

New in

STATA®

16

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

The screenshot shows the Stata interface. The 'Frames Manager' window is open, displaying a list of frames: census, counties, default, health, insurance, persons, pop5, states, and work. The 'persons' frame is selected as the current frame. Below the list are buttons for 'Create...', 'Switch to', 'Rename...', 'Copy...', 'Put subset...', 'Drop', and 'Reset'. A note at the bottom indicates that frames marked with an asterisk contain unsaved data. In the background, a data viewer window shows a table with columns for 'averted', 'hours', 'enclgrad', 'tenure', and 'wages'. A command window in the foreground shows the following commands and their output:

```
. frame create persons
. frame change persons
. use persons
. frame create counties
. use counties
. frlink m:1 county, frame(counties)
(all observations in frame persons matched)
```

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

The screenshot shows the Stata 16.0 interface. The main window displays a list of states with population and median age. Overlaid windows include 'Rename existing frame' (renaming 'default' to 'census'), 'Copy selected variables or observations to a new frame' (copying 'pop5' to a new frame), and 'Frames Manager' (showing a list of frames like 'counties', 'default', 'persons', etc.).

state	pop	medage
1. California	23,667,902	29.90
2. Florida	9,746,324	34.70
3. Georgia	5,463,105	28.70
4. Illinois	11,426,518	29.90
5. Indiana	5,490,224	29.20
6. Massachusetts	5,737,037	31.20
7. Michigan	9,262,078	28.80
8. New Jersey	7,364,823	32.00
9. New York	17,558,072	31.90
10. N. Carolina	5,881,766	29.60
11. Ohio	10,797,630	29.90
12. Pennsylvania	11,863,895	32.10
13. Texas	24,229,191	28.20
14. Virginia	5,346,818	29.80