advanced WORKFLOW tips

A reminder about the principles of WORKFLOW
Research requires that analyses be ROBUST & REPLICABLE. 'Workflow' simply refers to the process of performing, document & archiving analyses so that results can be confirmed, replicated, &/or built upon in the future.

Two guiding questions for a good WORKFLOW

- If I were forced to hand the documentation for this project off to another analyst, would I be confident they could replicate my results?
- Could I return to my analyses in 5-7 years & replicate my results? (N.B.: For many research projects, this timeline is very realistic…)

If the answer to either question is no, your WORKFLOW process needs improvement. Review the WORKFLOW primer first, then read on for further tips.

TIPS FOR MANAGING DATA SETS

Data management sometimes requires managing or providing data in multiple formats. Keeping track of these different sources is important; so is ensuring that data have been converted from one format to another correctly.

Tips for converting data & verifying

- **Explore software-specific commands for converting between platforms:** Stata, in particular, has a number of options for importing data from other programs, like SAS, SPSS or Excel. Spend some time researching these commands to determine which one is best. For SAS → Stata transfers, for example, the `fdause` command (developed for use by the U.S. Food & Drug Administration) can import value labels from a `formats.xpf` SAS XPORT file. Or, you may find that using an intermediary (e.g., StatTransfer or DBMS/Copy) best preserves information across formats.

- **Convert the data in more than one way:** If you are unsure about the conversion method you are using, convert the file using two methods (e.g., use the `-insheet-` command & then the `-import excel-` command to import Excel data). Then use Stata's `cf` command to compare the data files. If the files match, you can feel confident about your conversion procedure(s); if not, verify differences & spend some time exploring why they occurred.

- **Verify data conversion by checking summary statistics & missing values:** Prior to conversion, print out a simple list of summary statistics & sample N for all variables in the data set using the original program. After converting, run these same commands in the new program & check the new numbers against the old. Note that extended missing categories can be especially troublesome in conversions, so be sure to check these.

  - **HINT:** In Stata, be sure to use the option `missing` or `m` after the `tab` command to include marginal distributions for missing categories.

Using data signatures: The `-datasignature set-` command adds metadata to a saved dataset that allows verification of correct, unchanged data. Datasignatures should be added to all datasets used for personal analyses; however datasignatures can also be used to help keep track of shared data.

- **For personal use:** Add a datasignature to each version of a dataset. Keep track of each version's signature in the research log file. Confirm the dataset being used (with the command `-datasignature confirm-`) prior to any analyses.

- **For working with a data manager:** Request that your data manager add a datasignature to any data s/he shares. Confirm that datasignature prior to performing analyses to ensure you have received the correct version of shared data. If you make further changes to the data, create a new datasignature for the changed data. Again, confirm that datasignature prior to running analyses with this changed data. Both you & your data manager should be sure to keep track of datasignatures in your respective project &/or data-specific log files.

MAKING DATA ANALYSIS EASIER

One of the paradoxes of data analysis is that including more information about your variables can make things both better AND worse: better in that more information = more efficient decision-making; worse in that most software packages aren't designed to handle more than 8-15 characters before outputs become difficult to read or tables wrap. Below are a couple of quick tips for managing these dual needs.

Using multiple languages to manage value labels

Stata includes the capability to attach multiple 'languages' of labels to the same set of variables, using the `-label language-` command. Certainly this can be useful if dealing with cross-national data or international collaborators; however the same language commands can be used to attach two sets of (differently useful) labels to variables. Two examples of this below:

- **Old & new labels:** Say you inherit an already-labeled data set from a colleague, professor or organization. The data is labeled, but not in a way you find optimal (or even useful). Rather than changing all of the labels, simply save the original labels as one language (e.g., `-label language old-`), then define a new language (`-label language new, new`) and change labels as desired. Both languages remain active & the `-label language-` command allows toggling between labels as desired.

- **Source & analysis labels:** Short labels are great for those who know the data intimately; however, long(er) labels may be necessary for presenting results to new colleagues or audiences. I frequently define an `analysis` label language and a `source`. `Analysis` includes only the important information about the variables; `source` includes, e.g., the exact wording of the question or response options.

Extended missing codes

We all know that not all missing data is equally missing, or created through the same processes. In some cases, response options that are considered missing for one analysis will provide useful information for another. Stata allows for the use of extended missing categories, which can help with keeping track of sources of missingness or data 'skip-patterns.' These extended labels consist simply of Stata's typical "." followed by a single letter (case-sensitive). The table below provides a few of the most common extended-missing categories that I use (*cf.* Long, 2009: 230).

Missing code	Value label	Meaning
`.c`	`catskip`	Categorical response not needed
`.d`	`nodebrief`	Refused to answer debriefing questions
`.f`	`femskip`	Females not asked question
`.m`	`maleskip`	Males not asked this question
`.p`	`priorref`	Question not asked as lead-in question refused
`.r`	`refused`	Current question refused
`.z`	`prior_0`	Not asked as answer to previous question was 0

DOCUMENTING PROVENANCE

In addition to keeping track of *which* decisions are made, it's also helpful to keep track of *where* & *when* decisions are made. A couple of tricks can make documenting this trail, from do-file to data to manuscript, much easier to follow.

- **Within your data:** In addition to making comments in your do-file about new variable construction, it's also important to keep track of which do-file created which variables. I do this by adding a "tag" local at the beginning of my do-file, which includes the name of the do-file, the date, and my initials, using the following commands:

  ```
  local     dte   "date"
  local     job   "filename"
  local     tag   "`job'.do-sns-`dte'"
  ```

  Then, at the bottom of each do-file, I create a loop that adds the tag as a note to each new variable. For example, if I had created 10 new variables var1 through var10, I would use the following loop:

  ```
  foreach v of varlist var1-var10 {
        note `v': `tag'
        }
  ```

  Then, at any point in the data cleaning or analysis process, if I want to double-check which file created var5, I simply need to use the command:

  `notes var5` which will list the note(s) for var5 only; or
  `notes` which will list the notes for all variables & the dataset.

  - **HINT:** Track the provenance of graphs &/or figures with the same `tag` by using the `note` option (with very small [`vsmall`] text!) in your graph commands.

- **Within your manuscripts:** Given the time that the review process can take, It can also be helpful to track do-file provenance of tables, figures and in-text numbers within manuscript drafts. The easiest way to do this is simply to write the do-file name that corresponds to each result within the manuscript. Then, prior to submitting papers for review (or finalizing manuscript formatting prior to submission), change the font of the provenance text to 'hidden.' In further editing, simply use the 'show all nonprinting characters' option (¶ button) to toggle showing/hiding the hidden text.

LEARNING MORE ABOUT WORKFLOW

Interested in more WORKFLOW tools? A few more resources:

- **Read the source of many of these tips.** Long's (2009) *The Workflow of Data Analysis Using Stata* expands on these tips & provides many more. Available for purchase through Stata Press & most booksellers.

- **Learn to automate your analyses using loops, locals & matrices.** There are lots of resources online, including a seminar I give regularly. See http://www.shawnasmith.net/teaching/ for materials.

- **Learn to write ado files.** Ado files are **a**utomatically-loading **do**-files, or do-files that run just like native Stata commands. Ado files can be written for any purpose imaginable, and numerous examples are available online. Learning some basic programming skills (locals & loops are a great start!) can help you customize the examples online as necessary.