporary name will automatically be discarded when the program ends. You can also specify `_estimates hold`'s `restore` option when you hold the estimates, and then the held estimates will be restored when the program ends, too.

Stored results

_estimates hold removes the estimation results—e() items.

_estimates unhold restores the previously held e() results.

_estimates clear permanently removes all held e() results.

_estimates dir returns the names of the held estimation results in the local r(names), separated by single spaces.

_estimates dir also returns r(varnames), which has the corresponding variable names for esample().

Also see

[P] **makecns** — Constrained estimation

[P] **mark** — Mark observations for inclusion

[P] **matrix** — Introduction to matrix commands

[P] **matrix rownames** — Name rows and columns

[P] **return** — Return stored results

[R] **estimates** — Save and manipulate estimation results

[R] **ml** — Maximum likelihood estimation

[R] **Stored results** — Stored results

[U] **13.5 Accessing coefficients and standard errors**

[U] **18 Programming Stata**

[U] **20 Estimation and postestimation commands**