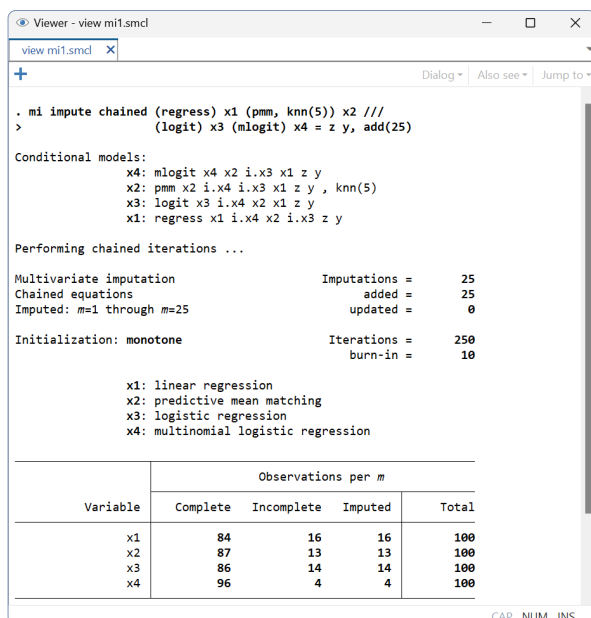# Multiple imputation

- **Support for all three stages of multiple imputation (MI):** Impute missing values, estimate model parameters, and combine estimation results
- **Imputation**
  - Nine univariate methods
  - Multivariate methods: MICE (FCS) and MVN
  - Monotone and arbitrary missing-value patterns
  - Add your own methods
- **Estimation:** Estimate and combine in one easy step
- **Inference:** Linear and nonlinear combinations, hypothesis testing, predictions
- **MI data:** Efficient storage, verification, import, full data management
- **Control Panel to guide you through your MI analysis**
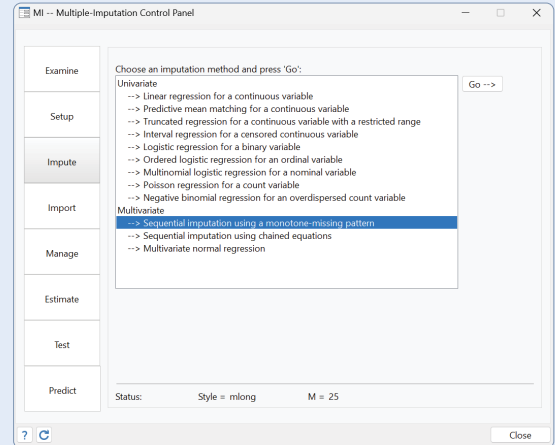
Missing data occur frequently in practice. MI is one of the most flexible ways of handling missing data. Its three stages are multiply imputing missing values, estimating model parameters from each imputed dataset, and combining multiple estimation results in one final inference. In Stata, you can use the **mi** command to perform these three stages in two simple steps.

Impute missing data

Impute missing values using **mi impute**.

Use predictive mean matching, linear, logistic, Poisson, and other regressions to impute variables of different types. Use multiple imputation using chained equations (MICE), multivariate normal imputation (MVN), and monotone imputation to impute multiple variables. Add your own imputation methods. With MICE, build flexible imputation models—use any of the nine univariate methods, customize prediction equations, include functions of imputed variables, perform conditional imputation, and more.

Already have imputed data? Simply import them to Stata for further MI analysis. For example, to import imputed datasets **imp1**, **imp2**, ..., **imp5** from NHANES, use

```
. mi import nhanes1 mymidata, using(imp{1-5}) id(obs)
```

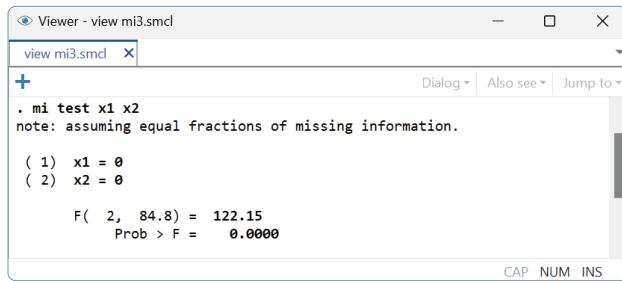## Estimate and combine: One easy step

Estimate model parameters from each imputation, and combine the results in one easy step using **mi estimate**.
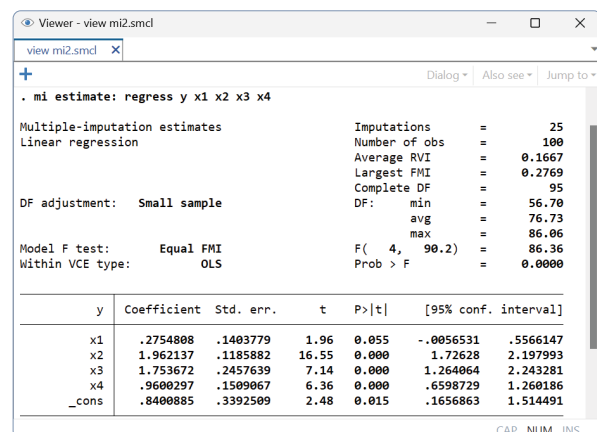
Choose from many supported estimation commands, and simply prefix them with **mi estimate**. Select how many imputations to use during estimation, request a detailed MI summary, and more.

Inference

After estimation, for example, perform hypothesis testing.

```
. mi test x1 x2
note: assuming equal fractions of missing information.

 ( 1)  x1 = 0
 ( 2)  x2 = 0

      F(  2,  84.8) =  122.15
           Prob > F =    0.0000
```
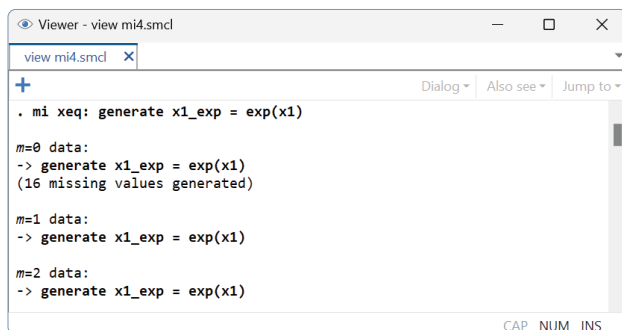
Estimate transformations of coefficients, compute predictions, and more.

In addition...

Stata's **mi** command uniquely provides full data management support, verification of integrity of MI data at any step of the analysis, and multiple formats for storing MI data efficiently. And you can even add your own imputation methods!

Manage imputed data

At any stage of your analysis, perform data management as if you are working with one dataset, and **mi** will replicate the changes correctly across the imputed datasets. Stata offers full data management of MI data: create or drop variables and observations, change values, merge or append files, add imputations, and more.

```
. mi xeq: generate x1_exp = exp(x1)

m=0 data:
-> generate x1_exp = exp(x1)
(16 missing values generated)

m=1 data:
-> generate x1_exp = exp(x1)

m=2 data:
-> generate x1_exp = exp(x1)
```
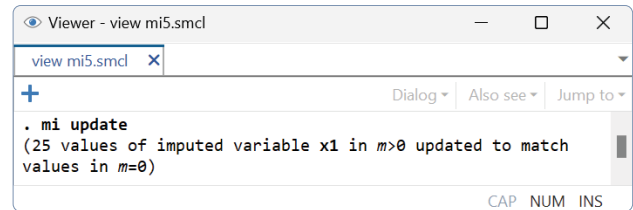
Verify imputed data

Accidentally dropped an observation from one of the imputed datasets, changed a value of a variable, or dropped a variable? Stata verifies the integrity of your MI data each time the **mi** command is run. (You can also do this manually by using **mi update**.) For example, Stata checks that complete variables contain the same values in the imputed data as in the original data, that incomplete variables contain the same nonmissing values in the imputed data as in the original, and more.

If an inconsistency is detected, Stata tries to fix the problem and notifies you about the result.

```
. mi update
(25 values of imputed variable x1 in m>0 updated to match
values in m=0)
```

Multiple storage formats

Stata offers several styles for storing your MI data: you can store imputations in one file or separate files or in one variable or multiple variables. Some styles are more memory efficient, and others are more computationally efficient. Also, some tasks are easier in specific styles.

You can start with one style at the beginning of your MI analysis, for example, "full long", in which imputations are saved as extra observations.

```
. mi set flong
```

If needed, switch to another style during your **mi** session, for example, the wide style, in which imputations are saved as extra variables.

```
. mi convert wide
```

Add your own imputation methods

Can't find an imputation method you need? You can easily program your own. Write a program to impute your variables once, and then simply use it with **mi impute** to obtain multiple imputations.

```
program mi_impute_cmd_mymethod
        ... program imputing missing values once ...
end

. mi impute mymethod ..., add(5) ...
```

Control Panel

Use an intuitive MI Control Panel to guide you through all the stages of your MI analysis—from examining missing values and their patterns to performing MI inference. The corresponding Stata commands are produced with every step for reproducibility and, if desired, later interactive use.