frame post — Post results to dataset in another frame

Description Syntax Remarks and examples Also see

Description

These commands are utilities to assist Stata programmers in performing Monte Carlo-type experiments. They are similar to Stata's `postfile` facilities (see [P] **postfile**) but operate on a dataset in a frame in memory rather than on disk.

`frame create` declares the variable names and frame name of a new Stata frame where results will be stored.

`frame post` adds a new observation to the dataset in the declared frame.

After you have posted all the observations you wish to the declared frame, you should `save` the data in it to disk; see [D] **save**.

These commands manipulate the data in the new frame without disturbing the data in memory in the current frame.

Syntax

Create new frame with specified variables for use with `frame post`

`frame create` *newframename newvarlist*

Add new observation to declared frame

`frame post` *framename* `(`*exp*`) (`*exp*`) ... (`*exp*`)`

Remarks and examples

The typical use of the `frame post` command is

```
tempname memhold
...
frame create 'memhold' ...
...
while ... {
        ...
        frame post 'memhold' ...
        ...
}
save ...
...
```

In our example, we obtain the new frame name from Stata's temporary name facility (see [P] **macro**). We recommend that *newframename* usually be obtained from `tempname`. This ensures that your program can be nested within any other program and ensures that the memory used by `frame post` is freed if anything goes wrong. You can of course substitute a hard-coded *newframename* for some programming situations.

Because `frame create` accepts a *newvarlist*, storage types may be interspersed, so you could have

```
frame create 'memhold' a b str20 c double(d e f)
```

Note that `frame create` allows `strL` as a variable storage type, unlike [P] **postfile**.

Example 1

We wish to write a program to collect means and variances from 10,000 randomly constructed 100-observation samples of lognormal data and save the results in `results.dta`. Suppose that we are evaluating the coverage of the 95%, t-based confidence interval when applied to lognormal data. As background, we can obtain a 100-observation lognormal sample by typing

```
drop _all
set obs 100
generate z = exp(rnormal())
```

We can obtain the mean and standard deviation by typing

```
summarize z
```

Moreover, `summarize` stores the sample mean in `r(mean)` and variance in `r(Var)`. It is those two values we wish to collect. Our program is

```
program lnsim
        version 19.5      // (or version 19 if you do not have StataNow)
        tempname sim
        frame create 'sim' mean var
        quietly {
                forvalues i = 1/10000 {
                        drop _all
                        set obs 100
                        generate z = exp(rnormal())
                        summarize z
                        frame post 'sim' (r(mean)) (r(Var))
                }
        }
        frame 'sim': save results.dta
end
```

The `frame create` command creates a new frame with a temporary name (`'sim'`); `mean` and `var` are the names to be given to the two variables that will contain the information we collect. Because two variable names were specified on the frame create line, two expressions must be specified following `frame post`. Here the expressions are simply `r(mean)` and `r(Var)`. If we had wanted, however, to store the mean divided by the standard deviation and the standard deviation, we could have typed

```
frame post 'sim' (r(mean)/r(sd)) (r(sd))
```

There is no need for a command to conclude the simulation. When the dataset in frame 'sim' has everything in it we wish to have in it, we can either switch to frame 'sim' to do what we wish with the data or save it to disk to examine later. Here we saved the new data in frame 'sim' to a file named `results.dta`.

```
. set seed 12345
. lnsim
file results.dta saved
. use results, clear
. describe
Contains data from results.dta
 Observations:        10,000
    Variables:             2                  27 Mar 2025 18:53
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
mean            float   %9.0g
var             float   %9.0g
-------------------------------------------------------------------------------
Sorted by:
```

We set the random-number seed to an arbitrary value, 12345, so that this example would be reproducible.

Also see

[P] **postfile** — Post results in Stata dataset

[D] **frames intro** — Introduction to frames

[D] **frames** — Data frames

[D] **frame create** — Create a new frame

[R] **simulate** — Monte Carlo simulations