Reproducible Results: A Workflow for Data Analysis

Scott Long

wfclass2017 lecture parts01-07 2017-07-10.docx

Table of Contents

PART 1: REPRODUCIBLE RESULTS 1
- Open science & reproducible research 1
- Two dimensions of "the same results" 2
- What is *workflow*? 4
- Challenges for reproducible results 10
- WF is built on ironical optimism 14
- Stages in your WF 16
- The goal for choosing your WF 31
- As you develop your workflow 33
- Challenges when learning workflow 34
- How good should your workflow be? 35
- Whose workflow? 36
- What you can accomplish in this class 37
- Overview 38

PART 2: TOOLS 1
- Macro programs 4
- Dual pane file managers 5
- Text editors 8
- Password managers 9
- The cloud 10

PART 3: DIGITAL ASSET MANAGEMENT 1
- Digital assets and workflow 2
- Organizing digital assets 4
- Naming files and directories 6
- Overview of organizing files 11
- Directory structures 11
- General directory structure 17
- Moving into a new directory structure 22
- Using the cloud 25
- Directories for collaborative projects 27
- Overview of DAM 27

PART 4: PROTECTING FILES 1
- Causes of data loss 2
- Recovering lost files 7
- Examples of lost data 8
- Protecting your data 11
- Flow of file preservation 19
- Keeping track of what is backed up 26
- Tools for backups 27
- My backup plan (more or less) 27

PART 5: USING STATA 1
- Stata interface 3
- Command syntax 10
- Three ways to run commands 13
- Why use script files? 19
- Stata's do-file editor 20

PART 6: PLAN, ORGANIZE & DOCUMENT 1
- Overview of POD 2
- Planning 5
- Collaboration 19
- Organizing 30
- Documentation 36
- Overview of POD 53

PART 7: A WORKFLOW FOR COMPUTING 1
- Script files for computing 3
- Project directory structure 4
- Strategies for computing 6
- The essential posting principle 7
- Dual workflows 14
- Run order naming and a dual workflow 19
- Master do-files 26

Part 1: Reproducible Results

Open science & reproducible research

1. Open science and reproducible research are new standards.
   - *The Center for Scientific Integrity* (retractionwatch.com)
   - *Data Access & Research Transparency* (DA-RT) (www.dartstatement.org)
   - *The Berkeley Initiative for Transparency in the Social Sciences* (www.bitss.org)
2. Expectations are changing.
   - Fields are archiving results and data.
   - Some journals require submission of data and analysis files.
   - The White House called for shared data.
   - Funding agencies have requirements for open science and reproducibility.
3. Your work will received greater scrutiny to certify reproducibility and accuracy.
4. These positive changes demand an effective workflow.

Two dimensions of "the same results"

Reproduction: same results with the same data

- Short term requirement with your data
- Easy to achieve with a proper workflow
- This class focuses on reproducibility

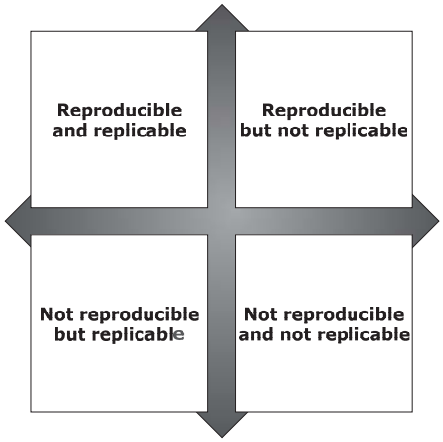
Replication: equivalent findings with new data

- Long term goal with new data
- Results are not repliacable if they exploit peculiarities of the sample

Without R&R, knowledge cannot accumulate

- Every aspect of your workflow must *anticipate* producing R&R results.

Does reproducible imply replicable?



What is *workflow*?

A workflow for data analysis is a set of *coordinated* procedures to work *efficiently* to create analyses that are *accurate* and *reproducible*.

WF includes the entire process of research

1. Planning research
2. Documenting work
3. Importing data
4. Managing data
5. Analyzing data
6. Presenting results
7. Revising analyses
8. Reproducing results
9. Preserving files

Why care about WF?

1. WF is essential for reproducible results.
2. WF prevents errors and retractions.
3. WF helps you find errors.
4. WF facilitates correcting errors and making revisions.
5. WF saves time.

Origins of WF project

- **Incorrect results** with clever explanations
- **A delayed dissertation** to determine why results changed
- **Irreproducible results** from a 743 line script file without comments
- **The wrong dataset**: "The datasets are *exactly* the same except that I changed the married variable."
- **The wrong variable** while writing an NAS report
- **Collaborations** that multiply the ways things can go wrong
- **Misleading output**...

Definitel a problem in a $3M study

```
. tabulate female sdchild_v1
```

R is female?	Q15 Would let X care for children: Defintel	Probably	Probably	Definitel	Total
0Male	41	99	155	197	492
1Female	73	98	156	215	542
Total	114	197	311	412	1,034

Which number is which?

```
. tab occ ed, row
```

Occupation	Years of education: 3	6	7	8	9	10	11	12	13	Total
Menial	0	2	0	0	3	1	3	12	2	31
	0.00	6.45	0.00	0.00	9.68	3.23	9.68	38.71	6.45	100.00
BlueCol	1	3	1	7	4	6	5	26	7	69
	1.45	4.35	1.45	10.14	5.80	8.70	7.25	37.68	10.14	100.00
Craft	0	3	2	3	2	2	7	39	7	84
	0.00	3.57	2.38	3.57	2.38	2.38	8.33	46.43	8.33	100.00
WhiteCol	0	0	0	1	0	1	2	19	4	41
	0.00	0.00	0.00	2.44	0.00	2.44	4.88	46.34	9.76	100.00

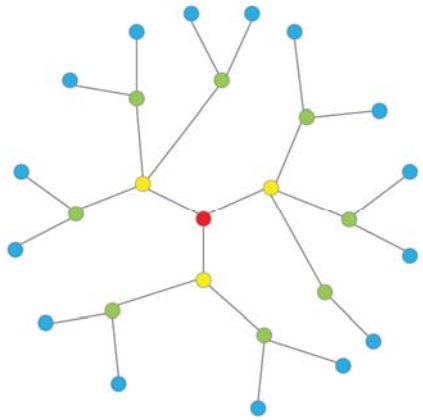
Key terms

Workflow (WF):	Coordinated procedures for data analysis
Reproducible:	Producing *identical* results with the same data
Replicable:	Producing *similar* results with different data
Script files/do-files:	Text files that control software
Template:	File with a standard form or an example of a procedure
Posted file:	File that will never change
Metadata:	Data that describes data

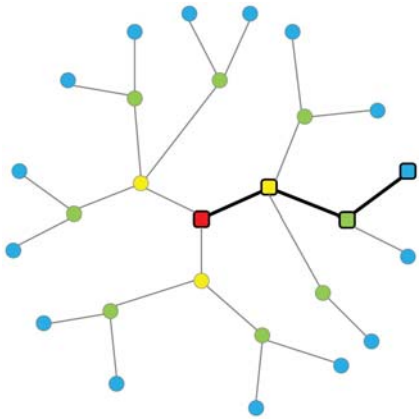
Challenges for reproducible results

1. **The curse of dimensionality**: research involves many decisions that must be repeated exactly.
   - Value at which variables is truncated
   - Seed for the RN generator
   - How scale is created with partially missing data
   - Cases chosen for analysis
   - Coding of education
   - Top coding for income greater than $200,000
   - And so on...

Decisions in the path to analysis: possible choices



Decisions in the path to analysis: choices made

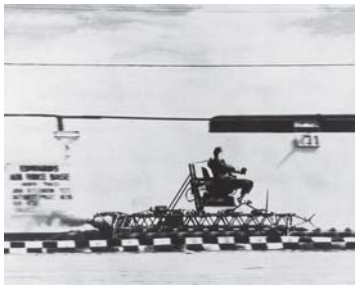


Challenges for reproducible results (continued)

2. **Documentation** is required to remember each decisions.
3. **Changing software** makes it hard to get exactly the same answers.
4. **Lost files** make reproduction is impossible.

WF is built on ironical optimism

The *universal aptitude for ineptitude* makes any human accomplishment an incredible miracle. *Dr. John Paul Stapp*



40G's: 1000mph & back in 3 seconds

"Only lost my vision for a few days..."

Stages in your WF

Stage 1: Substantive goals

Stage 2: Data management

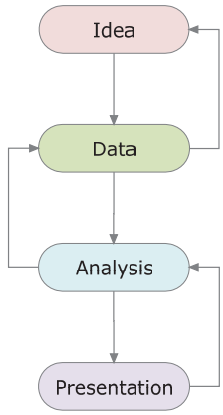
90% of the work

Stage 3: Data analysis

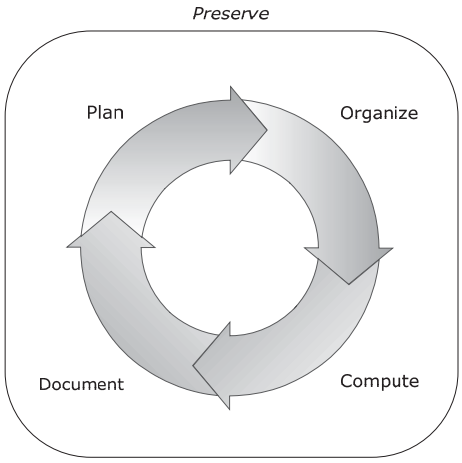
Accurate & reproducible results

Stage 4: Presentation

Effective while maintaining provenance

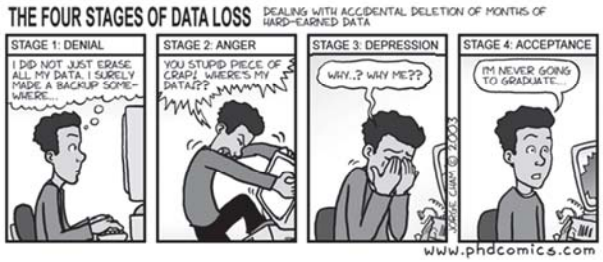


Tasks within each step



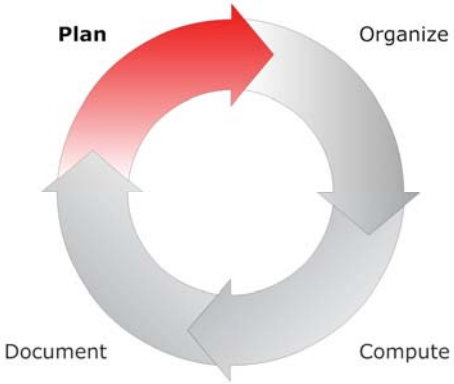
Preserving your work

At each step you must preserve what you have done.

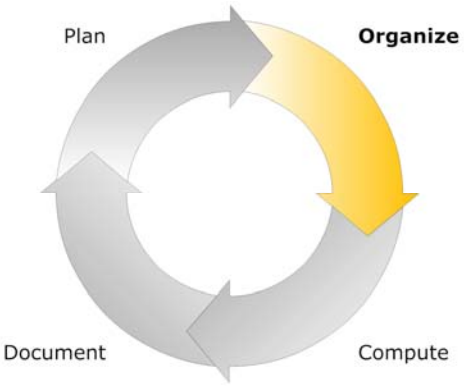


http://www.phdcomics.com/comics/archive.php?comicid=382

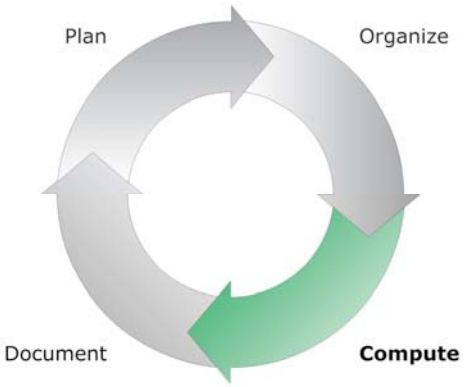
Tasks within each step



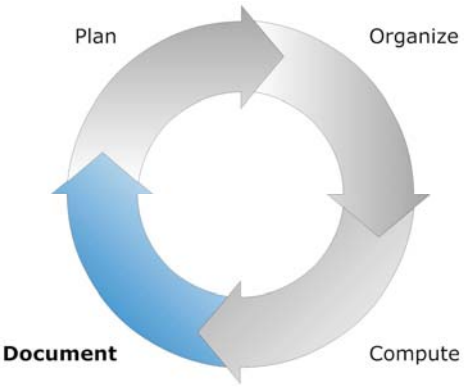
Tasks within each step



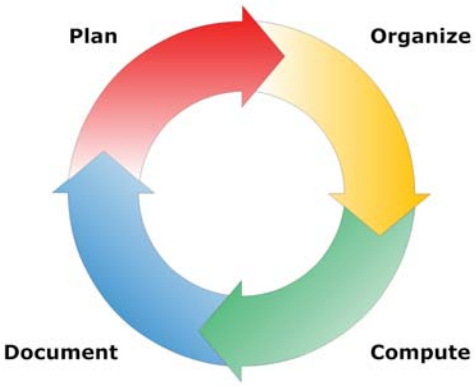
Tasks within each step



Tasks within each step



Tasks within each step



Task 1: Planning

1. Planning helps you
   - Stay on track
   - Finish the research
   - Publish the results
2. Planning considers goals, deadlines, division of labor, names and labels, missing data, modeling, documentation, presentation, and protecting materials.
3. Planning occurs in the *big*, in the *middle*, and in the *small*.
4. Most people plan too little and compute too much.

Task 2: Organization

1. Helps you *find things* and *avoid duplication*.
2. Requires *thinking systematically* about procedures.
3. Rewards *consistency* and *uniformity*.
4. Is *contagious* and so is the lack of organization.
5. Makes you more *efficient*.
6. Reverses the effects of *entropy*.

Signs of poor organization

1. You can't find a file and think you deleted it.
2. You and a colleague are working on different versions of the same paper. You changed what she changed and now you have three versions of the paper.
3. You need the final version of the paper that was submitted for review, but you have two (or 16) files with "final" in the name.

Task 3: Documentation

1. Without documentation, *reproduction* is impossible, *mistakes* are more likely, and *time is wasted*.
2. More codified fields place greater the emphasis on documentation.
   - The Research Log by the American Chemical Society.
3. Universal truths about documentation:
   - It is faster to document it today than tomorrow.
   - Nobody likes to write documentation.
   - Nobody regrets having written documentation.
4. Effective and efficient documentation occurs on multiple, *reinforcing* levels.
   - Research diary
   - Commented do-files
   - Metadata
   - Provenance in presentations

Stage 4: Execution/Computing

1. Effective execution requires the *right tools*.
2. Effective use of tools considers the trades-off between:
   - Time spent learning the tool
   - Time saved and accuracy gained by the tool
3. Execution in the absence of POD makes things go slower.
   - Computing is compelling
   - Resist computing without organization and a plan

Consider the recent history of computing...

Cornell 1975: the entire computing infrastructure

IBM 370 with 240K memory

Winchester drives with 3MB storage

Equipment cost: $1,000,000

Mean time to degree: 7.6 years

Indiana 2009: a disposable PC

Asus 1000HE with 2GB memory

: 10,000 times more

FreeAgent with 1TB storage

: 350,000 times more

: 1,000,000 times more in 2015

Equipment costs: $400.

Mean time to degree: 7.6 years

A thought experiment: planning & computing

1. Imagine two groups of doctoral students

   Computers have unlimited computing.

   Planners compute eight hours per week.
2. Who finishes first?

The goal for choosing your WF

To efficiently produce statistical results that are reproducible and accurate.

The guiding principles

1. *Reproducibility* is obtaining the same results using your data and script files.
   - You and others must be get exactly the same results.
2. *Replication* is needed to show that your results can be generalized.
   - Others must understand enough to do approximately the same things you did and then obtain similar results with new data.

Criteria for choosing your WF

Accuracy	Efficiency	Scalability
Standardization	Automation	Simplicity
Usability		

Accuracy

1. WF helps find, correct, and prevent errors.

Efficiency

1. WF increases the time you have to work.
2. Efficiency balances *speed* and *care*.
3. Efficiency assesses *fixed* and *variable* costs relative to *frequency of use*.

Scalability

1. Works for small and large projects.
2. Works for single researcher and research team.

Standardization

1. Procedures for doing things consistently and efficiently.
2. If you know how things should look, you see problems more easily.
3. When you do something often, *create a template*.

Automation

1. Automation prevents errors and makes errors easier to fix.
2. Automation scales.
3. Time learning to automate is recovered with repetitions.

Simplicity

1. Procedures that are too complicated cause mistakes and are soon abandoned.
2. Things that are *simple to use* can be *hard to learn*.

Usability

1. A good WF compliments the way *you* like to work.
2. An effective WF only helps if it is used.

As you develop your workflow

1. How does your WF relate to these criteria?
2. How do your tools support these criteria?

Challenges when learning workflow

Tacit knowledge

- WF depends on things learned "at the bench" through experience.
- Personal computers make this learning harder.

Undifferentiated heavy lifting

- Developing your WF requires hard work, but it make later work easier and more efficient.

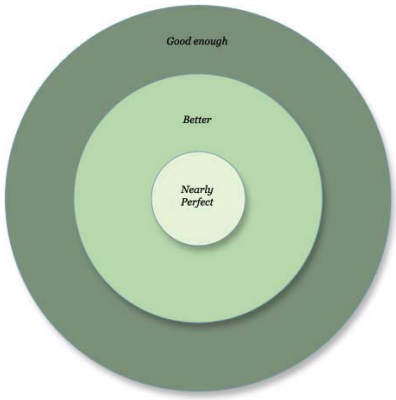
It takes time to practice

- This class provides that time.

Change is hard

- Initially, the changes seem hard, inefficient, and unnecessary.
- Not changing your WF is harder.

How good should your workflow be?



Whose workflow?

1. Each part of your workflow must be coordinated.
   - A WF is as effective as its weakest link.
2. Is my WF better than yours?
   - Do you want to spend your time discovering the mistakes I made?
   - Are you willing to think through the implications of every modification you make?
   - Will you document your WF?

What you can accomplish in this class

1. Assess your current workflow
2. Learn about new tools and strategies
3. Plan how to integrate this workflow into your work
4. The class is too short to fully adopt a new workflow

Adopting a new WF

1. New ways of doing things are uncomfortable.
   - Vic Bradon: "How does that feel?"
   - Student: "Awful!"
   - Vic Bradon: "Remember that awful—it will make you famous."
2. Stick with what you learn until the new procedures become routine
3. It is slower in the short run, but faster in the long run

Overview

What is workflow?

A workflow for data analysis is a set of coordinated procedures to work *efficiently* to create analyses that are *accurate* and *reproducible*.

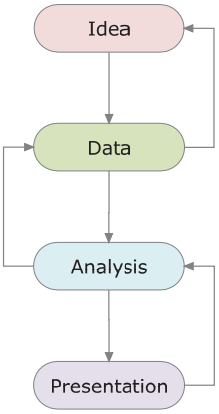
Guiding principle

The universal aptitude for ineptitude makes any human accomplishment an incredible miracle.

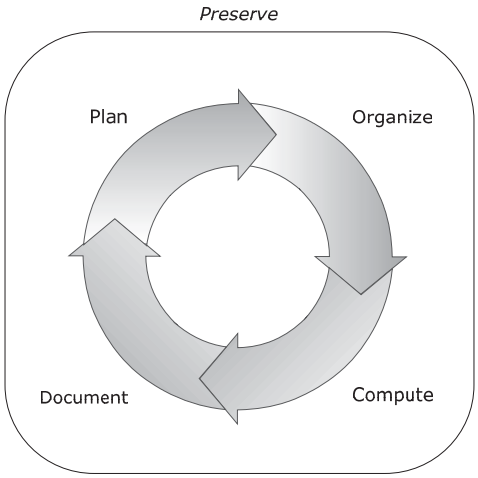
Criteria

Replicability and reproducibility, accuracy, efficiency, standardization, automation, and simplicity.

Stages in your WF



Tasks



Part 2: Tools

A scientist who does not love the tools is doomed to failure.

1. *Tools define tactics.*
   - o Tools make repetitions faster and more accurate.
   - o Tools make work easier and more enjoyable.
2. Types of tools:
   a. Low tech: 3-hole punch, stapler, Post-its, notebooks, etc.
   b. Hardware: Computers, monitors, scanners, storage, etc.
   c. Software: Statistical software, file managers, text editors, macro programs, etc.
   d. Automation: Work faster with fewer errors.
   e. Templates: Save time and make you more efficient and accurate.

Tools are an investment

1. Start with:
   - o The most critical requirements for reproducible results
   - o Easy things you can fix quickly
   - o Procedures that save the most time
2. Avoid learning a little of everything.
3. Evaluate how much time a tool will save (next slide).

The time-frequency assessment.

TIME per USE = How much time does the tool save each time you use it?
FREQ of USE = How often will you use it?
PAYOFF = TIME per USE x FREQ of USE

Example

1. I add dates to documents and to name files 1000s of times a year.
2. I invest time learning a macro program.
3. I create two simple shortcuts:

;ts	inserts	2015-04-15
;ds	inserts	2015-04-15_13-13

4. Three benefits:
   a. Each use saves time since I don't need to type or remember the date.
   b. The tool encourages me to use dates.
   c. All dates are in a uniform format.

Macro programs

Macro programs let you to complete a complex task or enter a short text string with a few keystrokes.

- o For example: **;ds** → 2015-04-05
  ;ts → 2015-04-05_10-17

Stand-alone macro programs

1. Win: AutoHotkey is free and powerful.
2. Mac: Automator does many things. Keyboard Maestro is more powerful.

Macros within programs

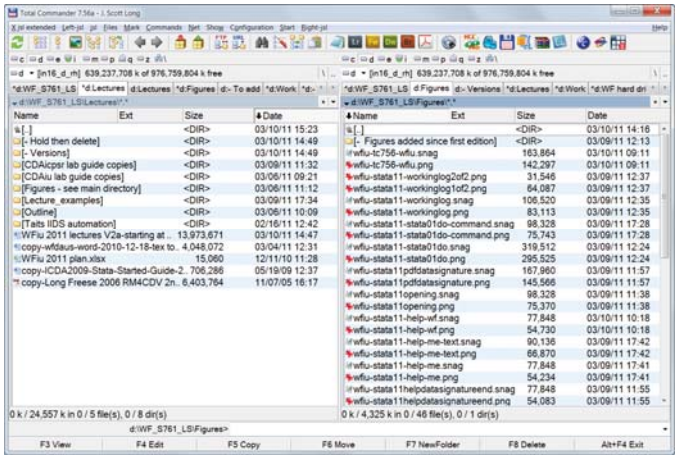
1. Many programs let you remap keystrokes and create new commands.
2. What things do you do often?

Dual pane file managers

1. The file managers in Mac and Win work poorly for data analysis.
   - o *Do not use them*.
2. A dual pane file manager is much better.
   - o Win: Total Commander
   - o Mac: Path Finder

Example…

Example of dual pane file manager



Key features

1. Dual panes
2. Projects and directory shortcuts
3. Keystrokes for any command -- you do not have to click
4. Utilities for compressing, searching, file comparison, directory comparisons, file viewing, file renaming, sorting, and more
5. Program launcher and OS commands

Please try it!

1. You *should* install and use one of these when you have time to use it consistently for several weeks. DO THIS!
2. When learning a new file manager, resist jumping back and forth between Explorer/Finder and the new file manager.
3. Gradually learn new features.

Text editors

Learning one to use with all statistics programs is an advantage.

Key features

1. Syntax highlighting
2. Templates
3. Auto indentation
4. Column mode
5. Sophisticated find and replace

Suggestions

1. *Notepad* is too simple and *Word* is too complex.
2. *Stata editor* is good if you only use Stata.
3. *notepad++* is Win freeware and excellent.
4. *TextWrangler* is Mac freeware and excellent.

Password managers

1. Rarely generate your own password.
   - o Do not use an algorithm to generate your passwords.
2. Be carefully about writing down passords.
3. Never use the same password twice.

The cloud

1. Files are stored offsite on servers unaffected by local disasters.
2. You can sync files to multiple local machines.
3. You share files with collaborators via the internet instead of attachments.
4. Backups are automatic.
5. **The cloud might *not* be secure enough for your work**.
6. You might have to pay for storage.
7. You can delete a file on many machines!

Part 3: Digital asset management

It is easier to create a file than to find a file.

It is easier to find a file than to know what is in the file.

It is harder to delete a file than create a file.

With disk space so cheap, it is tempting to create a lot of files.

WFDAUS pages 19-33.

How cheap is storage?

- GSS cumulative file has 57,061 cases and 5,570 variables (27,178,137 bytes)
- A $120 6TB drive holds 242,734 copies of the GSS at $0.00053 per copy

Digital assets and workflow

1. We are buried in digital assets.
   - *Digital Asset Management: The DAM Book* by Peter Krogh
2. Sound familiar?
   - You have multiple versions of a file and don't know which is which.
   - You think an important file was deleted.
   - You and a co-author both have the "latest" draft of your paper, but the two files differ.
   - You have two copies of a dataset named data.dta that are differnt.
3. DAM might seem to be a minor issue, but it has a huge impact on efficiency, accuracy, and reproducibility.
4. Operating systems provide little help when organizing files. For example, ...

The Win/Mac difference

Win	Mac
Desktop	Desktop
Music	Music
Pictures	Pictures
Videos	Movies
Documents	Documents

Organizing digital assets

Objectives in DAM

1. Find files
2. Avoid duplicates
3. Know what a file is
4. Preserve files

Stages in DAM

1. Name files consistently in ways that support DAM
2. Create a planned directory structure
3. Use naming templates and planned directory structures

Copy and move

1. Copying create duplicates.
   - Why do you want two copies?
   - What are the costs of duplicates?
   - How do know you which one to delete?
   - If you change a file, do you always need to give it a new name?
2. Moving files does not create duplicates.
   - Why would you want two copies instead of one?
3. Finder and Explorer make it too easy to copy when you want to move and move when you want to copy.

Thought experiment: solution later

1. You download the article Chen (2012). Where do you put it with what name?
2. Project 1 uses Chen (2012). Do you copy or move it to that project folder?
3. Project 2 uses Chen (2012). Do you copy or move it to that project folder?
4. Project 1 no longer needs Chen (2012). Do you delete it?

Naming files and directories

http://www.phdcomics.com/comics/archive.php?comicid=1531

Goals when naming files and directories

1. Names contain critical information.
2. Names are not too long or hard to type.
3. Naming conventions are consistent across file types.

Characters to use in names

1. The characters a-z, A-Z, 0-9, underscore _, and dash – generally work.
2. Blanks cause problems with some programs.
3. Which do you prefer and why?
   a. `\My◦Documents\`
   b. `\My_documents\`
   c. `\Documents\`

Dates

1. Which is most effective? Why?
   a. 17may2013
   b. May 17, 2013
   c. 17-05-2013
   d. 2013-05-17
   e. 2013_05_17
2. Which is easiest to type?
3. Which is easiest to use? Why?

Naming PDFs

1. A typical list of names of PDFs.

```
03-19Greene.pdf
0OWENS94.pdf
12087810.pdf
12087811.pdf
Chapter03.pdf
faiq-example.pdf
gllamm2004-12-10.pdf
long2.pdf
Muthen99biometrics.pdf
```

2. A simple naming template:

   lastname year journal keywords.PDF

   `long2.pdf` → `Long 1978 ASR prod position.PDF`

3. Keywords are used to search for a paper.
   - If I have trouble finding a PDF, I add a new keyword to the name.

A project keyword

1. Pick a keyword for your project that will be used in files and files names.
   - **CWH**: Cohort, work, and health
   - **SDSC**: Sex differences in the scientific career
   - **PASL**: Paul Allison/Scott Long
   - **WFCLASS**: Workflow class
2. *Short* keywords keep names from getting too long.
3. *Unique keywords* facilitate searching for files.
4. *Common words* like "THE" are poor choices.

Keywords are metadata

1. The project keyword *in the file name* is information about the file.
   - You can search for files with the keyword in the name.
2. The keyword *inside a file* ties the file to the project.
   - Why inside the file? What happens when a file is deleted?

Overview of organizing files

Directory structures

1. An effective directory structure is a critical first step in an effective WF.
2. The *general directory structure* is the overarching structure for all files.
3. A *project directory structure* is used with every project.
   - Details given as part of the workflow for computing

Why is your directory structure important?

1. The structure tells you where files belong.
   - If you don't have a place for something, it won't be in the right place.
2. The structure makes it easier to find files.
3. A file's directory location provides documentation.

Is my structure good enough for your work?

1. Use my structure as a starting point.
2. Imagine where you would put different types of files.
   - If you don't know, why not?
   - Reluctantly, make changes that consider the entire structure.
   - Repeat the exercise.
3. Don't tweak the structure for minor preferences.
   - I suggest **\Vault**.
   - Is **\Safe** really that much better?

Using your directory structure

1. After you decide on a structure
   - Always use it or you will make things worse.
   - Change it reluctantly with very good reasons.
2. When files are created, put them in the correct directory.
   - If you put files in the wrong place, you will have duplicates or files with the same name and different content.
3. Do not move into your new structure until you have:
   a. *Backed up* your files multiple times
   b. *Finalized* the structure
   c. Made a *commitment* to use the new structure
   d. Have a *detailed plan* on how to move into the new structure
   e. Have *time* to do move into the structure slowly and carefully

Tool: naming directories provide documentation

1. Normally a directory holds files, but can also be used as documentation.
2. I call these *naming directories* with the syntax

```
\-◦◦<naming directory>
```

3. The name of the subdirectory tells me what is in a directory.

```
\Posted
    \-  Never change these files
```

4. I use two spaces after the **–** so the directory sorts to the top.

\- History starting <date>

1. A history directory documents a project's history.

```
\Couples3
   \-◦◦History starting 2009-03-01
      \2009-03-09 begin data cleaning
      \2009-04-13 begin analysis
      \2009-05-01 analysis after project review
      \2009-07-19 analysis complete for 1st draft
      \2010-01-21 respond to coauthors
      \2010-03-17 archive files-paper submitted
      \2010-06-12 snapshot to MDSS
      \2010-11-09 revision analyses
      \2010-12-12 revision submitted; backups made
```


General directory structure

```
\-◦◦History starting 2015-06-03  ≪ Naming directory
    \2015-06-03 drive purchased

\-◦Hold then delete        ≪ Recycle bin/trashcan

\-◦To clean
   \Review                 ≪ What are these?
   \Shelve                 ≪ Put these away later

\Active
   \-◦◦Active projects
   \WFclass                ≪ Active class folder
   \Groups                 ≪ Active research folder
```



```
\Admin
   \-◦◦Administrative things
   \Travel
   \Vita

\Bookshelf                 ≪ Lending library
   \-◦◦Copy but do NOT move files
   \-◦Review
   \-◦Shelve
   \Articles
   \Books
   \Manuals
```

```
\Dropbox
   \Scott Long shared with
      \-◦◦Shared DB directories
      \Mason, Fred
         \To Fred
         \To Scott

\Inactive
   \-◦◦Inactive incomplete projects
   \Couples
   \SGC

\Program_support
   \-◦◦Software support files
   \Stata14                ≪ ado directories
   \TotalCommander         ≪ exe and customization
```



```
\Templates
   \-◦◦Templates and examples
   \Directory general
   \Directory project
   \Documents – LaTeX
   \Documents - Word
   \Stata

\Vault
   \-   History
   \-◦◦Files that never change

\Web
   \-◦◦Local copy of web site
```


A workflow for PDFs: copy and move

1. The problem
   a. You have multiple copies of the same PDF that have different names.
   b. You download a n that you already have rather than looking for the one you have.
2. The solution
   a. All PDFs of are saved in **\Bookshelf** using the naming template: lastname year journal keywords.
   b. If I need a PDF for a project, I *copy* it to the project's **\Resources**
      - The orginal stays in **\Bookshelf** so it is never lost.
      - This is what libraries do with digital checkout.
   c. When the project ends, I can delete PDFs in the project directory since they are copies, not originals.
   d. If I annotate a PDF for the project, I give it a new name and move it to **\Bookshelf** or keep it with the project.

Moving into a new directory structure

Overview

1. *All files* are *copied* to multiple backups
2. *Move* all files to reorganize into **\Old**
3. *Plan and create* your directory structure inside **\New**
4. *Populate* **\New** by *moving* things from **\Old**

Steps for moving into a new structure

Backups

1. Years of work can be lost moving into a new directory structures.
2. *Backup everything at least twice before starting; save backups different places.*

Move current files to \Old

1. *Move* all files to **\Old** retaining their original directories.
   - "All files" means files you control and want to organzie
   - E.g., **\Dissertation** and files are moved to **\Old\Dissertation**
2. Make a *backup* of **\Old** before proceeding.

Create the new directory structure

1. Create the new directory structure on the drive where you want your files.
   - I call this drive **\New**
   - It might by the D: drive, your Dropbox sync, or Box sync folder, etc.

Move from \Old to \New

1. *Move* (not copy) files from **\Old** to their new location.
   - Do this slowly for a few directories at a time.
   - Have a plan for you you will work
2. You can use files in **\Old** but:
   - Do not add new files to **\Old**
   - Do not change existing files
3. New work is saved in **\New**
   - Even if you did not move project files from **\Old** to **\New**, place new work in **\New**.
   - Better yet, move the entire project to **\New**.
4. Eventually, odds and ends will be left in **\Old**
   - Move these to **\New\- To clean\Old**
   - Ignore these till you need them.
   - They are in the right location in your new structure.

Using the cloud

Costs

1. Is the cloud secure enough for your work?
2. You might need to pay for it, but it may be cheaper than buying hardware.

Benefits

1. Files are off site with copies at multiple physical locations.
2. You can sync files across multiple machines.
   - o Changes on each machine are synchronized across all machines.
   - o You can carelessly delete files on multiple machines!
3. Check how long deleted files are saved
   - o Files are restored one at a time which can be tedious
   - o Test restoring a file before you have need to recover a file
4. You can share files without using attachments.

How I use Dropbox with automatic syncing

```
D:\Dropbox
   \-  History starting 2015-04-05
      \1. Serious work goes here
      \2. Box holds snapshots
      \3. Sync to laptop, home, office computers
   \Active
   \Admin
   \Bookshelf
   \...
   \Scott Long shared with
      \Bussberg, Nic
      \Manago, Bianca
   \...
```


Directories for collaborative projects

1. How does collaboration affect you directory structure?
   - o Consistency is more critical
   - o Common standards are necessary
2. This is discussed later.

Overview of DAM

1. Tools: a dual pane file manager makes this vastly easier.
2. Structure: plan your directory structure; let it age; test it; commit to it.
3. File naming: decide on naming templates.
4. Move into the new structure: do it slowly with a plan.
5. It takes time before the new structure become intuitive. Persist!
6. Avoid tweaking the structure and naming templates.
7. Why bother?

Part 4: Protecting files

WFDAUS Chapter 8.

Will you lose a file?

Pr(lost file) = [**Pr(lost file | backup plan used)** X **Pr(backup plan used)**]

+ [**Pr(lost file | plan not used)** X **Pr(plan not used)**]

Cost of losing a file

E(cost of loss) = [**Pr(lost file)** X **(cost of losing file)**]

+ [**Pr(file not lost)** X **(cost of protecting file)**]

Causes of data loss

Deleted and lost files

1. A file was mistakenly *deleted*.
   - o *Recycle Bin* in Windows works sometimes...
   - o *Trash* and *Time Machine* in Mac OS X are imperfect
   - o I prefer **`\- Hold then delete`**
2. You cannot find a file.
3. Backups are missing or corrupt.
   - o How long are backups kept?
   - o Where are they stored?
   - o Do they work?

Is the LAN backed up?

How often are backups made and how long are backups kept?

Corrupted files

1. A file corrupt when bits are stored incorrectly due to a write error, deterioration, or a virus.
   - o A single incorrect bit can make a file unreadable.

Solutions

1. Use *bit comparison* to verify that the copy is *exact*.

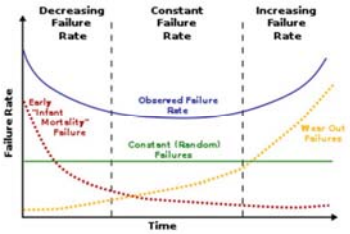
   Source: `00100101010010101001010001010000101010101 10...`

   Copy: `00100101010010101001010001010000101010101 00...`

2. Eject USB drives.
3. Do not move your laptop when the disk is spinning.
4. Virus software.

Hardware failures

1. Electronics are amazingly robust and amazingly fragile.
2. Failure rates follow a bathtub curve.
3. With 25,000 drives, Backblaze found 74% of one model failed in 3 years.
4. If a drive develops a noise, replace it immediately; otherwise replace it every five years (standards may differ for SSD).



en.wikipedia.org/wiki/Bathtub_curve

Obsolete media and formats

Hardware

1. Digital data can be more fragile than paper and are easier to misplace.
   - o An 80-year-old snapshots is in an album, but the hard drive with last summer's JPGs crashed and the pictures were lost.
   - o I have the binder with printouts from a 1978 paper. Where are the unprinted log files from a paper published two years ago?
   - o Ironically, digital files can also be harder to completely eliminate.
2. Technology changes and old technology disappears.
   - o It cost $2,000 to read a file from a backup tape!
   - o Zip drives disappeared in just a few years.
3. *Use multiple devices and migrate files from old to new media.*

Files formats

1. Old formats might not be supported by new software
   - o OSIRIS has disappeared.
   - o 508K volumes at the British Museum in obsolete formats.
2. There is no easy solution since archival formats have not been established.
3. *Store datasets in multiple formats.*
   a. SAS Transport format (the FDA standard)
   b. Stata format
   c. ASCII
   d. XML
   a. Maybe more formats
4. Put important datasets in an archive like ICPSR.
5. Save text as docx, rtf, txt or tex, and PDF.

Recovering lost files

1. Someday you will lose a file.
2. Attempts to recover a file or entire drive can make things worse.
3. When a file is lost *PAUSE!*
4. If in doubt, ask for advice. If you get advice, don't trust it completely.

Examples of lost data

LISTEN. Do you hear it? The bits are dying. -- Schwartz 2008 NY Times

British Museum

508K volumes in obsolete formats

ICPSR Data Archive

A hose above the server room

Toy Story 2

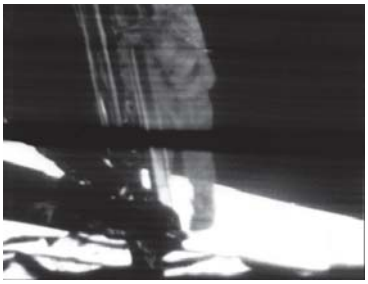
Saved by a maternity leave

Moving apartments

A grad student was moving and computer and backups were stolen.

NASA: one giant blunder

Neil Armstrong's walk on the moon on July 20, 1969 and the lost moon tapes.



"*fuzzy* gray *blob* wading through an inkwell"

Looked like this!

Pink Floyd, Dark Side of the Moon (1973)

Protecting your data

The *universal aptitude for ineptitude* makes any human accomplishment an incredible miracle. --Dr. John Paul Stapp

1. If you expect the worst, you might prevent it.
2. When it comes to preserving your work
   - Expect things to go wrong.
   - Expect to delete an important file at the worst possible time.
   - Expect a water leaks above your computer.
   - Expect that it will be worse than that
3. Preserving files is hard without a plan
4. Preserving files is easy if you have a plan and follow it

Issues to consider when creating a backup plan

1. Cost of data loss
2. What to preserve
3. Types of protection and how to protect files
   - Tools and technology
4. Documentation of what is backed up
5. Tools and technology

Options for storage

1. *Enterprise storage*: Scholarly Data Archive (SDA), LANs...
   a. How long backups are kept?
   b. Are files kept when you leave?
2. *Hard drives: On line storage*
   a. After 5 years
      - Copy the old drive to the new with bit verification.
      - Retire the drive or keep it for deep backup (stored elsewhere).
   b. Write the birth date on the physical drive. Add the date to the drive's history directory.
3. *Cloud storage*
   a. This can be cheaper than buying multiple physical drives.
   b. Will a vendor go out of business or change policies?
   c. Is the data private?
   d. Other risks?

Comparing files: are copies exactly the same?

1. Bit verification is needed to verify copies are exact.
2. Built into Total Commander for Win;
   possible with Mac's Unix shell or ChronoSync Express (I am told).

Compressing files

What is compression?

1. Let: *symbol*(#) mean # repetitions of *symbol*.
2. You have this data which requires 44 symbols:

   11111111111111111111000000000000000000002333

3. You can represent it with 18 symbols:

   1(20)0(20)2(1)3(3)

Combining files

1. More importantly compression combines many files into one.

```
\Stata11 (15,924 files; 145 folders) 325,594,198
Stata11.rar                          115,657,681
```

How to make compressed files?

1. I use WinRar (www.win-rar.com) in Windows
2. Check with IT staff for programs where you work

Posting and file preservation

1. I use to change, rename, or relocate files that were "backed up"
   - o I *organized after the fact*.
   - o After changes I needed to back up the files again.
   - o This created problems
2. What about the files I backed up before with different names?
   - o If I delete them, old script files will not work and my project diary points to a file that no longer exists.
   - o If I keep them I have files with the same information but different names.
3. The solution is *posting*

   Rule 1: After a file is posted, it never changes.

   Rule 2: If I find a mistake in a posted file, create a new file with a new name, but never change the original file.

This is discussed in detail later.

Considerations in recovering files

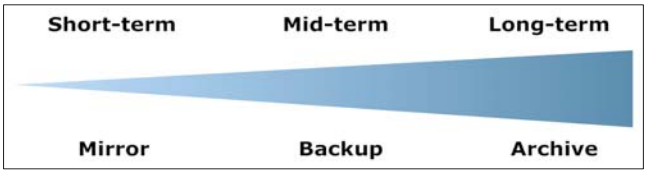
Active files that you are working on.

Posted files that you are done working on.

	Active files	Posted files
How long to preserve file?	0-5 years	5+ years
How difficult to preserve?	Trivial	Easy with a plan
Concern with media/format?	Minor	Some
How to recover missing file	Redo recent work	Redo old work
Time to recover missing file	Not much	Weeks or years

Types of protection

How long are files to be preserved?



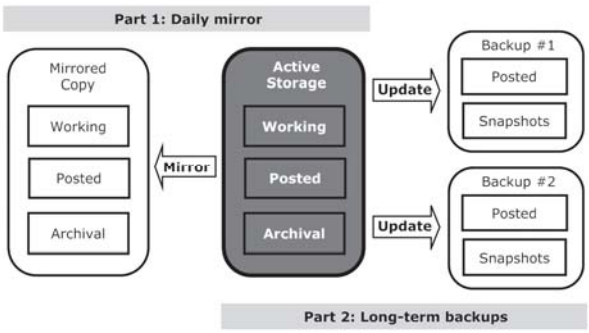
What type of protection do you need?

Mirror: Real-time duplicate copy

Backup: Multiple copies saved in multiple locations

Archive: You can't do this; Google "ICPSR data preparation"

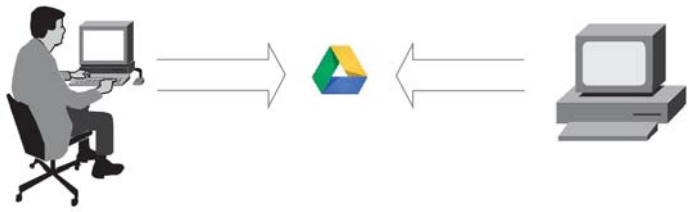
Flow of file preservation



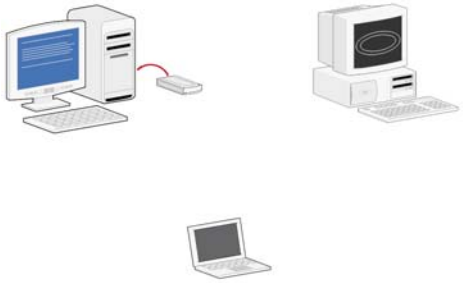
Active storage: mirroring files as they change

1. Risks are hardware failure, power surges, viruses, theft & accidental deletion.
2. The solution is a *mirror* that duplicates files at multiple locations.

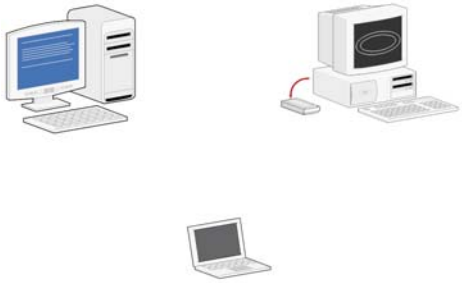
Mirroring with the cloud



Mirror to external drive



Mirror to external drive



Snapshots: short term preservation

1. Shapshots are folders and files compressed into a single zip or rar file.
2. I take *snapshots* of active files associated with a project.
   - Before a trip I take a snapshot of what I'm working on
   - Before moving a project to **\Inactive**
3. The file is named date keyword description.rar and saved to **\Snapshots**

```
\Snapshots
    2014-06-04 WF2nd work on hold.zip
    2013-03-01 EPSL to clean.zip
    2015-06-03 WFiu2015 precleaning.zip
```

4. I save these files on a local computer or on offline storage
5. I hope I never need them and delete them when a project is completed.

Long-term backups

Selected files are stored more securely than mirrors or snapshots.

What to back up?

1. You need to know what files should be backed up.
   - If don't know what to backup, you probably won't do it.
2. The easiest method I know is post files in **\Posted**
   - These files are never changed
3. When a project is complete, move the project's posted folder to **\Vault**
4. Only these files that need to be backed up and only backed up once!

How to make backups

1. Make *verified* backups.
2. Store them in *multiple locations*.
3. Backed up files should *never change*.

Example of backups using the posting principle

Locations

Local active: where I do work

Local backups: backups located on my computer

Offline backups: backups on offline storage

Example

Local active		Local backups		Offline backups
\EPSL				
\Posted	→	\Vault\EPSL\	→	\Vault\EPSL\
\Work	→	\Snapshots\EPSL\	→	\Snapshots\EPSL\

Keeping track of what is backed up

For mirrors

1. You simply decide which folders to sync across machine.
2. Regularly check that the sync program is working.
3. Reboot at least once a week.

For posted files

1. Posted files go to **\Posted** or **\Vault**
2. The project's history records what was done.

```
\-  History starting 2009-10-03
    \2009-10-03 initial snapshot to SDA
    \2010-01-23 snapshot to EX21
    \2010-07-02 snapshot to SDA before ICPSR
    \2010-08-28 snapshot pre move to Box
    \2011-01-16 posted files to SDA
```

Tools for backups

My backup plan (more or less)

Storage locations

1. Active drives — Office; Home
2. Synced cloud storage — Cloud ↔ Office ↔ Home
3. Offline storage — Offline

Classes of files

1. Vault and Posted — Posted files
2. Active — What I'm doing now
3. Software — Customizations for installed software
4. OS — Operating system and software
5. Mundane — Things I don't need to preserve

Preserving Active Files

1. **Cloud Sync** provides continuous mirrors of active work.

   Office ↔ Cloud ↔ Home
2. **Snapshots** are made at strategic times in a project such as:

   Active\EPSL\ → Offline\Snapshots\2015-11-29 EPSL preRR.rar

Preserving Vault Files

1. Vault files are copied offline:

   Vault\project\ → Offline\Vault\EPSL\EPSL 2015-11-29.rar
2. Cloud Sync is another copy of vault files (if I have enough cloud storage):

   Office\Vault\ ↔ Home\Vault\ ↔ Cloud\Vault\

Preserving Other Files

1. **OS and software**: Might copy Offline
2. **Mundane**: Might copy Offline

Part 5: Using Stata

Goals

1. Key ideas for using Stata
2. General overview:
   - Interface
   - Do-file editor
   - Command syntax and basic commands
   - Ways to run commands
   - Working directory

Essential knowledge for an effective WF

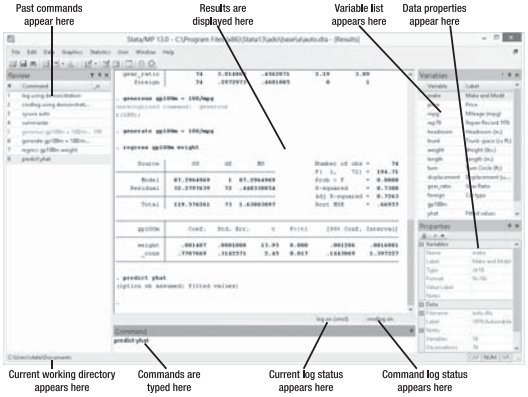
1. Robust script files
2. Using the *working directory*
3. Macros, returns, and loops for automation

Learning Stata

You learn Stata by using Stata and trying the examples from lecture.

1. Stata on YouTube: *youtube.com/user/statacorp*
2. Stata resources: *www.stata.com/support*
3. Stata Press: *www.stata-press.com*
4. The PDF manuals are installed with Stata.
5. *R for Stata Users* by Muenchen & Hilbe is a PDF download from the library.

Stata interface



Command window

- Enter to run the command; it is moved to review window.
- Page Up and Page Down retrieves prior comamnds; then edit and run.

Results window

- Echos command
- Shows results

Review Window

- Commands are echoed
- Double-click to immediately execute a command
- Single-click to paste to command window; revise and submit with Enter

Variables Window

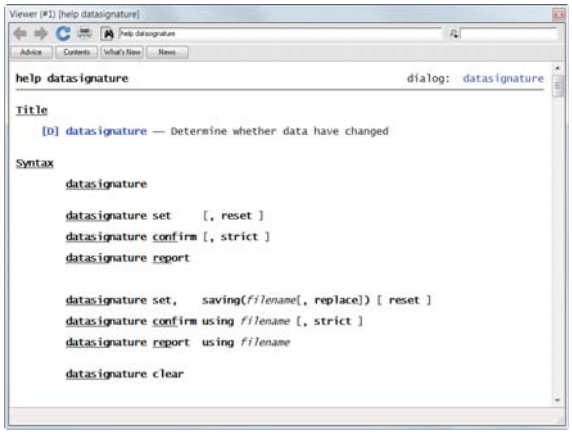
- Double click on a name to paste to command window

Copy and paste (shortcuts ctrl/cmd-c; ctrl/cmd-v)

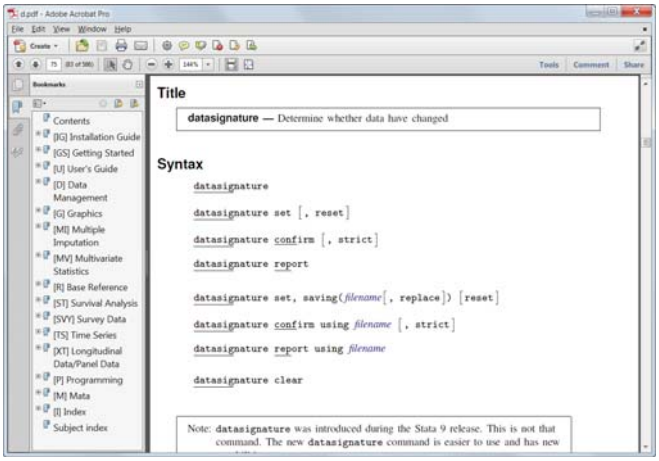
- Highlight and copy text from the Command Window or the Results Window

Paste into your editor

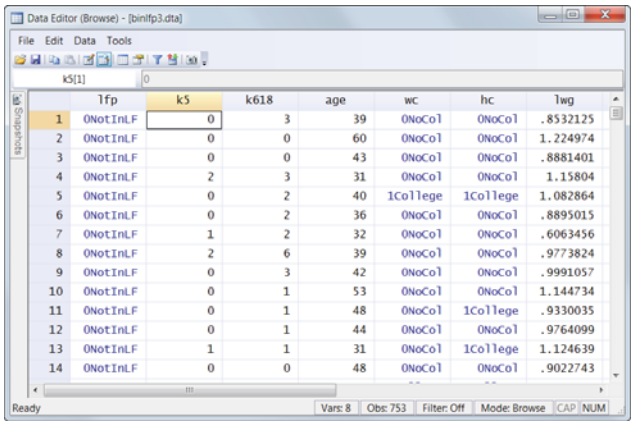
Type help from the command line



To open the manual click [D] datasignature



Stata data viewer: enter browse



Working directory: critical concept for CWF

1. By default, Stata looks for files in the working directory.
2. You change working directories with commands or from the menu.
3. To determine your working directory check the interface or

```
. pwd
d:\Dropbox\active\WF stat 2015\work
```

4. To change the working directory (use quotes if there are spaces):

```
cd d:\Groups\work
```


5. To list files in the working directory:

```
. dir
<dir>   4/14/08 13:25  .
8.7k    4/14/08 13:24  wf-acjob.dta
0.2k    5/19/12 11:13  wf-test02-cd.do
:::
```


Command syntax

command things cases , options

command	Action to take
things	Things on which action is performed
cases	Observations used by action
options	Options controlling how action is performed

Example

```
tabulate hc wc if age>40, row
```

Basic commands with selected options

use: load a dataset

use *filename* [, clear]

dotplot: create histogram

dotplot *varname* [if] [in]

summarize: descriptive statistics

summarize [*varlist*] [if] [in] [, detail]

codebook: labels and descriptive statistics

codebook [*varlist*] [if] [in] [, compact]

tabulate: twoway table

tabulate *varname1 varname2* [, options]

tab1: oneway table

tab1 *varlist* [, missing nolabel]

regress: LRM

regress *depvar* [*indepvars*] [, noconstant]

logit: logit model

logit *depvar* [*indepvars*] [, noconstant or]

log: save result to file

log using *filename* [, replace [text|smcl]]

capture: keep errors from stopping program

capture *command-name*

twoway scatter: scatter plot

twoway scatter *xvar yvar* [, jitter(#)]

Three ways to run commands

Command Window

Type commands in the Command window. Press Enter to run them.

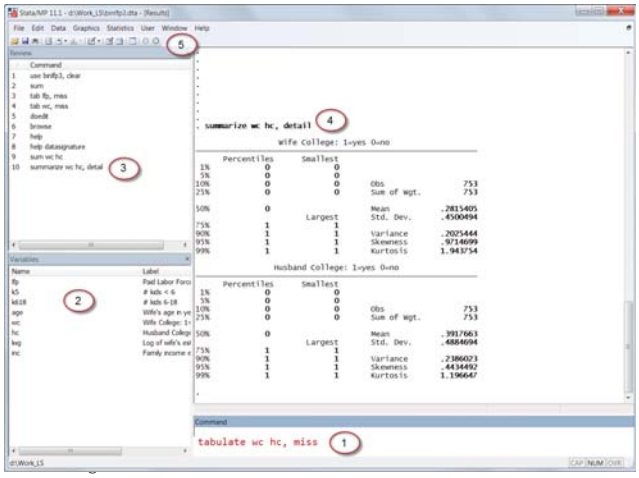
Dialog boxes

Construct commands with dialog boxes and click Submit to run them.

do-files

Run text files that contain your commands.

Run from Command window



Recording interactive work

1. Open a log file

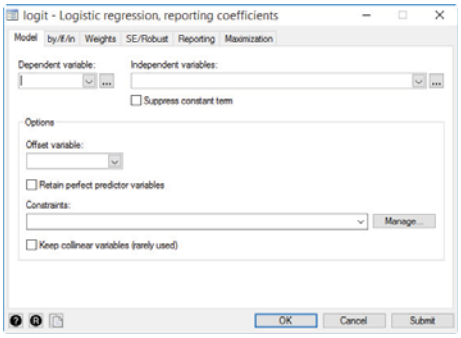
```
log using mywork, text replace
```

2. Enter commands from the Command window.
3. Close the log file when you are done.

```
log close
```

4. You can use this information to create a do-file, considered below.

Run command from with a dialog box



1. Easy to learn and slow to use.
2. Useful for finding options and commands.
3. After submit, command are shown in results.

Run commands in a do-files

1. A simple do-file **wfx-intro.do**:

```
capture log close
log using wfx-intro, replace text

use wf-lfp, clear
summarize lfp age

log close
```

3. To execute the commands,

```
do wfx-intro
```

4. The Results window and the text file **wfx-intro.log** contain the results…


```
------------------------------------------------------------------------------
      name:  <unnamed>
       log:  d:\Dropbox\Active\wfclass\work\wfx-intro.log
  log type:  text
 opened on:  16 Jun 2017, 10:27:33

.
. use wf-lfp, clear
(Workflow data on labor force participation \ 2008-04-02)

. summarize lfp age

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         lfp |        753    .5683931    .4956295          0          1
         age |        753    42.53785    8.072574         30         60

.
. log close
      name:  <unnamed>
       log:  d:\Dropbox\Active\wfclass\work\wfx-intro.log
  log type:  text
 closed on:  16 Jun 2017, 10:27:33
------------------------------------------------------------------------------
```


Why use script files?

1. You have a record of what you did.
2. You can easily re-run commands:
   - To revise them
   - To allow others (e.g., journals) to reproduce your work
   - To create new scripts based on old script
3. Do-files make documentation easier. The project diary chronicles do-files.
4. They are faster
   a. Use a text editor to write them
   b. Use templates
5. Journals are starting to require the submission of do-files.

Stata's do-file editor

Since Stata 12, the Stata editor is adequate for serious work.

Three ways to open the do-file editor

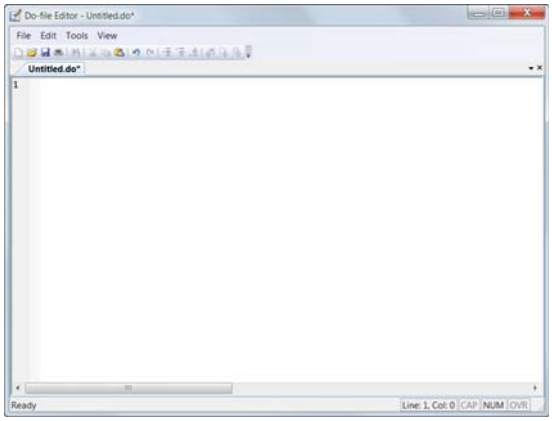
1. Enter **doedit** from the command line.
2. Select *Do-file editor* from the Windows menu.
3. Click 

Features of the editor

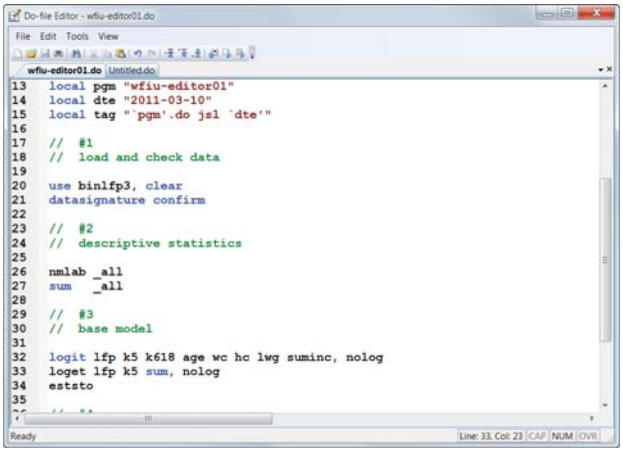
1. Syntax highlighting
2. Multiple do-files
3. Auto-indentation
4. Bookmarks
5. Execution of commands from within editor

Opening the editor…

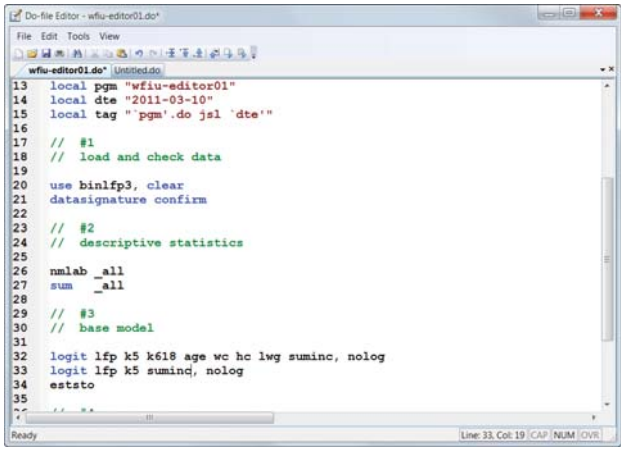
```
. doedit
```



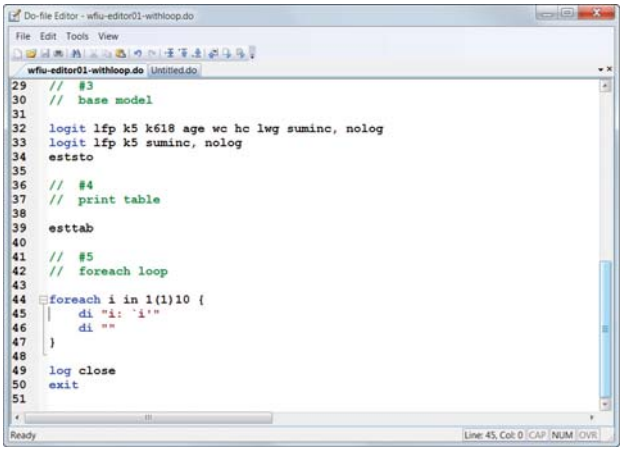
Syntax highlighting: notice the error in line 33



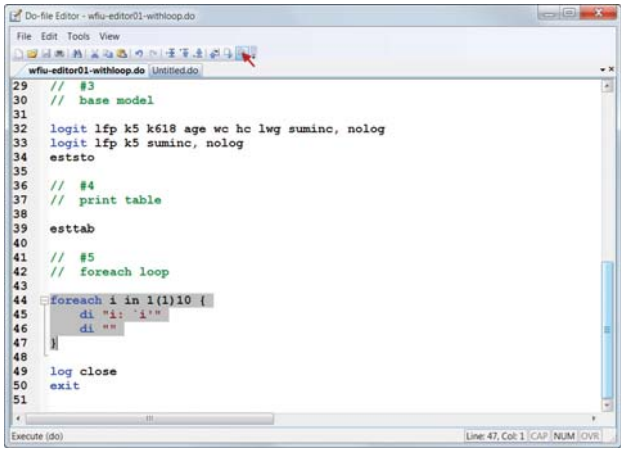
The error is fixed line 33



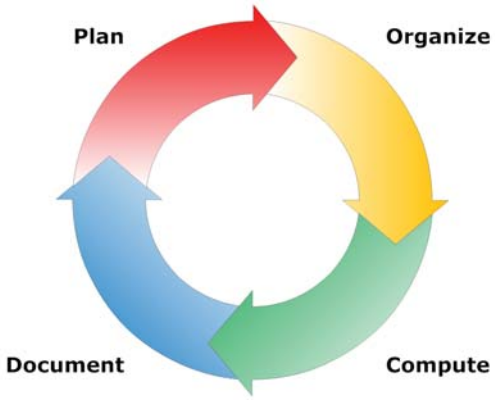
A loop is marked



You can highlight code to be executed



Part 6: Plan, organize & document



Overview of POD

WFDAUS pages 11-46.

1. The core tasks
   - **Planning** is strategic, focusing on broader objectives and priorities.
   - **Organization** is tactical, developing structures and procedures to complete your plan.
   - **Documentation** is book-keeping; recording who did what when.
   - **Computing** gets the numbers out of the computer.
2. The temptation is to jump into computing without planning, organizing, or documenting (POD).
   - Mainframe computer imposed time for POD.
   - Today, computing rarely imposes delays, so we compute instead of POD.
3. *Time spent on POD will improve your workflow and save time.*

Do you have time for POD?

1. We resist POD, but:
   - It takes longer to search for a lost file than to create a directory structure that prevents losing files.
   - More time is spent deciding who has the last draft of the paper than using naming conventions that avoid the problem.
2. You compute every day. Do you POD every day?
3. At each stage of your project
   - Assess your plan
   - Update your organization.
   - Document what you have done.
4. *POD pays huge dividends* in reproducibility, accuracy, and efficiency.

Multi-tasking and layering

1. Being "connected" interferes with POD.
2. Self-imposed interruptions let you avoid making hard decisions.
   - How should I organize my dissertation? Hmm, I think I'll check my e-mail.
3. Rotating tasks is helpful if it is not a way to avoid what is hard.
   - If programming isn't going well, I write the description of the sample.
   - If writing is going in circles, so I might update my bibliography.
   - Sometimes going in circles is what needs to be done!
4. A type of multi-tasking, called *layering*, can be effective.
   - I like to organize when I'm considering major decisions in a project.

Planning

1. It begins with *broad considerations* of goals.
   - "A goal without a plan is a dream." -- Antoine de Saint-Exupéry
2. *Anticipates* what needs to be done and when.
3. It *guides efficient completion*.
4. Planning is a reminder to stay on track.

The gold standard for planning

Blau and Duncan (1967) *The American Occupational Structure*

- All analyses were specified 9 months before output was received.
- The book was written based entirely on those analyses.

Dimensions of planning

Three dimensions of planning (Oliveira and Stewart, 2006: 59-70)

1. **In the large** *Objectives* of the work
2. **In the middle** *Manageable tasks* within objectives
3. **In the small** *Necessary details* to accomplish tasks

Things to plan

General goals and publishing plans

1. Plan papers and presentations to prioritize tasks and avoid "supply chain" problems.

Scheduling

1. Create a *time line* with target dates for major stages of the project.
2. If you fall behind, revise the plan.
3. Decide what is urgent and what is important.

Datasets

1. What data are required? Do you need permission?
2. How many panels? Countries? Anticipating complexity prevents problems.
3. What variables are needed? Avoid repeatedly extracting forgotten variables.

Variable names and labels

1. Plan names and labels: bad names and labels haunt for years.
2. Anticipate new variables and plan their names.

Choice 1:	myvarP1	myvarP2	myvarP3
Choice 2:	P1myvar	P2myvar	P3myvar
Choice 3:	myvarp1	myvarp2	myvarp3

3. If software restricts names, plan for this.

Missing data

1. What types of missing data are expected? How will types be coded?
2. How will multiple imputations be handled?

Data Analysis

1. What analyses are anticipated?
2. What software is needed? Is it available?
3. What restrictions does your software impose?

Documentation

I have joined projects where the computing was "done", but there was no documentation. We had to start over.

1. What documentation is needed?
2. When will you write documentation?
3. Without a plan for what to document, you probably won't do it.

Preserving files

1. How are backups made? When? Where? By whom?
2. How long do you need to preserve files?
3. Which files need to be preserved?

Summary on planning

Work. Finish. Publish. -- Michael Faraday

I have two kinds of problems, the urgent and the important. The urgent are not important, and the important are never urgent.
-- Dwight D. Eisenhower

1. Planning is a reminder to publish. Working is not enough.
2. Planning prevents work from being *interrupt driven* and *paged to death*.
3. Take time to plan -- turn your devices off.

Paper planning: Pavalko, Gong, & Long 2007

This example is *one way* to write an effective plan.

Terms

CWH: cohort, work and health

LFP: labor force participation

Employment status: Employed or not employed

Employment category: *Reason* for having a given employment status (e.g., not looking for a job, can't find a job, too ill to work)

NLS: National Longitudinal Survey

Planning in the large

1. Tie your objectives to *the potential of your data*.
2. It does little good to make a plan that requires data that you do not have.
   - A common disease of dissertation proposals
3. Do not dwell on details--yet.

The CWH large plan

1. The CWH paper examined the relationship between LFP and health for women.
2. The abstract describe the goals of the paper:

> Social change in women's labor force behavior in the past half century has been well documented. Coinciding with dramatic increases in women's labor force participation are increases in percentages of women with young children in the labor force.... In this paper we assess whether social change in women's employment has implications for women's physical health. While health benefits of employment for women are fairly clear, we know little about whether these effects have changed in concert with the changes in women's labor market experiences.

3. Writing the abstract *first* makes you focus on the important things.
   - Later, you will be glad that you do not have to write it!
4. Our earlier research found patterns we wanted to explore further.
   a. *Cohort variation by employment status versus employment category*. There is little cohort variation in health by employment status, but there is interesting variation by employment category.
   b. *Employment categories and health*. Relationships between employment category and health that vary between the earliest and latest cohorts.
5. Motivated by these findings, we had these research questions.
   a. *Employment status and health*. Are employed women healthier than non-employed women? Does this relationship vary by birth cohort?
   b. *Employment categories and health.* Does the effect of non-employment on health vary by the reason for non-employment? Is the health of women who are non-employed to care for family different than the health of employed women? Does this relationship vary by cohort?

   c. *Explaining change in health.* Are the relationships between employment and health due to (i) changes in the effects of employment on health? (ii) Changes in the distribution of women among employment categories? (iii) The greater selectivity of the 1991 sample compared to the 1971 sample?
   d. *Additional controls*. Do these relationships persist after controlling for variables such as workforce commitment, hours worked, and type of employment?
6. *Planning in the middle* comes next.

Planning in the middle

1. Middle level plans translate broad objectives into distinct analytic tasks and considers how each task can be divided among script files.
2. New tasks emerge when initial analyses show that things are more complex than anticipated.

CWH in the middle: overview

1. **Describe the sample and variables.** Compute descriptive statistics to describe the sample and the variables. Describe the distribution of health by cohort and employment status.
2. **Model health by cohort, employment status, and controls.** Estimate cohort differences in health by employment category after controlling for demographic and other variables. Compare alternative modeling approaches.
3. **Sensitivity analyses.** Determine if findings can be explained by sample attrition or measurement problems.
   - If results are sensitive to minor variations in model specification, they are unlikely to be replicable.

CWH in the middle: specific tasks

- **Red** is for data management; **Blue** is for data analysis.

cwh01: **Descriptive statistics.** Basic descriptive statistics

cwh02: **Compare count models for number of health limitations.**

cwh03: **Logit model for having any limitations.** Due to large percent 0 in outcome, run logit of no limitations or any.

cwh04: **Hurdle model for number of limitations.** Results from the logit and ZIP lead us to a hurdle model. After reviewing these findings, we returned to data management to add variables that made it easier to estimate models with interactions and to include data from the 1971 panel:

cwh05: **Data management.** Add interaction variables.

cwh06: **Data management.** Add data from the 1971 panel.

cwh07: **Count models with 1971 data included**.

cwh08: **Hurdle model using alternative parameterizations**. Since this model had best fit and made most substantive sense, we tried different parameterizations to make it easier to test hypotheses.

cwh09: **Sensitivity analysis of hurdle model**.

Using these results, we wrote the first draft

1. After discussing draft, we planned new analyses with new variables.
2. New analyses could have been added to the tasks above, be we created new tasks organized around tables in the revised paper.
   - When a paper is almost ready to circulate, having tasks organized around tables/figures is convenient for making changes later.
3. The results in the revised paper were all from these tasks:

***cwh10*: Data management**. Add additional variables.

cwh11: **Descriptive statistics for Tables 1, 2, and 3**.

cwh12: **Hurdle models and predictions for Figures 1 through 5**.

cwh13: **Supplementary analyses with the hurdle model for Table 4**.

4. We submitted the paper, received an R&R, added variables suggested by reviewers, refined the coding of other variables, and updated our figures and tables. Since the analyses in the paper were included in tasks **cwh10** through **cwh13**, revisions were simple.

***cwh14*: Data management.** Add work and smoking variables; revise some operational definitions.

cwh15: **Re-estimate models and create plots**.

cwh16: **Estimate additional models for Table 4.**

5. The paper went through two more revisions before it was published.
   - The project was on hold for months at a time while we waited for reviews.
   - Having the work divided into tasks and the provenance of all results documented made it easy to pick up the work where we had left off.

Planning in the small

1. Planning in the small implements the tasks from the middle level plan.
   - The *nitty-gritty details* (Oliveira and Stewart 2006:61)
2. The details are critical, but it is easy to get lost in these details.
3. Decide which variables to use, how to code analysis variable, and which commands to use (e.g., `logit` with `ztp` or `hhplogit`?).
4. Periodically step back to review the larger objectives.

Collaboration

1. Collaboration makes it more difficult to have an effective workflow.
   - Collaboration is a "stress test" for your workflow.
2. Everyone should understand the broad outlines of the project.
   - Collaborators bring different skills but need a common understanding of each person's expertise and perspective

Goals in collaborations

1. Agree upon standards and enforce them
2. Coordinate work; make action items explicit
3. Maintain good will, cordiality, and respect

Complaining about collaborators

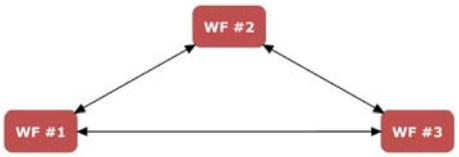
1. Gary King suggests that you could probably write another paper in the time you take complaining!

Why is it so hard to coordinate multiple WFs?

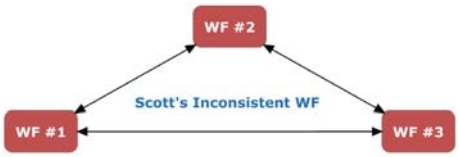
Coordinating multiple WFs

Coordinating multiple WFs

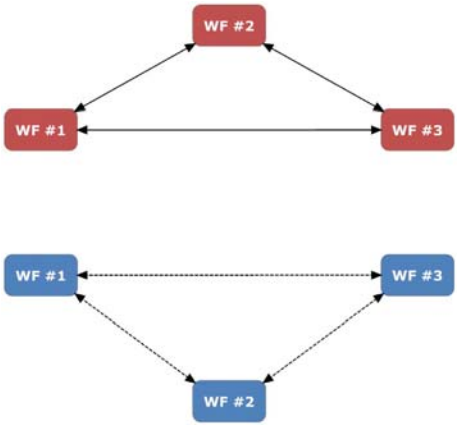
Coordinating multiple WFs



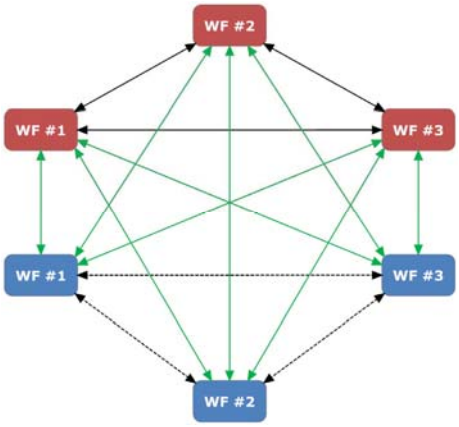
Coordinating multiple WFs



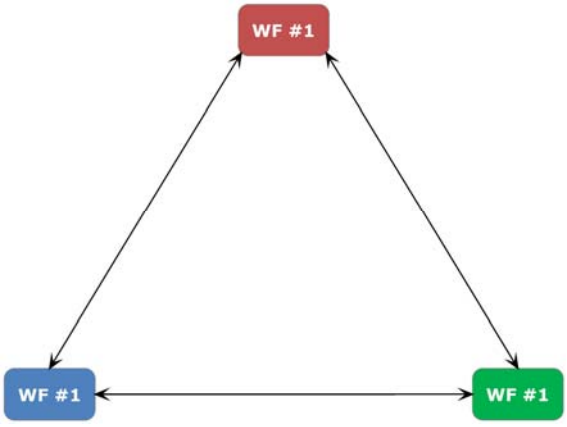
Coordinating multiple WFs



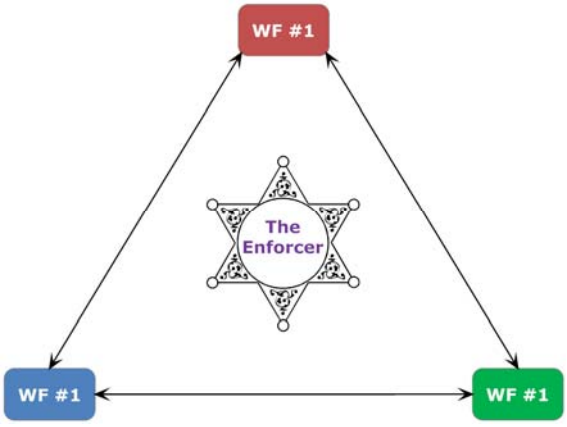
Coordinating multiple WFs



Coordinating multiple WFs



Coordinating multiple WFs



What is the enforcer?

1. Who should it be?
2. What authority does this person have?
3. What are the remedies for non-compliance?
4. If you are working alone, who is your enforcer?

Collaboration is harder when…

1. Individuals do not have an effective workflow.
2. Collaborators have different, un-coordinated WFs.

Collaboration is easier when…

1. Workflows are coordinated with agreed upon procedures.
2. Planning prevents avoid duplication and gaps.
3. Organization increases efficiency and accuracy.
4. Documentation keeps track of who did what when and what was not done.
5. Communication is regular; action items are clear; good will prevails.

Collaboration directories on shared space

\Mailbox: to exchange files

```
\Mailbox
   \To Fred
   \To Scott
```

\Private: files you aren't ready to share but others could check

```
\Private
   \Fred
   \Scott
```

\- To shelve

1. The data manager/enforcer verifies files before moving them to the appropriate location.
2. If you are working alone, you are the data manager for yourself.

Organizing

1. Organization involves deciding
   a. What goes where?
   b. What should you name it?
   c. How will you find it?
2. Planning helps organizing
   a. Plans for the *broader objectives* help you define how complex your organization needs to be.
   b. Plan for *specific issues*, such file naming, help you complete the work accurately and quickly.
3. Organization makes documentation easier.
   a. It is easier to explain what files are.
   b. You don't have to document what is clear from your organization.
   c. You know where things are located.

Principles for organization

Start early

Organization is contagious. Disorganization is also contagious.

Consistency

1. Consistency pays dividends.
2. Use the same organizational methods for all projects.

Simple, but not too simple

1. More elaborate organization is not always better.
   - I prefer a more elaborate organization at the start.
   - You might prfer a simple structure that is expanded thoughtfully as needed.
2. Organizing without a plan for doing it makes things worse.

Organizing digital assets

1. Can you find it?
2. Are name clear and unique?
3. To search by content, you need distinct keywords *inside* files.

Templates and exemplars

1. Examples of how things should look.
2. Shell documents to populate.

Tools for organization

Software

File manager, macro program, and text editor make things more efficient and uniform.

Cyberinfrastructure

Cloud storage makes sharing files and authoring documents easier.

Names & metadata are critical for organization

A project keyword

1. Pick a mnemonic for your project.
   - CWH: Cohort, work, and health
   - SDSC: Sex differences in the scientific career
   - PASL: Paul Allison/Scott Long
   - WF20xx: Class taught in 20xx
2. *Short* to keep names from getting too long since they are part of other names.
3. *Unique so you can search for it.* Capitals might help.
4. The project keyword should be included *in names* and *inside files*.

E-mail

Folders

```
\Active
\Consulting
\Information
\Junk
\Personal
\Projects
\Service
\Teaching
\Todo
```

Subject lines

1. Use keywords in subject line such as *WF2017: <describe content>*
   - E-mail to me about class should have "WF20xx" in the subject
   - This allows scripts, sorting, and searching to find related e-mails.

Tips for e-mail

1. E-mail is not private. It gets forwarded.
2. Be careful with bcc, cc, and avoid reply all.
3. Use the subject line to indicate content and purpose.
   - Do not reply with a new topic but the same subject line
4. One subject per message when possible.
5. Keep messages to the point. Summarize long discussions.
6. Be informal, not sloppy.
7. Would personal contact be better?
8. Delete e-mails you don't want!

Documentation

If you were hit by a bus, would a colleague be able to reconstruct what you were doing and keep the project moving forward? -- Terry White

1. Insights from months of work make it *harder* to write documentation.
2. Details now obvious are later obscure.
   - Was cohort 1 the youngest cohort or the oldest?
   - Which is the latest version of a variable?
   - What assumptions were made about missing data?
   - Is **ownsex** or **ownsexu** the variable for the 1st question asked?
   - Does JM refer to Johnny Miller or Janice McCabe?

Do it today

1. Writing documentation takes time, but eventually saves time.
2. Without documentation
   - Advantages of planning and organization are reduced.
   - Reproduction is harder or impossible.
3. To keep up with documentation make it a regular part of your WF.
   Option 1: Have a fixed time for reviewing documentation.
   Option 2: Tie documentation to major events in the project.

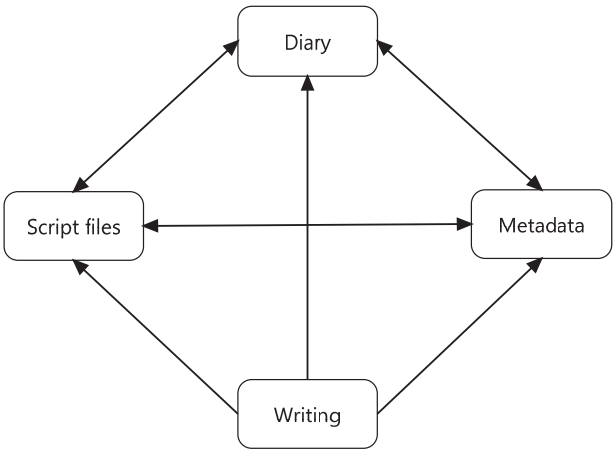
Document what is needed for reproducible results

Data sources	Which release of the data did you use?
Data decisions	How and why were variables created and cases selected the way they were? Why did you dichotomize at 2 not at 3?
Statistical Analysis	What steps were taken in the data analysis, in what order, and what guided those analyses? If you explored an approach but did not use it, keep a record of that as well.
Software	Details on software and computing environment used for analysis.
Storage	Where are the results preserved?
Ideas and plans	Ideas for future research and tasks to be completed should be documented. What seems obvious today isn't next year.
Progress	Document what has been done.
Other things	Document everything that is essential for reproduction.

Levels of documentation

1. Multiple types of documentation
   - Diary
   - Script files
   - Metadata
   - Provenance in presentations
2. Levels should *reinforce, not duplicate*
   - Links among types make documentation efficient and accurate.
   - Duplication wastes time and risk having conflicting information.

Overview...



Project diary or research log (details below)

1. The diary is the cornerstone of documentation.
2. It chronicles who did what, when, what files, and where things are.
3. It contains pointers to other information.

Metadata for datasets and variables (details below)

1. Metadata describes other data.
2. It resides within the data structure so it "follows the data".
3. Metadata can describe the datasets and the variables.
4. It can provide links to the research diary.

Script files (details below)

1. Your project diary points to script files.
2. Comments in script files document what is done.
3. Balance the details in the project diary and comments in the do-file.

Documenting your documents

1. Provenance of results
2. Internal documentation is "metadata".
   - Author names/e-mails
   - File name
   - Date created
   - Who has control of the document
3. Without this information it is easy to:
   - Revise a document and forget to change the internal date, then save the file with the same name
4. Add a history section to the document.

   History

   2011-03-23: S Long created couple2 2011-03-23.docx after JHSB review.

   2013-04-18: S Long final update for resubmission.

#1 project diary or research log

Goal 1: Keep your work on track

Include research plan to help you set priorities and complete work efficiently.

Goal 2: Let you move between projects and manage interruptions

When you return to a project, the project diary helps you pick up the work where it ended without wasting time or forgetting critical information.

Goal 3: Facilitate reproduction

Record what was done and the files that were used.

What should your diary look like?

1. As long as the goals are met, your diary is *good enough*.
2. Many formats work: bound books, loose leaf notebooks, computer files

My project diaries contain

1. Broad description of what I have done, why I did it, when, and how.
   - Do-files and metadata have the specific details.
2. Research plan with a "to do" list that evolves into the "what was done list."
3. Information on what I decided not to do and why.
4. Ideas for new projects so I don't get off track working on them now or forget about them later.
5. Comments and emails from collaborators and colleagues.
6. People tell me "after <some disaster>, I started including information on <*whatever*> in my diary."

For example...

Old format for project diary for FLALT

```
First complete set of analysis for FLIM measures paper

f2alt01a.do - 24May2002
  Descriptive information on all rhs, lhs, and flim measures
f2alt01b.do - 25May2002
  Compute bic' for each of four outcomes and all flim measures.
   **  Outcome: Can Work                  global lhs "qcanwrk95"
   **  Outcome: Work in three categories  global lhs "dhlthwk95"
   **  Outcome: bath trouble              global lhs "bathdif95"
   **  Outcome: adlsum95 - sum of adls    global lhs "adlsum95"
f2alt01c.do - 25May2002
  Compute bic' for each of four outcomes and with only these restricted flim
measures.
   *     1.  ln(x+.5) and ln(x+1)
   *     2.   9 counts: >=5=5  >=7=7  (50% and 75%)
   *     3.   8 counts: >=4=4  >=6=6  (50% and 75%)
   *     4.   18 counts: >=9=9 >=14=14  (50% and 75%)
   *     5.   probability splits at .5; these don't work well in prior tests
f2alt01d.do - 25May2002
  bic' for all four outcomes in models that include all raw flim measures
(fla*p5; fll*p5);
  pairs of u/l measures; groups of LCA measures
f2alt01e.do - all LCA probabilities - 25May2002
<snip>
```

Newer style of project diary

First complete set of analysis for FLIM measures paper

2002-05-24: model flims as a count

f2alt01a.do – desc stats
f2alt01b.do – bic to compare models for four outcomes

2002-05-25: alternative outcome measures

***f2alt01c.do* - model with restricted flim measures**
f2alt01d.do – model with raw flim measures

2002-05-26: LCA

f2alt01e.do – LCA of flims

<snip>

Template for project diaries in /Templates

Research diary shortcuts¶

Ctrl-1¶

Normal text (ctrl-alt-shift-t)¶

Ctrl-2¶

Ctrl-3¶

Ctrl-4¶

¶
Normal text in proportional font (ctrl-alt-shift-p)¶
¶
o→ Bullet text (ctrl-alt-b)¶
¶
1.→ Numbered (ctrl-alt-n)¶
¶
Output in 10 point font (alt-0)¶
·····1·····2·····3·····4·····5·····6·····7·····8¶
12345678901234567890123456789012345678901234567890123456789012345678901234567890¶
¶
Output in 9 point font (alt-9)¶
·····1·····2·····3·····4·····5·····6·····7·····8¶
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890¶
¶
¶
Source: Template-research-diary-win-2015-11-21.docx¶

#2 Metadata in datasets

1. *Metadata* internally documents the dataset and points to the project diary and script files that create the dataset

   Details in later chapters

#3 Script files with comments

1. Script files do the work and are the best documentation for *exactly* what you did with your data and analysis.
2. Keeping track of these script files is essential.
   - My research diary points to script files.
   - Organization makes it easy for me to find them (details below)

Codebooks

1. A codebook summarizes information on the variables you started with and variables you add.
2. It reflects *your* decisions using existing data and new variables you created.
3. Guide to Social Science Data Preparation and Archiving: Best Practice Throughout the Data Life Cycle (ICPSR 2005) on the web.

What to record about new variables

1. *Variable name* and source variable
2. How the variable was created including assignment of missing
3. How missing data was imputed

Where to put this

1. In the PDF for the original codebook
2. As metadata in your data file or output from your statistics program

Codebook based on survey instrument

	Not at all Important								Very Important
Q43. Turn to family for help	1	2	3	4	5	6	8	9	9	10
tcfam **Q43 How Important: Turn to family for help**										
Q44. Turn to friends for help	1	2	3	4	5	6	8	9	9	10
tcfriend **Q44 How Important: Turn to friends for help**										
Q45. Turn to a minister, priest, Rabbi or other religious leader	1	2	3	4	5	6	8	9	9	10
tcrelig **Q45 How Important: Turn to a Minister, Priest, Rabbi or other religious leader**										
Q46. Go to a general medical doctor for help	1	2	3	4	5	6	8	9	9	10
tcdoc **Q46 How Important: Go to a general medical doctor for help**										
Q47. Go to a psychiatrist for help	1	2	3	4	5	6	8	9	9	10
tcpsy **Q47 How Important: Go to a psychiatrist for Help**										
Q48. Go to a mental health professional for help	1	2	3	4	5	6	8	9	9	10
tcmhprof **Q48 How Important: Go to a mental health professional**										

ALLOWED DEFINITION – PSYCHOLOGIST, THERAPIST, SOCIAL WORKER, OR COUNSELOR

INTERVIEWER NOTE: CODE "DON'T KNOW" AS 98 ABOVE SEQUENCE.

The next few questions deal with the government's responsibility to help people like NAME. For each statement please tell me if you think the government definitely should, probably should, probably should not, or definitely should not be responsible for helping people with situations like NAME.

General suggestions for documentation

1. **Do it today**: documentation is easier when work is fresh in your mind.
2. **Check it later**: What is obvious now is obscure later.
3. **Use full dates and names:**
   - On 3/26 it seems inconceivable that the project will continue three years.
   - Three people can have the initials JM on a four person project.

A thought experiment for documentation

1. Think of a variable created early in your project.
2. You finish a draft paper and discover a variable is wrong.
3. If it take more than 10 minutes to fix the problem and re-run all analyses, you need to improve your WF.

Project checklist

A useful way to keep track of POD is to have a checklist that you use at the completion of a major step (e.g., paper submission).

Project Checklist

__1. Project diary is complete and clear; clarify dates and abbreviations.

__2. Reproduce results by moving script files to a new directory and running the master script files. Compare to prior results.

__3. Verify that provenance of results in the presentation; verify links to scripts.

__4. Post supplementary analyses on the web.

__5. Clean stray files.

__6. Post files and make backups.

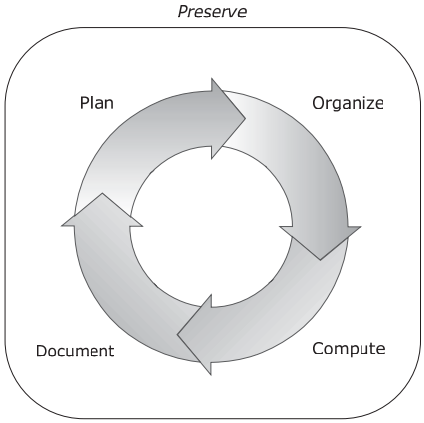
__7. Plan follow-up work.

__8. Write summary in diary of status of project.

Overview of POD

1. While people avoid these things, *POD plays huge dividends*.
2. You create datasets and variables faster with fewer mistakes.
3. Data analyses is more focused and efficient.
4. You spend less time deciding what to do.
5. You find and fix errors faster.
6. It is easier to recover from interruptions and move between projects.
7. *You can reproduce results with less effort.*
8. Revisions are much easier.
9. *Stress* is reduced.

Part 7: A workflow for computing



How I compute

1. Robust and legible script files
2. A project directory structure that supports reproducibility
3. Three strategies for computing
   a. The posting principle
   b. Dual workflows for data and analysis
   c. Run order naming of script files

Script files for computing

1. Script files are text files with commands. Also called “syntax files”, “do-files”, “command files”, and more.
2. To be reproducible, computing must use script files.
3. Script files should be:
   - *Robust* so they produce the same results on a different computer.
   - *Legible* to prevent errors and provide documentation.
4. Writing script files is discussed in Part 8.

Project directory structure

Every project uses the same directory structure.

Booking keeping and administrative folders.

```
\ProjectKeyword
    \-  History starting yyyy-mm-dd
    \- Hold then delete
    \- To clean
        \Review
        \Shelve
    \Admin
        \-  Administrative documents
    \Resources
        \-  Materials that support the project
```

Research folders (details on use follow)

```
\Posted
    \-  Never change these files
\Work
    \-  Stata working directory for project
    \PrePosted
    \ToDo
    \Versions
\Write
    \-  Writing that is in process
    \PrePosted
    \ToDo
    \Versions
```


Strategies for computing

The Posting Principle

Dual Workflows for Data and Analysis

Run Order Naming of Script Files

The essential posting principle

Reproduction is impossible if you change files used for *shared* results.

Posting is defined by two simple rules

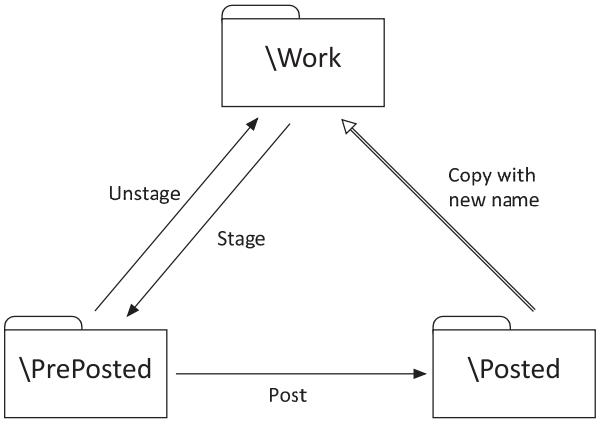
The share rule

Only share results *after* the files are posted.

The no change rule

Once a file is posted, *never* change it.

Overview of file movement with posting



1. Files evolve through three steps

Type of file	Location
Active files	`\Work`
PrePosted or staged files	`\PrePosted`
Posted or committed files	`\Posted`

2. **Staging** moves files from \Work to \PrePosted
   - Files "age" before I decide they are finished
3. **Recalling** moves files from \PrePosted to \Work
   - I make further revisions.
4. **Posting** moves files from \PrePosted to \Posted
   - Before sharing I move files used to create the results here
   - Also called committing
5. To **revise** a posted file, copy the file with a new name from \Posted to \Work

Activities consistent with posting

Debug files in \Work

1. You can change a do-file while debugging it.
2. You can work on multiple do-files at the same time.
3. You can let files sit for weeks and return to them.

Correct errors in posted files

1. **`pgm1.do`** in \Work creates **`data1.dta`** with **`var1`** through **`var9`**.
2. *After* posting **`pgm1.do`** to \Posted I discover **`var5`** is wrong.
3. Copy **`pgm1.do`** from \Posted to **`pgm1V2.do`** in \Work to correct the error and creates **`var5V2`** and in **`data1V2.dta`**.
4. I can delete **`data1.dta`** and **`pgm1.do`** but must *not* change them.

Sharing preliminary results

1. I want to share preliminary results
2. I could post files with preliminary results
   - Then create new versions of files after I get feedback.
3. Or I could share un-posted files
   - Rename files: **`pgm01.log`** to **`pgm01.logunofficial`**
   - *Inside* the renamed file add note that results are unofficial.
   - To be extra careful, only distribute a PDF
4. Sharing preliminary results is useful, but risky.

Deciding when to post a file

1. I work on multiple related do-files at the same time.
   - What I learn debugging one leads to changes in another.
2. Eventually I *move* the files to **`\PrePosted`** and let them *age*.
3. I post the files when:
   a. I share want to results with others
   b. I decide the results are acceptable
4. Before posting, I *move* files from **`\PrePosted`** to **`\Work`** and verify they are correct. Then they are moved to **`\Posted`**

Make no exceptions to the posting principle

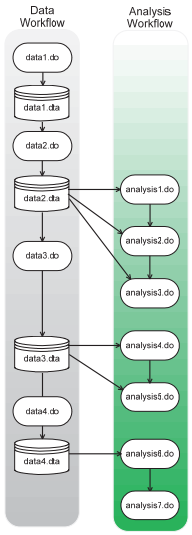
1. Posting is essential for RR
2. Posting saves times by reducing errors
3. It provides implicit documentation since you can easily determine what work was completed
   - This helps a lot when projects are interrupted
4. It makes it easier to backup your work.

You agree, but…

- "The data are *exactly* the same except I changed the married variable."
- "It was only there a few minutes before I changed it."

Discussion

Dual workflows



Data management & statistical analysis

1. Data management is more exacting, less exciting, and more fundamental.
2. Analysis sparks creative energy that encourages poor data management.

Advantages of a dual workflow

A dual workflow is not essential for a reproducible workflow, but

1. It facilitates reproducible results.
2. It prevents errors.
3. It simplifies organization and documentation.
4. It encourages planning.
5. It simplifies collaboration.
6. It prevents ad hoc variables…

Example of dual workflow

`wfx-dual-data01-list.do`	Use rr-issp-russia01.dta
`wfx-dual-data02-varselect.do`	Create wfx-dual-data01.dta
`wfx-dual-data03-regvars.do`	Create wfx-dual-data02.dta
`wfx-dual-stat01-ologit.do`	Analyze wfx-dual-data02.dta
`wfx-dual-data04-lhsbinary.do`	Create wfx-dual-data03.dta
`wfx-dual-stat02-binlogit-BNall.do`	Analyze wfx-dual-data03.dta
`wfx-dual-stat03-binlogit-BHall.do`	Analyze wfx-dual-data03.dta
`wfx-dual-stat04-irtbin.do`	Analyze wfx-dual-data03.dta
`wfx-dual-data05-binscales.do`	Create wfx-dual-data04.dta
`wfx-dual-stat05-regscales.do`	Analyze wfx-dual-data04.dta
`wfx-dual-stat06-lpoly.do`	Analyze wfx-dual-data04.dta

The danger of ad hoc variables

1. Variables created "on the fly" can be created differently in different do-files.
2. For example,
   a. **data1.dta** has **educyears**.
   a. In **logit1.do** using **data1.dta** I create **higheduc** for bachelor's degree or beyond.
   b. In **logit4.do** using **data1.dta** I create **higheduc** for high school or beyond.

Creating a scale with dual workflow

1. I use IRT to create a scale.
2. Specifying, estimating, and testing an IRT model is in the analysis WF.
3. After I finalize the model, I use that model in the data WF to create a new dataset with the new scale.

Creating a "dual" dataset

1. It is annoying to interrupt analysis to create a dataset.
2. To minimize the interruption, before I begin analysis I make a clone of the current dataset, called the *dual dataset*.

How to use a dual dataset

1. **data2.do** uses **data1.dta** and saves **data2.dta** as a duplicate.
2. **analysis1.do** uses **data2.dta**.
3. I need a new variable **agesq**.
4. I modify **data2.do** using **data1.dta** to create **agesq** in the unposted **data2.dta**.
5. **analysis1.do** continues to use **data2.dta** which now includes **agesq**.
6. I cycle through the programs until I am satisfied with the analysis do-files.
7. I pre-post the data management files, data analysis files, and data files.

Run order naming and a dual workflow

This is not essential, but simplifies documentation and reproducibility.

Run Order Rule: Name do-files so they *exactly* recreate your datasets and reproduce your data analyses when run in *alphabetical order*.

Data workflow	*Analysis workflow*
data1-ingest.do	desc1-sumstats.do
data2-labels.do	desc2-graphs.do
data3-missing.do	
data4-scales.do	scales1-irtbinary.do
data5-sesvars.do	scales2-irtordinal.do
data6-samplesel.do	scales3-cfaordinal.do
	logit1-baseline.do
	logit2-educ.do
	logit3-age.do
	logit4-indirect.do

Must scripts be run alphabetically?

1. **Data management** scripts must be run in a specific order.
   - Run order naming makes this easy. The names are documentation.
2. Robust do-files for **data analysis** do *not* depend on the run order.
   - I sequentially name analysis do-files so that the last do-file in the sequence produces the latest analyses. Thus, the names provide documentation.
3. Run order helps documentation since the names reflect the history.
4. Beyond sorting alphabetically, how should you name do-files?

Naming do-files that create datasets

1. Creating a dataset requires that do-files are run in a specific order.
2. Suppose I need two do-files to create a dataset.
   do-file 1: (a) Extract **hlthexpend** from **medical.dta**
   (b) Extract **popsize** from **census.dta**
   (c) Save **health1.dta**
   do-file 2: (a) Generate **hlthpercap = hlthexpend/popsize**
   (b) Save **health2.dta**
3. What should I call the do-files?

Option 1	Option 2	Option 3
merge.do	data1.do	data1-merge.do
addvar.do	data2.do	data2-addvar.do

Template for naming script files

project-taskstepVversion-description.do

- cwh-clean01-CheckLabels.do
- cwh-logit02V2-BaseModel.do

Parts of the name

project: a mnemonic such as **cwh** for a study of cohort work (optional).

task: As needed, I divide projects into tasks. For example, **cwh-clean** for jobs related to cleaning data for the CWH project.

step: Two digits to indicate the run order. For example, **desc01.do**, **desc02.do**, etc.

version: If there is a revision I add a version number. If **desc01.do** was posted, the revision is **desc01V2.do**.

description: *NOT needed to uniquely identify the file*, but is convenient.

Minimal and full name

1. The *minimal name* uniquely identifies a file.
2. The description in the full name is for convenience and documentation.
3. I refer to the minimal name in documentation and provenance notes.

Full name	Minimal name
`fl-clean01-checklabels.do`	`fl-clean01.do`
`fl-logit01V2-basemodel.do`	`fl-logit01V2.do`

Ease of use

1. I prefer names that remind me of what is in the file.
   - o **`logit01.do`** is better than **`pgm01.do`**.
2. Names should be easy to type.

Collaborative projects

You can add author's initials to the front of the job name.

`cwh-desc01.do` → **`jsl-cwh-desc01.do`**.

Names for complex analyses

1. Long and Pavalko (2004) examined how different measures of functional limitations affected substantive conclusions. We used 500 do-files to constructed hundreds of scales and fit 1000s of models.
2. Subdirectories were used for different tasks.
   - **`\flim0-data`** Datasets
   - **`\flim1-extr`** Extract data from source files
   - **`\flim2-scal`** Construct scales of FLIMs
   - **`\flim3-out`** Construct outcome measures
   - **`\flim4-desc`** Descriptive statistics of source variables
   - **`\flim5-lca`** LCA of FLIMs
   - **`\flim6-reg`** Regression models for FLIMs
3. Files were named for run order within each subdirectory.

Correcting mistakes in posted do-files

Run order naming makes it simpler to fix problems.

1. Scripts **`data01.do`** through **`data10.do`** create **`cwh01.dta`**.
2. After posting, I find a mistake in **`data06.do`**.
3. I create **V2** versions of **`data06V2.do`** through **`data10V2.do`**.
4. Because of the way files are named, I know exactly which do-files to run and in what order to create a corrected dataset **`cwh01V2.dta`**.
   - o I don't re-run **`data05.do`** or prior since they occur earlier in the run order.
5. Master do-files make this even simpler.

Master do-files

Running a sequence of do-files

1. You will often re-run a sequence of do-files.
   - o When I complete the do-files to create a dataset, I verify that all of the programs work correctly before pre-posting or posting the files.
   - o After discovering an error in one program from a sequence of related jobs, I fix the error and verify that all other programs still work correctly.
2. A *master do-file* makes this simple. For the data programs:

```
// data.do: do-file for data management
// Scott Long | 2008-03-14

do data01.do
do data02.do
do data03.do
do data04.do
exit
```

3. To rerun the four do-files in sequence, I run:

```
do data.do
```


4. Similarly:

```
// analysis.do: do-file for statistical analysis
// Scott Long | 2008-03-14

* descriptive statistics
do stat01.do
do stat02.do
do stat03.do

* logit models
do logit01.do
do logit02.do

* graphs of predictions
do graph01.do
do graph02.do
```

5. To rerun the do-files in sequence, I run:

```
do analysis.do
```


Fixing mistakes with a master do-file

1. I find a problem in **`data03.do`** that affects **`data03.dta`** and consequently **`data04.dta`**.
   - o This invalidates analyses based on these datasets.
2. I create **V2** versions of some data management do-files.

```
// data.do: do-file for data management
// Scott Long | 2008-03-14; revised 2008-03-17

do data01.do
do data02.do
do data03V2.do
do data04V2.do
exit
```

3. I change the analysis master file to use the new datasets:

Next page...

```
// analysis.do: do-file for statistical analysis
// Scott Long | 2008-03-17

* descriptive statistics
do stat01.do
do stat02.do
do stat03.do

* logit models
do logit01.do
do logit02V2.do

* graphs of predictions
do graph01V2.do
do graph02V2.do

exit
```

4. I can rerun everything with the commands:

```
do data.do
do anaysis.do
```

5. This is very useful when revising a paper.

A single master files for data and analysis

```
//  Reproducible Results: Dual Workflow
//  Master do-file

//  2017-06-16 Scott Long

do wfx-dual-data01-list.do
do wfx-dual-data02-varselect.do
do wfx-dual-data03-regvars.do

do wfx-dual-stat01-ologit.do

do wfx-dual-data04-lhsbinary.do

do wfx-dual-stat02-binlogit-BNall.do
do wfx-dual-stat03-binlogit-BHall.do
do wfx-dual-stat04-irtbin.do

do wfx-dual-data05-binscales.do

do wfx-dual-stat05-regscales.do
do wfx-dual-stat06-lpoly.do
```


Conclusions

1. These strategies are simple, save time, and prevent problems.
2. They work with any software
   - They are a simple way to implement "versioning"
3. Posting is essential, while the rules make work easier and less error prone.
4. The rules take time, but prevent errors that can take days or weeks to correct.
5. They make it much easier to revise a paper.
6. They simplify coordination of work in collaborative projects.