Title

stata.com

include — Include commands from file

Syntax Description Remarks and examples Also see

Syntax

`include` *filename*

Description

`include` is a variation on `do` and `run`—see [R] **do**—that causes Stata to execute the commands stored in *filename* just as if they were entered from the keyboard.

`include` differs from `do` and `run` in that any local macros (changed settings, etc.) created by executing the file are not dropped or reset when execution of the file concludes. Rather, results are just as if the commands in *filename* appeared in the session or file that included *filename*.

If *filename* is specified without an extension, `.do` is assumed.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Use with do-files
Use with Mata
Warning

Use with do-files

`include` can be used in advanced programming situations where you have several do-files among which you wish to share common definitions. Say that you have do-files `step1.do`, `step2.do`, and `step3.do` that perform a data management task. You want the do-files to include a common definition of the local macros `‘inname’` and `‘outname’`, which are, respectively, the names of the files to be read and created. One way to do this is

```
------------------------------------------------------------ begin step1.do -----
...
include common.doh
...
-------------------------------------------------------------- end step1.do -----

------------------------------------------------------------ begin step2.do -----
...
include common.doh
...
-------------------------------------------------------------- end step2.do -----
```

begin step3.do

```
...
include common.doh
...
```

end step3.do

begin common.doh

```
local inname  "inputdata.dta"
local outname "outputdata.dta"
```

end common.doh

Presumably, files `step1.do`, `step2.do`, and `step3.do` include lines such as

```
. use ‘inname’, clear
```

and

```
. save ‘outname’, replace
```

Our use of the `.doh` suffix in naming file `common.doh` is not a typo. We called the file `.doh` to emphasize that it is a header for do-files, but you can name the file as you wish, including `common.do`.

You could call the file `common.do`, but you could not use the `do` command to run it because the local macros that the file defines would automatically be dropped when the file finished executing, and thus in `step1.do`, `step2.do`, and `step3.do`, the macros would be undefined.

Use with Mata

`include` is sometimes used in advanced Mata situations where you are creating a library of routines with shared concepts:

begin inpivot.mata

```
version 13
include limits.matah

mata:
real matrix inpivot(real matrix X)
{
        real matrix     y1, yz
        real scalar     n

        if (rows(X)>‘MAXDIM’ | cols(X)>‘MAXDIM’) {
                errprintf("inpivot:  matrix too large\n")
                exit(1000)
        }
        ...
}
end
```

end inpivot.mata

begin limits.matah

```
...
local MAXDIM   800
...
```

end limits.matah

Presumably, many `.mata` files include `limits.matah`.

Warning

Do not use command `include` in the body of a Stata program:

```
program ...
        ...
        include ...
        ...
end
```

The `include` will not be executed, as you might have hoped, when the program is compiled. Instead, the `include` will be stored in your program and executed every time your program is run. The result will be the same as if the lines had been included at compile time, but the execution will be slower.

Also see

[R] **do** — Execute commands from a file

[R] **doedit** — Edit do-files and other text files