Reproducibility, Documentation, and Project Workflow

Best Practices for Transparent Social Science Research

Shyam Raman

11/5/2021

Cornell University Brooks School of Public Policy

Roadmap for Today’s Talk

Why Does Reproducibility Matter?

Documenting Research Decisions

My Project Workflow

Q&A For Remaining Time

Why Should I Make My Project Reproducible?

Many journals already have requirements for code & data supplements

- All open-source journals and most social science journals mandate

Data-driven analyses often involve data manipulation as well

- Researcher decisions at this stage are often not communicated in manuscript
- These same decisions often make large impacts to analysis (will get to this)

You should want to!

- Reproducibility habits make your code and workflow better
- This also makes things easier on you, going back to a project

Reproducibility vs. Replicability

These are often conflated, but we are interested in the former

Replicability has to do with the research method

- ie. using new data, similar results can be found with same analysis

Reproducibility has to do with transparent coding and documentation

- ie. using the exact same data and analysis, identical results can be found

Both are super important!

- You want people to be able to replicate your study with any data
- This is the definition of good science - we aren't there yet

Documenting Research Decisions

We Make Mistakes - Documentation Finds Them

Research involves a lot of decision-making at the researcher level

- Data sources, manipulation, imputation
- Outcome transformations, RHS variable creation (dummies)

Effective documentation on all these decisions prevents issues

- Clearly cataloging differences between raw and cleaned data
- Providing rationale for variable transformations and creation

Researcher decisions drive estimates: what could go wrong?

- A lot.
- Nick Huntington-Klein and coauthors test this (Economic Inquiry 2021)

The influence of hidden researcher decisions in applied microeconomics (2021)

Project recruited economists to replicate papers with kits available

- Papers to replicate were broadly pulled from T5s and large contribution
- Replication code & files returned by participants to the authors

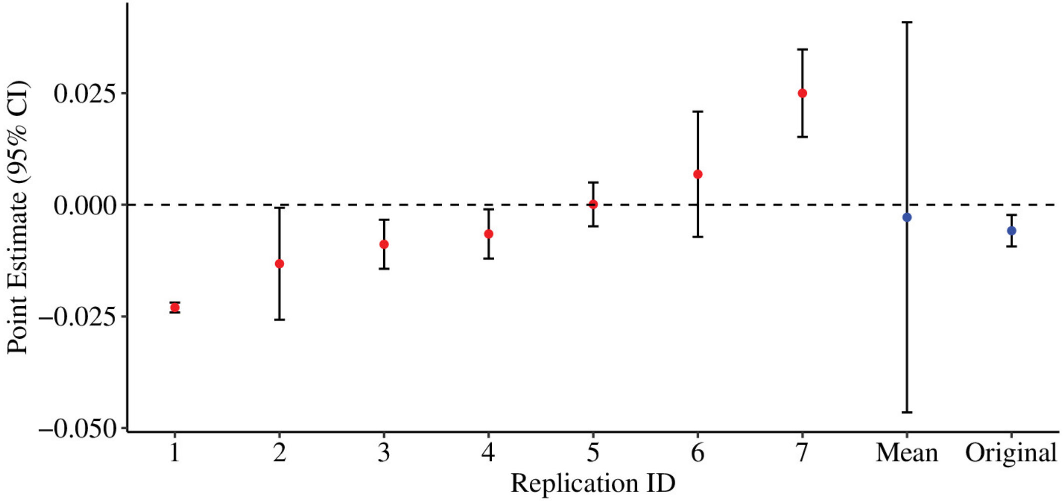
Participating economists were told to replicate and given the same instructions

- Question: Do researchers make differing decisions which create variation in results?
- Answer: Yes.

Authors find large variation in point estimates across separate replications

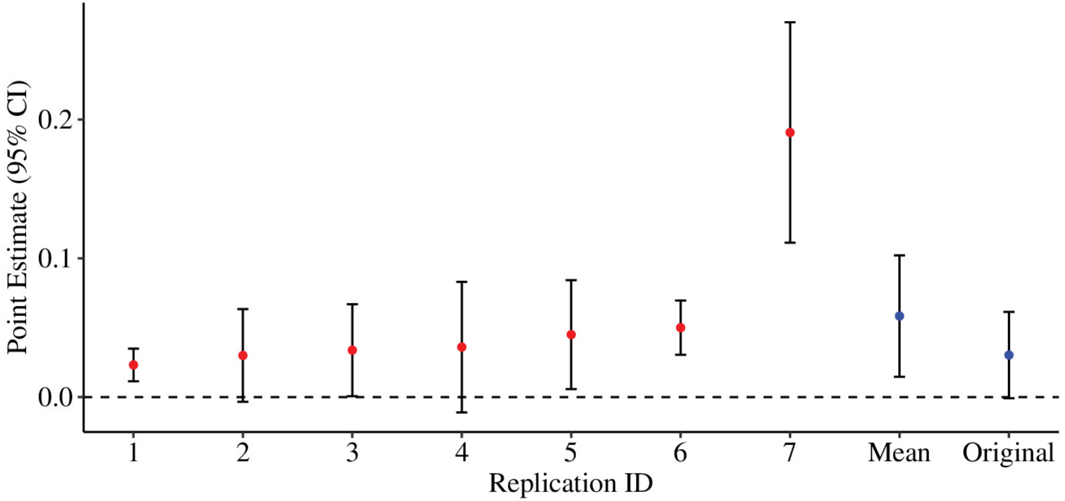
- Most large differences were in data preparation and analysis decisions
    - Hard to measure/capture when a manuscript is the final product
- No two replicators had the same sample size (for a given rep. paper)
- SD of point estimates for replications were 3-4x that of original mean standard error

Var(Replicated Results): Compulsory Schooling & Pregnancy (Black et.al 2008)



"Mean" shows mean of replications and confidence interval for that mean.
"Original" is from the original publication.

Var(Replicated Results): Health Insurance & Self Employment (Fairlie et.al 2011)



"Mean" shows mean of replications and confidence interval for that mean.
"Original" is from the original publication.

Scan for Access to (Left) The Effect; (Right) Replication Paper

My Project Workflow

What is My Goal Workflow? I want to be able to...

Come back to a project after 6 months and seamlessly restart work

- Perpetual data storage, known file structure, updated code
- Self-contained research environments: [Rproj]
- Dynamic coding, dynamic data collection (!), version control

Compress the directory to a [.zip] and send it off for replication

- A separate user should be able to reproduce results fully
- This loops in the importance of readME files - we'll get there

Perform all of my project from a single file: [0.master_run]

- Allows reproduction by simply setting the home directory path

Disclaimer! Results May Vary!

This is what works for me, it may not comport to your workflow

- Most workflows, anyway, are learned over time

Some of what I will discuss have large learning curves

- It was worth it for me when I learned, probably wouldn't be now
- If you don't want to use Git, don't.
    - You probably should, though.

You've probably seen some (if not most) of this before

- My goal is to offer options

My workflow is an accumulation of many projects and colleagues

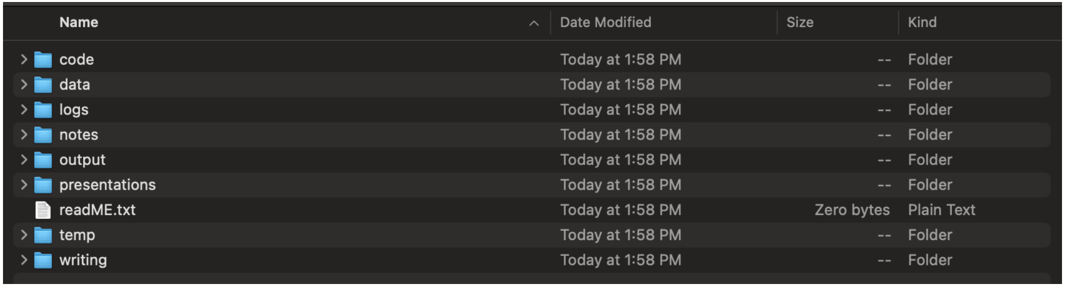
- Pick what works, but have a broad goal of reproducibility

Directory Structure

Uniform directory structures help my workflow

- Being able to look for files in the same sub-folders
- Having consistent data cleaning workflows across projects

A sample directory that I would use:



Name	Date Modified	Size	Kind
code	Today at 1:58 PM	--	Folder
data	Today at 1:58 PM	--	Folder
logs	Today at 1:58 PM	--	Folder
notes	Today at 1:58 PM	--	Folder
output	Today at 1:58 PM	--	Folder
presentations	Today at 1:58 PM	--	Folder
readME.txt	Today at 1:58 PM	Zero bytes	Plain Text
temp	Today at 1:58 PM	--	Folder
writing	Today at 1:58 PM	--	Folder

File Naming Conventions and Storage

Avoid leaving spaces or special characters in file names

- I default to all lowercase and "_"

For data files, I try not to change anything about the source

- Names are left same (unless there are issues outlined above)
- Cleaned data are stored with my naming conventions

Rule of Thumb: If you have to reference it in code, make it easy.

- Names should be: concise, informative, and accurate
- Avoid version #s or other extraneous details on names
    - Sometimes this is unavoidable, I try to avoid it

The [initialize_directory] R Function

I got really tired of manually creating a directory for every project

- I tried to place an “empty” directory on my desktop
    - Forgot about it immediately

I wrote a function in R which creates the directory from before

- To run it, all you need to do is pass:

```
initialize_directory(root_path)
```

This is available on my github, but I just save the function

- I use this every time, it also creates the [root_path]

ReadME File & Template

The [initialize_directory] function creates [readME.txt]

- Populate this with your documentation
- I fill this once the project analysis is done

Many templates exist for social science research readME files

- Most are a skeleton for (roughly) the same info

All readME files need (based on most journal reqs):

- Details on data retrieval and storage
- Full walkthrough of code from start to finish
    - Quick description of each file: what, which data, how
- Details on how a replicator would use the underlying code

Data Documentation

The [initialize_directory] function ALSO creates [data_documentation.txt]

- This file is meant to hold all info relating to data
- I populate this as data comes in - makes life easier

Data documentation is (arguably) even more important

- Where does it come from? Is a web source updated consistently?
- I include (at min) a link to source, variables, and time captured
    - Just passing the [desc] command in STATA will give almost all info

Document how your final analysis file is composed/created

- Note the specific manipulations and imputations and respective code
- Just providing code is not enough, justify your decisions here

Version Control vs Cloud Storage

A sad but necessary note that Box/GDrive/DropBox $\neq$ Version Control

- This is okay! Cloud storage is often a really good option.

Pro/Con: Version Control (Github/Gitlab)

- Pro: track changes in code, great for many authors, link to Overleaf
- Con: learning curve (steep), decentralized data storage (annoying)

Pro/Con: Cloud Storage (Box/GDrive/DropBox)

- Pro: it's easy, you already use it, syncs to cloud automatically
- Con: can't see specific code changes, no recovery for local deletions

Dynamic Coding for Replicability

Dynamic coding is a fancy name and is implemented easily

- Basically, we want all our main code to be path-agnostic

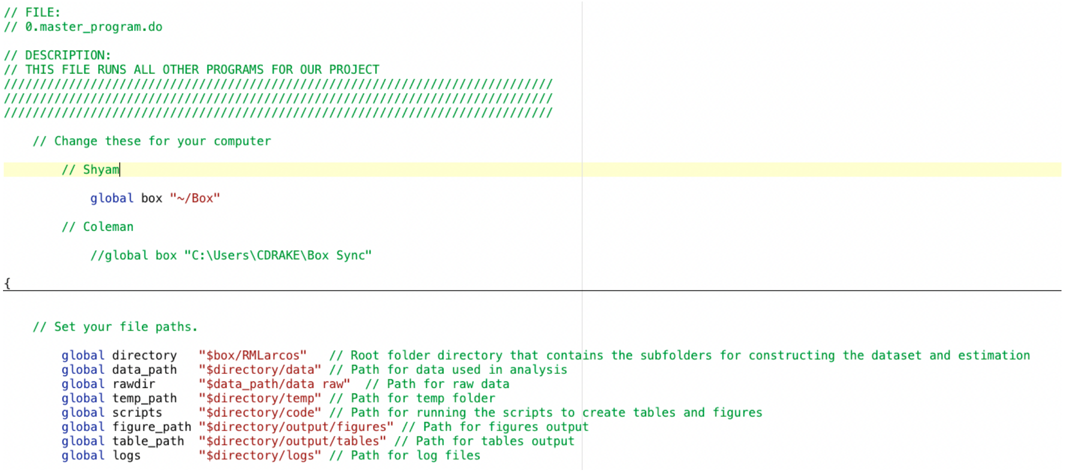
Path-agnostic code uses $global values or project spaces

- STATA: Set a [root_path] and create macros for all subdirectories
- R: Create an [.Rproj] file, sit back, and relax

Why do this? Isn't a [root_path] easy enough to paste in?

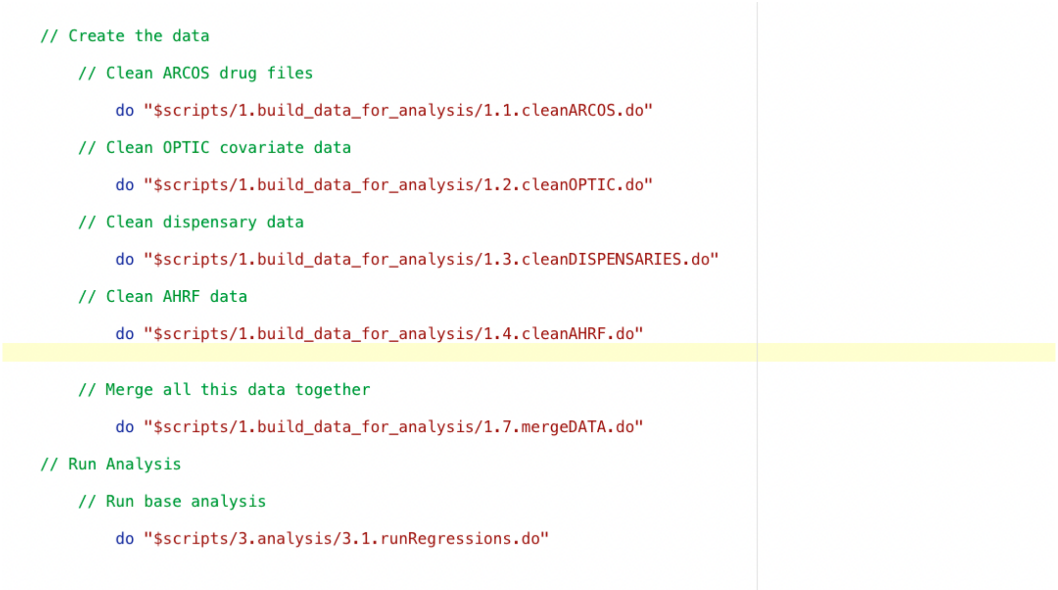
- Hypothetical: You move your project to CISER from local drive.
- Problem: Your filepaths now need to be changed, in every script
- Solution: You dynamically code your paths, only change one line
    - Where? We create and edit the [0.master_run] file

The [0.master_run] File - Setup



```
// FILE:
// 0.master_program.do

// DESCRIPTION:
// THIS FILE RUNS ALL OTHER PROGRAMS FOR OUR PROJECT
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////

    // Change these for your computer

        // Shyam

            global box "~/Box"

        // Coleman

            //global box "C:\Users\CDRAKE\Box Sync"

{

    // Set your file paths.

        global directory   "$box/RMLarcos"   // Root folder directory that contains the subfolders for constructing the dataset and estimation
        global data_path   "$directory/data" // Path for data used in analysis
        global rawdir      "$data_path/data raw"  // Path for raw data
        global temp_path   "$directory/temp" // Path for temp folder
        global scripts     "$directory/code" // Path for running the scripts to create tables and figures
        global figure_path "$directory/output/figures" // Path for figures output
        global table_path  "$directory/output/tables" // Path for tables output
        global logs        "$directory/logs" // Path for log files
```

The [0.master_run] File - Run Scripts



```
// Create the data

    // Clean ARCOS drug files

        do "$scripts/1.build_data_for_analysis/1.1.cleanARCOS.do"

    // Clean OPTIC covariate data

        do "$scripts/1.build_data_for_analysis/1.2.cleanOPTIC.do"

    // Clean dispensary data

        do "$scripts/1.build_data_for_analysis/1.3.cleanDISPENSARIES.do"

    // Clean AHRF data

        do "$scripts/1.build_data_for_analysis/1.4.cleanAHRF.do"


    // Merge all this data together

        do "$scripts/1.build_data_for_analysis/1.7.mergeDATA.do"

// Run Analysis

    // Run base analysis

        do "$scripts/3.analysis/3.1.runRegressions.do"
```

The Pitfall of R: Package Requirements and Version

STATA has R beat squarely when it comes to package consistency

- You frequently need multiple packages to do work in R
- Making sure those packages are available to replicators is important!

R packages (thankfully) are easy enough to install and load

- Problem: doing this is verbose and unnecessary if already installed
- Solution: the [pacman] package's [p_load] function

The [pacman::p_load] function checks if a package is installed and if...

- TRUE: loads the package (equiv to [library] command in R)
- FALSE: installs and then loads the package

Sublime Text & STATA Integration

My first coding exposure was in R, so STATA feels worse

- Please don't come for me
- I just really missed autocompletion on everything

Enter: The Sublime Text/STATA Integration

- Sublime Text is a (free) IDE with multiple build modes
    - Also has package control and user written packages
- The integration is easy, free, and has autocomplete

This is not necessary, but it 100% is for me

- Building from ST also shows the underlying file structure
- You can reference live file paths from within a DO file

Zotero, Better BiBTeX, and LaTeX integration

Manuscript writing is arguably the most miserable part of research

- For me, this was the case because I absolutely hated formatting citations

LaTeX has its drawbacks, but it handles citations so smoothly

- A [.bib] file can be linked to a document and contain all necessary citation info
    - The [.bib] file can be automatically created and updated by Zotero (free)
- My setup: Zotero folder with all relevant papers $\longrightarrow$ [.bib] file export

But wait! I have a new citation and need to add to the [.bib] file.

- Problem: you don't want to create a new [.bib] file every time (this would suck)
- Solution: Better BiBTeX integration with Zotero and Project-Specific Folders
    - Create a perpetual export using BBT from Zotero project folder to project directory

My Project Setup Procedure

Identify Question, Write a Research Sketch, Create Directory

- Use the [initialize_directory] function, add your sketch
- Add this folder to Git/Box/GDrive/DropBox

Gather data, populate documentation files, create analysis file

- Collect data dynamically (where possible) using API/URL downloads
- Store stable raw data in the [data/raw] folder
- Generate analysis data file, fill data documentation

Run analysis, output results within directory, write manuscript

- Analysis ideally could be run via the [0.master_run] file
- Output should be created and exported from script (makes life easier)
- Write manuscript in Sublime with Zotero integration - auto citations

Thank You! Questions?