New in

Data frames: Multiple datasets in memory

- Load datasets in memory simultaneously
- Store each dataset in a frame
- Link related frames
- Multitask
- Record results in another frame
- Use frames interactively
- Program with frames in both ado and Mata
- Access data in Stata frames from Java and Python

Creating and modifying frames
Datasets in memory are stored in frames, and each frame is named. When Stata launches, it creates a frame named default.

Create frame named myframe
  . frame create myframe

Drop existing frame named oldframe
  . frame drop oldframe

Rename existing frame oldname to newname
  . frame rename oldname newname

Copy only variables x1, x2, and x3 into a new frame named subset1
  . frame put x1 x2 x3, into(subset1)

Copy only observations where z > 50 into a new frame named subset2
  . frame put if z > 50, into(subset2)

Exploring frames
List all frames in memory, along with the label and dimensions of the data in each frame
  . frames dir

Switching frames
Make myframe the active frame, execute Stata commands on data in myframe, and make default the active frame again
  . frame change myframe
  . stata_command
  . stata_command
  . frame change default
  
Use the frame prefix to run a Stata command on the data in myframe
  . frame myframe: one_stata_command

Run multiple commands on data in myframe
  . frame myframe {
    stata_command
    stata_command
  }
Storing multiple datasets in memory allows you to multitask, work with separate but related datasets simultaneously, record results from one dataset into another, and more.

**Work with separate but related datasets simultaneously**

You have two files, *persons.dta* and *counties.dta*, that are related. The persons live in the counties. You can load the datasets into separate frames and link them.

Open *persons.dta* in the default frame

```
. use persons
```

Create a new *counties* frame and open *counties.dta* in it

```
. frame create counties
. frame counties: use counties
```

Link observations in the active frame (default) to the corresponding observations in the *counties* frame using variable *countyid*

```
. frlink m:1 countyid, frame(counties)
```

Copy variable *med_income* recording each county’s median income from the *counties* frame to the active frame

```
. frget med_income, from(counties)
```

**Use frames to make your work easier**

You have data for cities and countries around the world. You want to analyze the data for Germany efficiently without modifying your current data:

```
. frame put if country==”Germany”, into(subset)
. frame change subset
. stata_commands
. frame change default
. frame drop subset
```

**Record results in another frame**

Create a new frame named *results* with variables *t* and *p*

```
. frame create results t p
```

Perform 1000 simulations, draw 100 random normal variates, perform a *t*-test comparing the mean with 0, post the *t* statistic and *p*-value into the results frame

```
. forvalues i=1(1)1000 {
  2. quietly set obs 100
  3. quietly generate x = rnormal()
  4. quietly ttest x=0
  5. frame post results (r(t)) (r(p))
  6. drop _all
  7. }
```

Count the observation in the *results* frame with a *p*-value less than 0.05

```
. frame results: count p <= 0.05
```

**Use commands or point and click**

![Stata Frame Manager](image1)

![Stata Frame Command](image2)

Stata is a registered trademark of StataCorp LLC, 4905 Lakeway Drive, College Station, TX 77845, USA.