Prepping Files for Sharing

Validation, Cleaning, and De-Identifying Data

POLICY DESIGN
& EVALUATION LAB

Julia Clark

BITSS Transparency and Reproducibility Workshop
New Delhi
16 March 2017

Overview

- Motivation
- 5 Steps to Prepping Files for Sharing:
    1. Set-up
    2. Initial replication
    3. De-identify
    4. Edit
    5. Final replication

Motivation

Congratulations!

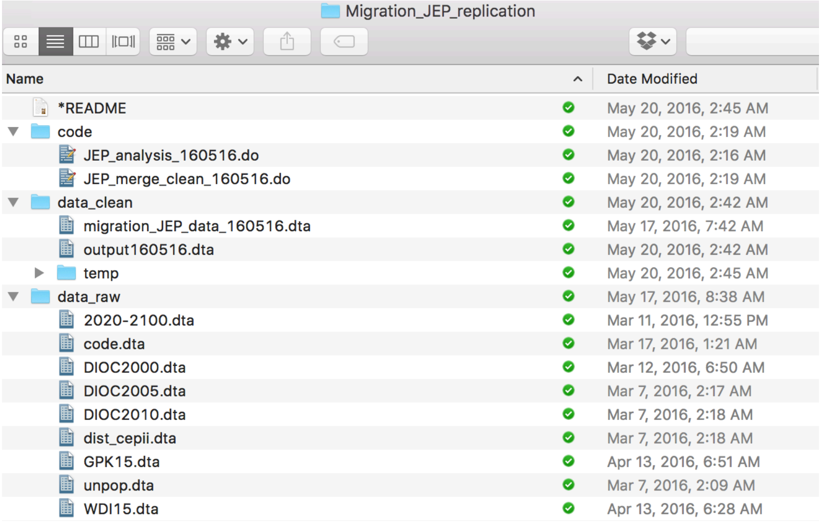
You've completed a study.

…but now, you have to share your data and code for replication, sending these files to colleagues, to a journal, or posting in an online repository.

Ideally

You’ve been using GitHub, and maintaining your files and code with replication in mind, and so they are already (1) complete, (2) replicate, (3) legible, and (4) protect PII.

And maybe they look something like this …

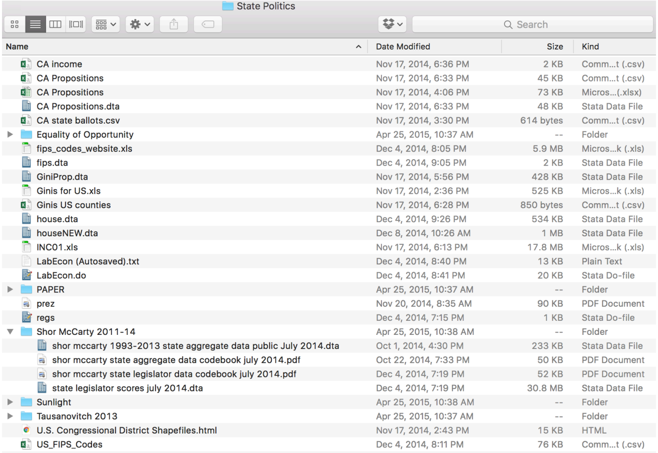


Migration_JEP_replication

Name		Date Modified
*README	✓	May 20, 2016, 2:45 AM
▼ code	✓	May 20, 2016, 2:19 AM
JEP_analysis_160516.do	✓	May 20, 2016, 2:16 AM
JEP_merge_clean_160516.do	✓	May 20, 2016, 2:19 AM
▼ data_clean	✓	May 20, 2016, 2:42 AM
migration_JEP_data_160516.dta	✓	May 17, 2016, 7:42 AM
output160516.dta	✓	May 20, 2016, 2:42 AM
▶ temp	✓	May 20, 2016, 2:45 AM
▼ data_raw	✓	May 17, 2016, 8:38 AM
2020-2100.dta	✓	Mar 11, 2016, 12:55 PM
code.dta	✓	Mar 17, 2016, 1:21 AM
DIOC2000.dta	✓	Mar 12, 2016, 6:50 AM
DIOC2005.dta	✓	Mar 7, 2016, 2:17 AM
DIOC2010.dta	✓	Mar 7, 2016, 2:18 AM
dist_cepii.dta	✓	Mar 7, 2016, 2:18 AM
GPK15.dta	✓	Apr 13, 2016, 6:51 AM
unpop.dta	✓	Mar 7, 2016, 2:09 AM
WDI15.dta	✓	Apr 13, 2016, 6:28 AM

Reality

More often than not, however, our files look like this …



Name	Date Modified	Size	Kind
CA income	Nov 17, 2014, 6:36 PM	2 KB	Comm...t (.csv)
CA Propositions	Nov 17, 2014, 6:33 PM	45 KB	Comm...t (.csv)
CA Propositions	Nov 17, 2014, 4:06 PM	73 KB	Micros...(.xlsx)
CA Propositions.dta	Nov 17, 2014, 6:33 PM	48 KB	Stata Data File
CA state ballots.csv	Nov 17, 2014, 3:30 PM	614 bytes	Comm...t (.csv)
▶ Equality of Opportunity	Apr 25, 2015, 10:37 AM	--	Folder
fips_codes_website.xls	Dec 4, 2014, 8:05 PM	5.9 MB	Micros...k (.xls)
fips.dta	Dec 4, 2014, 9:05 PM	2 KB	Stata Data File
GiniProp.dta	Nov 17, 2014, 5:56 PM	428 KB	Stata Data File
Ginis for US.xls	Nov 17, 2014, 2:36 PM	525 KB	Micros...k (.xls)
Ginis US counties	Nov 17, 2014, 6:28 PM	850 bytes	Comm...t (.csv)
house.dta	Dec 4, 2014, 9:26 PM	534 KB	Stata Data File
houseNEW.dta	Dec 8, 2014, 10:26 AM	1 MB	Stata Data File
INC01.xls	Nov 17, 2014, 6:13 PM	17.8 MB	Micros...k (.xls)
LabEcon (Autosaved).txt	Dec 4, 2014, 8:40 PM	13 KB	Plain Text
LabEcon.do	Dec 4, 2014, 8:41 PM	20 KB	Stata Do-file
▶ PAPER	Apr 25, 2015, 10:37 AM	--	Folder
prez	Nov 20, 2014, 8:35 AM	90 KB	PDF Document
regs	Dec 4, 2014, 7:15 PM	1 KB	Stata Do-file
▼ Shor McCarty 2011-14	Apr 25, 2015, 10:38 AM	--	Folder
shor mccarty 1993-2013 state aggregate data public July 2014.dta	Oct 1, 2014, 4:30 PM	233 KB	Stata Data File
shor mccarty state aggregate data codebook july 2014.pdf	Oct 22, 2014, 7:33 PM	50 KB	PDF Document
shor mccarty state legislator data codebook july 2014.pdf	Dec 4, 2014, 7:19 PM	52 KB	PDF Document
state legislator scores july 2014.dta	Dec 4, 2014, 7:19 PM	30.8 MB	Stata Data File
▶ Sunlight	Apr 25, 2015, 10:38 AM	--	Folder
▶ Tausanovitch 2013	Apr 25, 2015, 10:37 AM	--	Folder
U.S. Congressional District Shapefiles.html	Nov 17, 2014, 2:43 PM	15 KB	HTML
US_FIPS_Codes	Dec 4, 2014, 8:11 PM	76 KB	Comm...t (.csv)

Goal

Use this process a few times on old projects (or other people’s datasets), then structure any new projects with these principles in mind from the beginning, making the back-end process much easier.

PDEL Process

See our GitHub wiki for a summary of these steps:

1. Set-up
2. Initial replication
3. De-identify
4. Edit
5. Final replication

Caveat

There is, of course, no ***single, perfect*** way to organize or prepare files for replication. We find this process to be relatively efficient, but do what works for you (and keeps those files complete, replicable, legible, and protecting PII)!

Note: This process assumes you haven't been using GitHub or other version control software; if you do, some of these steps will become obsolete (yay!).

1. Set up

Start Fresh for Replication

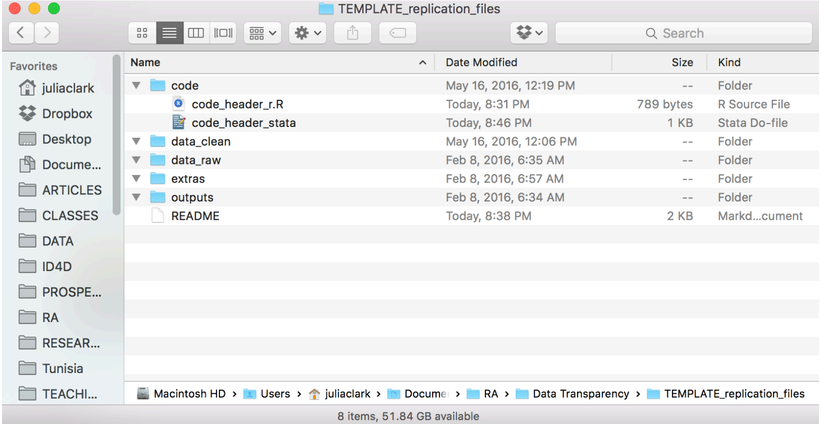
Create a ***new***, clearly organized folder structure for replication that you add to selectively.

- Purpose:
  - Ensure files are complete/parsimonious, legible
  - Protect original files [if you're using GitHub, you don't have to worry about this!]

Create

1. **A new, empty replication folder** within your project directory (e.g., "`replication_files` ")
2. **Subfolders:**
   - `/code` — scripts
   - `/data_clean` — manipulated data
   - `/data_raw` — original data
   - `/output` — generated tables, graphs, etc.
   - `/extra` — misc. extras (e.g., code book)
3. **A "README.txt" or "README.md" file** to document contents, sources, software/system versions, other info necessary for replication/comprehension.

PDEL Template



Name	Date Modified	Size	Kind
▼ code	May 16, 2016, 12:19 PM	--	Folder
code_header_r.R	Today, 8:31 PM	789 bytes	R Source File
code_header_stata	Today, 8:46 PM	1 KB	Stata Do-file
▼ data_clean	May 16, 2016, 12:06 PM	--	Folder
▼ data_raw	Feb 8, 2016, 6:35 AM	--	Folder
▼ extras	Feb 8, 2016, 6:57 AM	--	Folder
▼ outputs	Feb 8, 2016, 6:34 AM	--	Folder
README	Today, 8:38 PM	2 KB	Markd...cument

Macintosh HD › Users › juliaclark › Docume› › RA › Data Transparency › TEMPLATE_replication_files

8 items, 51.84 GB available

Note

If you're beginning a project, this is also a good way to start organizing your files! In that case, you might also want a folder called "`/draft`" to keep your paper drafts.

See also "reproducible_workflow.md" in the training folder for more suggestions on setting up a one-click system for new projects.

2. Initial Replication

Populate and run replication files

Copy (don't move!) over data and code files into the replications folder and try to replicate your results.

Purpose:

- Make sure your code actually runs and reproduces before you tinker with structure and formatting
- Build up your replication folder with complete and parsimonious data/code files

Step 1. Check Analysis

Easier to start with final analysis and work backwards to data cleaning/merging.

1. Copy original analysis script(s) into `replication_files/code`
2. Copy cleaned dataset(s) used for analysis into `replication_files/data_clean`
3. Run code without changes (except for wd)
4. Fix any bugs in the code, address discrepancies with previous results

Step 2. Check Data Clean/Merge

1. If separate from analysis, copy original merge/cleaning script(s) into `replication_files/code`
2. Copy original dataset(s) to `replication_files/data`
3. Run merge/clean code without changes (except for wd)
4. Rerun the analysis code from above on the newly cleaned/merged data file
5. If you get different results than step #1, there is a discrepancy with merging/cleaning code—fix it!

3. De-Identification

De-Identifying Individual-Level Data

Now you know the code works and replicates, congratulations! The next step is to ensure that any shared files *do not contain* data that could be used to identify individuals.

Purpose:

- Ensure you are protecting individuals' identity and private information—this is an ethical issue for researchers, and a potential safety issue for participants
- Comply with legal, research board or funder requirements (e.g., HIPAA and IRB in the US)

What does “de-identifying” mean?

Two types of identifiers:

1. Direct: Variables that is explicitly linked to the subject—*e.g., name, email, address, Aadhaar number, phone number, etc.*

2. Indirect: Variables that, in combination, could be used to identify individuals—*e.g., gender, dates (birth, program admission, etc.), geographic location (village, GPS), unusual occupations or education, etc.*

See this useful infographic.

Example of Indirect Identifiers

- You survey teachers and collect information on *gender*, *classes taught*, and *age*.

- If there is only one *female*, *third-grade* teacher *aged 40-49* at a particular school, she is not anonymous in your data

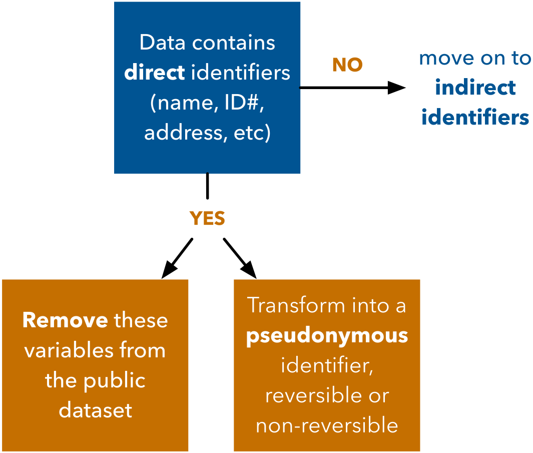
Dealing with Direct Identifiers

In general, direct identifiers—e.g., name, address, mobile number, ID number—should ***never*** be made public.

Options:

- Remove variables from shared dataset
- Pseudonymize data: replace identifiers with “pseudonyms” that may be reversible or non-reversible—e.g., give people random names or ID numbers—goal is to be able to link datasets

Solutions for Direct Identifiers



Data contains **direct** identifiers (name, ID#, address, etc)

NO

move on to **indirect identifiers**

YES

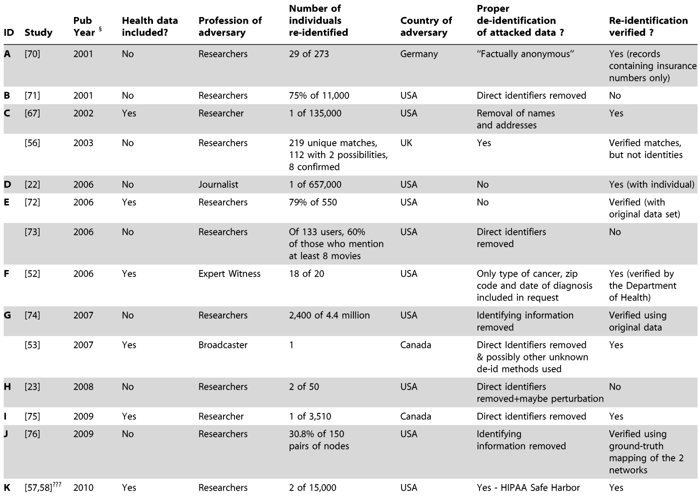
Remove these variables from the public dataset

Transform into a **pseudonymous** identifier, reversible or non-reversible

What is Sufficient De-Identification for Indirect Identifiers?

1. **Determine Risk** = Pr(de-identifying) × sensitivity of data
2. **Set k-anonymous level:** each record cannot be distinguished from at least $k-1$ other individuals who also appear in the data set
3. **Select appropriate method(s) of de-identification:** aggregating data, removing certain variables or observations, reducing information/detail, adding random noise or values

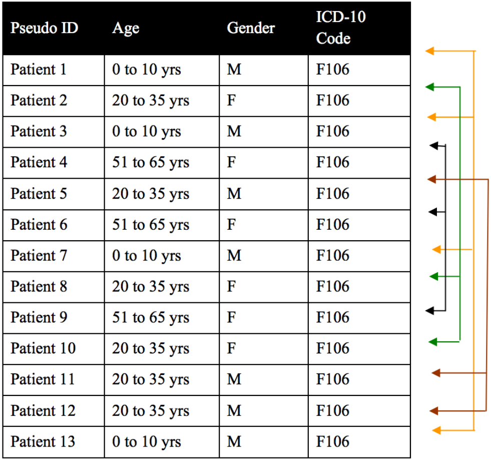
The Problem



ID	Study	Pub Year [§]	Health data included?	Profession of adversary	Number of individuals re-identified	Country of adversary	Proper de-identification of attacked data ?	Re-identification verified ?
A	[70]	2001	No	Researchers	29 of 273	Germany	"Factually anonymous"	Yes (records containing insurance numbers only)
B	[71]	2001	No	Researchers	75% of 11,000	USA	Direct identifiers removed	No
C	[67]	2002	Yes	Researcher	1 of 135,000	USA	Removal of names and addresses	Yes
	[56]	2003	No	Researchers	219 unique matches, 112 with 2 possibilities, 8 confirmed	UK	Yes	Verified matches, but not identities
D	[22]	2006	No	Journalist	1 of 657,000	USA	No	Yes (with individual)
E	[72]	2006	Yes	Researchers	79% of 550	USA	No	Verified (with original data set)
	[73]	2006	No	Researchers	Of 133 users, 60% of those who mention at least 8 movies	USA	Direct identifiers removed	No
F	[52]	2006	Yes	Expert Witness	18 of 20	USA	Only type of cancer, zip code and date of diagnosis included in request	Yes (verified by the Department of Health)
G	[74]	2007	No	Researchers	2,400 of 4.4 million	USA	Identifying information removed	Verified using original data
	[53]	2007	Yes	Broadcaster	1	Canada	Direct Identifiers removed & possibly other unknown de-id methods used	Yes
H	[23]	2008	No	Researchers	2 of 50	USA	Direct identifiers removed+maybe perturbation	No
I	[75]	2009	Yes	Researcher	1 of 3,510	Canada	Direct identifiers removed	Yes
J	[76]	2009	No	Researchers	30.8% of 150 pairs of nodes	USA	Identifying information removed	Verified using ground-truth mapping of the 2 networks
K	[57,58][???]	2010	Yes	Researchers	2 of 15,000	USA	Yes - HIPAA Safe Harbor	Yes

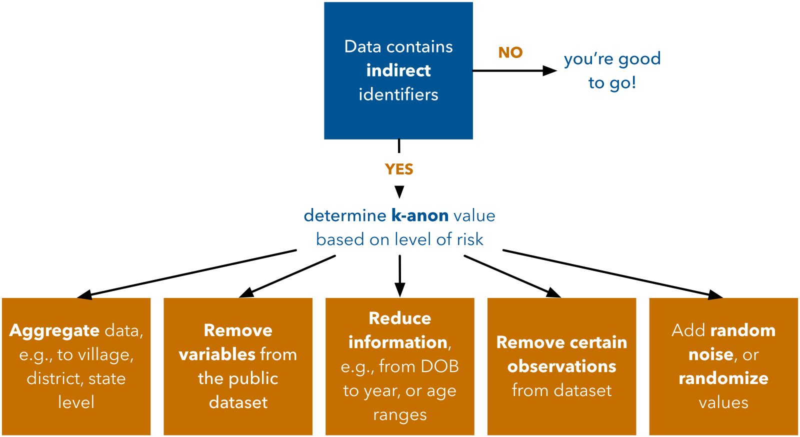
Source: El Emam et al. 2015. "A Systematic Review of Re-Identification Attacks on Health Data." PLOS One.

Example of K-anon where k=3



Pseudo ID	Age	Gender	ICD-10 Code
Patient 1	0 to 10 yrs	M	F106
Patient 2	20 to 35 yrs	F	F106
Patient 3	0 to 10 yrs	M	F106
Patient 4	51 to 65 yrs	F	F106
Patient 5	20 to 35 yrs	M	F106
Patient 6	51 to 65 yrs	F	F106
Patient 7	0 to 10 yrs	M	F106
Patient 8	20 to 35 yrs	F	F106
Patient 9	51 to 65 yrs	F	F106
Patient 10	20 to 35 yrs	F	F106
Patient 11	20 to 35 yrs	M	F106
Patient 12	20 to 35 yrs	M	F106
Patient 13	0 to 10 yrs	M	F106

Solutions for Indirect Identifiers



Data contains **indirect** identifiers

NO

you're good to go!

YES

determine **k-anon** value based on level of risk

Aggregate data, e.g., to village, district, state level

Remove variables from the public dataset

Reduce information, e.g., from DOB to year, or age ranges

Remove certain observations from dataset

Add **random noise**, or **randomize** values

Trade-off: Usefulness $\Longleftrightarrow$ Anonymity

- **Aggregating**—lose ability to replicate any individual-level analysis
- **Removing variables**—may not be able to replicate specific models
- **Reducing information**—adds noise to models
- **Remove observations**—adds bias if non-random
- **Adding random noise/values**—adds noise

See here and here for more discussion of appropriate thresholds, methods, and tools for de-identification.

Good Practices

- Include code for de-identified data for transparency (as long as the code itself doesn't compromise anonymity)
  - e.g., censor code that sets the seed for a random draw to generate new ID numbers and could be used to re-identify individuals
- If identifiers ***aren't*** used for analysis, de-identify early in merging/cleaning process
- Store original data with PII securely—if you're using Dropbox, see PDEL GitHub wiki for tips on sharing with RAs in a way that protects PII data

4. Editing

Edit and Organize Files for Clarity

Now we have working files that are de-identified; the next step is to clean and annotate so the so they are organized and written in a logical, user-friendly way.

Purpose:

- Ensure files are legible in terms of structure and content

Basic steps

1. Structure and name files
2. Streamline and annotate code
3. Document file and folder contents

Step 1: Structure and Name Files

- Create separate scripts for merging/cleaning and data analysis, with a master-script for running it all
- Give code and data files logical names where possible (and remember to change file paths in code where necessary!)
  - e.g., Number folders/files sequentially in the order they should be run

Step 2: Streamline & Annotate Code

- Use working directories (and R projects)
- Move exploratory analysis to end of script—good for posterity, but shouldn't obscure main code
- Add headers (see PDEL template)
- Format scripts so they're easily readable—e.g., indent code, use ample line breaks and spaces, standardize comment syntax

Step 2: Streamline & Annotate Code (Cont.)

- Add comments to improve reader understanding; remove unhelpful/embarrassing comments
- Clearly label code sections, main analyses, outputs
- Give variables intuitive names like `edu_percent` rather than `v76`
- Give output objects intuitive names like "table_main_results"
- Label variables and values in Stata

Working directory

R: `setwd("~/Documents/replication_files")`
Stata: `capture cd "~/Documents/replication_files"`

- Saves you time, since you only have to change the path once if the files move AND your code will be shorter
- Someone replicating your files also only needs to change the file path once
- Particularly helpful if switching between Mac ("/") and Windows ("\") file extensions

Step 3: Document File and Folder Content

- Update the README file to describe contents of replication folders
- If necessary, include codebook in "`/extra`" folder
- Track and document packages, software versions
  - R: `sessionInfo()`
  - Stata: `version`

5. Final Replication

One more time

Now that you have cleaned/reorganized script files …

- Shutdown or clear your Stata/R memory
- Rerun the entire process—including data merging, cleaning and analysis—to make sure the editing process didn't break anything
- Testing on a friend (or RA's) computer can also be a final check
- Once discrepancies are addressed, the files are ready for sharing!

Activity

Prepping Replication Files

1 **Set up:** create separate replication folder structure

2 **Initial replication:** first analysis, then cleaning/merging

3 **De-Identify:** remove/ alter identifying data

4 **Edit:** clean and organize files

5 **Final replication:** double check that everything runs

References

- Tools for De-Identification
- El Emam. 2010. Risk-based De-Identification of Health Data.
- Christensen. 2016. Manual of Best Practices In Transparent Social Science.
- Gentzkow & Shapiro. 2014. Code and Data for the Social Sciences: A Practitioner's Guide
- J. Scott Long. 2008. The Workflow of Data Analysis Using Stata.
- Christopher Gandrud. 2013. Reproducible Research in R and R Studio.