Software and Workflow for Reproducible Research

Discussion

Garret Christensen[1]

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

APHRC, Summer 2015

Outline

1. Introduction
2. Problems
   - Irreproducible Workflow
3. Solutions
   - Workflow
   - Literate Programming
   - Version Control
   - Dynamic Documents
   - Data Sharing
4. Conclusion

Reproducibility & Transparency

- What are problems associated with reproducibility?
- What are solutions to these problems?
- What are practical tools to implement these solutions?

Problems

- Publication bias (see previous talk)
- Specification Searching (see previous talk)
- Data not available
- Code not available/unintelligible
- Code and data cannot reproduce original results

Irreproducible Workflow

- Even with the original authors’ help, 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.

Solutions

- Study Registry (see previous talk)
- Pre-Analysis Plan (see previous talk)
- Reproducible Workflow

Solutions

- Study Registry (see previous talk)
- Pre-Analysis Plan (see previous talk)
- Reproducible Workflow

Solutions

- Study Registry (see previous talk)
- Pre-Analysis Plan (see previous talk)
- Reproducible Workflow

Reproducible Workflow

- Literate Programing
- R Markdown and R Studio to write dynamic documents.
- Version control with Github or OSF.
- Data Sharing
  - Harvard's Dataverse

Literate Programming

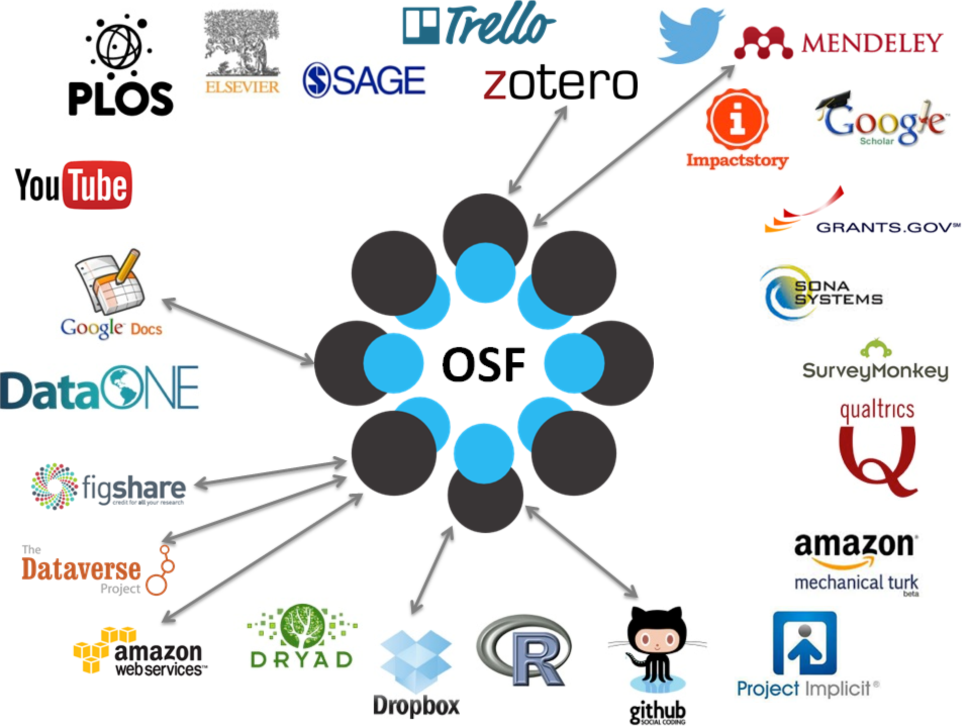
First, you should be *programming*. Working in Excel is not reproducible. Reinhart and Rogoff. If you are using SPSS, there is a ‘syntax’ command to record all the commands you run. Similarly in Stata, commandlog.

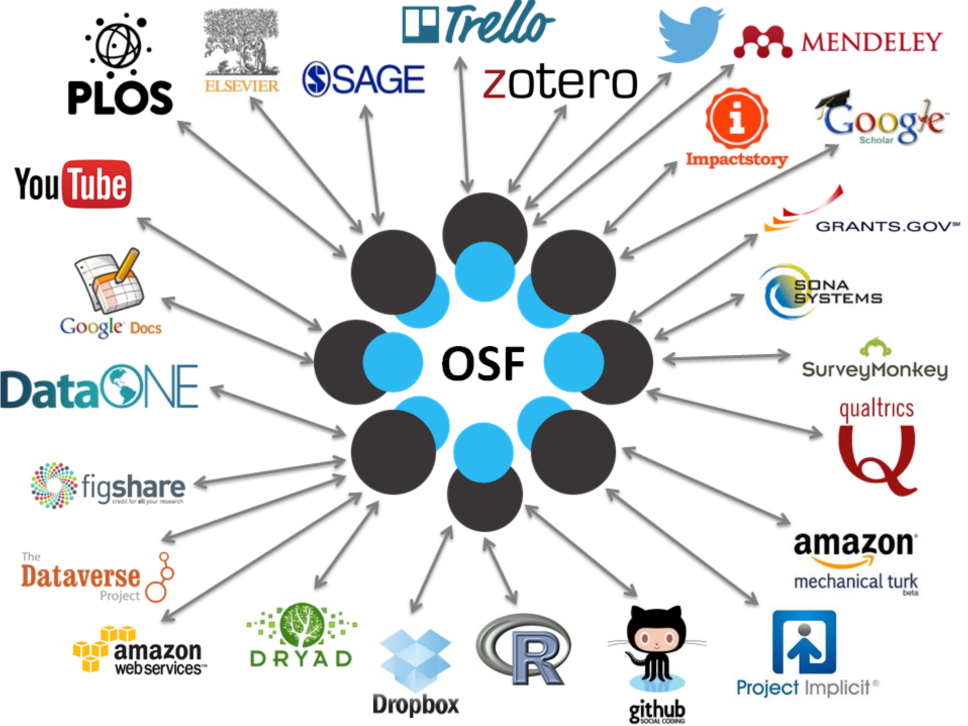
Better is to write scripts. R, Stata, SAS, Python, whatever. Open source has some advantages (being free, for one) but you’re going to use what everyone in your field uses.

Second, literate programming. Write code to be read by a human being, with the code for the computer secondary.

Version Control

Use version control.
DCVS distributed version control system
gentzkow and shapiro’s RA guide


PLOS
ELSEVIER
SAGE
Trello
zotero
MENDELEY
Impactstory
Google Scholar
YouTube
GRANTS.GOV
Google Docs
SONA SYSTEMS
OSF
SurveyMonkey
DataONE
qualtrics
figshare credit for all your research
The Dataverse Project
amazon mechanical turk beta
amazon web services
DRYAD
Dropbox
R
github SOCIAL CODING
Project Implicit®


PLOS
ELSEVIER
SAGE
Trello
zotero
MENDELEY
Impactstory
Google Scholar
GRANTS.GOV
SONA SYSTEMS
SurveyMonkey
qualtrics
amazon mechanical turk beta
Project Implicit
github SOCIAL CODING
R
Dropbox
DRYAD
amazon webservices
The Dataverse Project
figshare credit for all your research
DataONE
Google Docs
YouTube
OSF

GitHub and OSF Examples

Dynamic Documents

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 and R Studio or Ketchup in Stata.

R Studio Example

Stata Example

Data Sharing

Make all your data and code publicly available.
Put it in a place where people can find it.
For APHRC that might be the APHRC repository.

Conclusion

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

- Open Science Framework
- Trial Registries
- Version Control
- Dynamic Documents
- Trusted Public Data Archive

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