Reproducibility and backward compatibility

Want your current analyses to run seamlessly in the future?
Want your old analyses to benefit from recent improvements?
Tired of holding on to old versions of your software?

Learn how Stata's unique integrated versioning can help you with reproducible research.

Integrated version control

Many people talk about reproducible research. Stata has been dedicated to it for over 30 years.

We constantly add new features; we have even fundamentally changed language elements. No matter. Stata is the only statistical package with integrated versioning. If you wrote a script to perform an analysis in 1985, that same script will still run and still produce the same results today. Any dataset you created in 1985, you can read today. And the same will be true in the future. Stata will be able to run anything you do today.

Unlike other software, you do not have to keep multiple installations of old versions of Stata, hoping they will still run on a modern operating system, to be able to run code from years or decades before. You can simply use modern Stata, and it will understand any old code or dataset from the past.

Version control in Stata is seamless. Simply include a **version** statement at the beginning of your script or program, or prefix your command with **version:**, and you will be able to run it, without modification, in any future version of Stata.

For instance, in Stata 13 (released in 2013), to compute confidence intervals (CIs) for means of normally distributed variables **y1** and **y2**, we used to type

```
. ci y1 y2
```

and to compute CIs for proportions of binary variables **z1** and **z2**, we used to type

```
. ci z1 z2, binomial
```

In modern Stata, we would instead type

```
. ci means y1 y2
```

and

```
. ci proportions z1 z2
```

But rest assured that the old syntax still works. All you need to do is to prefix the old commands with the appropriate **version** statement. In our example, we could type

```
. version 13: ci y1 y2
. version 13: ci z1 z2, binomial
```

to run the old commands, as they are, in modern Stata.

You can version-control an entire program or script by simply including the appropriate **version** statement at the beginning:

```
program myci
        version 13
        ci y1 y2
        ci z1 z2, binomial
end
```

No broken scripts. No broken programs. No additional effort.

Stata was designed from its very first version with reproducible research in mind. We want users to be confident that years down the road, the files they used to produce a particular analysis will continue to work even if they change operating systems or computer architecture and move to a much newer version of Stata.

Stochastic reproducibility

If your analyses use stochastic (random) procedures, such as simulations, you can set a random-number seed for reproducibility. You can use **set seed** to set the seed before a stochastic command or a block of code,

```
. set seed 12345
. simulate, reps(1000): mysimprog …
```

or you can specify the **seed()** or **rseed()** option with the command:

```
. simulate, reps(1000) seed(12345): mysimprog …
```

You can even safely run simulations in parallel using random-number streams and reproduce your results:

```
. set rngstream 10
. set seed 123
```

Numerical reproducibility

Stata ensures that deterministic (nonrandom) procedures reproducibly return accurate results when run repeatedly by

- using high precision for internal computations,
- using consistent stopping and convergence criteria,
- accounting for operations that depend on ordering,
- normalizing scale of values to minimize finite precision rounding,
- and more.

Computer reproducibility

Different computers have different chips, libraries, etc., that may lead to slight numerical differences across systems. Using a single core versus multiple cores may affect numerical results too.

Stata is extensively and continually tested for accuracy and reproducibility of results on all supported platforms (Windows, Mac, Linux, etc.) through its extensive certification process. A suite of tests, including over four million lines of test script, produces over five million lines of output. This output is electronically compared with previous runs of the test suite and with runs on other platforms. Any differences are examined and resolved by Stata's statisticians and software engineers before a new release of Stata (or any subsequent free updates) is sent to users.

Visit **stata.com/whystata/#trusted** for details.

Stata takes reproducibility seriously!

stata.com/whystata