Programming Practices

Tyler Ransom[1]

June 12, 2012

[1]Taken from Justin Trogdon

Steps to good programming

- Plan
- Develop
- Test
- Time should be spent equally on all three steps

Plan

- Make a road map
  - What is the desired output?
  - What variables will be needed?
  - Are there components that will be used again?
- Keep components small and manageable

Develop

- Take advantage of existing routines (e.g. `regress`)
- Be efficient
  - Vectorize code in Matlab
- Include detailed comments (see below)

Test

- Test each component of a program separately
- Use small subset of data

Reproducibility

- All work should be reproducible
  - Accidental deletion of results
  - Computer/hard drive crashes
  - Referee wants a slightly different specification but you can't find the .log file with the results

Master file

- In Stata, create a master .do file ('master.do') that contains:
  - Descriptions of each step in analysis
  - Calls to each .do file in order
- Could run this file and reproduce all files and results
- Useful for documentation
  - File structure
  - History of analysis decisions
- Can do something similar in Matlab, but Matlab's treatment of functions makes this a bit unnatural

Organization of files

- Rule 1: One directory, and one directory only, for a project
  - Some people use separate subdirectories for:
    - Data files
    - Program files (e.g., .sas, .m and .do files)
    - Log files
  - I tend to keep them all in the same directory

Organization of files

- Rule 2: No interactive exploratory analysis is official until it's in a .do (.sas, .m) file
  - Explore the data interactively, but anything that you really want to remember should be in a .do (.m) file
  - The record of previously used commands makes this easy

Organization of files

- Rule 3: All .do (.sas, .m) files create logs
  - The standard preamble always checks that any open logs are closed and a new log is started
  - Do not have to be as protective of .log files as they can always be re-generated
    - Unless of course the program takes awhile
  - In Matlab, use `diary` to create a log file
    - Downside: `diary` only appends and does not ever replace
    - As a result, I prefer to create Matlab log files through the `matsub` command

Organization of files

- Rule 4: Naming conventions can help in locating files
  - Prefixes indicating purpose
    - Creating data sets (‘cr *filename. ext*’)
    - Analysis files (‘an *filename. ext*’)
  - Use same name for log files as for program files (.m or.do)

File naming conventions

- Creating data sets
  - When creating datasets, use the same name of the data for the program and log files
  - Make sure to compress (pack) the data before storing to the server

File naming conventions

- Analysis files
  - No creation of permanent datasets
  - an*.ext files can be run in any order and will either:
    - Produce the same results
    - Not work at all
  - All datasets created within an*.ext files should be temporary
    - Stata: tmpfile *lclname* [*lclname* [...]]

File naming conventions

- Use suffixes that indicate order files
  - Version number (_v1)
  - Date (_20120708)
- Common to have several files doing very similar things with slight changes
  - Different estimation samples
  - Different variable definitions
  - Sensitivity analysis for different specifications

Organization of files

- Rule 5: Once a .do file is added to the master.do file, it is *never* edited again.
- If a change is necessary (e.g., add a variable to a specification), write a new .do (.sas, .m) file
- .do (.sas, .m) files are typically very small –> low memory cost to store many of them

Reproducibility

- Side note: make sure that commands that are random are reproducible
  - Set the random number seed before generating random number or drawing random samples (e.g., bootstrapping)
  - In Stata, 'sort' will randomly break ties.
    - Use 'isid *varlist*, sort' if existing variables uniquely identify an observation
    - Create a unique identifier otherwise and include it in sort

Reproducibility

- There are significant lags in publication process. You will likely have to return to code that is months old and understand it.
- Keep all programs (.sas, .m and .do) and .log files (these are small)
- Make sure all programs are well documented

Documenting Code

- Why document?
  - Because you may write code and have to return to it 1 year later and alter it.
  - Those who do not document suffer a harsh punishment later...
- The holy trinity of documentation in coding computer programs:
  - Comments
  - Meaningful Identifiers
  - White Space

Comments

- Comment in such a way that someone else could understand what you're doing, including yourself months from now.

Commenting in Matlab

- Begin with a % and then write whatever you want
- Multi-line comments can be offset with '%{' and '%}'
  - These are only useful when you are inside a continued line (see `http://blogs.mathworks.com/loren/2006/08/30/commenting-code/`)
- Matlab's text editor will color these comments green
- Comments falling just beneath a function's header, i.e., [FunctionName]([Arguments]), are printed when one types:
  - help [FunctionName]

Meaningful Identifiers

- Variable names should clearly indicate what the variable does or is for
- One convention:
  - GradSchoolIsFun
- In Stata, variable and value labels make it easier to understand and data set
- Matlab doesn't have such utilities, so variable names are that much more important

White Space

- Use blank lines to separate your code into the appropriate (commented) blocks, e.g.

```
%%%%%%%%%%%%%%%%%%%%%%%%
% Initializing Variables
%%%%%%%%%%%%%%%%%%%%%%%%
```

- Use indentation of command nests (e.g., if-else-end, while-end, switch-case-end)
  - Matlab's text editor will perform these indentations automatically

White Space

- Separate your code blocks into cells, e.g.

```
%% Variable Initialization
educ = data(:,1);
...
%% OLS Estimation
...
```

- These cells can then be “folded” so that the structure of your program is immediately visible
- Whatever you put after the %% is automatically bolded

White Space

- Finally, wrap long command lines to ensure readability
  - Matlab: `...[enter]`
  - Stata:
    - /* [enter]
    - */
    - Can also reset delimiter (for a line) from 'enter' (default): "#delimit ;"
    - /// [enter] will also do the trick

Network manners: Memory

- Work with subsets when developing
- Compress data when saving
- Delete intermediary data files (you can always reproduce them exactly!)
- Zip files that are not in use

Network manners: CPU

- Check server status before beginning long jobs
- Avoid looping over observations (i.e., vectorize code)
- Background jobs in batch mode
- 'nice' option in unix