The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$ 98	1-year subscription	$138
2-year subscription	$165	2-year subscription	$245
3-year subscription	$225	3-year subscription	$345
1-year student subscription	$ 75	1-year student subscription	$ 99
1-year institutional subscription	$245	1-year institutional subscription	$285
2-year institutional subscription	$445	2-year institutional subscription	$525
3-year institutional subscription	$645	3-year institutional subscription	$765
Electronic only		**Electronic only**	
1-year subscription	$ 75	1-year subscription	$ 75
2-year subscription	$125	2-year subscription	$125
3-year subscription	$165	3-year subscription	$165
1-year student subscription	$ 45	1-year student subscription	$ 45

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2014)
14, Number 2, pp. 449–450

Stata tip 120: Certifying subroutines

Maarten L. Buis
Wissenschaftszentrum Berlin für Sozialforschung (WZB)
Berlin, Germany
maarten.buis@wzb.eu

When you write your own program in Stata, it is good practice and useful to create and run a certification script. A certification script is a do-file that runs your program and compares the results either with a result that is known to be true or with results from a previous run (Gould 2001). It is also good practice and useful to divide your program into smaller subroutines; you can store these subroutines in the same ado-file. These subroutines will be visible only to other programs defined within the same ado-file; the only program that is visible to all other programs in Stata will be the first program defined in an ado-file. This can be useful for subroutines that make sense only within the context of the main program. For example, you may want to delegate the parsing of some complicated syntax element to a subroutine. Moreover, putting a subroutine inside the ado-file of the main program protects users against accidentally running that subroutine. This can be important when, for example, the subroutine changes the data, and the main program has various safeguards in place to ensure that this will not corrupt the user's data.

Sometimes, it is helpful to certify some subroutines in isolation. How would you do that if the subroutines are not visible outside the ado-file? You could copy the subroutine and store it in its own file, thus making the subroutine globally visible. However, as mentioned above, there can be good reasons why you would not want the subroutine to be globally visible in the final program.

Another solution is to `do` or `run` (see [R] **do**) the ado-file. This treats the ado-file as a regular do-file, which in this case defines only a set of programs. So after you `do` or `run` an ado-file, all of its subroutines will also be available. Now you can certify the subroutines from the file that will be released to the general public without having to copy and paste parts out of and into that file. This trick can also be useful when debugging a subroutine.

Consider the example ado-file below:

```
*! version 1.0.0 26Feb2014 MLB
program mainprog
version 13
args input
subprog `input´
di `"`s(output)´"´
end

program subprog, sclass
version 13
args input
sreturn local output `"do something smart with "`input´""´
end
```


If you store this file where Stata can see it (see [P] **sysdir**) or if you change the working directory to where this ado-file is stored (see [D] **cd**), the command `mainprog` works directly. However, if you try to call `subprog`, Stata will return an error.

```
. clear all
. mainprog "this"
do something smart with "this"
. subprog "this"
unrecognized command:  subprog
r(199);
```

We can look at the names of the programs stored in memory by using `program dir` (see [P] **program**), and we see that `subprog` exists but only as part of the `mainprog` command.

```
. program dir
  ado      232  mainprog.subprog
  ado      213  mainprog
  (output omitted)
```

If we `run` this ado-file first, then we can directly access both `mainprog` and `subprog`.

```
. clear all
. run mainprog.ado
. mainprog "this"
do something smart with "this"
. subprog "this"
. di `"`s(output)´"´
do something smart with "this"
. program dir
           232  subprog
           213  mainprog
  (output omitted)
```

Reference

Gould, W. 2001. Statistical software certification. *Stata Journal* 1: 29–50.