

# Claude Code for replications

Shiro Kuriwaki (Yale)  
March 10, 2026

Brief presentation to Yale [DISSC ORR](#) group



# Summary

- LLMs appear to have made a big leap in late 2025
  - May 2025: Claude Code, Nov 2025: Claude Opus 4.5, similar with OpenAI
  - Claude Code / Codex: integration with local project files
  - LLMs as *harness* / agent, not just an app (Ethan Mollick [blog](#), Feb 2026)
- Claude Code has quickly taken over parts of my workflow as an applied social scientist
  - Read and keep track of file structure, variable names, and dependencies in replication files
  - Check for errors and sometimes investigate them
  - Do preliminary analysis and write code the way I want it to
- Today: More on benign, QOL improvements rather than radical changes
  - Three use cases

# 1. Reading other people's replication files



Innovations for Poverty Action Dataverse

(Innovations for Poverty Action)

## A Multifaceted Program Causes Lasting Progress for the Very Poor: Evidence From Six Countries

Version 3.1



Banerjee, Abhijit; Duflo, Esther; Goldberg, Nathanael; Karlan, Dean; Osei, Robert; Parienté, William; Shapiro, Jeremy; Thuysbaert, Bram; Udry, Christopher, 2015, "A Multifaceted Program Causes Lasting Progress for the Very Poor: Evidence From Six Countries", <https://doi.org/10.7910/DVN/NHIXNT>, Harvard Dataverse, V3, UNF:6:K0OUadzW2QoGUc5HkaWwUQ== [fileUNF]

[Cite Dataset](#) ▾

Learn about [Data Citation Standards](#).

[Access Dataset](#) ▾

[Contact Owner](#)

[Share](#)

[Dataset Metrics](#) ?

19,868 Downloads ?

### Description ?

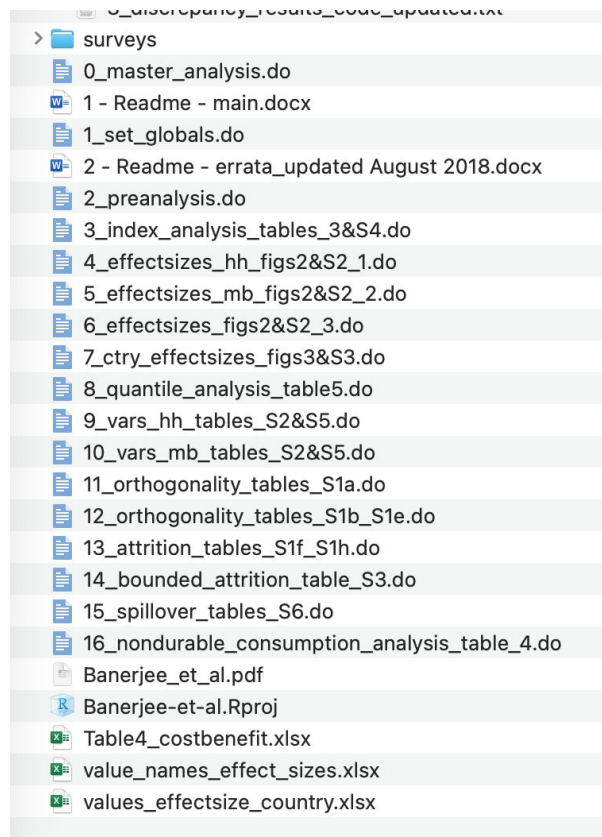
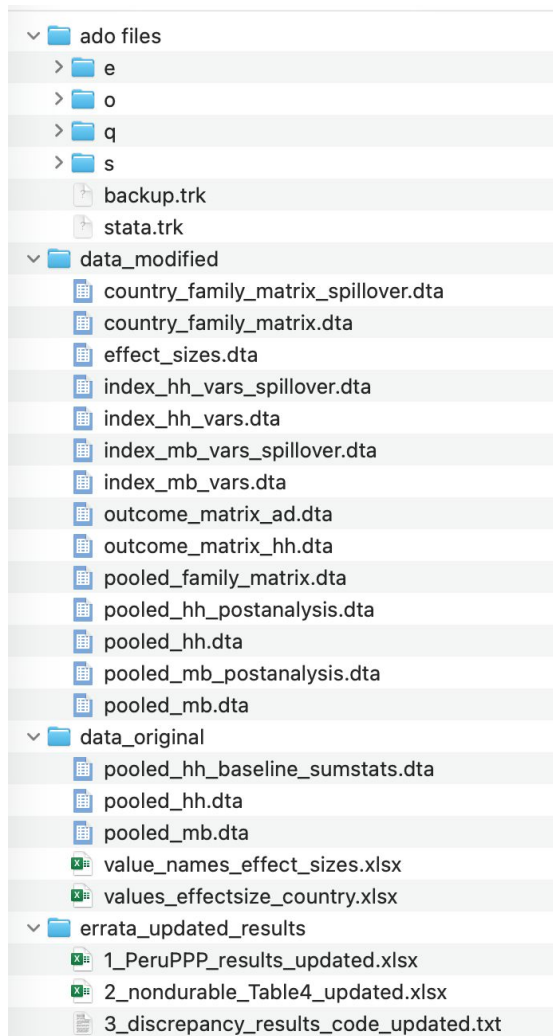
We present results from six randomized control trials of an integrated approach to improve livelihoods among the very poor. The approach combines the transfer of a productive asset with consumption support, training, and coaching plus savings encouragement and health education and/or services. Results from the implementation of the same basic program, adapted to a wide variety of geographic and institutional contexts and with multiple implementing partners, show statistically significant cost-effective impacts on consumption (fueled mostly by increases in self-employment income) and psychosocial status of the targeted households. The impact on the poor households lasted at least a year after all implementation ended. It is possible to make sustainable improvements in the economic status of the poor with a relatively short-term intervention.

**(Live demo of claude code in repository)**

# 1. Reading other people's replication files

- How do I replicate table 1?
- I understand better in R – what does the regression look like?

LLMs are very good at  
keeping track of everything

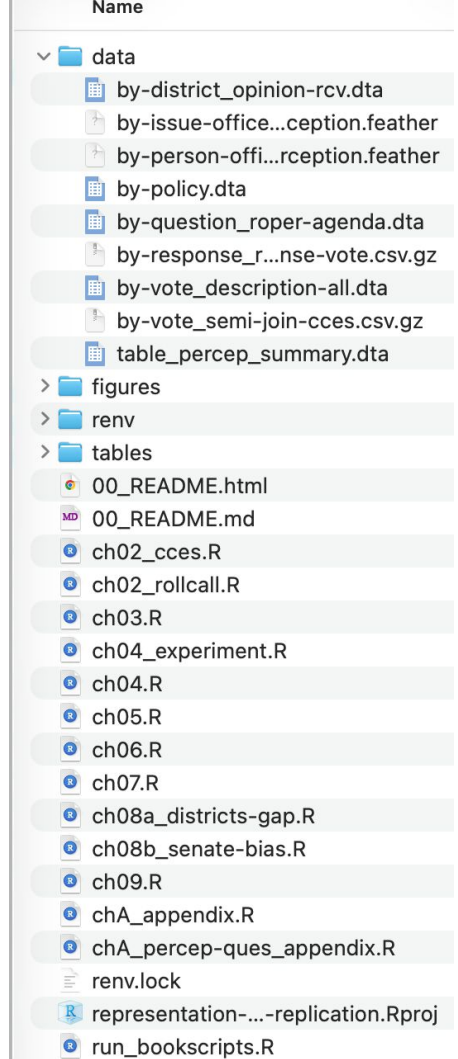


## 2. Checking my own replication files

### Common Issues

- Script 1 depends on datasets **X**, **Y**, **Z** from outside the project folder
- Dataset **X** is huge, and you only need a subset of them for public release (but not sure which ones)
- Want to rename a variable name for public use, but need to update it everywhere (scripts, **README**, etc..)
- Change in script **1** could affect script **9**

Claude code with Opus 4.6 seems to “just get it”



> look at the R scripts. These scripts have been copy-pasted from a different directory. I want to make this new directory into a renv-compliant replication package. I also imported the datasets in the "data/" folder. As a first step, propose changes to all file paths so that every script can run.

- Let me explore the project structure and understand the current state of the R scripts and data files.
- **Explore**(Explore project structure)  
 L Done (20 tool uses · 65.7k tokens · 40s)  
 (ctrl+o to expand)
- Let me verify the exact data directory contents and check for a few specific files.
- Read 2 files (ctrl+o to expand)
- Here's the full picture and my proposed path changes:

#### Current problems

##### 1. Data read paths: `data/book/` → `data/`

All scripts read from `data/book/...` but your files are directly in `data/`. Every script needs the `book/` subdirectory removed.

**Script(s):** `ch02_cces`, `ch03`, `ch06`, `ch07`, `ch08a`, `ch09`, `chA_appendix`  
**Current path:** `data/book/by-policy.dta`  
**Proposed path:** `data/by-policy.dta`

**Script(s):** `ch02_rollcall`  
**Current path:** `data/book/by-vote_description-all.dta`  
**Proposed path:** `data/by-vote_description-all.dta`

**Script(s):** `ch03`  
**Current path:** `data/book/by-issue-office_perception.dta`  
**Proposed path:** `data/by-issue-office_perception.dta`

**Script(s):** `ch04`, `ch05`  
**Current path:** `data/book/by-person-office_perception.dta`  
**Proposed path:** `data/by-person-office_perception.dta`

**Script(s):** `ch07`, `ch08a`, `ch08b`  
**Current path:** `data/book/by-district_opinion-rcv.dta`  
**Proposed path:** `data/by-district_opinion-rcv.dta`

**Script(s):** `ch06`  
**Current path:** `data/book/by-question_roper-agenda.dta`  
**Proposed path:** `data/by-question_roper-agenda.dta`

##### 2. Output paths: `book/Draft4-for-production/...` → `figures/` and `tables/`

All scripts write figures/tables to `book/Draft4-for-production/figures/` or `book/Draft4-for-production/tables/`. These should point to the local `figures/` and `tables/` directories you already created.

Script(s)	Current pattern	Proposed pattern
All scripts	<code>book/Draft4-for-production/figures/...</code>	<code>figures/...</code>
All scripts	<code>book/Draft4-for-production/tables/...</code>	<code>tables/...</code>

##### 3. `run_bookscripts.R`: `analyze/book-analysis/` → `root`

The `source()` calls reference the old directory structure. Since scripts are now in the project root:

Current	Proposed
<code>source("analyze/book-analysis/ch02_cces.R")</code>	<code>source("ch02_cces.R")</code>
(same pattern for all 12 scripts)	

##### 4. Missing data files (not in `data/`)

Several scripts reference files that don't exist in your `data/` directory:

**File:** `data/source/qv_key.Rds`  
**Used by:** `ch02_cces`, `chA_appendix`  
**Notes:** Supplementary key file

**File:** `data/output/intermediate/by-response_response-vote.feather`  
**Used by:** `ch03`  
**Notes:** Intermediate dataset

**File:** `data/output/intermediate/by-vote_semi-join_cces.feather`  
**Used by:** `chA_appendix`  
**Notes:** Intermediate dataset

**File:** `data/output/intermediate/by-response_perceptions.dta`  
**Used by:** `chA_percep-ques_appendix`  
**Notes:** Intermediate dataset

**File:** `data/output/analysis/by-policy.dta`  
**Used by:** `ch08b`  
**Notes:** Different version of `by-policy`

These files were likely generated by upstream build scripts in the original project. You'll need to either:

- (a) Include them in `data/` and update paths, or
- (b) Exclude the scripts that depend on them (`ch03` partially, `chA_appendix`, `chA_percep-ques_appendix`, `ch08b` partially), or
- (c) Include the upstream build scripts that generate them.

› If you think it runs with the data provided, just put the name of that script in a bucket ("should be replicable") without any formatting. For the ones that do not run, just list the scripts that do not run and, separately, a unique list of the datasets needed.

● **Should be replicable:**

ch02\_cces.R, ch02\_rollcall.R, ch04.R, ch04\_experiment-replication.R, ch05.R, ch06.R, ch07.R, ch08a\_districts-gap.R, ch08b\_senate-bias.R, ch09.R

**Cannot run:**

- ch03.R (Table 3.1 only; rest should work)
- chA\_appendix.R (Table A.2 only; Table A.1 should work)
- chA\_percep-ques\_appendix.R

**Missing datasets:**

- data/output/intermediate/by-response\_response-vote.feather
- data/output/intermediate/by-vote\_semi-join-cces.feather
- data/output/intermediate/by-response\_perceptions.dta
- data/source/qv\_key.rds

# Smart Ctrl+F -> Replace updates

This is a nice touch that mechanical Ctrl+F would not have done

> I replaced the data, but also changed it to a feather format. Edit the scripts so that it reads correctly from the feather file where relevant

- Searched for 1 pattern (ctrl+o to expand)

- **Update(ch03.R)**

- └ Added 1 line, removed 1 line

```
12
13 # Read in Datasets
14 by_policy <- read_dta("data/by-policy.dta")
15 -by_issue_all <- read_dta("data/by-issue-office_perception.dta")
15 +by_issue_all <- read_feather("data/by-issue-office_perception.feather
+")
16 by_issue <- by_issue_all |>
17 tidylog::filter(
18   issue_self != "",
```

- **Update(ch04\_experiment-replication.R)**

- └ Added 1 line, removed 1 line

```
98 left_join(budget_act1, by = "cd") |>
99 select(-cd)
100
101 -iss_self <- read_dta("data/by-issue-office_perception.dta") |>
101 +iss_self <- read_feather("data/by-issue-office_perception.feather"
+)|>
102 filter(office == "rep", year %in% c(2009, 2018)) |>
103 inner_join(q_label_map, by = "q_label") |>
104 select(year, case_id, q_label = q_label_expt, issue_prpc, issue_se
lf, party_self, issue_act1) |>
```

- Now let me make sure both scripts have `library(arrow)` loaded (needed for `read_feather`).

- Searched for 2 patterns, read 1 file (ctrl+o to expand)

- `ch03.R` already has it. Need to add it to `ch04_experiment-replication.R`.

- Read 1 file (ctrl+o to expand)

- **Update(ch04\_experiment-replication.R)**

- └ Added 1 line

```
2
3 library(tidyverse)
4 library(haven)
5 +library(arrow)
6 library(dataverse)
7 library(fixest)
8 library(modelsummary)
```

- Done. Both `ch03.R` and `ch04_experiment-replication.R` now read from `data/by-issue-office_perception.feather` using `read_feather()`, and both have `library(arrow)` loaded.

# Summarizing necessary variables

> for by-issue-office\_perception.dta, what are the variables that are actually needed for the replication of these scripts? Give me a select() command with efficient dplyr verbs that will narrow down the dataset. I want to make it smaller

- Searched for 2 patterns (ctrl+o to expand)
- Wait – ch07.R matched? Let me check. Also need to look at ch04\_experiment-replication.R.
- Searched for 2 patterns (ctrl+o to expand)
- That's just a variable name, not the dataset. Now let me trace all actual column usage from the two scripts that read it.
- Searched for 2 patterns, read 1 file (ctrl+o to expand)
- Let me read the relevant sections to see exactly which columns are used.
- Read 2 files (ctrl+o to expand)
- Now let me also check the full columns in the actual file to know what we're trimming from.
- **Bash**(Rscript -e ' library(haven)...  
\_ Columns:  
year  
case\_id  
... +33 lines (ctrl+o to expand)

- Now I have the complete picture. Columns used across both scripts:

**ch03.R:** issue\_self, issue\_act1, office, year, q\_label, issue\_prpc, agrmt\_issue\_act1, agrmt\_issue\_prpc

**ch04\_experiment-replication.R:** office, year, case\_id, q\_label, issue\_prpc, issue\_self, party\_self, issue\_act1

Combined unique set:

```
read_dta("data/by-issue-office_perception.dta") |>  
  select(year, case_id, q_label, office, issue_self, issue_act1, issue_prpc,  
         party_self, agrmt_issue_act1, agrmt_issue_prpc)
```

That takes you from 33 columns down to 10.

> what about by-person-office\_perception.dta? what are the variables that are necessary?

- Searched for 1 pattern, read 1 file (ctrl+o to expand)
- Let me trace the column usage in each script.
- Read 4 files (ctrl+o to expand)
- Now let me also check the full column list of the file.
- **Bash**(Rscript -e ' library(haven)...  
\_ Columns:  
year  
case\_id  
... +22 lines (ctrl+o to expand)
- Here's the usage across all 4 scripts:

Column	ch03	ch04	ch05	ch04_experiment
year	x		x	x
case_id		x		x
office	x	x		x
agrmt_issue_act1	x	x	x	
agrmt_issue_prpc	x	x	x	
agrmt_party_act1	x	x	x	
agrmt_party_prpc			x	x
ideol_a_nominate	x	x		x
approve		x	x	
intent_rep_inc		x	x	
dataset		x		
cd			x	x

# Manual Edits

- Done. Here's what changed:

**ch02\_cces.R** – Replaced `read_rds("data/source/qv_key.Rds")` with derivation from `by-policy.dta`: splits `qIDs` into long form and extracts year from the `qID` prefix.

**chA\_appendix.R** – Replaced the `qv_key.rds + by-vote_semi-join-cces.feather` section with `by-policy.dta` columns:  
– `vname` extracted from `voteview_H/voteview_S` URLs  
– `wording` added from `by-policy.dta` directly  
– Dropped `date` and `votes` (Yea-Nay tallies) columns since those required the missing feather file  
– Removed the now-unused `library(rcces)` and `library(arrow)` imports

**Remaining missing files** (down from 4 to 2):

- `data/output/intermediate/by-response_response-vote.feather` → needed by `ch03.R`
- `data/output/intermediate/by-response_perceptions.dta` → needed by `chA_percep-ques_appendix.R`

\* Cooked for 1m 15s

```
> oh but you can never delete a column that we had been using in the final table
(in A2). If you can't replicate the column, then don't delete it and keep the
data that is required.
```

- You're right, sorry. The old Table A.2 had `Date` and `Yea-Nay` columns – I shouldn't have dropped them. Let me restore them as empty columns so the table structure is preserved, and add a comment noting the missing data source.

- Read 1 file (ctrl+o to expand)

```
• Update(chA_appendix.R)
  L Added 15 lines, removed 6 lines
    38   gt::gtsave(glue("{dir_tbl}/table-A.1.docx"))
    39
    40   # Appendix Table A2 ----
    41   +# NOTE: date and votes (Yea-Nay) require data/output/intermediate/by-vote_semi-join-cces.feather
    42   pol2 |>
    43     mutate(
    44       vname_H = str_extract(voteview_H, "[^/]+$") |> str_sub(6, 9),
    45       vname_S = str_extract(voteview_S, "[^/]+$") |> str_sub(6, 9),
    46 -     across(c(vname_H, vname_S), \(x) na_if(x, NA_character_))
    47 +     across(c(vname_H, vname_S), \(x) na_if(x, NA_character_)),
    48 +     date_H = NA_character_,
    49 +     date_S = NA_character_,
    50 +     votes_H = NA_character_,
    51 +     votes_S = NA_character_
    51   ) |>
    47 -   select(cong_fmt, q_label = issue, natl_supp, wording,
    48 -          vname_H, passed_H, vname_S, passed_S, nation_outcome_agree) |>
    52 +   select(cong_fmt, q_label = issue, natl_supp,
    53 +          date_H, vname_H, votes_H, passed_H,
    54 +          date_S, vname_S, votes_S, passed_S,
    55 +          nation_outcome_agree) |>
    56   mutate(passed_H = zap_label(passed_H),
    57           passed_S = zap_label(passed_S)) |>
    58   arrange(cong_fmt, natl_supp) |>
  ...
    60   gt::gt() |>
    61   gt::fmt_percent(decimals = 1, columns = natl_supp) |>
    62   gt::tab_style(style = cell_text(weight = "bold"), locations = cells_row_groups
    63   ( ) ) |>
    56 -   gt::tab_spanner("House", columns = c(vname_H, passed_H)) |>
    57 -   gt::tab_spanner("Senate", columns = c(vname_S, passed_S)) |>
    58 -   gt::cols_label(starts_with("vname_") ~ "ID",
    63 +   gt::tab_spanner("House", columns = c(date_H, vname_H, votes_H, passed_H)) |>
    64 +   gt::tab_spanner("Senate", columns = c(date_S, vname_S, votes_S, passed_S)) |>
    65 +   gt::cols_label(starts_with("date_") ~ "Date",
    66 +                  starts_with("vname_") ~ "ID",
    67 +                  starts_with("votes_") ~ "Yea-Nay",
    68 +                  starts_with("passed_") ~ "Pass.",
    69 +                  nation_outcome_agree ~ "Succ.",
    70 +                  q_label ~ "",
```

# Finishing touches

```
> I've cleaned up my folder. Check if this looks good as a replication file for
dataverse (I will upload the version on github)
```

- Read 4 files (ctrl+o to expand)
- Now let me do a final audit of all script data references against the actual data files.
- **Explore**(Final audit of all scripts)
  - └ Done (49 tool uses · 74.4k tokens · 4m 44s)(ctrl+o to expand)
- The package looks good. Everything checks out:
  - All 9 data files exist and are correctly referenced
  - All output paths point to [figures/](#) or [tables/](#)
  - All `source()` calls in `run_bookscripts.R` match existing scripts
  - All packages are in `renv.lock`
  - No stray hardcoded external paths (only in commented-out/disabled code)

Two minor things to consider cleaning up before upload:

1. `ch02_cces.R` lines 6–16: The `if (FALSE) { ... }` block references `data/source/` and Google Sheets. It's dead code – you may want to remove it for a cleaner replication file.
2. `ch02_cces.R` lines 60–71: Commented-out block referencing `~/Dropbox/cces_cumulative/...`. Same – dead code with an external path.

Neither will cause errors, but they could confuse replicators. Up to you whether to strip them.

```
> Checking back in. Draft a 00_README.md that explains how to use renv and how to
run the package, etc..
```

- Let me first review the current state of the project to make sure the README reflects everything accurately.
- Read 3 files (ctrl+o to expand)
- Let me also check the R version and quickly scan what outputs each script produces.
- **Bash**(`Rscript -e 'cat(R.version$version.string)'`)
  - └ R version 4.5.1 (2025-06-13)
- Searched for 3 patterns (ctrl+o to expand)
- **Write**(`00_README.md`)
  - └ Wrote 75 lines to `00_README.md`

```
1 # Replication Package
2
3 This repository contains the replication code and data for all figures
and tables in the book.
4
5 ## Requirements
6
7 - R (>= 4.5.0)
8 - [renv](https://rstudio.github.io/renv/) for package management
9
10 ## Setup
... +65 lines (ctrl+o to expand)
```
- Drafted `00_README.md`. Take a look and let me know if you want to adjust anything – e.g., adding the book title, author names, a license, or a citation block.

# Tailoring code style

- Can tailor how it writes code with user-specific code style

CLAUDE.md

×

## # Global preferences

### ## Git

- Never commit on behalf of the user. Only commit when explicitly asked.

### ## Code style

- Add ``# Written by Claude Code`` as the first line of every new script file

### ## R / tidyverse

- Use RStudio-compatible section headers: ``# Section name ----`` for top-level sections, ``## Subsection name ----`` for subsections (four trailing dashes minimum). Do NOT use box-style headers with ``=====`` or ``####``.

- Always specify ``relationship`` argument in joins (e.g., ``relationship = "many-to-one"``)

- Prefer ``glue::glue()`` over ``sprintf()`` or ``paste0()`` for string interpolation

- Use ``scales::number()``, ``scales::percent()``, ``scales::comma()``, etc. for formatting numbers – not ``sprintf()``

- Use ``ggh4x::facet_wrap2()`` / ``facet_grid2()`` with ``axes = TRUE``, ``remove_labels = "all"`` instead of ``facet_wrap()`` / ``facet_grid()``

- Use ``summarize()`` (American spelling), not ``summarise()``

- Use tidyverse/readr I/O functions: ``read_rds()`` / ``write_rds()`` instead of ``readRDS()`` / ``saveRDS()``

### ## Citations

- When reporting a citation, triple-check that the authors, year, title, and journal are correct. Misattributing a citation is very serious. Verify against the actual source (e.g., the paper's webpage or PDF) rather than relying on memory or search snippet summaries.

# 3. Exploratory subgroup analysis

## Replication Data for: Local News, Information, and the Nationalization of U.S. Elections

Version 1.0



Moskowitz, Daniel J., 2020, "Replication Data for: Local News, Information, and the Nationalization of U.S. Elections", <https://doi.org/10.7910/DVN/HDDPTB>, Harvard Dataverse, V1, UNF:6:Rh5DGdtpz+cc70zx4CyXrA== [fileUNF]

Cite Dataset ▾

Learn about [Data Citation Standards](#).

Access Dataset ▾

Contact Owner

Share

Dataset Metrics ⓘ

1,905 Downloads ⓘ

Description ⓘ









Replication materials for "Local News, Information, and the Nationalization of U.S. Elections."

Subject ⓘ

Social Sciences

Related Publication ⓘ

MOSKOWITZ, DANIEL J. 2020. "Local News, Information, and the Nationalization of U.S. Elections." American Political Science Review, 1–16. [doi: 10.1017/S0003055420000829](https://doi.org/10.1017/S0003055420000829)

1 to 10 of 14 Files	
<input type="checkbox"/>	 <a href="#">cces_1216.tab</a> Tabular Data - 17.5 MB Published Nov 5, 2020 224 Downloads 55 Variables, 120554 Observations UNF:6:Vz9p...c8A== 
<input type="checkbox"/>	 <a href="#">cces_analysis.do</a> Stata Syntax - 38.9 KB Published Nov 5, 2020 157 Downloads MD5: 850...e06 
<input type="checkbox"/>	 <a href="#">county_covariates.tab</a> Tabular Data - 998.5 KB Published Nov 5, 2020 149 Downloads 40 Variables, 3141 Observations UNF:6:B/pq...7mg== 
<input type="checkbox"/>	 <a href="#">county_dma_analysis.do</a> Stata Syntax - 7.7 KB Published Nov 5, 2020 140 Downloads MD5: 0d8...53c 

- Does the effect of news on voting affect low information or high information voters?

> How does the effect vary by a respondent's level of news interest (newsint)? Report results for each of the four levels, and for each year, 2012-2016, and 2012-2024.

```
● Bash(Rscript 17_moskowitz-replication.R 2>&1 | grep -A 80 "=== By news interest ===")
└─ === By news interest ===
    --- Most of the time (newsint == 1) ---
        N = 82,869
        ... +55 lines (ctrl+o to expand)
└─ (timeout 5m)
```

● Here are the full results:

	newsint=1 (Most)	newsint=2 (Some)	newsint=3 (Now&then)	newsint=4 (Hardly)
2012	-0.012 (0.017)	<b>0.064 (0.027)</b>	0.046 (0.038)	-0.082 (0.106)
2016	0.022 (0.014)	<b>0.060 (0.025)</b>	0.028 (0.053)	0.087 (0.051)
2020	0.003 (0.022)	0.044 (0.058)	-0.023 (0.066)	0.212 (0.097)
2024	-0.025 (0.023)	-0.074 (0.046)	-0.171 (0.081)	0.092 (0.146)
<b>2012-16</b>	0.009 (0.011)	<b>0.064 (0.019)</b>	0.025 (0.034)	0.029 (0.059)
<b>2012-24</b>	0.006 (0.010)	<b>0.049 (0.018)</b>	-0.007 (0.032)	0.023 (0.054)
N (pooled)	47,943	17,654	5,639	2,100

**Takeaway:** The effect is entirely concentrated in **newsint=2 ("some of the time")** – the only group with consistent, significant positive coefficients across 2012 and 2016, and significant pooled estimates. newsint=1 (most attentive) shows zero effect, consistent with these voters getting information from many sources beyond local TV. newsint=3 and 4 have small samples and erratic estimates.

\* Churned for 31s

# Additional data collection analysis

```
> Merge in the TV news variable from the 2016 CCES (using get_cces_dataverse), which should be CC16_300_2. Merge in with V101 caseid. Do people who are CC16_300_2 == 1 have a higher effect?
```

- ```
Bash(Rscript -e 'library(dataverse); ?get_dataframe_by_name' 2>&1 | head -5)
└─ get_dataframe_by_name      package:dataverse      R Documentation

   _D_o_w_n_l_o_a_d_d_a_t_a_v_e_r_s_e_f_i_l_e_a_s_a_d_a_t_a_f_r_a_m_e
   ... +2 lines (ctrl+o to expand)
```
- ```
Bash(Rscript -e '
# Check if cumulative/common already has CC16_300...
└─ Attaching package: 'arrow'

The following object is masked from 'package:utils':
... +7 lines (ctrl+o to expand)
```
- Not in the common content. Let me check if there's a [cumulus/ccesMRPprep](#) or similar package with [get\\_cces\\_dataverse](#), or if I should use the [dataverse](#) package directly.
- ```
Bash(Rscript -e 'cat(system.file(package = "ccesMRPprep"), "\n"); cat(system.file(package =
"cumulus"), "\n")' 2>&1)
└─ /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/ccesMRPprep
```
- ```
Bash(Rscript -e 'library(ccesMRPprep);
cat(paste(ls("package:ccesMRPprep")[grep("cces|dataverse|get", ls("package:ccesMRPprep"),
ignore.case=TRUE)], collapse="\n"))'...)
└─ cces_dv_ids
cces_join_slim
ed_ed3_cces
... +5 lines (ctrl+o to expand)
```
- [get\\_cces\\_dataverse](#) is in [ccesMRPprep](#). Let me download the 2016 CCES and extract CC16\_300\_2.
- ```
Bash(Rscript -e '
library(ccesMRPprep)...
└─ Downloading and reading large dataset, can take about 3-5 minutes to complete.
   Downloading 1% █ 1m
   Downloading 4% █ 1m
... +22 lines (ctrl+o to expand)
```

- CC16\_300\_2: 1=Yes (watches TV news), 2=No. The column is named differently – the caseid in 2016 individual file might be [V101](#) under a different name. Let me find it.
- ```
Bash(Rscript -e '
library(ccesMRPprep); library(haven)...
└─ * Column `V101` doesn't exist.
   Backtrace:
     █
   ... +12 lines (ctrl+o to expand)
└─ (timeout 5m)
```
- It's [case\\_id](#), not [V101](#). Let me save the file properly.
- ```
Bash(Rscript -e '
library(ccesMRPprep); library(haven)...
└─ Saved. Rows: 64600
   case_id class: numeric
   CC16_300_2 distribution:
   ... +3 lines (ctrl+o to expand)
└─ (timeout 5m)
```
- Good. Now [case\\_id](#) is numeric (matches [V101](#) in Moskowitz). 1=Yes watches TV news (71%), 2=No (29%). This is 2016 only, so we can only run it for the 2016 subsample. Let me add this to the script.
- Now also need to add [case\\_id](#) from [analysis\\_new](#) so the column exists in [pooled](#) (even though we'll only use it for 2016). Let me check – [analysis\\_new](#) doesn't have a [case\\_id](#) column. The TV news merge is only for 2016 which all comes from [mosk\\_clean](#), so [V101\\_chr](#) being NA for [analysis\\_new](#) rows is fine. But [bind\\_rows](#) will handle the missing column. Let me run it.
- ```
Bash(Rscript 17_moskowitz-replication.R 2>&1 | grep -A 10 "=== TV news")
└─ === TV news viewership (CC16_300_2, 2016 only) ===
   2016 with TV news variable: 34,192 (TV=1: 24,955, TV=0: 9,237)
   NOTE: 11,488 observations removed because of NA values (LHS: 11,455, RHS: 39, Fixed=ef
   ... +9 lines (ctrl+o to expand)
└─ (timeout 5m)
```
- **TV news viewership results (2016 only):**

Group	Coef	SE	p	N
Watches TV news (CC16_300_2=1)	<b>0.040</b>	0.015	<b>0.009</b>	13,467
Does NOT watch TV news (CC16_300_2=2)	0.016	0.018	0.373	4,246

Yes – the Moskowitz effect in 2016 is concentrated among people who report watching TV news. The coefficient is 2.5× larger and significant (p=.009) for TV viewers, versus small and insignificant for non-viewers. This is exactly what the theory predicts: higher in-state DMA share → more local TV news exposure → more candidate-specific information → more split-ticket voting, but only if you actually watch TV news.

# Extensions

- Full automation
- Multiple skills, agents
- Automated checks before posting

# Scaling Reproducibility: An AI-Assisted Workflow for Large-Scale Reanalysis

Yiqing Xu  
(Stanford)

Leo Yang Yang  
(HKBU)

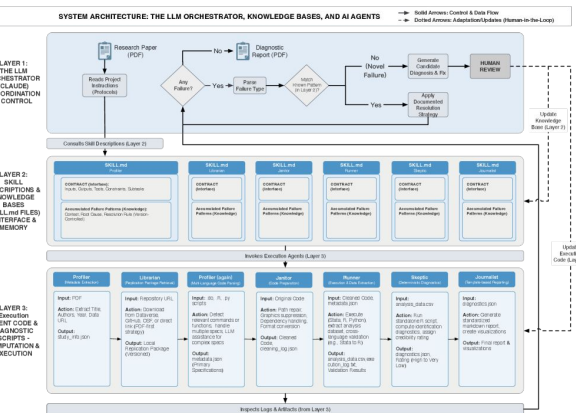
Yiqing Xu, Assistant Professor, Department of Political Science,  
Stanford University. Email: [yiqingxu@stanford.edu](mailto:yiqingxu@stanford.edu).

Leo Yang Yang, Research Assistant Professor, Department of  
Accountancy, Economics and Finance, School of Business,  
Hong Kong Baptist University, Kowloon, Hong Kong SAR.

Email: [leoyang@hkbu.edu.hk](mailto:leoyang@hkbu.edu.hk).

## Abstract

Reproducibility is central to research credibility, yet large-scale reanalysis of empirical data remains costly because replication packages vary widely in structure, software environment, and documentation. We develop and evaluate an agentic AI workflow that addresses this execution bottleneck while preserving scientific rigor. The system separates scientific reasoning from computational execution: researchers design fixed diagnostic templates, and the workflow automates the acquisition, harmonization, and execution of replication materials using pre-specified, version-controlled code. A structured knowledge layer records resolved failure patterns, enabling adaptation across heterogeneous studies while keeping each pipeline version transparent and stable. We evaluate this workflow on 92 instrumental variable (IV) studies, including 67 with manually verified reproducible 2SLS estimates and 25 newly published IV studies under identical criteria. For each paper, we analyze up to three two-stage least squares (2SLS) specifications, totaling 215. Across the 92 papers, the system achieves 87% end-to-end success overall. Conditional on accessible data and code, reproducibility is 100% at both the paper and specification levels. The framework substantially lowers the cost of executing established empirical protocols and can be adapted in empirical settings where analytic templates and norms of transparency are well established.



# Caution / Disclaimers

- This presentation uses Claude Code, but is not an endorsement of Claude Code over Codex
- This uses the expensive Max plan – Pro plan (\$20/month) does have Claude Code but is time limited, and Free plan does not have Claude Code
- You need to know what you want to make Claude Code *do* in order to effectively use it
- Analysis: More for quick “what if I did ...” questions than doing the main analysis
- Relying too much on automated analyses can atrophy your skills