Saving Time

Bill Rising

StataCorp LLC

2018 Stata Conference
Columbus, OH
July 20, 2018

Saving time

- Saving time is a Good Thing
- Using time to save time can be a good thing
  - It can also be a bad thing if it takes too much time to save time

Automation in Stata

- For Stata, saving time means automating repetitive tasks
- Do-files can be used for this
- Ado-files are not very hard to write
- Mata can also be used
- At all times, one needs to thing of the time saved vs the time used to save time

Other Tools

- There are other tools outside of Stata which are useful when working with Stata
- These include other text editors and version control
- We'll brush by all of these

Built-in Time Savers

- Stata has some time-savers
- Dialog boxes
    - Save time for complicated graphs
- Command-window shortcuts
    - Reusing commands with page up and page down
    - Tab-completion of variable names
    - Tab-completion of file names

Looking at Files

- It's nice to look at the files in your working directory

```
. dir
```

- Sometimes, however, it would be nice really see the files

opendir

- Here is a small community-contributed command for opening up an Explorer/Finder/File window in any OS

```
. opendir
```

- This can open other folders/directories, also

```
. opendir ..
```

- This works in any OS

A Fractured Existence

- Imagine a computer where
    - Hobbies are stored one place
    - Official projects are stored another place
    - Author Support projects are another place
    - Homebrewed projects are another place
    - Bug reports are in another place
    - etc.
- So... a typical computer, but possibly with different types of projects

Jumping from Place to Place

- Now suppose that we would like to move from one place to another
- This can be done via the OS
    - On the Mac, this is not too onerous
    - In Windows it is
        - The dialog has no remembrance of things past
- It can be done via the Command window, using tab completion

```
. cd "~/Desktop/2018columbus/data"
```

Making a Quick Visit

- Sometimes it is worth visiting quickly . . .

```
. cd "~/Documents/Scratch"
```

- . . . doing some work . . .

```
. * work work work
```

- . . . and coming back

```
. cd "~/Desktop/2018columbus/data"
```

- Doing this by hand is miserable
  - Copying and pasting can help, but you need to remember to copy!

pushd and popd

- Here are two simple commands for jumping back and forth:
    - `pushd` changes directory, but keeps track of the current directory for later
    - `popd` jumps back to the last pushed directory
- You can push multiple times in a row and build a stack of directories through which you can then backtrack
    - Though this isn't all that useful
- These get used just like `cd`

Example of Pushing and Popping

- Here is the above example of jumping around using these commands
- First: go to the Scratch directory

```
. pushd "~/Documents/Scratch"
```

- Do some work

```
. * work work work
```

- Come back

```
. popd
```

- This is nice, but not that nice

Known Special Locations

- Better than this is some way to jump to specially named places
- For this, there is the user-written go
- Here is my current state of shortcuts

```
. go list
```

- I could jump to the scratch directory . . .

```
. go scratch
```

- . . . and come back

```
. popd
```

Aside: How go Works

- It creates a do-file in your `PERSONAL` folder named `golookup_`*OS*`.do`
    - The *OS* gets replaced by your operating system
        - This oddity is needed for someone working/testing for multiple operating systems on one machine
- The do-file gets read when setting up a Mata object to hold the lookups
    - The object is called an associative array by Stata or a heap by some other languages

Aside: Where the Shortcuts Get Saved

- By default, the do-file gets written every time you make a change
  - You can squelch a write with the `nowrite` option
    - But then you should `go write` at some point before quitting Stata
  - This is in case someone is, say, writing shortcuts en masse
- The do-file is useful because it allows hand-editing
- We can take a look at it; first jump to my `PERSONAL` folder

```
. go personal
```

- Then look

```
. doedit "golookup_MacOSX"
```

Wrapup of go

- I find go very handy, and it saves many many many small bits of time
- It did take a while to write, but it was done as an exercise to learn the programming methods in Bill Gould's book about programming Mata
- Let's go back to the talk directory

```
. popd
```

Emacs and ado-mode

- If you find the Do-file Editor limited, try looking for other text editors
- I use Emacs, and edit my do-files with a "mode" called `ado-mode`
  - I use Aquamacs (`http://aquamacs.org`) which makes Emacs much nicer, but is Mac-only
- This is available at `https://www.louabill.org/Stata/`

Advantages

- Can submit code to Stata and have the commands in the Review window
- Can submit code with // and /// comments without issue
- Can open help and/or code for commands easily
    - Even personal or downloaded commands
- Has better syntax highlighting
- Has supplied templates for ado, do, and help files

Disadvantages

- Installation is not friendly
- Emacs is an old text editor built in the early 1980’s
  - So it has strange keyboard shortcuts

Version Control

- Version control in Stata means using a `version` command to keep syntax valid for the future
- Version control outside of Stata means keeping track of edits you make to files
    - This is also called “revision control”
- Using version control saves headaches and heartaches when changing files, because it allows gracefully backing out of changes
    - It also allows working on long changes to critical files, because of easy reversion
- Version control allows many people to work on the same files

What Tools Are There?

- There are any number of tools; these are open source
    - Subversion (or SVN)
    - Git
    - Mecurial
- I use Git

The Good and Bad of Git

- Good
    - It is decentralized, so it is better for someone who travels
    - It takes up very little disk space because it stores incremental changes instead of complete backups
    - It has an active user base
        - Microsoft just bought github for several billion dollars
- Bad
    - It redefines "cryptic"

What to Do?

- In situations where things are really cryptic, try buying software!
- I use Tower `https://www.git-tower.com`
  - This is paid software
- It is meant as a frontend to Git which can be used by graphic designers
- So... time for a small demo which does not show in the slides

Presentations as Outlines

- Presentations such as this are nothing more than outlines
    - With graphics
    - Sometimes with callouts
- While there are a few outlining programs, not many save their data in a useful form
- OmniOutliner is a program which can save an outline in OPML (Outline Processor Markup Language)
- The OPML can be post-processed and turned into a LaTeXfile
- OmniOutliner is also commercial software
    - `https://omnigroup.com`

StatWeave

- StatWeave was the first piece of software for integrating Stata input and output into either LATEXor docx files
- It has a useful quirk which allows it to split input from output
  - This makes it good for, say, including output in handouts, but not in slides
- This is open source software
  - Old version: `http://homepage.divms.uiowa.edu/~rlenth/StatWeave/`
  - New version: coming in September to github

A Brief Example

- We can typeset this lesson now as an example

Statistics

- I save time by using Stata

Conclusion

- Saving time is a worthwhile endeavour
- Saving time should not be at the cost of using more time
- The trick is assessing the effort and the longevity of the shortcuts