BITSS

BERKELEY INITIATIVE FOR TRANSPARENCY
IN THE SOCIAL SCIENCES

Software and
Workflow

Christensen

Software and Workflow for Reproducible Research

Garret Christensen[1]

[1]UC Berkeley: Berkeley Initiative for Transparency in the Social Sciences
Berkeley Institute for Data Science

Annual Meeting, December 2015

- **What are practical tools to implement reproducibility solutions?**

BITSS

BERKELEY INITIATIVE FOR TRANSPARENCY
IN THE SOCIAL SCIENCES

Software and
Workflow

Christensen

- Data not available
- Code not available/unintelligible
- Code and data cannot reproduce original results

- Even with the help of the original author (yourself?), you can't get the data to reproduce the published results. Or you just can't find the data to begin with.
- *Journal of Money, Credit, and Banking* Project. (Dewald et al., AER 1986)
- Martin Feldstein on Social Security and private savings, Reinhart and Rogoff on debt and GDP growth.

- Literate Programing
- Version control
  - Github
  - OSF
- Dynamic Documents
  - R Markdown and R Studio
  - Ketchup in Stata
- Data Sharing
  - Harvard's Dataverse

- First, *programming* is key to reproducibility. Working in Excel is not reproducible.
- See Reinhart and Rogoff "Growth in a Time of Debt" controversy:
  - Original Paper, *AER P & P* 2010
  - Herndon et. al (2013) finding.
  - *New Yorker* summary.
- Random number generation in Excel–set seed with Data Analysis Toolpak.

- If you are using SPSS, use of 'syntax' to record all the commands you run is simple. (See UCLA tutorial.) Similarly in Stata, 'commandlog'.
- Better is to write scripts. R, Stata, SAS, Python, or whatever you please.
- Open source has some advantages (being free, for one) but you're going to use what everyone in your field uses.

- Second, *literate programming* is key to reproducibility. Write code to be read by a human being, with the code for the computer secondary.

"I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: "Literate Programming."
Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

(cont.)

"The practitioner of literate programming can be regarded as an essayist, whose main concern is with exposition and excellence of style. Such an author, with thesaurus in hand, chooses the names of variables carefully and explains what each variable means. He or she strives for a program that is comprehensible because its concepts have been introduced in an order that is best for human understanding, using a mixture of formal and informal methods that reinforce each other."

–Donald Knuth *The Computer Journal*, 1984 Quotes Original

*"Reproducibility is just collaboration with people
you don't know, including yourself next week"*

—Philip Stark, UC Berkeley Statistics

Practical coding and organizational suggestions

- Long (2008) *The Workflow of Data Analysis Using Stata*
- Making any changes to a file that has been posted/shared means it gets a new name.
- Use version commands to ensure others get same results.
- Keep a daily research log.

- Using version control (AKA revision control) can help to make your work more reproducible.
  - What is version control?

    Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

    –Git, About Version Control

- Using version control (AKA revision control) can help to make your work more reproducible.
- What is version control?

  Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

  –Git, About Version Control

- Using version control (AKA revision control) can help to make your work more reproducible.
- What is version control?

    Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

    –Git, About Version Control

With version control you can:

- Collaborate
- Track who made every change
- Easily switch between versions of files
- Compare versions of files
- Backup
- Work with the same files on different machines
- Experiment with a new version of code without breaking things

Link1 Link2 Link3

Places you're already using version control without knowing
it:

Bobbie Spellman

March 27, 5:53 PM
anonymous

March 27, 1:01 PM
anonymous

March 27, 5:21 AM
Arthur Lupia

March 26, 10:22 AM
Bobbie Spellman

- (cur | prev) ○ 03:54, 1 August 2006 SlimVirgin (talk | contribs) **m** . . (53,895 bytes) (0) . . *(Protected Elephant: wasn't actually protected; as requested [edit=autoco*
- (cur | prev) ○ 03:54, 1 August 2006 SlimVirgin (talk | contribs) **m** . . (53,895 bytes) (0) . . *(Protected Elephant: wasn't actually protected; as requested [edit=autoco*
- (cur | prev) ○ 03:53, 1 August 2006 SlimVirgin (talk | contribs) **m** . . (53,895 bytes) (0) . . *(Protected Elephant: wasn't actually protected; as requested [edit=autoc*
- (cur | prev) ○ 03:51, 1 August 2006 RasputinAXP (talk | contribs) . . (53,895 bytes) (+14) . . *(protecting from vandalism)*
- (cur | prev) ○ 03:47, 1 August 2006 Stevenj (talk | contribs) . . (53,881 bytes) (-15) . . *(whoops, unrevert; I accidentally re-added the vandalism instead of removing*
- (cur | prev) ○ 03:46, 1 August 2006 Crzrussian (talk | contribs) . . (53,896 bytes) (-46) . . *({{protected}})*
- (cur | prev) ○ 03:41, 1 August 2006 Stevenj (talk | contribs) **m** . . (53,942 bytes) (+61) . . *(Reverted edits by Xaosflux (talk) to last version by Fire Star)*
- (cur | prev) ○ 03:40, 1 August 2006 Xaosflux (talk | contribs) . . (53,881 bytes) (-61) . . *(-THE NUMBER OF ELEPHANTS HAS TRIPLED IN THE LAST SIX MONTH*

Places you're already using version control without knowing it:

- Google Docs
- Wikipedia
- Every piece of software you use.

Isn't this just a complicated version of the "date and initial"
method?

- regressions2015.08.24.do
- regressions2015.08.25.do
- regressions2015.08.25GC.do
- Hassle
- Confusion

Here is a good rule of thumb: If you are trying to solve a problem, and there are multi-billion dollar firms whose entire business model depends on solving the same problem, and there are whole courses at your university devoted to how to solve that problem, you might want to figure out what the experts do and see if you can't learn something from it.

...

Not one piece of commercial software you have on your PC, your phone, your tablet, your car, or any other modern computing device was written with the "date and initial" method.
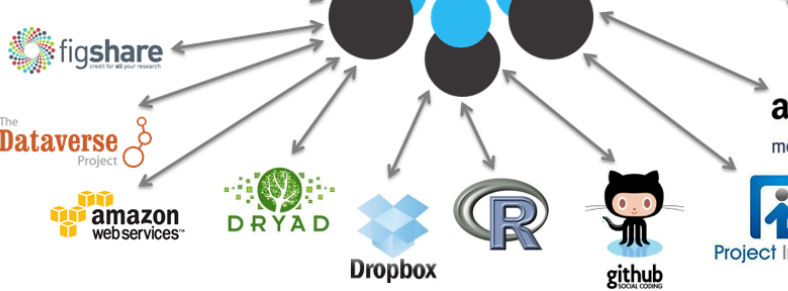
–Matthew Gentzkow and Jesse M. Shapiro "Code and Data for the Social Sciences: A Practitioner's Guide"
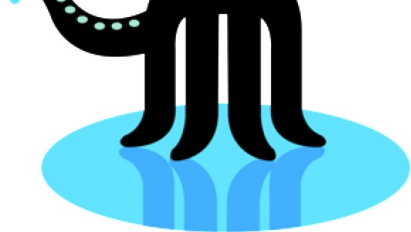
GitHub and OSF Examples:

- Slides for this workshop on Github.com
- `http://www.github.com/bitss/annual2015`
- Slides also available on the Open Science Framework
- `https://osf.io/fillthisinlater/fordemo`

- Even if you write perfect (version controlled) code, you can still run into problems going from your code to paper. This is where *dynamic documents* come in.

- A dynamic document includes your data, code, analysis, and output all in one place. Fully automated, you can guarantee no mistakes from copying and pasting.

- Do this with R Markdown in R Studio or Ketchup in Stata.

- Even if you write perfect (version controlled) code, you can still run into problems going from your code to paper. This is where *dynamic documents* come in.
- A dynamic document includes your data, code, analysis, and output all in one place. Fully automated, you can guarantee no mistakes from copying and pasting.
- Do this with R Markdown in R Studio or Ketchup in Stata.

- Even if you write perfect (version controlled) code, you can still run into problems going from your code to paper. This is where *dynamic documents* come in.
- A dynamic document includes your data, code, analysis, and output all in one place. Fully automated, you can guarantee no mistakes from copying and pasting.
- Do this with R Markdown in R Studio or Ketchup in Stata.

BITSS

BERKELEY INITIATIVE FOR TRANSPARENCY
IN THE SOCIAL SCIENCES

Software and
Workflow

Christensen

- Include tables by linking to a file, instead of a static image.
- Include number by linking to a value calculated by an analysis file, instead of a static number typed manually.
- Automatically update tables and numbers.
- Produce entire paper with one or two clicks.

- R Studio Example
- Stata Example

Simple tools exist to help you transparently and reproducibly take your research from beginning to end.

- Trial Registries (previous slides)
- Pre-Analysis Plans (previous slides)
- Version Control
- Open Science Framework
- Dynamic Documents
- Trusted Public Data Archive (next slides)

Read more in my *Manual of Best Practices in Transparent Social Science Research* on GitHub.