Title

stata.com

signestimationsample — Determine whether the estimation sample has changed

Description Syntax Remarks and examples Stored results Also see

Description

`signestimationsample` and `checkestimationsample` are easy-to-use interfaces into `datasignature` for use with estimation commands; see [D] **datasignature**.

`signestimationsample` obtains a data signature for the estimation sample and stores it in `e()`.

`checkestimationsample` obtains a data signature and compares it with that stored by `signestimationsample` and, if they are different, reports "data have changed since estimation"; r(459).

If you just want to know whether any of the data in memory have changed since they were last saved, see [D] **describe**. Examine stored result `r(changed)` after `describe`; it will be `0` if the data have not changed and `1` otherwise.

Syntax

`signestimationsample` *varlist*

`checkestimationsample`

Remarks and examples

stata.com

Remarks are presented under the following headings:

Using signestimationsample and checkestimationsample
Signing
Checking
Handling of weights
Do not sign unnecessarily

Using signestimationsample and checkestimationsample

Estimators often come as a suite of commands: the estimation command itself (say, `myest`) and postestimation commands such as `predict`, `estat`, or even `myest_stats`. The calculations made by the postestimation commands are sometimes appropriate for use with any set of data values—not just the data used for estimation—and sometimes not. For example, predicted values can be calculated with any set of explanatory variables, whereas scores are valid only if calculated using the original data.

Postestimation calculations that are valid only when made using the estimation sample are the exception, but when they arise, `signestimationsample` and `checkestimationsample` provide the solution. The process is as follows:

1. At the time of estimation, sign the estimation sample (store the data's signature in `e()`).
2. At the time of use, obtain the signature of the data in memory and compare it with the original stored previously.

Signing

To sign the estimation sample, include in your estimation command the following line after `e(sample)` is set (that is, after `ereturn post`):

```
. signestimationsample 'varlist'
```

`'varlist'` should contain all variables used in estimation, string and numeric, used directly or indirectly, so you may in fact code

```
. signestimationsample 'lhsvar' 'rhsvars' 'clustervar'
```

or something similar. If you are implementing a time-series estimator, do not forget to include the time variable:

```
. quietly tsset
. signestimationsample 'r(timevar)' 'lhsvar' 'rhsvars' 'othervars'
```

The time variable may be among the `'rhsvars'`, but it does not matter if time is specified twice.

If you are implementing an `xt` estimator, do not forget to include the panel variable and the optional time variable:

```
. quietly xtset
. signestimationsample 'r(panelvar)' 'r(timevar)' 'lhsvar' 'rhsvars' 'clustervar'
```

In any case, specify all relevant variables and don't worry about duplicates. `signestimationsample` produces no output, but behind the scenes, it adds two new results to `e()`:

- `e(datasignature)`—the signature formed by the variables specified in the observations for which `e(sample)` $= 1$
- `e(datasignaturevars)`—the names of the variables used in forming the signature

Checking

Now that the signature is stored, include the following line in the appropriate place in your postestimation command:

```
. checkestimationsample
```

`checkestimationsample` will compare `e(datasignature)` with a newly obtained signature based on `e(datasignaturevars)` and `e(sample)`. If the data have not changed, the results will match, and `checkestimationsample` will silently return. Otherwise, it will issue the error message "data have changed since estimation"; r(459).

Handling of weights

When you code

```
. signestimationsample 'lhsvar' 'rhsvars' 'clustervar'
```

and

```
. checkestimationsample
```

weights are handled automatically.

That is, when you `signestimationsample`, the command looks for `e(wexp)` and automatically includes any weighting variables in the calculation of the checksum. `checkestimationsample` does the same thing.

Do not sign unnecessarily

signestimationsample and checkestimationsample are excellent solutions for restricting postestimation calculations to the estimation sample. However, most statistics do not need to be so restricted. If none of your postestimation commands need to checkestimationsample, do not bother to signestimationsample.

Calculation of the checksum requires time. It's not much, but neither is it zero. On a 2.8-GHz computer, calculating the checksum over 100 variables and 50,000 observations requires about a quarter of a second.

Stored results

signestimationsample stores the following in e():

Macros

e(datasignaturevars)	variables used in calculation of checksum
e(datasignature)	the checksum

The format of the stored signature is that produced by datasignature, fast nonames; see [D] **datasignature**.

Also see

[D] **datasignature** — Determine whether data have changed

[D] **describe** — Describe data in memory or in file