

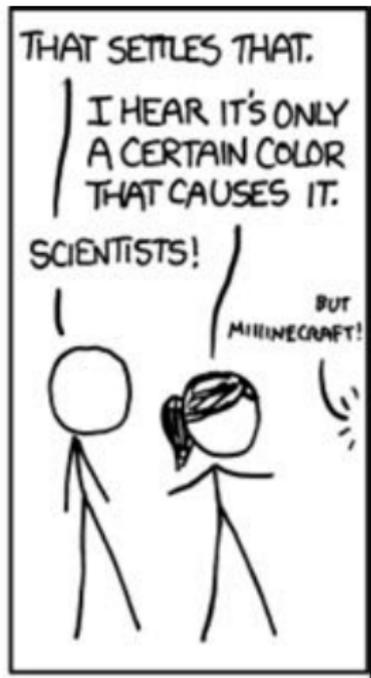
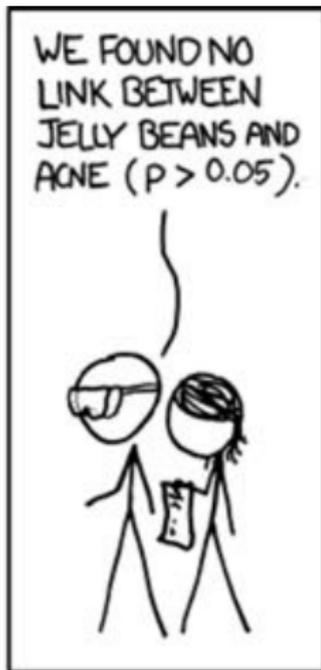
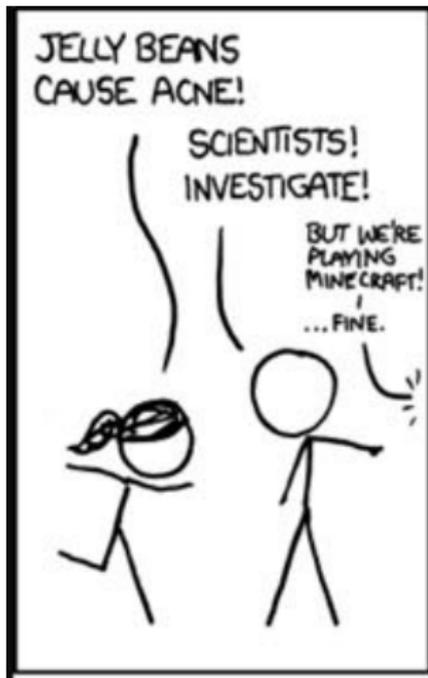
How to improve the credibility of (your) social science

A practical guide for researchers



October 2, 2017

The Problem



WE FOUND NO
LINK BETWEEN
PURPLE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BROWN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
PINK JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BLUE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TEAL JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
SALMON JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
RED JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TURQUOISE JELLY
BEANS AND ACNE
($P > 0.05$).

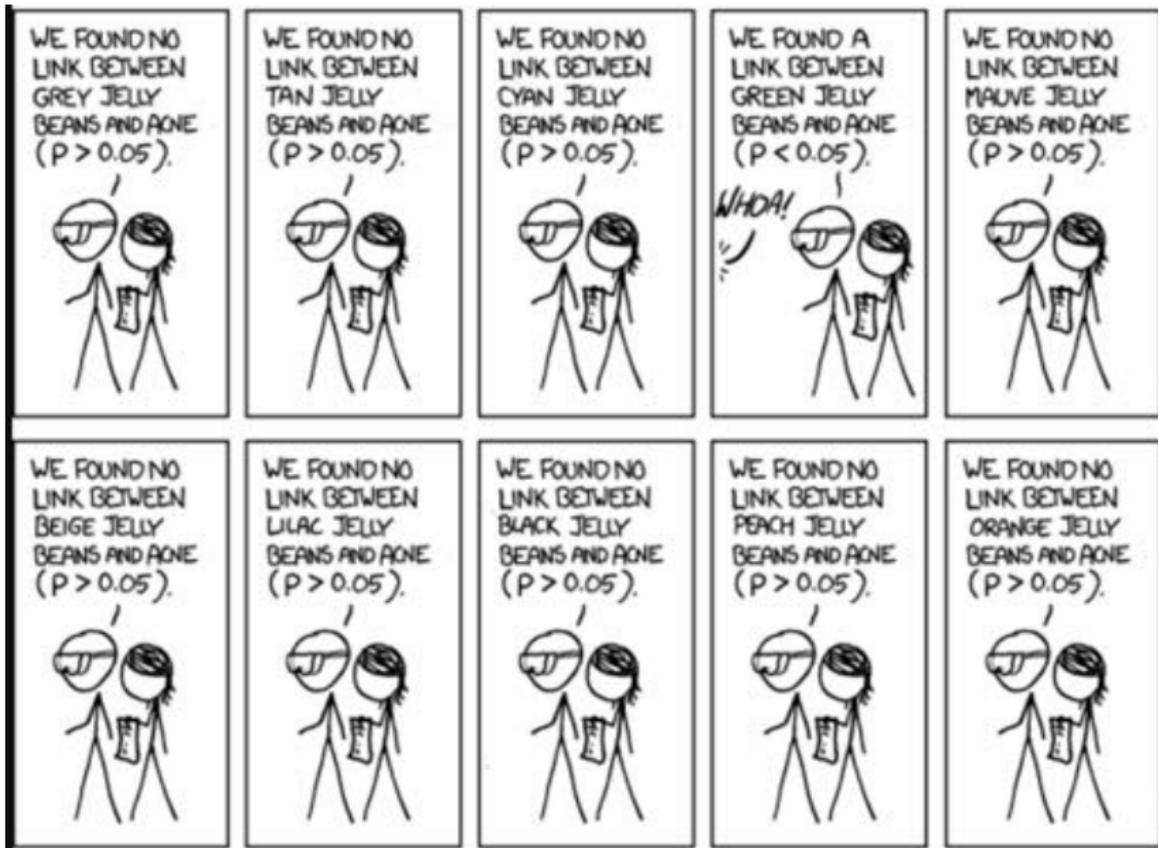


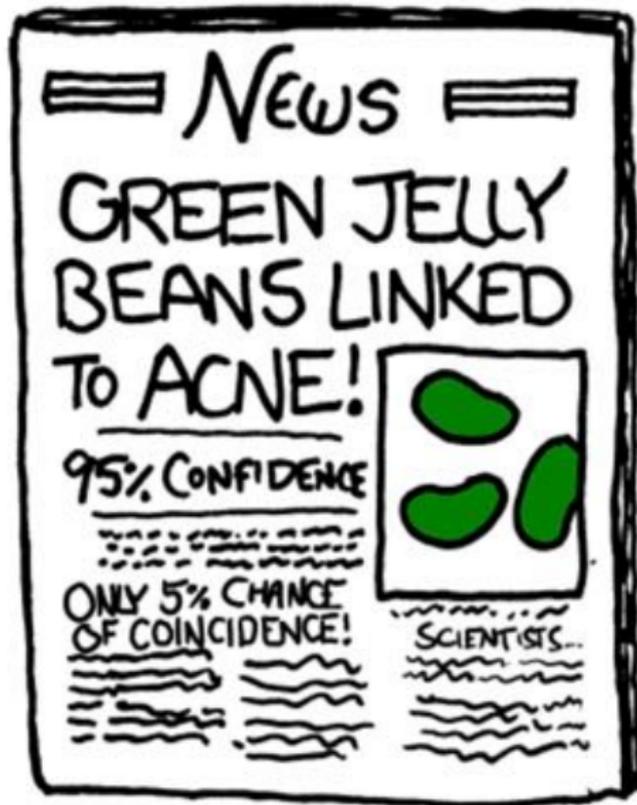
WE FOUND NO
LINK BETWEEN
MAGENTA JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
YELLOW JELLY
BEANS AND ACNE
($P > 0.05$).







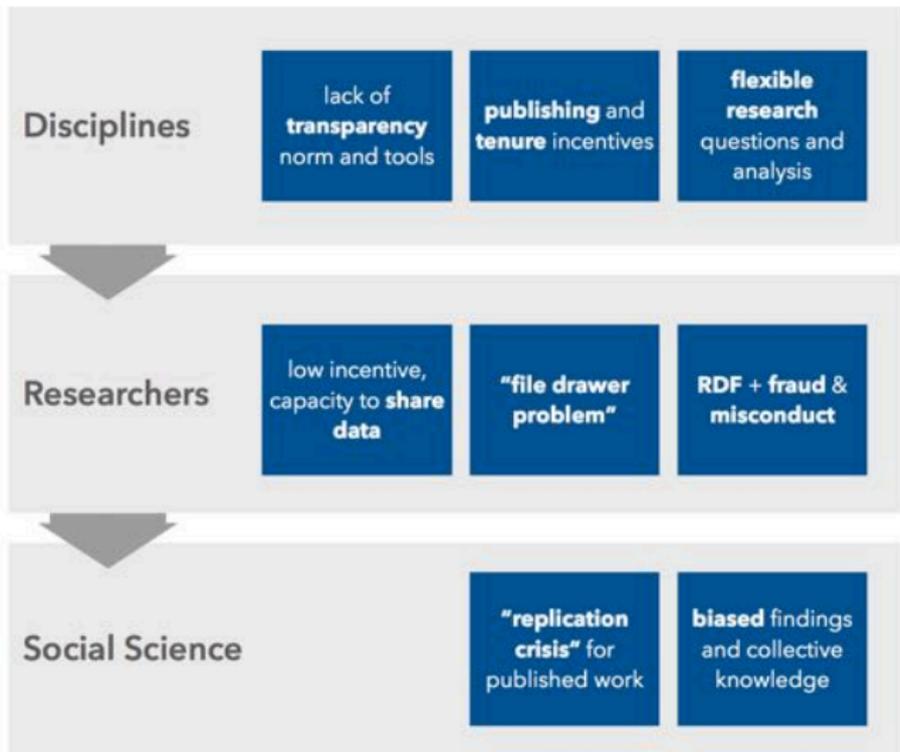
Source: XKCD

We often hear about ...

1. **“Replication crisis”**—studies fail to replicate (psych, econ, polisci, medicine, etc.)
2. **Publication bias**—published studies only represent fraction of results, biased toward significant positive findings
3. **P-hacking/researcher degrees of freedom**—published studies use only a fraction of possible specifications, biased toward significance
4. **Misconduct/fraud**—relatively easy to get away with!

→ adds up to **biased body of knowledge**

Why do we have this credibility crisis?



1. “Replication Crisis”

Social, behavioral, and medical studies often don't replicate

- ▶ **Ideally**, replications determine if original results are robust to alternative specifications or sample if they were due to *random chance*.
- ▶ **In reality**, failure to replicate often a result of ...
 - ▶ Lack of transparency in sharing data/code
 - ▶ Errors in data/code
 - ▶ Misconduct or fraud

Dewald et al. (1986)

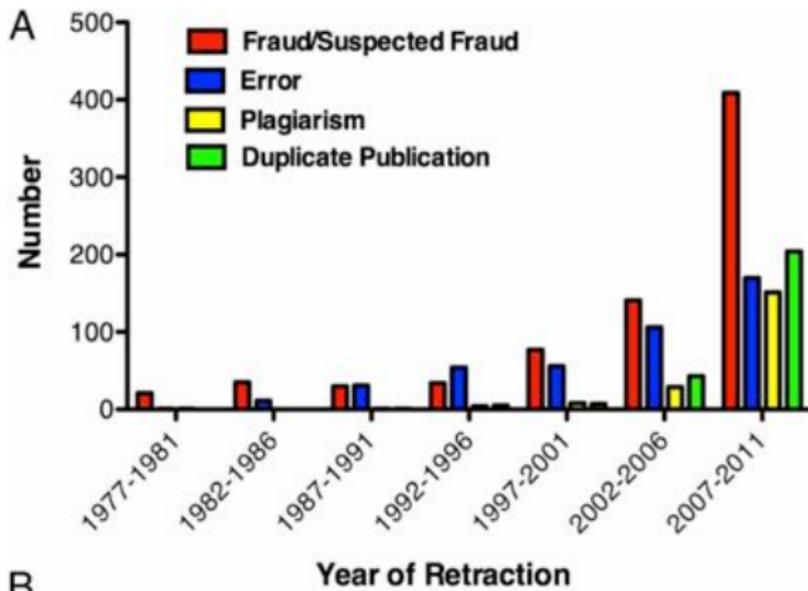
Attempted to replicate papers submitted to *Journal of Money, Credit and Banking*:

TABLE 2—PROBLEMS IN SUBMITTED DATA SETS

	Published before Data Requested	Accepted before Data Requested	Under Review when Data Requested
No Problems	1	3	4
Problems Identified:			
Incomplete Submission	6	3	5
Sources Cited Incorrectly	0	4	4
Sources Cited Imprecisely	11	7	10
Data Transformations	3	4	1
Described Incompletely			
Data Element Not Clearly Defined	2	3	2
Other	0	3	1
Problems	<u>22</u>	<u>24</u>	<u>23</u>
Data Sets Examined	19	14	21

Fang et al. (2012)

Review of 2,047 retracted biomedical and life-science articles on PubMed:



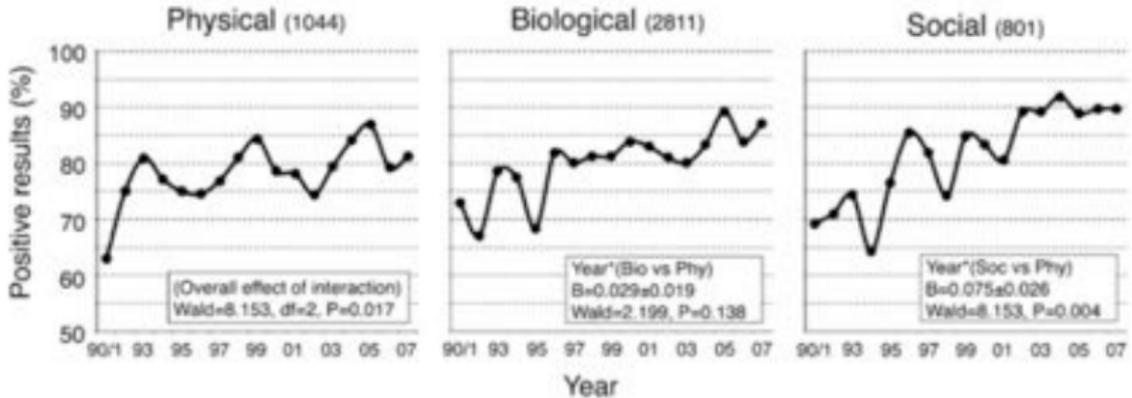
2. Publication Bias

AKA the “file drawer problem”

- ▶ **Problem:** Studies more likely to be submitted/published when findings are significant → studies with null (or negative) findings are hidden
- ▶ **Result:** Bias evidence base—we’re missing full universe of studies and results; what gets published could be due to random chance (e.g., if we expect 5% of results of all studies to be significant)

Fanelli (2010 & 2011)

Increase in % of papers with positive results over time, across scientific disciplines:



Franco, Malhotra, Simonovits (2014)

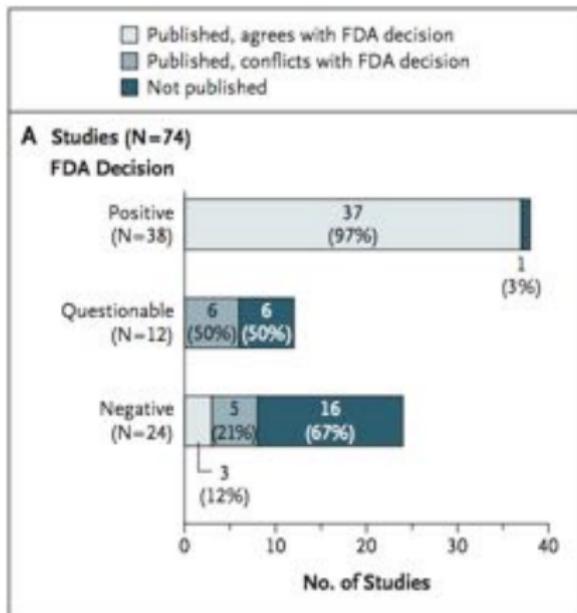
Strong results 60pp more likely to be written up than null results, 40pp more likely to be published:

Table 3. Cross-tabulation between statistical results of TESS studies and their publication status (column percentages reported). Pearson χ^2 test of independence: $\chi^2(6) = 80.3, P < 0.001$.

	Null (%)	Mixed (%)	Strong (%)
Not written	64.6	12.2	4.4
Written but not published	14.6	39.0	34.1
Published (non-top-tier)	10.4	37.8	38.4
Published (top-tier)	10.4	11.0	23.1
Total	100.0	100.0	100.0

This has consequences!

E.g., studies that agree with FDA decisions more likely to be published (Turner et al. 2008):



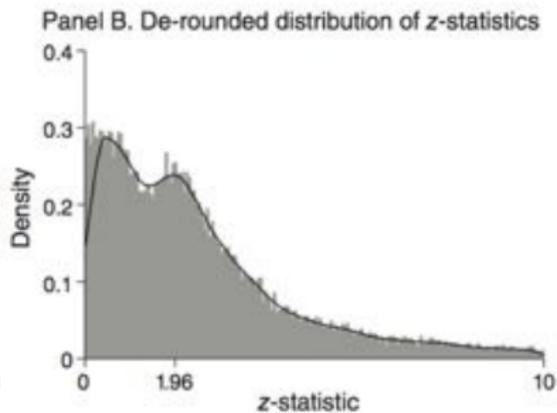
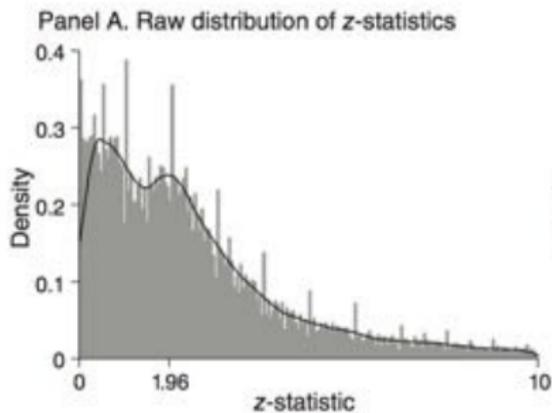
3. P-hacking—AKA fishing, data mining, specification searching, etc.

“Torture the data until it tells you what you want to hear”

- ▶ **Opportunity:** Researchers also have many “degrees of freedom” (RDF) in the design and analysis of a study → p-hacking (may not always be intentional, see Gelman & Loken 2013)
- ▶ **Motive:** Researchers have incentives (from journals, tenure requirements, etc.) to find significance
- ▶ **Result:** Biased evidence base (also contributes to replication crisis)

Brodeur et al. (2016)

Evidence of P-Hacking:



Wicherts et al. (2016)

Identify 34 key researcher DFs (see [article](#) for full list):

Table 1

Checklist for different types of degrees of freedom in the planning, executing, analyzing, and reporting of psychological studies

Code	Related	Type of Degrees of Freedom
Hypothesizing		
T1	R6	Conducting explorative research without any hypothesis
T2		Studying a vague hypothesis that fails to specify the direction of the effect
Design		
D1	A8	Creating multiple manipulated independent variables and conditions
D2	A10	Measuring additional variables that can later be selected as covariates, independent variables, mediators, or moderators
D3	A5	Measuring the same dependent variable in several alternative ways
D4	A7	Measuring additional constructs that could potentially act as primary outcomes
D5	A12	Measuring additional variables that enable later exclusion of participants from the analyses (e.g., awareness or manipulation checks)
D6		Failing to conduct a well-founded power analysis
D7	C4	Failing to specify the sampling plan and allowing for running (multiple) small studies ...

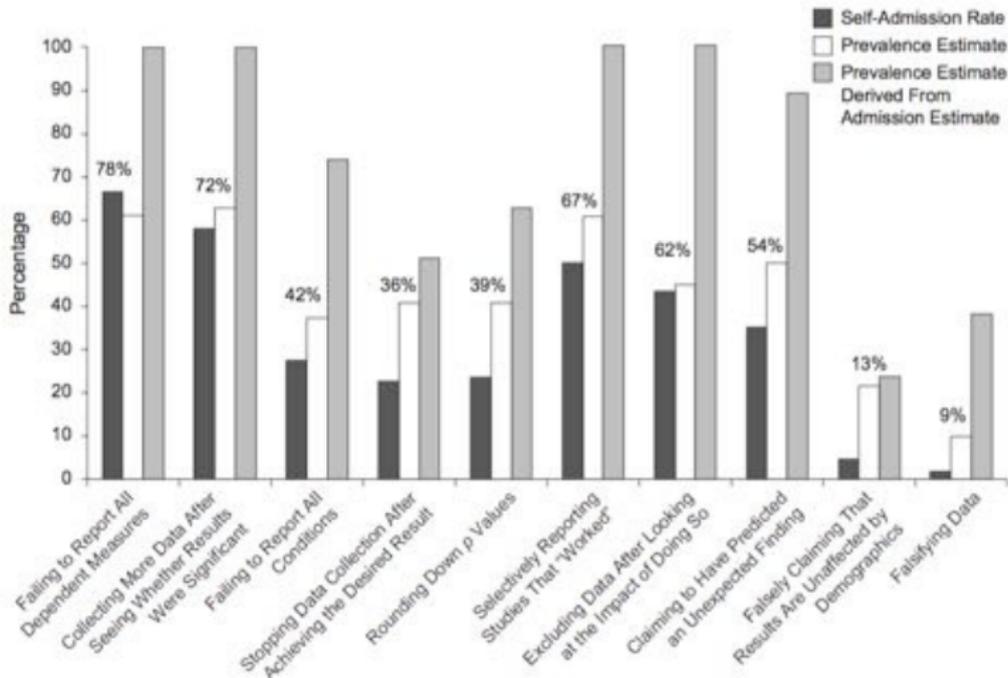
4. Misconduct & Fraud

Rare(?) but serious

- ▶ **Includes:** Falsifying some or all data and/or results, as well as plagiarism and other forms of misconduct
- ▶ **Result:** False or biased evidence base, (also contributes to replication crisis)
- ▶ **Note:** Fabrication of data (e.g., **LaCour**, **Fujii**, **Foster**, **Staple**) less common than other “questionable research practices”

John et al. (2012)

Survey of 2000 psychologists on questionable practices:



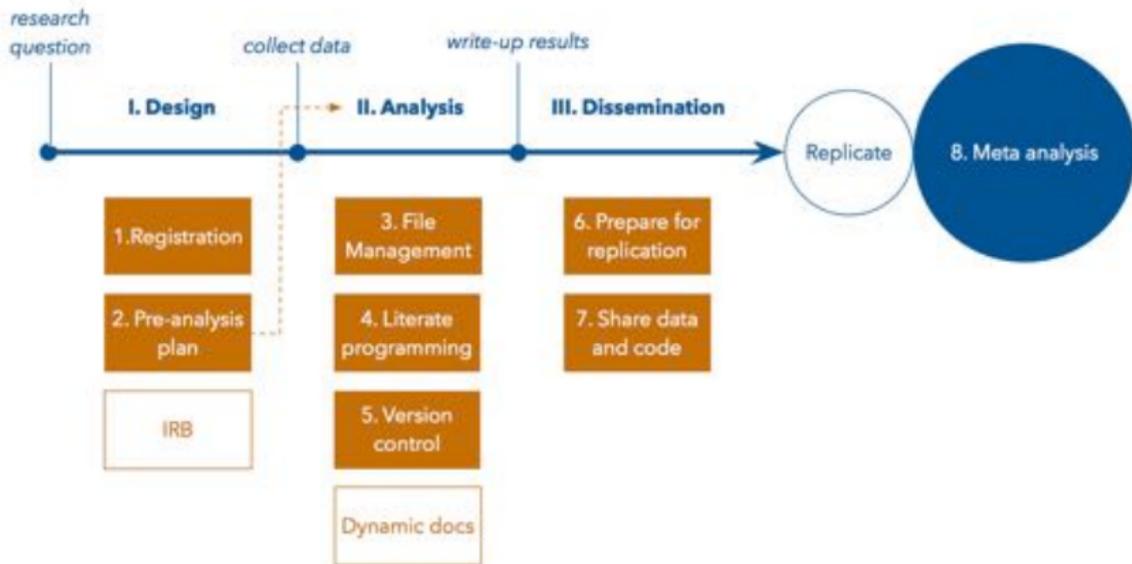
But all hope is not lost ...

Norms are changing

Smart people are working on these issues and developing standards and tools to help throughout the **research lifecycle**.

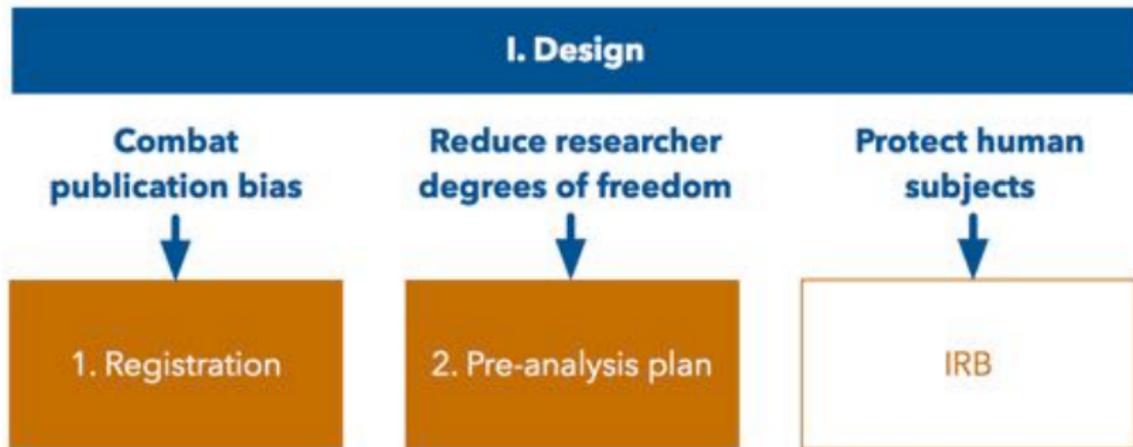
- ▶ PDEL, BITSS, OSF, DART, Dataverse, EGAP, etc. etc.

Research lifecycle: Individual-level solutions



Solutions I: Design

Steps



1. Registration

About Registration

- ▶ **What:** Enter your study into the appropriate disciplinary “registry”—basically a requirement for experiments (especially in medicine)
- ▶ **Why:** To combat the file-drawer problem, publication bias— also, stake out intellectual claim!

Where to Register

- ▶ American Economics Association (AEA):
<http://socialscienceregistry.org>
- ▶ Experiments in Governance and Politics (EGAP):
<http://egap.org/design-registration>
- ▶ Registry for International Development Impact Evaluations (3ie): <http://ridie.3ieimpact.org>
- ▶ Open Science Framework: <http://osf.io>—OSF is integrated with other formats, soon with AEA!
- ▶ <http://aspredicted.org>

AEA

To register an experimental study with AEA ...

1. Create an account at <https://www.socialscienceregistry.org>
2. Click on “register a trial” and enter basic information—including title, country, status, keyword, abstract, start and end dates, outcomes, experimental design, whether treatment clustered, planned number of clusters and observations, IRB information

EGAP

To register an experimental (or non-experimental) study with EGAP ...

1. If you're not already in the EGAP author database, go to <http://egap.org/node/add/people> to add your name and basic information
2. Go to <http://egap.org/node/add/registration> and complete the registration form—including faculty affiliation, prospective vs. retrospective, whether experimental, start date, background on study, hypotheses to be tested, basic research design, sample size, whether power analysis, IRB information, and keywords

2. Pre-Analysis Plan

About Pre-Analysis Plans (PAPs)

- ▶ **What:** Detailed description of research design and data analysis plans, submitted to a registry BEFORE looking at the data.
- ▶ **Why:**
 - ▶ Tie your hands for data analysis (address researcher degrees of freedom, etc.)
 - ▶ Distinguish between *confirmatory* and *exploratory* analysis
 - ▶ Boost credibility of research (get a badge from OSF!)
 - ▶ Transparent methods make it easier for others to build on your work

PAP vs. Registration

Registration often—but not always—includes a pre-analysis plan. BUT, purpose is different ...

- ▶ **Registration addresses publication bias**—study enters the universe, no matter the outcome
- ▶ **PAP addresses p-hacking**—limiting degrees of freedom

Where to Submit a PAP

Generally, upload as *part* of registration process ...

- ▶ American Economics Association (AEA):
<http://socialscienceregistry.org>
- ▶ Experiments in Governance and Politics (EGAP):
<http://egap.org/design-registration>
- ▶ Registry for International Development Impact Evaluations (3ie): <http://ridie.3ieimpact.org>
- ▶ Open Science Framework: <http://osf.io>

OSF

- ▶ Goal is one-stop hub for transparency across scientific disciplines
- ▶ Make an account and explore at <https://osf.io/>
- ▶ Win \$1000 with **Preregistration Challenge**

No universal standard, can include ...

Background	abstract, motivation, questions
Design	treatment, sampling & randomization, attrition, spillover, survey instruments, power calculations, plan for data collection, processing & management
Analysis	hypotheses (main, auxiliary), outcome measures (primary, secondary), variable operationalization, balance checks, estimation of treatment effects (ATE, ITT, TOT, etc.), HTEs (subgroups, interactions), covariates, standard errors, corrections for multiple hypothesis testing, missing values, outliers
Team	members, affiliations, conflicts of interest
Logistics	fieldwork, timeline, budget

Olken's PAP Checklist (2013)

<i>Item</i>	<i>Brief description</i>
Primary outcome variable	The key variable of interest for the study. If multiple variables are to be examined, one should know how the multiple hypothesis testing will be done.
Secondary outcome variable(s)	Additional variables of interest to be examined.
Variable definitions	Precise variable definitions that specify how the raw data will be transformed into the actual variables to be used for analysis.
Inclusion/Exclusion rules	Rules for including or excluding observations, and procedures for dealing with missing data.
Statistical model specification	Specification of the precise statistical model to be used, hypothesis tests to be run.
Covariates	List of any covariates to be included in analysis.
Subgroup analysis	Description of any heterogeneity analysis to be performed on the data.
Other issues	Other issues include data monitoring plans, stopping rules, and interim looks at the data.

Tie your hands in the right places



→ requires a lot of thought!

Ongoing Debate

- ▶ **Olken (2013)** on “Promises and Perils of Pre-analysis Plans”
- ▶ **Coffman & Niederle (2015)** argue that “Pre-analysis Plans Have Limited Upside, Especially Where Replications Are Feasible”
- ▶ More debate on utility for observational work but can be done (see **Neumark 2001**)

[IRB]

Not covered here, but ...

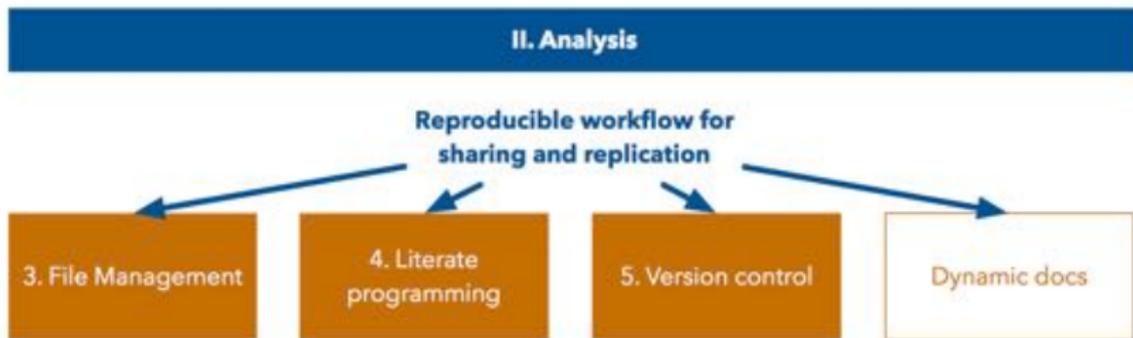
Don't forget **IRB requirements** to protect human subjects!

Necessary for ethical research, though not sufficient (see <http://desposito.org/ethicsfieldexperiments.pdf> for more on ethics in experiments).

Solutions II: Analysis

Steps

“**Reproducibility** is just collaboration with people you don’t know, including yourself next week” — Philip Stark, UC Berkeley

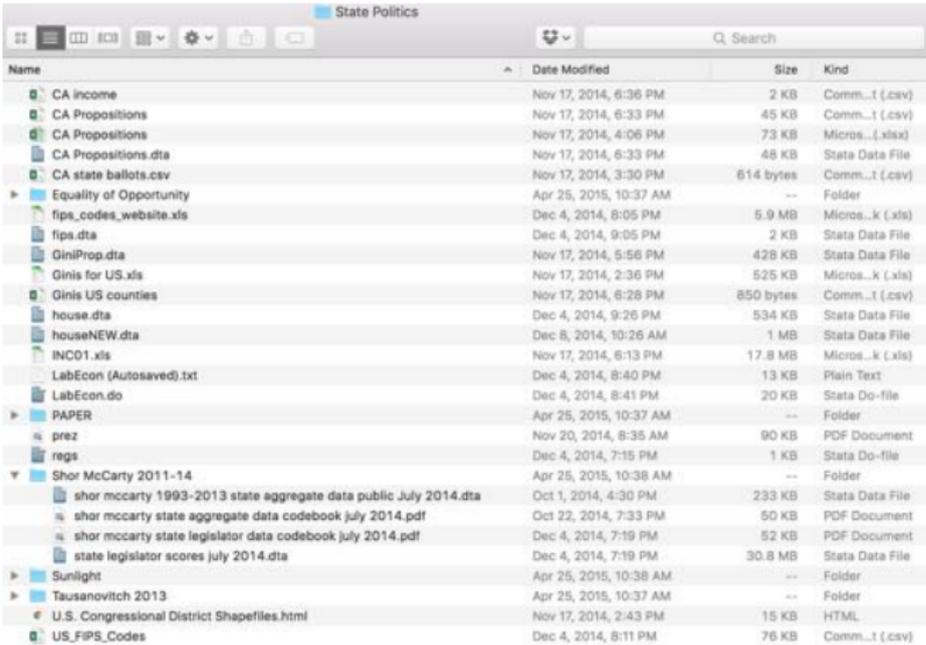


3. File Management

About File Management

- ▶ **What:** Organizing and managing files cleanly and intuitively
- ▶ **Why:** To preserve original data, streamline workflow, and reduce prep time when sharing files

Don't let your files look like this ...



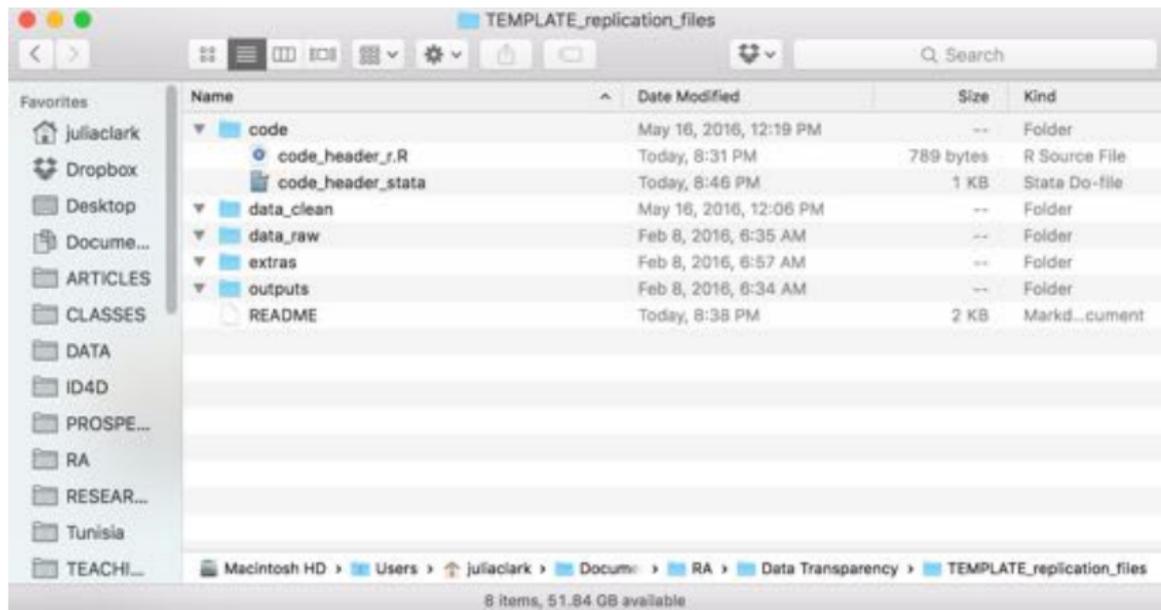
The screenshot shows a Finder window titled "State Politics" with a search bar and various icons. The file list is as follows:

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
INCO1.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)

Instead, use PDEL template (or similar)

Download at [https://github.com/](https://github.com/PolicyDesignEvaluationLab/Transparency-Initiative)

PolicyDesignEvaluationLab/Transparency-Initiative



4. Literate Programming

About Literate Programming

- ▶ **What:** Writing code that it's legible to *humans*
- ▶ **Why:** So you and others can better replicate your work (and to help you avoid mistakes!)

(The Most) Basic Principles

- ▶ Structure and name files intuitively
- ▶ Make the contents of files easy to navigate
- ▶ Streamline code to avoid repetition

Structure and Name Files

- ▶ Create separate scripts for merging/cleaning and data analysis, with a master-script for running it all
- ▶ Give code, data files, and output logical names where possible
 - ▶ Number scripts sequentially in the order they should be run (e.g., `1_main_analysis.R`, `2_robust_checks.R`)
 - ▶ Label output figures with descriptive names, but ones that aren't likely to change (e.g., `figure_hte.png` is better than `figure_1.png`)

Improve navigation

- ▶ 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
- ▶ Add comments to improve reader understanding
- ▶ Clearly label code sections, main analyses, outputs
- ▶ Give functions, objects, and variables intuitive names like `edu_percent` rather than `v76`
- ▶ Label variables and values in Stata

Streamline Code—e.g., working directories

R: `setwd("~/Documents/replication_files")`

Stata: `capture cd "~/Documents/replication_files"`

- ▶ Saves you time, since you (or someone replicating your study) only have to change the path once if the files move AND your code will be shorter
- ▶ Particularly helpful if co-authors alternate between Mac (“/”) and Windows (“\”) file extensions

5. Version Control

"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JOBAS Coker © 2012

WWW.PHDCOMICS.COM

About Version Control

- ▶ **What:** A system for managing iterative versions of files (code, data, manuscripts) over time and across collaborators
- ▶ **Why:** Keep original files, protect work, collaborate efficiently, streamline workflow, etc., etc.

Principles of Version Control

- ▶ Vault original, raw data files—do not save over!
- ▶ Changes to files should be documented and reversible
- ▶ Keep “master” versions of files in working order; create copies before experimenting
- ▶ Reconcile independent changes by different users

Manual Solutions (not ideal, but better than nothing)

- ▶ Create dated versions of files (save-as) for each substantive change
- ▶ With each modification, re-run ALL code to make sure nothing is broken—helps if you have a master file to run all scripts!
- ▶ Check-in with coauthors to ensure multiple people aren't working on the same files at the same time
- ▶ Keep a simple log to remind yourself of the location/content of major changes

Or let version control software do this for you!



GitHub

Version control software > Git > GitHub

- ▶ **Version control software:** helps manage versions and edits to files (e.g., Microsoft Word’s “track changes”, or Google Doc’s “suggestion” feature)—**many options!**
- ▶ **Git:** Open-source, “distributed model” of version control developed by creator of Linux
- ▶ **GitHub:** Free, web-based service that hosts Git “repositories” and offers a variety of features for collaboration

Common problems that GitHub helps solve

- ▶ **Tracking changes in code/text files**—who, what, where, when, preserved forever
- ▶ **Selectively reverting changes**—better than `ctrl + Z`
- ▶ **Experimenting**—easier than “my_code_v2_new.R”
- ▶ **Collaborating**—sharing/vetting/reconciling changes

How do I use GitHub?

- ▶ **GitHub website**—necessary for collaboration, but limitations
- ▶ **GitHub Desktop**—free desktop client for Windows/Mac, more user friendly than website
- ▶ **Command line (shell)**—optimal for advanced users

How to think about Git



Tell Git to **watch a set of files** (“repository”) and it tracks every change within them, line-by-line.*

*If they are text/code files (e.g., .txt, \LaTeX , Markdown, Stata, .R, etc.). Git’s not really useful for PDF, Word, Excel (sorry).

GitHub is NOT ...



(GitHub.com looks like cloud-based drive, but primary purpose is collaboration, not storage)



(Desktop app looks like file manager, but use to view changes, not to navigate to/open files)

(The most) basic vocabulary

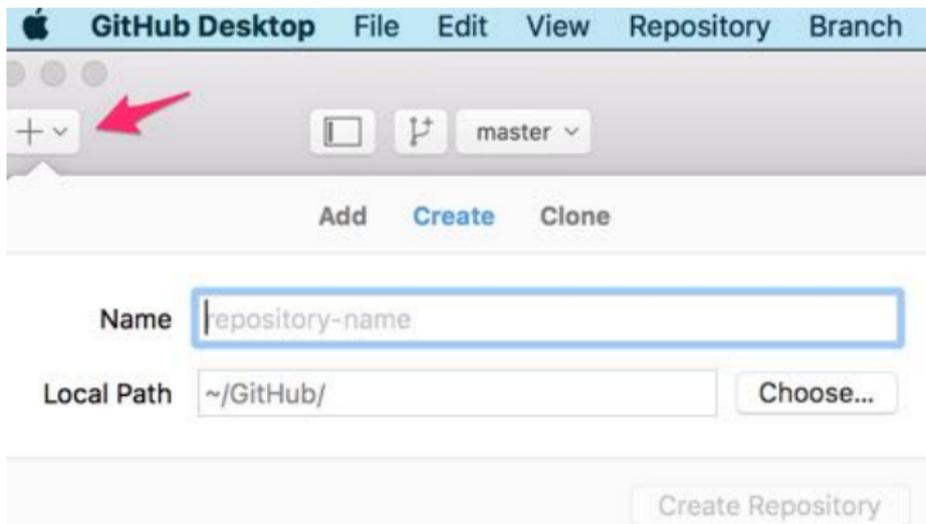
- ▶ **Repository:** A set of files (in a folder) that you have told Git to track, along with its associated .git files. **Local** repository = copy on your computer; **remote** repository = copy synced online.
- ▶ **Commit:** A labeled change or series of changes to files. Git tracks every change you make, and then you group these changes as desired into a “commit” that can be commented on, reverted, etc.

10 Baby Steps in Git—Prep

- ▶ **Make sure you have a good text editor.** Notepad or TextEdit will work (if you set TextEdit to `.txt` and not `.rtf`). Or get a more powerful editor like **Atom**.
- ▶ **Create an account at GitHub.** This gives free *public* repositories, but click “request a discount” at [for free private](#) repositories.
- ▶ **Download and install GitHub Desktop.** Then open and log in using your GitHub account.

1. Create a NEW repository

Within **GitHub Desktop**, click on “+” and then “create” to make a new repository with a name and location of your choice. This creates a new folder that will be empty except for some hidden files (e.g., a .git directory).



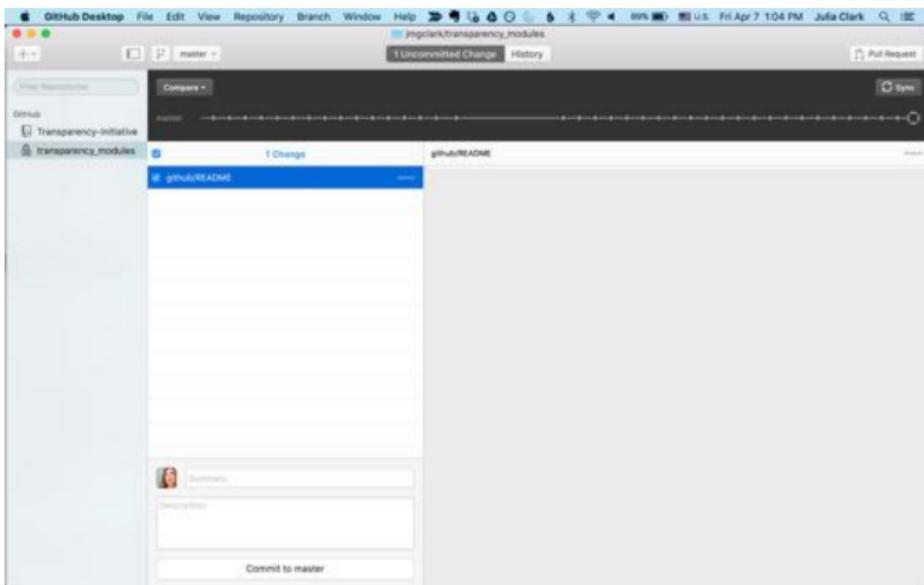
2. Add a text file to your repository

Leave the Desktop app and go to your text editor:

- ▶ Create a new text file called “README” and *save it in your repository location*.
- ▶ This should be a plain text file (.txt) or Markdown file (.md), NOT a rich text format file (.rtf).

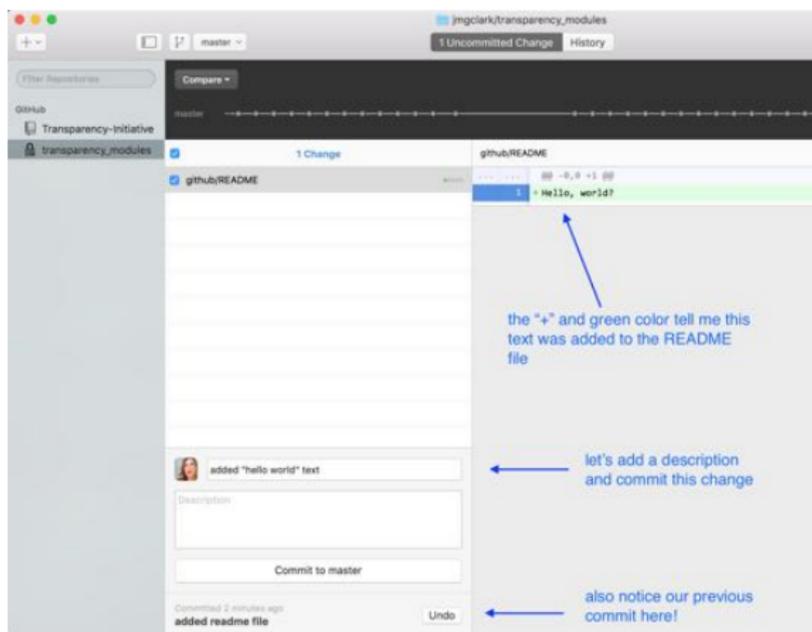
3. Commit this change in GitHub Desktop

Commit (i.e., record) your change of adding README by writing a summary and clicking "Commit to master".



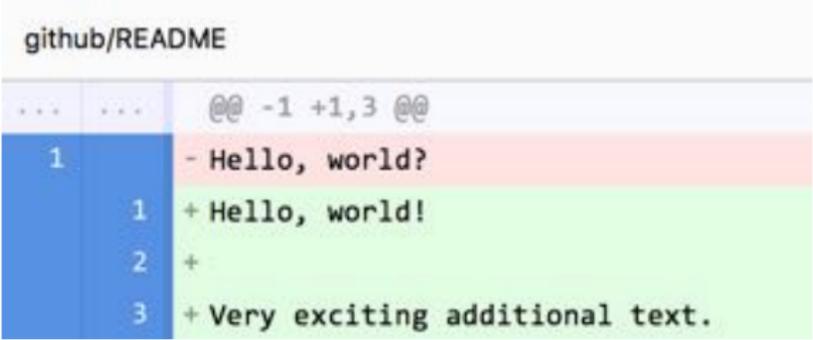
4. Add text to README and commit changes

Add some text to your file and save. If you go back to the Desktop client, you will now see something like this:



5. Edit README text and commit changes

Make and save changes to your text, then go back to GitHub Desktop. In the right-hand pane, additions will appear in green and deletions will appear in red:

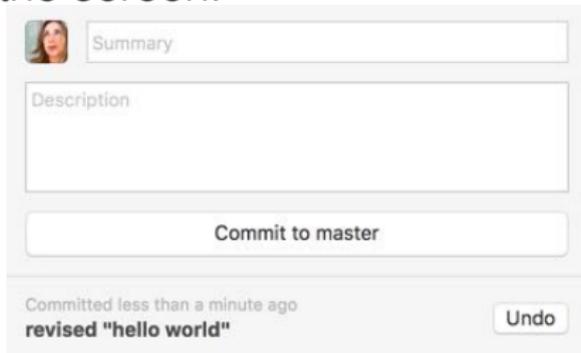


```
github/README
...  ...  @@ -1 +1,3 @@
1    - Hello, world?
    1    + Hello, world!
    2    +
    3    + Very exciting additional text.
```

Note that the unit of change is the *paragraph*, so changing “?” to “!” involved deleting/adding the whole phrase.

6. Undo the last commit

If you're unhappy with your LAST commit (i.e., you disliked how it was grouped or labeled), click **“Undo”** at the bottom of the screen:

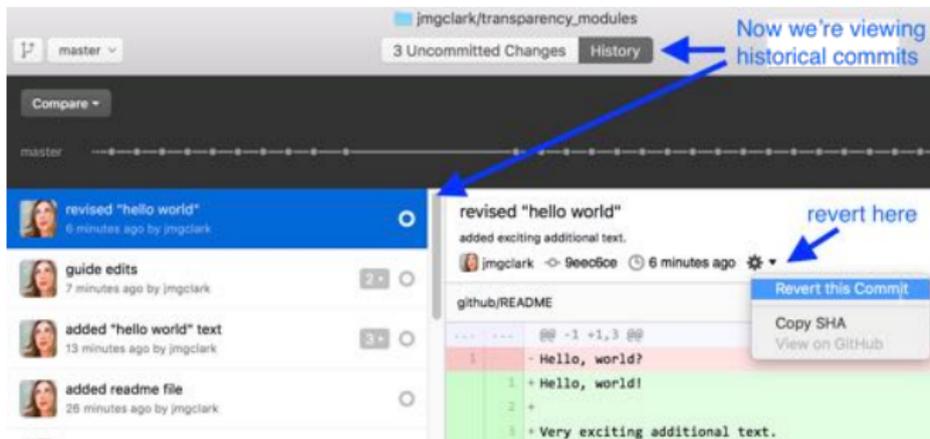


The screenshot shows a commit interface with a profile picture on the left, a 'Summary' text box, and a larger 'Description' text box. Below these is a 'Commit to master' button. At the bottom, it says 'Committed less than a minute ago' followed by 'revised "hello world"' in bold. To the right of this text is an 'Undo' button.

Now, these changes will appear again as “uncommitted changes” for you to regroup or relabel.

7. Revert a previous commit

If you're unhappy with the CHANGES in a commit themselves, you can “revert” them. → **switch to the “History” tab** at the and view all your previous commits. Select one, navigate to the dropdown menu, and click “Revert”:



8. Publish repository to your online account

We've been working in a **local repository**—one that that you created on your computer.

To collaborate you'll need to publish the repository to the web (i.e., make a **remote repository**). → **Click “publish”**:



9. View your repository & changes online

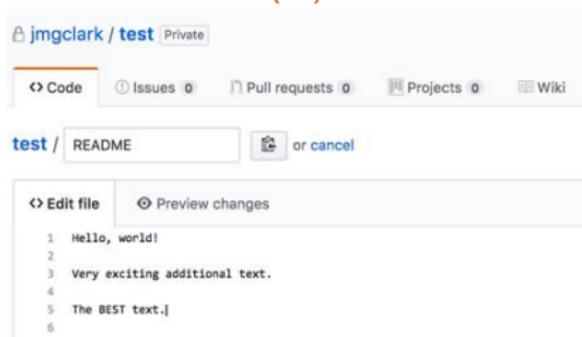
When you login to GitHub online, you'll see the new repository and file you've added.

The screenshot shows a GitHub repository page for 'jmgclark / test' (Private). At the top, there are navigation links for Code, Issues, Pull requests, Projects, Wiki, Settings, and Insights. Below these are repository statistics: 3 commits, 1 branch, 0 releases, and 1 contributor. A 'Branch: master' dropdown and a 'New pull request' button are visible. Action buttons include 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A commit history section shows 'jmgclark edited intro' with the latest commit '49f4a7' from 5 minutes ago. Below this, a file named 'README' is listed as 'edited intro' from 5 minutes ago. The content of the README file is displayed in a code block: 'Hello, world!' followed by 'Very exciting additional text.'

10. Edit the file online & sync with local repository

Click on the README file and then click the edit button (the pen). **(A) Make some changes and then commit.** Then go back to the Desktop client and click “Sync”. **(B)** Your new commit will appear in the **history tab**.

(A)



The screenshot shows the GitHub web editor interface for a file named 'README'. At the top, it says 'jmgclark / test Private'. Below that are navigation tabs for 'Code', 'Issues', 'Pull requests', 'Projects', and 'Wiki'. The file path 'test / README' is shown with a 'cancel' button. The main area has two tabs: 'Edit file' (selected) and 'Preview changes'. The code editor shows the following content:

```
1 Hello, world!  
2  
3 Very exciting additional text.  
4  
5 The BEST text.]  
6
```

(B)



The screenshot shows the GitHub commit history for the README file. It displays a commit by 'jmgclark' (918f165) made '2 minutes ago'. The commit message is 'added one more line'. Below the commit message is a diff view for the README file:

...	...	@@ -1,3 +1,5 @@
1	1	Hello, world!
2	2	
3	3	Very exciting additional text.
	4	+
	5	+ The BEST text.

What's next?

That was very very basic. To really use Git, explore these great features with weird names ...

- ▶ **Forking** online repositories—duplicates *someone else's* shared repository so you can use/change/build on it without affecting their original work
- ▶ **Cloning** online repositories—copies an online repository onto your local hard drive
- ▶ **Branching** a repository—lets you (and others) experiment with changes that can later be merged into the “master” version
- ▶ **Initiating a pull request**—submits your commits to be merged into a forked/branched repository (accepted/rejected by collaborators)

Git Resources

Too many to name, but some good places to start:

- ▶ Gentle intro to version control
- ▶ GitHub and collaborative writing in academia
- ▶ Forks and pull requests
- ▶ Non-programmer's intro to Git using command line
- ▶ Fork-branch workflow using command line (but useful to read for Desktop as well)

[Dynamic Docs]

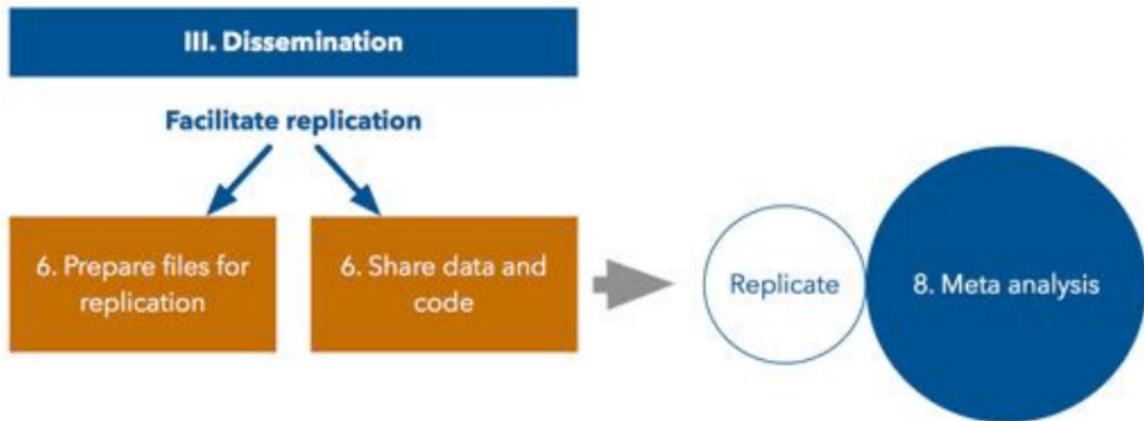
Not covered here, but ...

You can take reproducible research a step further by integrating code *into your manuscript*.

- ▶ RMarkdown
- ▶ Stata Markdoc or Stata texdoc

Solutions III: Dissemination

Steps



6. Prepare for Replication

Why do we care if our code is reproducible?

- ▶ **Unselfish reasons**—part of the scientific process and a public good
- ▶ **Selfish reasons**—make code more usable for yourself, catch potentially embarrassing errors before they become public, boost your transparency credibility

Replication files should ...

- ▶ Be complete but parsimonious
- ▶ Run and reproduce results with one click
- ▶ Be readable and interpretable by humans
- ▶ Protect personal information

Caveat: There is no single, perfect way to organize or prepare files for replication. Do what works for you (as long as it meets the above criteria)!

5 Steps for Prepping Files

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

1. Set Up

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

- ▶ Purpose:
 - ▶ Ensure files are complete/parsimonious, legible
 - ▶ Protect original files

Create

1. A new, empty replication folder *within* your project directory (e.g., “`replication_files/`”)
2. Subfolders: *Same as File Management tips!*
 - ▶ `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” file to document contents, sources, software/system versions, other info necessary for replication/comprehension.

2. Initial Replication

Copy (don't move!) data and code files into the replication 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

A. 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

B. 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-Identifying Individual-Level Data

If you haven't already, make sure replication files *do not contain* data that could be used to identify individuals.

Purpose:

- ▶ Protect individuals' identity and private information—ethical issue for researchers, 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 explicitly linked to subjects—*e.g., name, email, address, ID 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*, *grade-level 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

The Problem

ID	Study	Pub Year ¹	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] ⁷⁷	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.

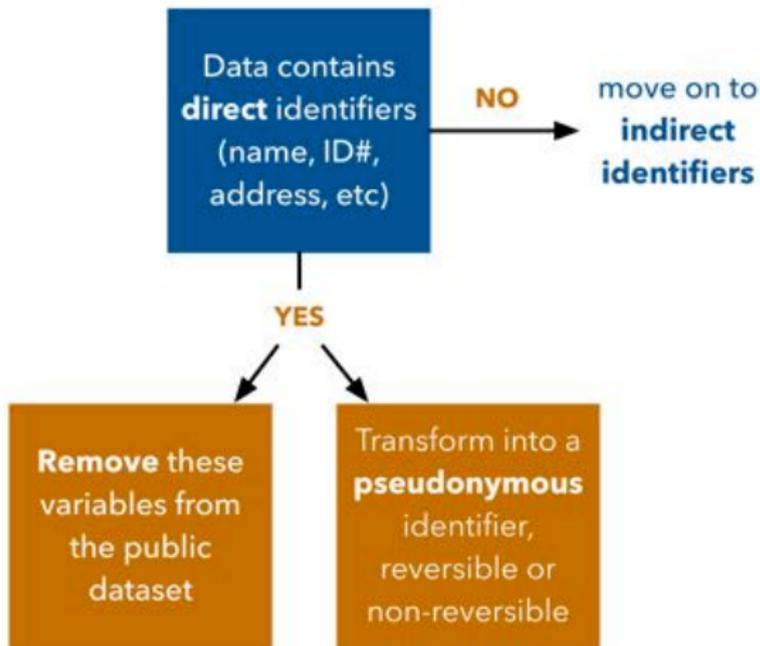
Dealing with Direct Identifiers

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 in order to be able to link datasets: replace identifiers with “pseudonyms” that may be reversible or non-reversible, e.g., give people random names or ID numbers

Solutions for Direct Identifiers

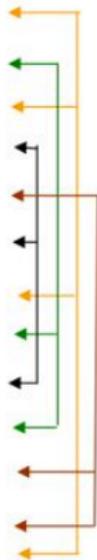


What is sufficient de-identification for indirect identifiers?

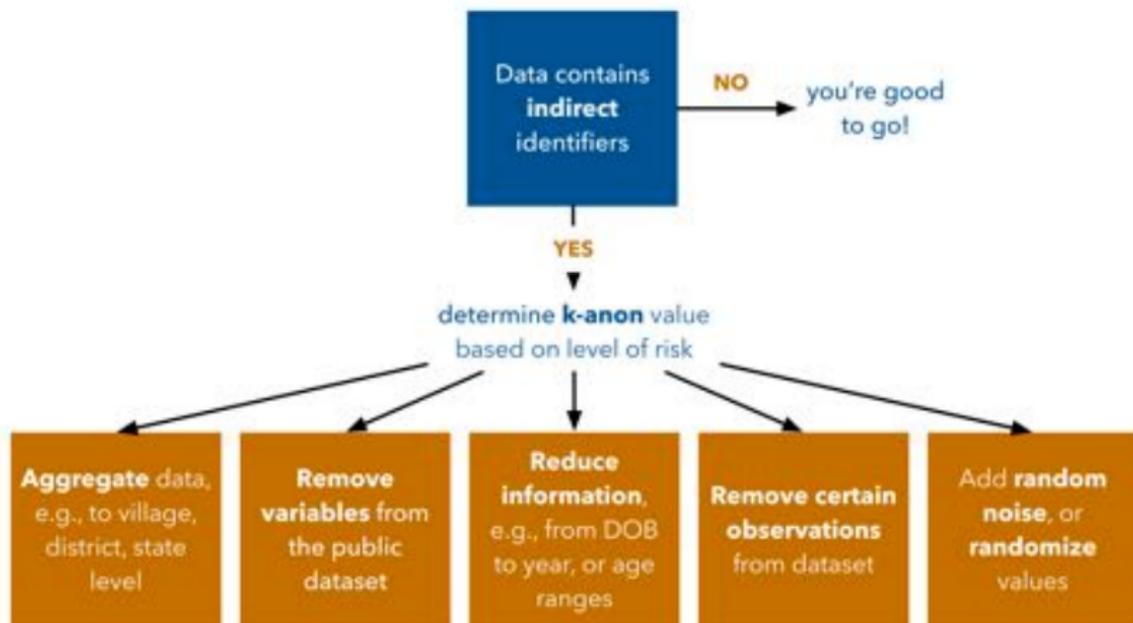
1. **Determine Risk:** $\text{Pr}(\text{being identified}) \times \text{sensitivity of data}$
2. **Set “k-anonymous” level:** each record cannot be distinguished from at least $k - 1$ other individuals who also appear in the data set
3. **Select appropriate method(s) of de-identification:** aggregating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

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



Trade-off: Usefulness \leftrightarrow Anonymity

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

See [here](#) and [here](#) for more discussion of appropriate thresholds, methods, and tools for de-identification.

Good Practices

- ▶ Include all code even if it manipulates/analyzes identified data, *as long as* it doesn't compromise anonymity—e.g., censor code that sets the seed for a random draw to generate pseudonymous ID numbers
- ▶ 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. Edit and Organize Files for Clarity

Next step is to clean and annotate data, code, and other files to improve usability.

Purpose:

- ▶ Ensure files are **legible** in terms of structure and content

Basic steps

- ▶ Structure and name files*
- ▶ Streamline and annotate code*
- ▶ Document file and folder contents

*Already done if you follow the literate programming tips in Phase II!

Document File and Folder Content

- ▶ Update the README file to describe contents of replication folders
- ▶ If necessary, include codebook in “`extra/`” folder
- ▶ Document packages & software versions used
 - ▶ R: `sessionInfo()`
 - ▶ Stata: `version`

5. Final Replication

- ▶ Shutdown or clear your Stata/R/etc. memory
- ▶ Rerun the entire process—merging, cleaning and analysis—to make sure your edits 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!

7. Share Data and Code

About Sharing Data and Code

- ▶ **What:** add replication files to an **online repository**
- ▶ **Why:** lasts longer than personal website, more searchable, future proof
- ▶ **Concerns:**
 - ▶ Can usually be embargoed, or provide only what is necessary for replication (e.g., unused survey Qs)
 - ▶ Biggest risk isn't having your data/ideas stolen, it's having your research ignored! (King 1995)
 - ▶ Difficult if proprietary

Where to Share

Depends on discipline: find appropriate registry at <http://www.re3data.org/>, or check out ...

- ▶ **Harvard's Dataverse**
- ▶ Open Science Framework
- ▶ OpenICPSR
- ▶ figshare
- ▶ Data Dryad
- ▶ University library (e.g., <http://library.ucsd.edu/dc/rdcp/collections>)

8. Meta-Analysis

About Meta-Analysis

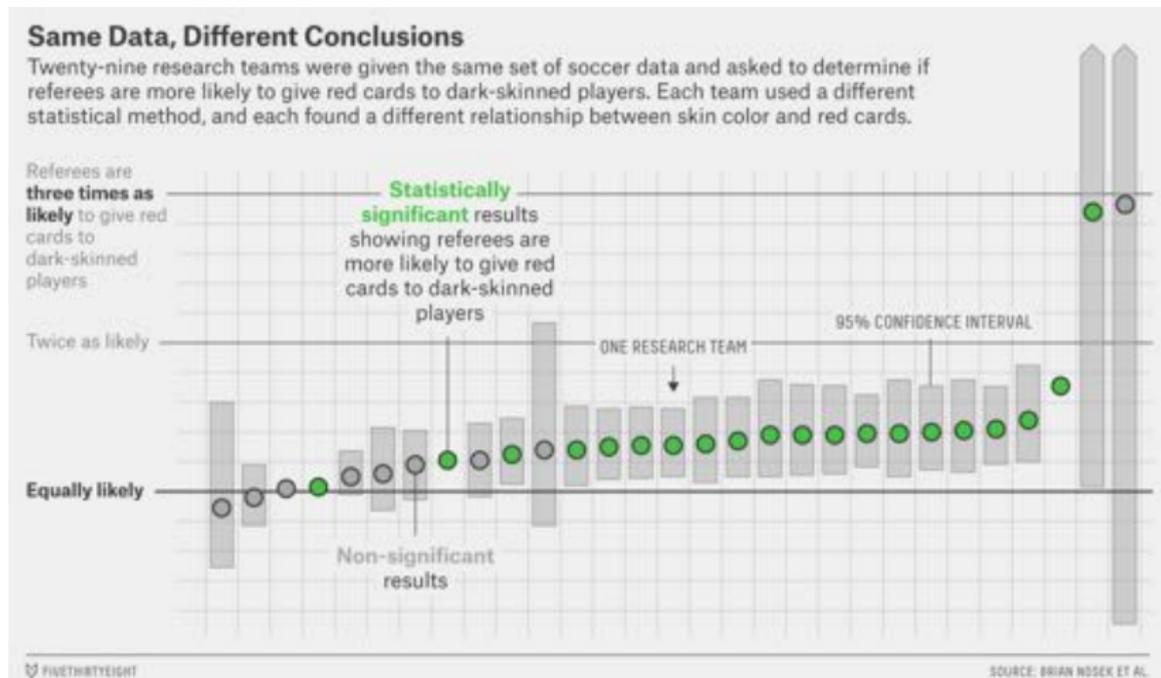
- ▶ **What:** Statistical analysis of a group of studies to derive a pooled estimate of the effect of a treatment; may be part of a “systematic review”
- ▶ **Why:** Because any estimate in an individual study may be biased or contain random error (note: assumes NO publication bias!)

One Study = One Data Point

That experiment you just ran with 3,685 participants? It's one data point among many other potential studies.

- ▶ What if the results are due to random chance?
- ▶ What if there was bias in your sample?
- ▶ What if someone else had analyzed your data?

Even with the same data, results may vary ...



Source: Graph = fivethirtyeight.com, see <https://osf.io/j5v8f/> for study materials

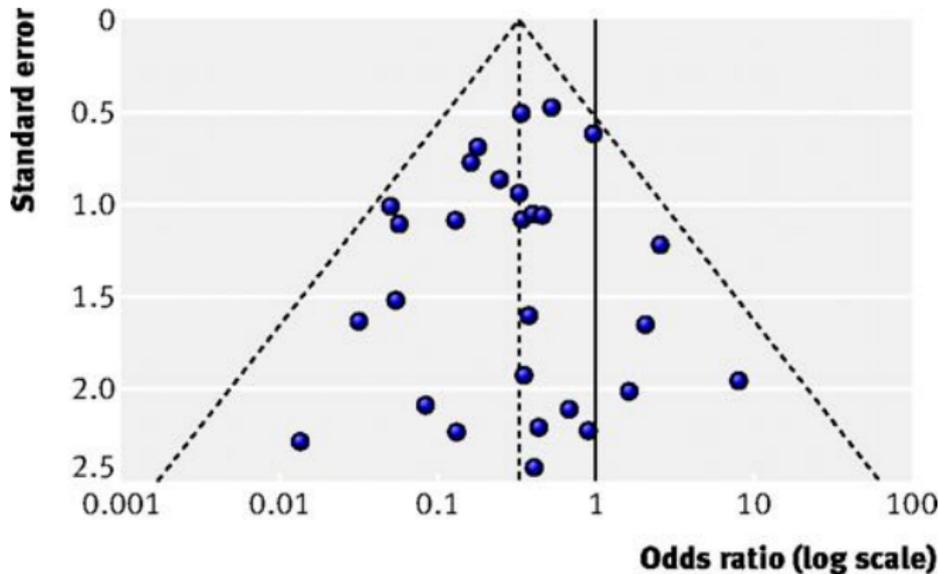
Basic Steps

Using a PAP or “protocol” ...

1. Determine which studies to include
2. Determine which outcomes to measure (e.g., discrete, continuous)
3. Select model for “meta-regression” (e.g., RE, FE, etc.)

Funnel Plots

Scatter plot of study effect sizes vs. precision (e.g., SE of treatment effect)



Source: *BMJ* 2011

Who does meta-analysis?

- ▶ Campbell Collaboration (policy)
- ▶ Cochrane Collaboration (medicine)
- ▶ 3ie (development)
- ▶ What Works Clearinghouse (US Gov't, Education)
- ▶ CLEAR (US Gov't, Labor)
- ▶ MAER-NET (Economics)
- ▶ You!

Extra

Solutions at the Institutional/Discipline Level

- ▶ **Design-based publication:** AKA “registered reports,” moves peer review before data analysis ([example](#))
- ▶ **Incentives for transparency, replication, meta-analysis:** See BITSS [prizes](#) and [awards](#), OSF [pre-registration challenge](#), etc.
- ▶ **Change norms:** e.g., journal/disciplinary standards for data sharing
- ▶ **Training:** Like this! More at BITSS, [Center for Open Science](#), etc.
- ▶ **Tenure:** “Adherence to the replication standard should be part of [tenure] judgment” (King 1995)

Selected Reading

- ▶ **Transparency:** BITSS Best Practices Manual
- ▶ **Replication:** Dewald et al. (1986), King (1995), Fang et al. (2012), FiveThirtyEight (2015), Clemens (2015)
- ▶ **Publication bias:** Turner et al. (2008), Gerber & Malhotra (2008) Fanelli (2010), Fanelli (2011), Franco et al. (2014)
- ▶ **P-hacking, fishing, researcher degrees of freedom, fraud:** Simons, Nelson, Simonsohn (2011), Gelmen & Loken (2013), Brodeur et al. (2016), John et al. (2012)
- ▶ **PAPs:** Olken 2013, Coffman & Niederle (2015), Neumark 2001
- ▶ **De-identifying data:** Tools for De-Identification, El Emam (2010)
- ▶ **Literate programming:** Long (2008), Gandrud (2013), Gentzkow & Shapiro (2014)
- ▶ **Meta-analysis:** Card & Krueger (1995), Stanlet & Doucouliagos (2012), BMJ (2011)

Thank you!

About this Presentation

This presentation was developed by Julia Clark, Scott Desposato, and Craig McIntosh of UCSD's Policy Design and Evaluation Lab (PDEL) as part of an effort to integrate good research transparency practices into methods training at UCSD.

Funding for this project was generously provided by the Berkeley Initiative for Transparency in the Social Sciences (BITSS) through a Catalyst grant.

This presentation and associated materials are available online at [GitHub](#) and are licensed under CC BY-NC 4.0 . You are free to share and adapt them for any non-commercial purpose with proper attribution. Please cite as “Clark, J., Desposato, S., and McIntosh, C. 2017. ‘How to improve the credibility of (your) social science: A practical guide for researchers’. Policy Design and Evaluation Lab (PDEL). University of California, San Diego.”