1 General Workflow Suggestions:

Here we offer some specific workflow organization suggestions that should be valid regardless of code or operating system.

- Do not use spaces in directory or file names, as it complicates referring to them in certain software.
- Use "naming directories", .i.e. a directory beginning with "-" (so that it will appear first alphabetically) inside each directory to explain the contents of the above directory.
- Add name, date, and describe contents, as well as updates, to all scripting files.
- Keep a daily research log, i.e. a detailed written diary of what research is done on a given day. You'll be surprised how often this will be useful to answer questions about whether you ran a certain test or not, when you did it, and what you called the file.
- Make sure that your script files are self-contained. That is, don't write a program that only works if you run a group of other files previously in a specific order and then leave things a certain precarious way. You should be able to run intermediate steps of the workflow without disrupting things downstream.
- You can never comment too much.
- Indent your code
- Once you post/distribute code or data, any changes at all require a new file name.
- Separate your cleaning and analysis files; don't make any new variables that need saving (or will be used by multiple analysis files) in an analysis file—it is better to only create the variables once so you know they're the identical when used in different analysis files.
- Never name a file "final" because it won't be.
- Name binary variables "male" instead of "gender" so that the name is more informative.
- Use a prefix such as x_ or temp_ so you know which files can easily be deleted.
- Never change the contents of a variable unless you give it a new name.
- Every variable should have a label.

1.1 Stata-specific Suggestions

- Use the full set of missing values available to you (".a"-".z", not exclusively ".") in order to distinguish between "don't know" and "didn't ask" or other distinct reasons for missing data.

- Make sure code always produces same result—if you use anything randomly generated, set the seed. When sorting or merging, you need to be sure to uniquely specify observations, because if you don't, Stata does something arbitrary and not repeatable. So instead of just sorting or merging on 'ID' when there are multiple observations per ID, sort by 'ID' and 'name.' You can use the 'duplicates' command to test whether the *varlist* you use uniquely indentifies observations. The 'sort, stable' command can be used, though it is slower.

- Use the 'version' command in your .do file to ensure that other researchers who run your code with a newer version of Stata get the same results.

- Don't use abbreviations for variables (which may become unstable after adding variables) or commands (beyond reason)

- It's a common piece of advice to avoid using global macros, and use locals instead. (Since globals can be accessed across different functions or spaces, they can create contradictions or inconsistent dependencies.) However, in Stata, it's possible to safely use them to define directory paths so collaborators can work across different computers. An alternative to globals is to use: 'capture cd C:/Users/garret/Documents/Researchproject' and have one line for each user/computer.

- Use locals for varlists to ensure that long lists of variables include the same variables whenever intended.

- Use computer-stored versions of numerical output instead of manually typing in numbers or copying and pasting. For example, instead of copying and pasting the mean after a 'summ' command, refer to 'r(mean)'. Use the 'return list' command to see a full list of stored values after a regular command and the 'ereturn list' after estimation commands.

- If you have a master .do file that calls other .do files, which each have their own .log file, you can run multiple log files at the same time (so you have a master .log file)

- Use the 'label data' and 'notes' commands to label datasets and help yourself and other researchers easily identify the contents.

- Use the 'notes' command for variables as well for identifying information that is too long for the variable label.

- Use the 'datasignature' command to generate a hash or checksum and help ensure that data is the same as before.

- In addition to labeling your variables, you should also use value labels for all categorical variables. Include the numerical value in the label, however, since without it, it can be hard to tell what numerical value is actually meant by a given category. 'numlabel , add' is your friend.

- Even though Stata is case sensitive, don't use capital letters in variable names since not all software packages are case sensitive.

- Make your files as non-proprietary as possible (use the 'saveold' command to enable those with earlier versions to use your data. This is why trusted repositories are so useful–they'll do this for you.)

- Use the 'tempfile' and 'tempvar' commands to save space and not clutter up datasets and folders with temporary files and variables.