mi estimate postestimation — Postestimation tools for mi estimate

Postestimation commands Remarks and examples Also see

Postestimation commands

The following postestimation commands are of special interest after `mi estimate` and `mi estimate using`:

Command	Description
`mi test`	perform tests on coefficients
`mi testtransform`	perform tests on transformed coefficients
`mi predict`	obtain linear predictions
`mi predictnl`	obtain nonlinear predictions

See [MI] **mi test** and [MI] **mi predict**.

The following standard postestimation command is also available:

Command	Description
`etable`	table of estimation results

Remarks and examples

After estimation by `mi estimate:` *estimation_command*, in general, you may not use the standard postestimation commands such as `test`, `testnl`, or `predict`; nor may you use *estimation_command*-specific postestimation commands such as `estat`. As we have mentioned often, `mi estimate` is its own estimation command, and the postestimation commands available after `mi estimate` (and `mi estimate using`) are listed in the table above.

Using the command-specific postestimation tools

After `mi estimate:` *estimation_command*, you may not use *estimation_command*'s postestimation features. More correctly, you may not use them unless you specify `mi estimate`'s `post` option:

```
. mi estimate, post: estimation_command ...
```

Specifying `post` causes many statistical issues, so do not be casual about specifying it.

First, the MI estimate of the VCE is poor unless the number of imputations, M, is sufficiently large. How large is uncertain, but you should not be thinking $M = 20$ rather than $M = 5$; you should be thinking of M in the hundreds. What is statistically true is that, asymptotically in M (and in the number of observations, N), the MI estimated coefficients approach normality and the VCE becomes well estimated.

Second, there are substantive issues about what is meant by *estimation_command*'s prediction after MI estimation that you are going to have to resolve for yourself. There is no one estimation sample. There are M of them, and as we have just argued, M is large. Do not expect postestimation commands that depend on predicted values such as `margins`, `lroc`, and the like, to produce correct results, if they produce results at all.

Which brings us to the third point. Even when you specify `mi estimate`'s `post` option, `mi estimate` still does not post everything the estimation command expects to see. It does not post likelihood values, for instance, because there is no counterpart after MI estimation. Thus, you should be prepared to see unexpected and inelegant error messages if you use a postestimation command that depends on an unestimated and unposted result.

All of which is to say that if you specify the `post` option, you have a responsibility beyond the usual to ensure the validity of any statistical results.

Also see

[MI] **mi predict** — Obtain multiple-imputation predictions

[MI] **mi test** — Test hypotheses after mi estimate

[MI] **mi estimate** — Estimation using multiple imputations

[MI] **mi estimate using** — Estimation using previously saved estimation results

[MI] **Intro** — Introduction to mi

[MI] **Intro substantive** — Introduction to multiple-imputation analysis

[MI] **Glossary**

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.