

Reproducible Workflow: Coding Strategies and Software [75mins]

Introduction, Hands-on with Version Control (Github) and Dynamic Documents (RMarkdown)

Fernando Hoces de la Guardia
BITSS

-

Slides at <https://goo.gl/aBQ3LR>

Inter-American Development Bank Workshop, March 2018

File Management & Coding Suggestions
(Christensen et al, 2018) [15 mins]

File Management [5 mins]

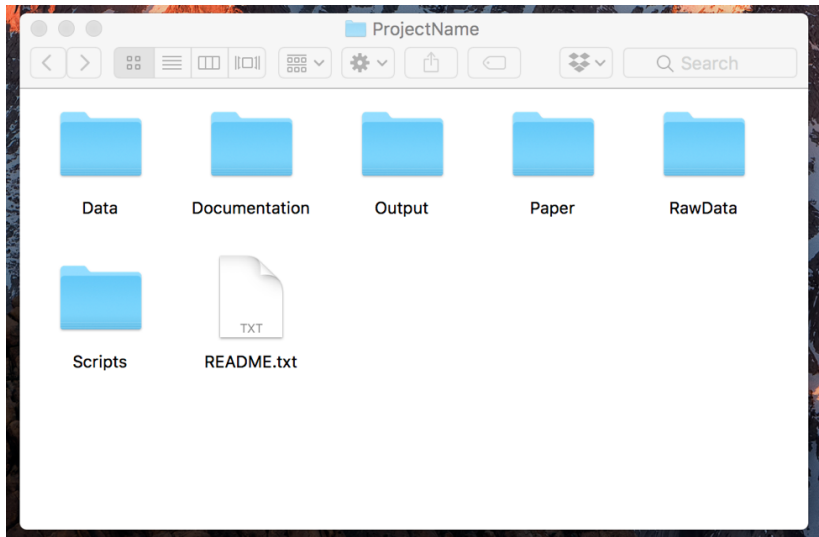


Figure 1:

Organizing Principles

- 1 - Use code (scripts), don't work by hand (GUI's or comand line).
- 2 - Consider not saving statistical output, and just saving the code and data that generates it.
- 3 - Reproducibility. Minimum: machine (laptop) independence. Ideal: analyst indepenence.

Coding Suggestions [10 mins]

1. Include tests in your code.
2. You can never comment your code too much.
3. Indent your code.
4. Once posted, any changes at all require a new file name. Or a version control system in place.
5. Separate your data cleaning and analysis files
6. Never name a file “final” because it won’t be.
7. Name binary variables “male” instead of “gender,” (1=Male and 0=Not)
8. Don’t leave clutter around-delete temporary or unnecessary intermediate objects.
9. Every variable should have a label.
10. Use relative directory paths (such as “./Data” and not “C:/Users/Fernando/Documents/Project/Data”)

Coding Suggestions: Stata-specific

1. Accurately and concisely capture missing values. (`.` and `.a-.z`)
2. Make sure code always produces the same result, and that merging and sorting is reproducible. `duplicates report; isid; sort, stable`
3. Run simple tests to alert yourself when results change.

Example:

```
count if _merge!=3
if r(N)!=74 {
display "Unmatched observations changed!"
there is an error here
}
```

4. Don't use abbreviations for variables or commands.
5. Use global macros to define directory paths so collaborators can readily work across different computers.
6. Use local macros for varlists.

Coding Suggestions: Stata-specific

7. Use computer-stored versions of numerical output (eg `r(mean)`). Use `return list` and `ereturn list`
8. If you have a master `.do` file that calls other `.do` files, which each have their own `.log` file capturing output, you can run multiple log files at the same time (so you end up with a master `.log` file)
9. Use the `label data` and `notes`.
10. Use the `notes` command for variables as well for identifying information that is too long for the variable label.
11. Validate data sources to ensure consistency. Use `datasignature` on `auto` data set (`sysuse auto.dta`, then `datasignature` set should give you this number:
`74:12(71728):3831085005:1395876116`)
12. Use value labels for all categorical variables. `numlabel [lblname-list]`, add `command`.
13. Don't use capital letters in variable names.
14. Make your files as non-proprietary as possible (use the `saveold` command)

Version Control + DD [30/35 mins]

Version Control Problem to avoid [15]

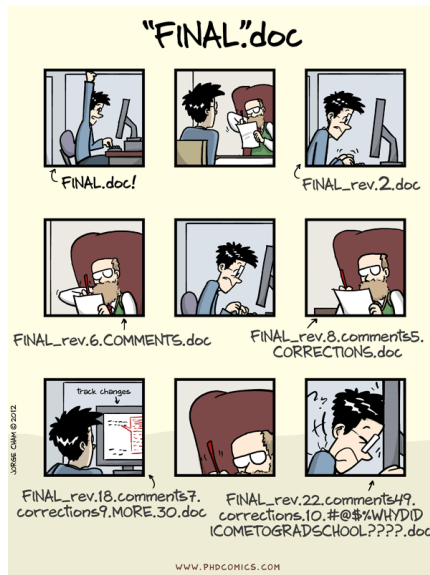


Figure 2: <http://www.phdcomics.com/comics/archive/phd101212s.gif>

Managing expectations



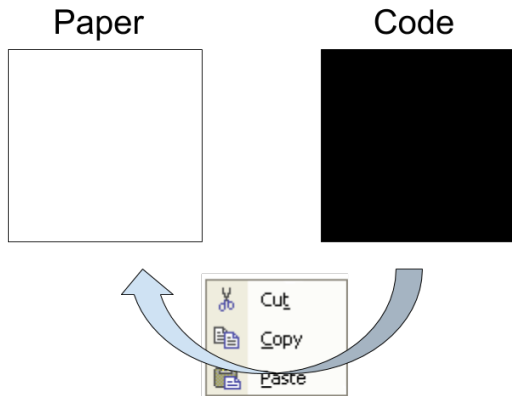
Figure 3: Git xkcd comic

Dynamic Documents [20 mins]

Dynamic Documents For Computational Reproducibility

- ▶ Based on principles of *literate programming* aims at combining code and paper in one single document
- ▶ Best framework to achieve the holy grail of **one-click reproducible workflow**
- ▶ Best two current implementations: RMarkdown (R) & Jupyter (Python). Stata is catching up (more at the end)

Currently code and narrative components live in separate universes



Dynamic Documents: integrate the two universes!

Paper + Code



Figure 5:

Dynamic Documents: A Recipe

- ▶ 1 simple language that can combine text and code: Markdown
- ▶ 1 statistical package to do the analysis (R, Python, 3S's?)
- ▶ 1 machinery to combine analysis and text to create a single output: Pandoc
- ▶ [Optional-but-not-really] 1 program to bring all the elements together: RStudio/RMarkdown, Jupyter

Hands-on!

CONSORT diagram of our little experiment

Balance of covariates

Estimated effect

The Stata version of all of the above:

Additional Resources

Garret, Ted and Jeremy's book

The Practice of Reproducible Research

Code and Data for the Social Sciences

The Workflow of Data Analysis Using Stata

Reproducible Research with R and R Studio

Project TIER

Great intro to GitHub by Jenny Bryan