

Stata Tip # XXX

Dealing with Convergence Issues and Other Seemingly-Inexplicable Problems

Richard Williams, Notre Dame Sociology, rwilliam@nd.edu,

[Forthcoming in [The Stata Journal](#) sometime in 2026. Used with permission. Last revised Sept 5, 2025]

Introduction

Many Stata runtime errors can be easily fixed. A quick look at the syntax will reveal a missing comma, an unmatched parenthesis, a misspelled option.

Errors such as the following, however, can be far more perplexing. After running a program seemingly forever, Stata finally gives the message

```
convergence not achieved
```

which usually means that even if you did get results, you can't trust them.

Convergence problems can have any number of causes. The model may be fundamentally flawed. The syntax may not specify the model correctly. There could be problems with the data. There may be too many variables in the model for Stata to handle. The sample size could be too big for the given task or too small. The statistical technique employed may not be the optimal one for this topic. Perhaps most frustrating of all, the user may be doing everything right but the software could be buggy.

There can be other seemingly inexplicable errors where Stata provides little or no guidance for addressing the problem. A personal favorite: Stata suddenly starts spewing messages in red that mean little to me (something about overflow problems) and then aborts. I assume Stata has encountered an error that it has failed to trap and give an informative error message for, but that does not do me any good in trying to figure out what to do about it.

It is easy to spend hours trying to solve convergence problems. Before doing so, however, I suggest trying some of the tips (arranged roughly in order of difficulty) given below. They might lead to a simple solution to the problem, or at least provide a template for identifying where the problem is. Some of these tips may also help when other seemingly-inexplicable errors are encountered. Many of these tips may seem obvious, but I've encountered numerous instances where less-experienced users knew nothing about them and even experienced users failed to try them.

Tips

1. **Make sure your software is up to date.** Use the `update all` and the `adoupdate` commands. These will make sure your copy of Stata and any user-written programs you have installed are up to date. If you are really lucky, the problem you are having has already been fixed, and if you are not so lucky you at least have the most up-to-date versions of all your software. If you do not have read/write access to your machine, e.g. you are working in a shared computer lab, you may have to beg some network administrator to update Stata for you.

This tip may seem too obvious, but not everyone follows it or even knows about it. Statalist occasionally has postings reporting a strange problem followed by a comment that the problem was fixed days earlier in the most recent maintenance release (or will be fixed in the next one). Speaking from personal experience, a few years ago a user reported a problem with my `gologit2` program (Williams 2006). She was semi-correct. The version of `gologit2` she was using did indeed have that problem; but I had already fixed it – in 2007.

Also, if you are using an older version of Stata, see if the problems persist in newer versions. I've had students come in with hand-me-down versions of Stata 12. One of the first things I do is see if I have the same problem with my version of Stata. I don't want to spend hours trying to solve a problem that StataCorp took care of ten years ago. Conversely, if a routine worked in the past, it sometimes helps to run it in an older version of Stata because new bugs can inadvertently get introduced.

2. Rescale variables. As Long (1997) notes, variable scaling can be very important. Even the most powerful computers do not offer infinite numeric precision. Long says the larger the ratio between the largest standard deviation and the smallest standard deviation, the more problems you will have with numerical methods. Even more extreme differences in values may be encountered if squared or interaction terms are included in the model. For example, if you have income measured in dollars, it may have very large values and a very large standard deviation relative to other variables. Recoding income to thousands of dollars may solve the problem. For example, you might have code like

```
gen incomek = income/1000
```

It can also sometimes help to subtract a constant from a variable. Variables like year can have very large values. Suppose, for example, year ranges from 1991 to 2010 in the data set. If you code

```
gen yearx = year - 1990
```

the new variable will have values ranging from 1 to 20, which Stata may find much easier to work with.

The choice of what constant to use is up to the researcher. Particularly when 0 is an impossible value (e.g. nobody weighs 0 pounds) centering the variable by subtracting the mean from each case is often a popular choice. Then, a value of 0 on the new variable means the person had the average score on the original variable, while a negative or positive value indicates the respondent had a below average or above average score.

I have seen many instances where rescaling variables instantly solved convergence problems. Further, whether you have convergence problems or not, you may want to rescale some variables for presentation purposes anyway. It may seem bizarre if the effect of income in dollars is reported as .000 and is significant at the .001 level. Income in thousands of dollars may instead yield a much more understandable coefficient such as .412.

Rescaling is NOT always a good idea though. For example, with count models, any transformation of the dependent variable (e.g. number of doctor visits) can be a serious mistake and violate the assumptions of the method. The user needs to understand what variable transformations are appropriate given the method used.

3. [Change the maximization options, e.g add the `difficult` or `technique` options to an estimation command](#). Stata has several options that affect how maximum likelihood estimates get computed. These options are rarely needed but they can be helpful when dealing with convergence problems. For example, the `difficult` option causes a different stepping algorithm to be used when the default algorithm is not working well. Adding the `difficult` option usually does nothing and may even make things worse, but it is extremely easy to try. This is often the first thing I do, and occasionally it works miracles. A `gologit2` user reported that `difficult` got one complicated model to finally converge and made another run much faster. Pitblado, Poi, and Gould (2024) explain the technical workings of the `difficult` option and further discuss its advantages and disadvantages.

A reviewer noted, and my own experience confirms, that the `technique` option sometimes helps, especially with more complicated programs like `sem` and `gsem`. `technique` specifies the algorithm by which the likelihood function is to be maximized. The default is `technique(nr)`, where `nr` stands for Stata's modified Newton–Raphson (NR) algorithm. There are other options though, including Berndt–Hall–Hall–Hausman (BHHH), Davidon–Fletcher–Powell (DFP), and Broyden–Fletcher–Goldfarb–Shanno (BFGS).

Some people may be reluctant to use the `technique` option because they have no idea what algorithm to use. I usually don't either, but I don't let that stop me! What I usually do is experiment and see what, if anything, works. Further, as Pitblado et al (2024, p. 74) helpfully point out, “Switching between techniques is often an excellent way of finding the maximum of a difficult-to-maximize function.” For example,

```
technique(nr bhhh bfgs)
```

will cause Stata to first try the Newton-Raphson algorithm, then Berndt–Hall–Hall–Hausman, and finally Broyden–Fletcher–Goldfarb–Shanno.

You can also try other maximization options, like increasing the number of iterations, changing the start values, or adjusting the tolerance levels. I rarely if ever have any luck with these but they could help in some situations. To learn more, type `help maximize` from within Stata or see the maximum likelihood book by Pitblado et al (2024).

4. [Clean the data](#). Long (1997) says that in his experience most problems with numerical methods are due to data that have not been “cleaned.” Does a variable have codes like 0, 1, and 99, where 99 is instead supposed to be a Stata missing data code? Are there extreme outliers that are actually coding errors? Has a variable been computed incorrectly?

This tip may also seem too obvious – it is basically Data Management 101 – but students in statistics classes are used to solving textbook problems where the data are in perfect shape, and even experienced users (especially those working on a research team) may just assume somebody else has adequately prepared the data. Be wary, too, of downloaded data. Even if the data are already in Stata dta format, that doesn't mean the provider has the data in perfect shape, e.g. users may have to fix the missing data coding themselves.

As an instructor, one of the first things I make students do is run descriptive statistics on their variables, e.g. use commands like `tab1`, `tab2`, and `summarize`. Further I tell them not to assume that any data transformations they themselves have done have been done correctly – run more descriptive statistics to make sure things look right. People who are otherwise brilliant are sometimes not so good at checking their work. As an undergraduate I briefly worked with an excellent mathematics professor who was nonetheless shocked at how much “real” data needed to be cleaned first.

5. Increase the number of observations. Long (1997) notes that convergence often occurs more rapidly when there are more observations. There usually isn't much you can do about sample size, but this may explain why you are having problems. You might also collect new data or find alternative data sets that are better suited for your task. Less restrictive sampling criteria may be called for if a subsample you want to analyze is too small. When appropriate, deleting variables with missing data that are causing you to lose cases or using alternative operationalizations of concepts may also help.

6. Check the distribution of outcomes. If one of the categories of a categorical variable has very few cases (e.g. only one or two cases have a value of “Never”), convergence may be difficult. Long (1997) says you can't do much about this. But, I think you can sometimes combine categories with sparse counts, e.g. “Never” and “Rarely,” or sometimes even collapse a variable into a dichotomy

Also, Stata has routines specifically designed to deal with rare events, e.g. a dichotomy where 99% of the cases are coded 0 and only 1% are coded 1. See Leitgöb (2019) and Allison (2012) for discussions.

7. Treat some ordinal independent variables as continuous. Williams (2020) notes that it is possible and sometimes even desirable to treat ordinal independent variables as continuous rather than as categorical. Fewer parameters need to be estimated (possibly helping with convergence) and interpretation can be more straightforward. Williams outlines procedures for testing whether an ordinal independent variable can be validly treated as continuous.

8. Check the command syntax. Don't just look for syntax errors, look for logical errors. Stata does what you tell it to do, which, unfortunately, is not always the same as what you want it to do. Again, otherwise brilliant people are not necessarily so good when it comes to being careful and accurate, or at translating what they want to do into Stata code. Perhaps a wrong variable is being used, a variable is being treated as continuous when it should be treated as categorical, or a subsample has not been chosen correctly. With things like Structural Equation Models it can be very easy to omit or mis-specify some parameters.

One of the above tips will hopefully provide a relatively quick and painless fix. But, if problems persist,

9. Simplify the model and gradually add more variables to it. If even the simplest models will not run, it may be that there is a fundamental problem with the theory, data, or method. Alternatively, by building the model gradually, you may be able to identify variables that are causing you grief and either fix problems with them or consider alternative operationalizations of concepts that can be used instead.

When building a model gradually, be careful the same cases are being analyzed throughout, since adding a new variable that has missing data can cause the sample size to shrink. This can be especially problematic if the new variable was only measured for a subsample of the respondents, e.g. women. The `nestreg` prefix is often useful for keeping the sample the same when sequences of models are estimated, e.g.

```
nestreg: logit y x1 x2 x3
```

will enter `x1`, `x2`, and `x3` one at a time into the model while using the same sample cases throughout.

10. Try alternative programs. Built-in Stata routines like `glm`, `sem`, `gsem`, `meglm`, and user-written routines like `oglm` (Williams, 2010) and `cmp` (Roodman, 2011) can estimate many of the same models that other Stata routines can. It is possible that an alternate routine may work when the program you are currently using does not. In some cases there may be programs that are actually better than what you are using, such as when dealing with rare events.

11. Delete extraneous and non-crucial variables. Some variables may be included in a model “just in case” they happen to be important. Other variables may be theoretically important but because they are so poorly measured or have too much missing data or have such small effects, they contribute little or nothing to the analysis. Sometimes sample sizes are so small that, even if a variable is relevant, there may not be enough statistical power to detect its effects. Often, including insignificant variables is harmless and even desirable (to demonstrate that their effects are small or nonexistent) but especially with small samples extraneous variables can dramatically inflate the standard errors for all variables, including the ones that really are important, and may also make it harder to achieve convergence.

Researchers are understandably reluctant to drop variables but they should not let the perfect be the enemy of the good: estimating a simplified model will generally be better than estimating no model at all. The user can always explain in the text why a variable was not included in the final model but was adequately examined in the earlier analysis.

12. Look for command-specific advice, especially for `sem` and `gsem`. I’ve primarily focused on generic problems here, but a few methods and commands have unique problems not encountered elsewhere. For example, I have seen the Stata `sem` command report convergence problems when the real problem was that the model was mis-specified and not identified. Indeed,

the Stata 19 Structural Equation Modeling Reference Manual (2025) claims (p. 123) that “It can be devilishly difficult for software to obtain results for SEMs” and speculates (p. 3) that “5% to 15% of complicated [SEM] models” will have difficulty converging. The manual urges users to not despair, though, because Intro 12 of the SEM Manual discusses “Convergence problems and how to solve them.”

Finally, if all else fails,

13. Rethink the theory, data, and method. Perhaps the theory isn’t as good as you thought it was. Or, even if it is, the data may not be adequate for testing it. I’ve often had users say that some variable is theoretically critical, but even if it is, it may have to be sacrificed because it isn’t measured well. You probably chose the method you are using because you thought it was the best, but it is often possible to try alternative approaches that also have their merits. For example, Liu et al (2017) argue that it is sometimes better to model continuous response variables using ordinal regression methods. Finally, remember that much of your model may be fine but there may be one piece of it that needs work, e.g. a complicated structural equation model may need tweaking to correct an identification problem.

Conclusions

Convergence problems and other-seemingly inexplicable errors can at first be baffling and the user could easily spend hours trying to figure them out. Before doing so, however, see if there are simple solutions that work. If you update your software you may find that your problem has already been fixed. The `difficult` option rarely helps, but you might be one of those lucky people for whom it works miracles. Similarly, trying different algorithms with `technique` may help to clear a convergence logjam. Cleaning data, rescaling variables, combining variable categories, or sometimes treating ordinal variables as continuous often helps, and are things you may want to do anyway. Indeed, if the data are problematic, it is probably better to get a serious error than to settle for results that seem plausible but are not correct.

Do not be too wedded to your model. Like it or not, some variables may need to be dropped because they are too poorly measured or contribute nothing to the analysis and may even make it difficult to get any estimates at all.

Consider alternative methods and software programs. They may be able to handle tasks your current program is struggling with and may even be more appropriate in some cases.

Finally, be open to the possibility that your theory is wrong, and even if it isn’t you may not have the data needed to estimate it. Consider what you can do instead or what you need to do differently in the future.

Acknowledgements

The author thanks Richard Campbell for his helpful comments and suggestions.

References

- Allison, Paul. 2012. Logistic Regression for Rare Events. <https://statisticalhorizons.com/logistic-regression-for-rare-events/>
- Leitgöb, H. 2020. Analysis of Rare Events. In P. Atkinson, S. Delamont, A. Cernat, J.W. Sakshaug, & R.A. Williams (Eds.), SAGE Research Methods Foundations. <https://doi.org/10.4135/9781526421036863804>
- Liu Q, Shepherd BE, Li C, Harrell Jr FE. 2017. Modeling continuous response variables using ordinal regression. Statistics in Medicine 36: 4316–4335. <https://doi.org/10.1002/sim.7433>
- Long, J.S. 1997. Regression Models for Categorical and Limited Dependent Variables. A volume in the SAGE Series for Advanced Quantitative Techniques. Thousand Oaks, CA: SAGE Publications.
- Pitblado, Jeffrey, Brian Poi, and William Gould. 2024. Maximum Likelihood Estimation with Stata, Fifth Edition. College Station, TX: Stata Press.
- Roodman, D. (2011). Fitting Fully Observed Recursive Mixed-process Models with cmp. The Stata Journal, 11(2), 159-206. <https://doi.org/10.1177/1536867X1101100202>
- StataCorp. 2025. Stata 19 Structural Equation Modeling Reference Manual. College Station, TX: Stata Press.
- Williams, Richard A. (2020). Ordinal Independent Variables. In P. Atkinson, S. Delamont, A. Cernat, J.W. Sakshaug, & R.A. Williams (Eds.), SAGE Research Methods Foundations. <https://doi.org/10.4135/9781526421036938055> . A pre-publication version can be found at <https://academicweb.nd.edu/~rwilliam/stats3/OrdinalIndependent.pdf>
- Williams, Richard. 2010. “Fitting Heterogeneous Choice Models with oglm.” The Stata Journal 10(4):540-567. <http://www.stata-journal.com/article.html?article=st0208>.
- Williams, Richard. 2006. “Generalized Ordered Logit/ Partial Proportional Odds Models for Ordinal Dependent Variables.” The Stata Journal 6(1):58-82. <http://www.stata-journal.com/article.html?article=st0097>.

More of Richard Williams’s work can be found at

<https://academicweb.nd.edu/~rwilliam/>