

Organizing a project analysis

Some suggestions

HIPRC Seminar, Feb 5, 1999

Peter Cummings MD, MPH

1. Directory structure:
 - a. Don't mix analysis files with program files. If your name is Joe, you should have an overall directory called "Joe" or "My files" or whatever. Within that directory, have a directory of the files for each research project. You might have subdirectories for human subjects, or forms, or correspondence, or the grant proposal, or whatever. One of those subdirectories might be called analysis or data. In that directory, you will put the data, the files that do the analysis, and all the output files. That way everything you need for the analysis is in one place. The directory structure might look like: D:\Joe\research project\data.
 - b. I like to keep the "raw" data inviolate and protected. I never change raw data files. By "raw" I mean the data that comes to me after key entry has been done. I tend to put these raw data files in a further subdirectory called rawdata. So the directory path might look like this: D:\Joe\research project\data\rawdata.
 - c. I usually put the suffix "raw" on raw data files: "*.raw"
 - d. If I am using Stata, I open Stata, then type: cd D:\Joe\research project\data. I then create data import commands that import the raw data from the subdirectory, saving the new files in data. By keeping Stata in the data file, I virtually eliminate the chance that I will save new files over my raw data, a situation that I want to avoid. When I type the save command, everything is saved to the data directory; things would only be saved to rawdata, if I typed a path, which I don't do and would be unlikely to do by mistake.
 - e. In this way, you end up with everything you need in one place. You can easily save the analysis directory to a laptop, a backup disk or the server, transfer to someone else, and so forth.

2. Get a real ASCII text editor:
 - a. This is worth it because Word, Word Perfect and most of the word processors love to insert command codes. You can force them to stop doing this, but their default settings make this a hassle. Since your statistical software cannot read your word processor's codes, you don't want to deal with this.
 - b. Windows comes with an editor, Notepad. I find it primitive and annoying.
 - c. A good solution is to use Programmer's File Editor (PFE), which is free shareware: <http://www.lancs.ac.uk/people/cpaap/pfe/>
 - d. This utility is powerful. You can open all sorts of files at once, do lots of cutting and pasting, search and replace, etc. And the web-site is amusing.
 - e. To save some time, set the preferences in PFE so that it always opens at the last directory used.
 - f. Stata (version 6.0) now has it's own built in do file editor. It is good, but I'm still using PFE, maybe just because I'm used to it. PFE has many more features.

- g. I set the Windows on my computer so that Stata takes up most of the screen. PFE takes about half the screen, to the right. It is positioned so that I can always just click on a portion of it that shows when Stata is on top, and I can always just click on the command window in Stata. This makes it fast to go from editing a file of commands to executing it, back to editing it. This is another example of why a big screen makes for faster work. Size does matter.
3. In a typical analysis session I open up Stata and my first command is `cd: d:\peter\project\data`. That puts me in the directory where I want to do the work. I then open PFE, go to the same directory, open a blank page and start typing. I save this file of commands to the same directory. Data, analysis command files, and analysis output are all in the same directory.
4. Creation files:
- a. Organize files so that the creation of variables and data files is separate from the actual analysis of the data.
 - b. I took Stata's Net Course #151, Programming in Stata. Much of the course is about programming in only the simplest sense and the course was full of tips on how to organize an analysis. I'm passing on some of what I learned.
 - c. There are basically two types of files. Files that create data and files that do analyses of data. Creation files all have the prefix "cr" (pretty original!). So if I am doing a study of air bags, I have `crair1.do`, `crair2.do`, `crair3.do`, etc.
 - d. The suffix, "do" is there because Stata knows it is supposed to "do" what is in a "do" file. You can use another suffix and Stata can still execute the file, but I find it handy to use the default suffix in the filename.
 - e. As you create your data set, the creation files can get long because:
 - i. I do all the editing in these files, never in the screen browser. That way I have a record of all changes made to the original data. In a moment I can re-create the entire new set of data files from scratch.
 - ii. I use these files to create all new variables, name them, label them, and label their values.
 - iii. I use these files to merge data sets and check on the merge process.
 - iv. So if the files are long, break them up into 1, 2, 3, etc. and at each stage save `air1.dta`, `air2.dta`, `air3.dta`, etc.
 - v. So `crair2.do` would start with `air1.dta`, make changes, and save `air2.dta`.
 - vi. No, this is not rocket science, just a simple way of keeping track of what you are doing.
 - f. Break the browser habit. Try not to look at your data in spreadsheet format to see what it looks like. This works only for small data sets. When files are large, this gets to overwhelming. So get in the habit of using `tabulate`, `list`, `summarize` and other commands to understand your data.
 - g. Never edit data in spreadsheet format. You will make mistakes and have no way of finding your errors. Make changes in a "cr" file.

5. Analysis files:
 - a. Analysis files, which produce tables, graphs, frequencies, statistics, and so forth all start with the prefix “an”.
 - b. A file called anair1.do might use air2.dta to produce some result that I want. Later, if I modify crair1, air2 will change - I can rerun the analysis of that file by simply running anair1.do.
 - c. Analysis files can also get long and so I usually break them up into several smaller files, rather than make Stata run through everything.
 - d. You will often make variables up in an analysis file, or redo a label, or make other changes to data. If you will want these changes for other analysis files, it is wise to cut out this creation material, put it into a creation file, and then run the “an” file on the new data set. Keep creation of permanent variables out separate from analysis.

6. How to keep track of all this:
 - a. At the start of each file, write yourself a memo, protected so that the software won't try to execute it, which tells you what the file does. I often put the date of file creation in the memo. Do this for both creation and analysis files.
 - b. Create a file called “master.do” for each analysis. There is only one master.do file in a directory. Master.do just has a list of the “cr” and “an” files with comments on what each one does. By telling Stata “do master.do” your entire analysis should run! I rarely run “master.do”, but I often refer to master.do to figure out what file does what and make needed changes.

7. How to make a Stata “do” file run better (“do” file format):


```

/* crair1.do */
/* this file imports raw data and assigns labels. It creates air1.dta
created on Feb. 3, 1999 */
version 6.0 /* doing this means the file will run under all future versions */
capture log close /* closes any open log file. Captures error code if a log is open */
set more off
log using crair1.log, replace

...whole bunch of stuff...

log close
exit /* remember to put a carriage return after each line, including the last */

```

8. How to be sloppy and get away with it.
 - a. Junk.log. As I work, I often want to see part of the data, do a little subanalysis, create a little table that I can hold in my hand, list part of a record to find a problem, and so on. I do this by typing:
 - i. log using junk, replace
 - ii. whatever it needed

- iii. Then I print the log file. And close it.
 - iv. I know that any file named “junk.log” can be safely erased. It is just something I wanted temporarily.
 - b. Temporary data files.
 - i. Often I need to create several files as part of a particular analysis, but I don’t want to save them.
 - ii. So a quick-and-dirty method is to save temp1.dta, temp2, etc. Since a file named “temp*.*” is never meant to be permanent, I can erase it or write over it at any time. If I want to clean up my act, I issue erase commands at the end of the analysis, but if I have been sloppy and forgot to do this, no harm is done.
- 9. When to write an actual program?
 - a. When you repeat some task over and over.
 - b. When you need to get a little fancier, parse things, etc. One example is checking ranges in data and other data cleaning chores.
 - c. Stata calls these “ado” files
 - d. In general I keep an “ado” file in the same data directory as the rest of the project files. It runs there, but not for any other project.
 - e. If your “ado” file is perfect and usable for many chores, you could put it in your personal “ado” directory, usually C:\ado.
 - f. Never put a personal “ado” file in Stata\ado. Bad things happen and you will die young and penniless.
- 10. When to bother with all these rules.
 - a. The answer is simple: always.
 - b. I’ve had student’s say “Oh, it’s just a simple analysis, I’ll just do it interactively and be done.” Most of them regret this. More important, when I ignore my own rules, I often regret it.
 - c. Many “simple” analyses, when kept in proper “do” files, have ended up saving me a ton of grief.
- 11. Other software: I use Stata and have given Stata examples. But the principles apply to SPSS, SAS, and any other statistical package.
- 12. If you follow these rules, you can just print your “do” files and your log files and put them in a data notebook. You will then have a paper record of the entire project, for yourself or anyone else.