

Data Science Applications to Politics Research

Zach Dickson & Daniel de Kadt

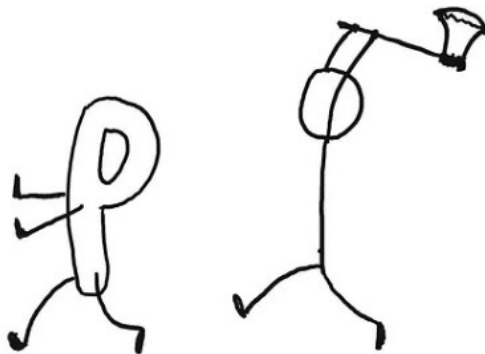
London School of Economics

GV330

Overview

- 1 Design
- 2 Analysis
- 3 Dissemination
- 4 Institutional/Discipline Level Solutions

Last week, we discussed...



P-HACKING

KC

p-hacking; image source: Twitter KC

The big problem

Science has **always** relied on **transparency**:

- Philosophy
- Mathematical/theoretical proofs
- Engineering (which required the creation of the patent system)
- Observational data
- Experimental data

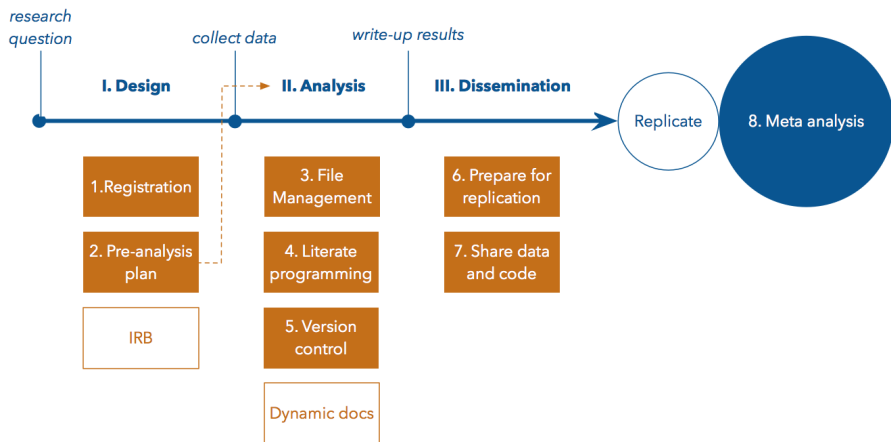
As science has become bigger (more scientists doing more science) and increasingly empirical (greater emphasis on data and analysis), the transparency problem has also become bigger.

What about solutions?

- Norms are ever-changing
- Tools and “standards” to be applied throughout the **research lifecycle**.
 - Berkeley Initiative for Transparency in the Social Sciences (BITSS): <https://www.bitss.org/>
 - Open Science Framework (OSF): <https://osf.io/>
 - Dataverse: <https://dataverse.org/>
 - Evidence in Governance and Politics (EGAP): <https://egap.org/>
 - Institute for Replication (I4R): <https://i4replication.org/>
- Main emphasis: Ways to improve **transparency** in why, what, and how researchers do science.

Research lifecycle: Individual-level solutions

The BITSS perspective looks like this:



Overview

- 1 Design
- 2 Analysis
- 3 Dissemination
- 4 Institutional/Discipline Level Solutions

The big picture

I. Design

**Combat
publication bias**



1. Registration

**Reduce researcher
degrees of freedom**



2. Pre-analysis plan

**Protect human
subjects**



IRB

- **What:** Enter your study into a “registry”
 - Now required for experimental studies submitted to *Journal of Politics*
 - In many other review contexts you will be penalised for not having one
- **Why:** Combat the file-drawer problem and publication bias.
- **Where:**
 - Open Science Framework: <http://osf.io>
 - American Economics Association (AEA):
<http://socialscienceregistry.org>
 - As Predicted (Wharton): <http://aspredicted.org>

Pre-Analysis Plans (PAPs)

- **What:** Detailed description of research design and data analysis plans, submitted to a registry *before* looking at the data.
 - Rule of thumb: PAP should be specific enough such that an expert from the same subfield could conduct and replicate all of the analyses on their own if they had 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
 - Transparent methods make it easier for others to build on your work

Registration typically includes a pre-analysis plan.

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

One step further: Registered reports

A registered report is a slightly newer idea, and is like a paper combined with a PAP.

- You literally write the paper, write your code, and then submit the paper without having collected or analysed any data.
- Editors and reviewers then judge the paper based purely on the motivation for writing it, not the results you get.
- APSR and JOP (among many other journals) both have dedicated registered report streams.

This takes a particular philosophical perspective to its (somewhat natural) conclusion: Science should be judged without knowing the results.

Discuss: What are the up-sides and down-sides of this perspective?

What to include in a PAP (or RR)

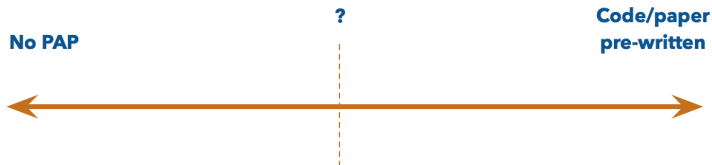
No universal standard, can include ...

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, etc.), heterogeneous treatment effects (subgroups, interactions), covariates, standard errors, corrections for multiple hypothesis testing, missing values, outliers
Team	members, affiliations, conflicts of interest

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 forethought!**

- 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"
- Ofosu & Posner (2023) weigh the arguments for and against
- More debate on utility for observational work but can be done (see Neumark 2001)

IRB requirements to protect human subjects → necessary for ethical research, though not always sufficient

For more on ethics in experiments, see:

- <http://desposato.org/ethicsfieldexperiments.pdf>
- **trade** <https://egap.org/wp-content/uploads/2020/05/egap-committee-memo-on-the-report-of-the-apsa-ad-hoc.pdf>

Overview

- 1 Design
- 2 Analysis
- 3 Dissemination
- 4 Institutional/Discipline Level Solutions

The big picture

Reproducibility \iff transparency

A **reproducible** workflow needs to account for at least the following steps:

- 1 Ingestion: Acquiring and reading raw data
- 2 Transformation: Turning raw data into usable data
- 3 Analysis: Extracting meaning from the usable data
- 4 Output: Formatting and producing results (e.g. tables or figures)
- 5 Reporting: Putting the results into a shareable document (e.g. slides or papers)

Worth knowing: In the data science/analytics engineering world, the first two parts are called an ELT (extract, load, transform) or ETL pipeline.

The big picture: Ultimate goal

Ultimately, a reproducible workflow should be:

- 1 Code-based, structured, fire-walled
- 2 Documented, literate, version-controlled
- 3 Functional, modular, tested, DRY
- 4 Dependency-proof
- 5 Automated

We will cover tiers **1** and **2** now (and do some of 3-5 in Seminar). If you are doing 1 and 2, you are doing better than most social scientists!

The big picture: The BITSS Approach

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

II. Analysis

Reproducible workflow for
sharing and replication

3. File Management

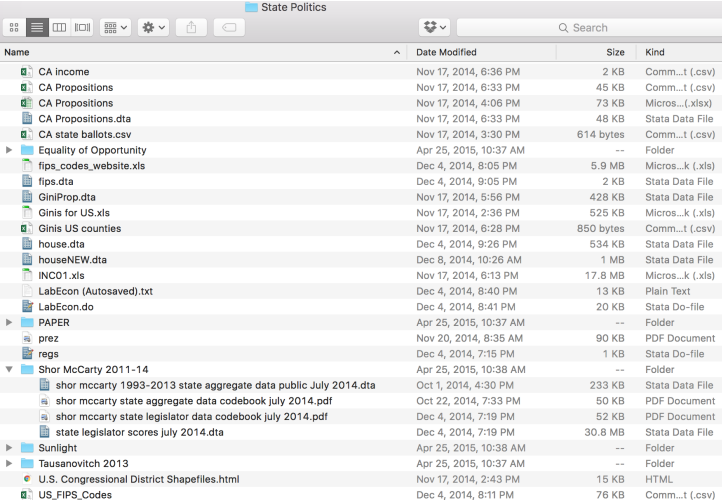
4. Literate
programming

5. Version control

Dynamic docs

- **What:** Organizing, manipulating, and managing files hygienically, transparently, and intuitively
- **Why:**
 - Preserve original data
 - Improve error detection
 - Streamline workflows
 - Reduce overhead

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

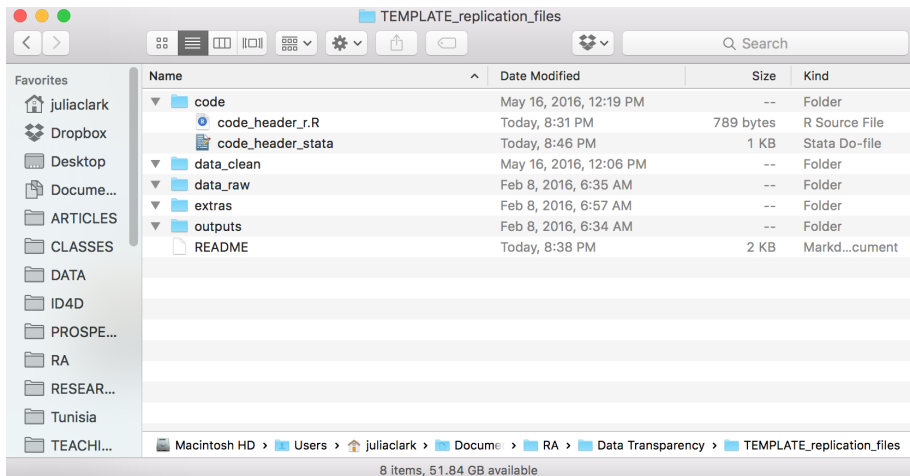


Name	Date Modified	Size	Kind
CA income	Nov 17, 2014, 6:36 PM	2 KB	Comm...t (.csv)
CA Propositions	Nov 17, 2014, 6:33 PM	45 KB	Comm...t (.csv)
CA Propositions	Nov 17, 2014, 4:06 PM	73 KB	Micros...(xlsx)
CA Propositions.dta	Nov 17, 2014, 6:33 PM	48 KB	Stata Data File
CA state ballots.csv	Nov 17, 2014, 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)

Instead, use (something like) PDEL template

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

PolicyDesignEvaluationLab/Transparency-Initiative

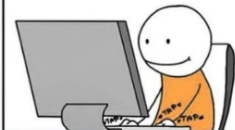


- **What:** Writing code that it's legible to *humans*.
- **Why:**
 - Document what you are doing and why
 - Allow others to reproduce (and ultimately replicate) your work
 - Avoid mistakes

Legible to humans, including your future self

UNFINISHED WORK

FRIDAY EVENING



PERFECT!
I'LL FINISH
THIS ON
MONDAY



MONDAY MORNING...



WHAT DOES
THIS MEAN!?!?

(The most) basic principles

- Structure and name files and variables intuitively
- Thoroughly comment code as you go, or use something like R Markdown or a Jupyter Notebook
 - Use this to document every decision you make.
 - Should cover the what, the why, and even the how.
- Make the contents of scripts and files easy to navigate
- Streamline (DRY) code to minimize repetition

Structure and naming

- Create separate scripts for processing and analysing data, with a runner-script for executing it all
- Give code, data files, variables, 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 with descriptive names that aren't likely to change (e.g., `figure_hte.png` is **much better** than `figure_1.png` – this is bad, don't do it!))

Improve navigation

- Add headers (see PDEL template)
- Format scripts so they're easily readable
- 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`

Header example

```
1 ▾ ##### INFO #####
2
3 # PROJECT
4 # Paper:|
5 # Authors:
6
7 # R Script
8 # Purpose:
9 # Created: <date> by <author> # you don't need this if using Git!
10 # Updated: <date> # you don't need this if using Git!
11 # Inputs: <files required>
12 # Outputs: <tables and figures>
13
14 ▾ ##### SETUP #####
15
16   rm(list = ls()) # clear workspace
17   setwd("~/Documents/replication_files")
18
19 ▾ ##### PACKAGES #####
20
21 # Check system and installs packages user doesn't have, load needed packages
22
23   need <- c("dplyr", "foreign", "ggplot2", "stargazer") # list packages needed
24   have <- need %in% rownames(installed.packages()) # checks packages you have
25   if(any(!have)) install.packages(need[!have]) # install missing packages
26   invisible(lapply(need, library, character.only=T)) # load needed packages
27
28 ▾ ##### ANALYSIS #####
29
```

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`

Streamline code

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

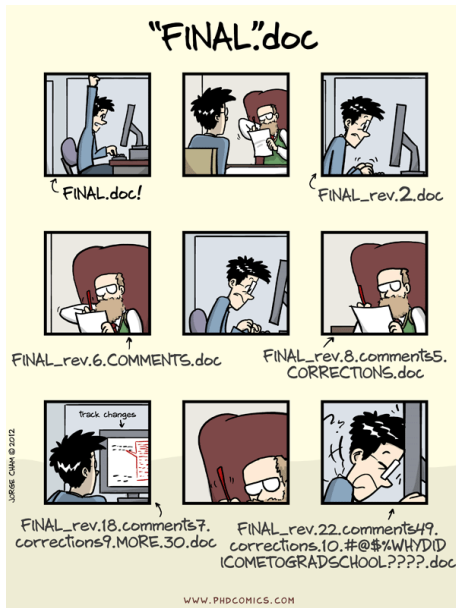
Python:

`os.chdir(os.path.expanduser("~/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
- Particularly helpful if co-authors alternate between Mac ("/") and Windows ("\\") file extensions
- Always best to use relative pathing as much as possible, but can be tricky (e.g. in R)

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.

Principles of version control

Principles of version control:

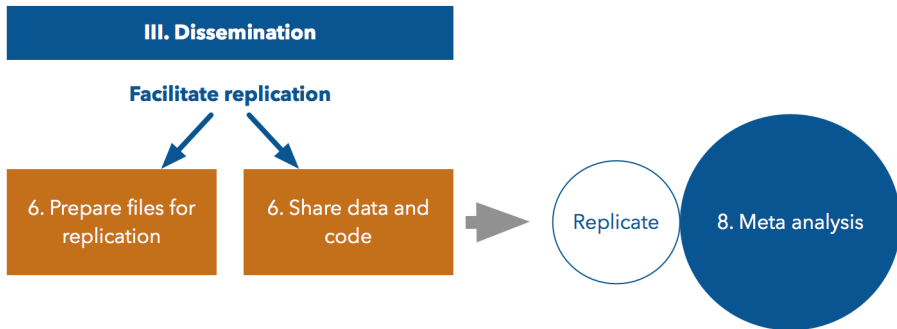
- Vault original, raw data files (do not save over raw, original data)
- Changes to files should be documented and reversible
- Keep “main” versions of files in working order; create copies before experimenting
- Reconcile independent changes by different users

While you can do this manually, in week 1 you learned about Git (and GitHub), so do that instead!

Overview

- 1 Design
- 2 Analysis
- 3 Dissemination**
- 4 Institutional/Discipline Level Solutions

The big picture



Reproducibility and replication

Note: These two terms are often interchanged (e.g. the BITSS diagram). Really, we are talking about reproducibility here, but the term “replication” is used so widely that we will interchange them a bit...

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 “cred”
- There are a lot of examples where big mistakes (and even fraud) have been detected through “replication” materials. Take this seriously – your scientific reputation may depend on it.

“Replication” files should ...

- Be complete but parsimonious
- Run and reproduce results, ideally from the raw data, 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)

Five 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. If you have been disciplined about the earlier parts of your process, this should not be hard.

Purpose:

- Ensure files are **complete/parsimonious, legible**
- Protect original files

1. Set-up

Create the following:

- ❶ **A new, empty replication folder** *within* your project directory (e.g., “`replication_files/`”)
- ❷ **Subfolders:**
 - `code/` — scripts
 - `data_clean/` — manipulated data
 - `data_raw/` — original data
 - `output/` — generated tables, graphs, etc.
 - `extra/` — misc. extras (e.g., code book)
- ❸ **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

2.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 (no, a small discrepancy is **not** ok!)

2.B. Check data clean/merge

- ➊ If separate from analysis, copy original merge/cleaning script(s) into `replication_files/code`
- ➋ Copy original dataset(s) to `replication_files/data`
- ➌ Run merge/clean code without changes (except for wd)
- ➍ Rerun the analysis code from above on the newly cleaned/merged data file
- ➎ If you get different results than step #1, there is a discrepancy with merging/cleaning code – find it, and 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

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.*

Data identifiability is not binary. Data lies on a spectrum with multiple shades of identifiability. See this useful infographic:

https://fpf.org/wp-content/uploads/2016/04/FPF_Visual-Guide-to-Practical-Data-DeID.pdf

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
- Birthdate (month, day and year of birth), gender, and 5-digit postal code (ZIP) uniquely identifies most people (87%) in the United States. (Sweeney 2000)

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.

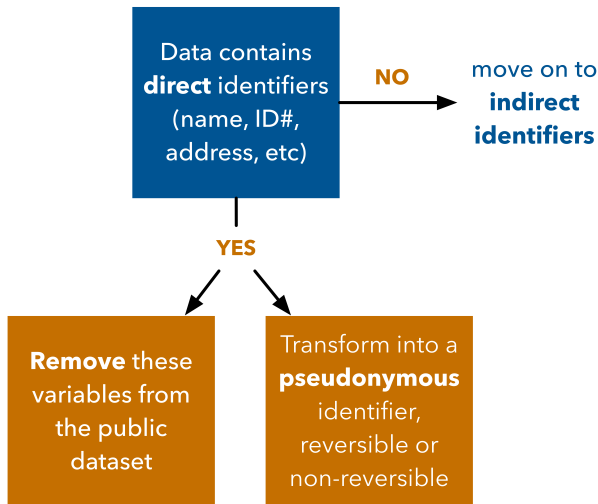
Dealing with direct identifiers

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

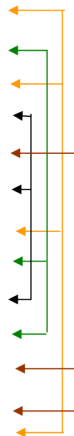


What is sufficient de-identification for indirect identifiers?

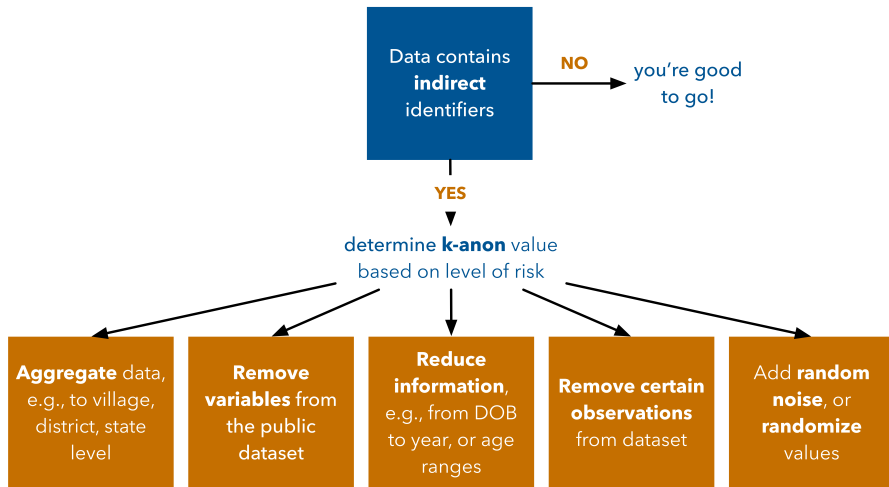
- 1 **Determine Risk:** $\text{Pr}(\text{being identified}) \times \text{sensitivity of data}$
- 2 **Set “k-anonymous” level:** each record cannot be distinguished from at least $k - 1$ other individuals who also appear in the data set
- 3 **Select appropriate method(s) of de-identification:** aggregating 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 \iff 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.

- 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 are not used for analysis, de-identify early in merging/cleaning process
- Store original data with personally identifiable information securely

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

- 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/python/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 any discrepancies are addressed, the files are ready for sharing

- **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 Dataverse
- Open Science Framework
- OpenICPSR
- And, of course, on your own GitHub page!

- **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

One study = one data point

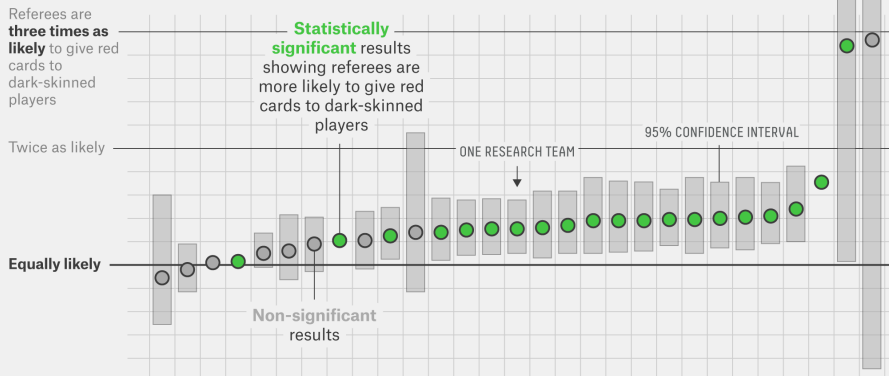
That experiment you just ran with 3,685 participants? It's one data point among many other potential studies.

- 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 ...

Same Data, Different Conclusions

Twenty-nine research teams were given the same set of soccer data and asked to determine if referees are more likely to give red cards to dark-skinned players. Each team used a different statistical method, and each found a different relationship between skin color and red cards.



FIVETHIRTYEIGHT

SOURCE: BRIAN NOSEK ET AL.

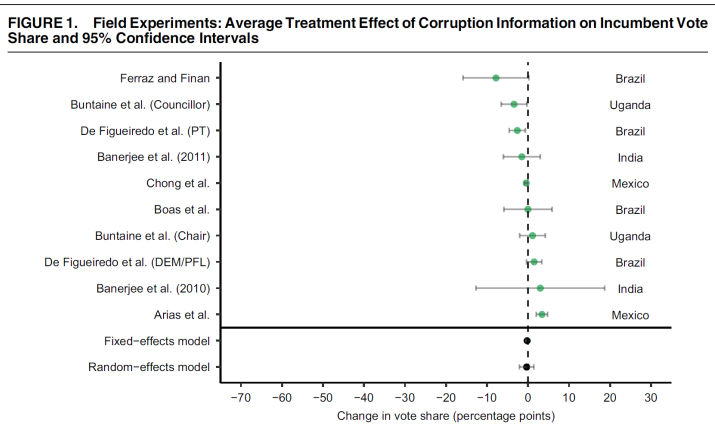
Source: Graph = fivethirtyeight.com, see <https://osf.io/j5v8f/> for study materials

Basic steps of meta-analysis

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.)

Meta-analysis: effect of corruption information



Incerti, T. (2020). Corruption information and vote share: A meta-analysis and lessons for experimental design. *American Political Science Review*, 114(3), 761-774.

Overview

- 1 Design
- 2 Analysis
- 3 Dissemination
- 4 Institutional/Discipline Level Solutions**

Solutions at the Institutional/Discipline Level

- **Design-based publication:** “registered reports” moves peer review before data analysis
- **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:** BITSS, Center for Open Science, etc.
- **Tenure:** “Adherence to the replication standard should be part of [tenure] judgment” (King 1995)

Selected Reading & Citations

- **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), Gelman & 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)

Materials adapted from: [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.](#)