Title

stata.com

mi XXXset — Declare mi data to be svy, st, ts, xt, etc.

Description Syntax Remarks and examples Also see

Description

Using some features of Stata requires setting your data. The commands listed below allow you to do that with `mi` data. The `mi` variants have the same syntax and work the same way as the original commands.

Syntax

`mi fvset ...`	see [R] **fvset**
`mi svyset ...`	see [SVY] **svyset**
`mi stset ...`	see [ST] **stset**
`mi streset ...`	
`mi st ...`	
`mi tsset ...`	see [TS] **tsset**
`mi xtset ...`	see [XT] **xtset**

Remarks and examples

stata.com

If you have set your data with any of the above commands before you `mi set` them, there is no problem; the settings were automatically imported. Once you `mi set` your data, however, you will discover that Stata's other set commands no longer work. For instance, here is the result of typing `stset` on an `mi set` dataset:

```
. stset ...
no; data are mi set
    Use mi stset to set or query these data; mi stset has the same
    syntax as stset.
    Perhaps you did not type stset.  Some commands call stset to obtain
    information about the settings.  In that case, that command is not appropriate
    for running directly on mi data.  Use mi extract to select the data on which
    you want to run the command, which is probably m=0.
r(119);
```

Also, you might sometimes see an error like the one above when you give a command that depends on the data being set by one of Stata's other set commands. In general, it is odd that you would be running such a command directly on `mi` data because what you will get will depend on the `mi` style of data. Perhaps, however, you are using `mi` wide data, where the structure of the data more or less corresponds to the structure of non-`mi` data, or perhaps you have smartly specified the appropriate `if` statement to account for the `mi` style of data you are using. In any case, the result might be

```
. some_other_command
no; data are mi set
    Use mi XXXset to set or query these data; mi XXXset has the same syntax as
    XXXset.

    Perhaps you did not type stset.  Some commands call stset to obtain
    information about the settings.  In that case, that command is not appropriate
    for running directly on mi data.  Use mi extract to select the data on which
    you want to run the command, which is probably m=0.
r(119);
```

Substitute one of the set commands listed above for XXXset, and then understand what just happened. You correctly used `mi XXXset` to set your data, you thought your data were set, yet when you tried to use a command that depended on the data being XXXset, you received this error.

If this happens to you, the solution is to use `mi extract` (see [MI] **mi extract**) to obtain the data on which you want to run the command—which is probably $m = 0$, so you would type `mi extract 0`—and then run the command.

Also see

[MI] **intro** — Introduction to mi