# A TAO of Data Wrangling: A Practical Routine for Getting Past the 'Janitor Work'*

**Preliminary version. Do not circulate without permission.**

Yue Hu[1] Yuehong Cassandra Tai[2] Frederick Solt[3]

This article focuses on a preliminary step in any ex-post data harmonization project—wrangling the pre-harmonized data—and suggests a practical routine for helping researchers reduce human errors in this often-tedious work. The routine includes three steps: (1) *T*eam-based concept construct and data selection; (2) Data entry *a*utomation; and (3) "Second-order" *o*pening—a "Tao" of data wrangling. We illustrate the routine with the examples of pre-harmonizing procedures used to produce the Standardized World Income Inequality Database (SWIID), a widely used database that uses Gini indices from multiple sources to create comparable estimates, and the Dynamic Comparative Public Opinion (DCPO) project, which creates a workflow for harmonizing aggregate public opinion data.

[1] Department of Political Science, Tsinghua University, Beijing, China

[2] Center for Social Data Analytics, Pennsylvania State University, University Park, USA

[3] Department of Political Science, University of Iowa, Iowa City, USA

*Corresponding author: Yuehong Cassandra Tai, yhcasstai@psu.edu. Current version: June 20, 2025. Replication materials and complete revision history may be found at https://github.com/fsolt/wrangling_data. The authors contributed equally to this work. Yue Hu appreciates the funding support from the National Natural Science Foundation of China (72374116) and Tsinghua University Initiative Scientific Research Program (2024THZWJC01).

# 1 A Wrangling Issue of Data Harmonization

Empowered by the spreading Internet and advancing computational power, researchers have entered an unprecedented age of data availability. A growing volume of social science research aims to take the benefit to extend the generality: they employ large quantities of data drawn from different sources. However, ensuring the quality of harmonized datasets remains a significant challenge in handling fitness to use and raw data quality monitoring among others (Slomczynsi, Tomescu-Dubrow, and Wysmulek 2025). Beyond the focus on the harmonization process itself, we argue that quality assurance must begin *earlier* to the data-wrangling step where raw inputs are selected, processed, and prepared for harmonization.

The wrangling step determines the quality of data in the harmonization process, and the challenges is how to properly and transparently clean the increasing amount and diversity of data. The conventional approach usually involves a notorious bulk of manual work on indicator identification, data merging, data scaling, and so on (see, e.g., Lohr 2014). The tiresome task is easy to introduce errors in data collection procedure. Manual wrangling make a full reproducibility of research pipeline more difficulty and undermine the transparency (Liu and Salganik 2019).

These challenges are amplified when raw data comes from heterogeneous sources and have been processed using various software environments over time. This is a common scenario in secondary data collection. For example, older survey files stored in SPSS's ASCII or portable formats often require extensive restructuring before they can be merged with new format of data. Such undocumented transformations make it difficult to track changes and undermine transparency.

Finally, even meticulous documentation cannot eliminate the influence of human discretion embedded in manual processing. Such discretion leaves behind few traces, making it difficult for collaborators or reviewers to verify the wrangling process or trace sources of error.

In short, poor source data quality, the absence of reproducibility, and untrackable human discretion in manual janitor work have collectively became the largest obstacle on the way to data harmonization, which yet have thus far gained little attention.

In this article, we provide a practical routine (a "TAO") taken advantage of automatic programming and teamwork to reduce such data-entry errors and improve the repro-

ducibility and transparency of the wrangling process for researchers and reviewers to check the errors. This TAO routine covers the three phases of data wrangling: data collection/selection, data entry, and opening. We illustrate how researchers use this routine on statistical (*hard*) and opinion (*soft*) data with two ongoing harmonization efforts, the Standardized World Income Inequality Database (SWIID) and the Dynamic Comparative Public Opinion (DCPO) project.

## 2 A 3-Step "Tao" for Data Wrangling

Our routine aims to helping researchers reach three goals for scientific research:

1. To reduce the manual entry errors to improve the accuracy of the harmonized data and analytic data;
2. To incorporate as much available data as possible to provide a base for comparable data and increase generality of the inferences; and
3. To improve the reproducibility of data wrangling process for the sake of transparency.

The routine decomposes a data-wrangling process into three steps:

1. **T**eam-based concept construct and data selection;
2. Data entry **a**utomation; and
3. "Second-order" **o**pening.

To illustrate the above routine, we use two data harmonization projects as examples, SWIID and DCPO. SWIID is a long-running project that seeks to provide harmonized income inequality statistics for the broadest possible coverage of countries and years (Solt 2009, 2015, 2016, 2020). As of its most recent update at the time of this writing, its source data consists of some 27,000 observations of the Gini coefficient of income distribution in nearly 200 countries over as many as 65 years, collected from over 400 separate sources including international organizations, national statistics bureaus, and academic studies.

DCPO is both a method and a database. Scholarship on comparative public opinion only rarely benefits from relevant items asked annually by the same survey in many countries (see, e.g., Hagemann, Hobolt, and Wratil 2017). To address the lack of cross-national

and longitudinal data on many topics, a number of works have presented latent variable models that harmonize available but incomparable survey items (see e.g., Caughey, O'Grady, and Warshaw 2019; Claassen 2019; Kołczyńska et al. 2024). Along this line, DCPO not only provides latent variable measurements but also automatized and reproducible data collection (Solt 2020), which has been applied in a complete pipeline for a variety of topics such as gender egalitarianism (Woo, Goldberg, and Solt 2023), political interest (Hu and Solt 2024), and support for gay rights (Woo et al. 2024), among other aspects of public opinion and open it freely for global researchers (see more updated data collections at https://dcpo.org/).

In the following sections we first address the common challenges for the phases of data wrangling and explain how our routine can help deal with it illustrated with the data wrangling processes of the SWIID and DCPO projects.

## 2.1 Step 1: Team-Based Construct Building and Data Selection

Large scale of data selection and cleaning is almost always tedious, as something to be delegated to research assistants, to someone—indeed anyone, but usually research assistants (RA)—else (see Torres 2017). This manual procedure is easy to make mistakes and errors. Haegemans, Snoeck, and Lemahieu (2019, 1) has demonstrated examples of misrouted financial transactions and airline flights. In a more systematic examination, Barchard and Pace (2011) found that RA assigned in an experiment to carefully enter data manually and instructed to prioritize accuracy over speed still had error rates approaching 1% in just a single roughly half-hour session. The consequences of such errors can be pernicious.

Our antidote for this issue is a combination of team work and automation. We will focus more on the team work and discuss the latter in OSM 2.2. The goal here is to have consistent understanding on conceptualized construct, select valid data for later measurement and/or analyses, and reduce biases caused by inconsistent human judgment. A team work framework for this end requires a deliberative set and a dual-entry process.

A deliberative set requires the members in a research team—regardless several coauthors or a primary author with one or two RAs—to have a clear and coherent understanding of the reseach questions and associated data goals. These understandings will help the team members identify the right data to collect and discover extra useful data

sources that are not in the initial plan.

In the SWIID program, for example, we told RAs that the goal of the research is to generate comparable statistics of country-level economic inequality. We provide a list of sources mainly from national statistic bureaus for them to start, but we also told them that updated statistics for some countries may come from academic papers, published documents, and other sources, and they are free to add them in while making sure a valid link of the new sources are also recorded.

Ensuring team members to understand how the data would use later is also important, as they could have a better sense of what data are analyticable and a forward perspective of how many situations would the later entry part need to take care. In the SWIID project, we told the RAs that the inequality statistics be recorded in four formats: Gini index in disposable (post-tax, post-transfer) income, Gini in market (pre-tax, pre-transfer) income, absolute redistribution (market-income inequality minus net-income inequality), or relative redistribution (market-income inequality minus net-income inequality, divided by market-income inequality). So, for later unification work, they need not only to record the digits but also seek documents to explain the methods of the statistics.

The SWIID project requires update for almost every year and we also often hire new RAs. Therefore, the cross-check is done in a rolling basis usually by the rookies who are in charge of checking the old data and updating malfunctional links. This is both a learning process and a way to improve data accuracy.

In the DCPO project, clearly defining and agreeing upon the latent construct among team members is a critical first step for ensuring theoretical comparability across countries and over time (Koc and Kołczyńska 2025). This process begins with a shared conceptual foundation established through literature review and corresponding pre-defined potential dimensions of the latent opinion. Each team member is then assigned survey datasets from specific geographic regions and tasked with identifying potentially relevant items and potential dimensions based on both general theoretical guidance and region-specific knowledge. This structure ensures that the construct is informed by both global theory and local context.

Before data selection begins, team members undergo hands-on training on how the method work and what type of data and detail they need to collect, such as data format and weighting types, which provide a valuable help of later build the automative data

preparation software.

Following the initial round of item selection and collection, the dural-entry section comes in. In this stage, each team member reviews and re-codes the survey data originally handled by another member. The independently coded versions are then compared to detect discrepancies, which may arise from misinterpretations of the construct, ambiguous item wording, or common entry errors.

Disputed cases are flagged for group discussion. Some mismatches may indicate items that may not be conceptually equivalent across cultures or regions, and others suggest multidimensionality that requires theoretical disaggregation. For the latter, we either categorize such items into pre-defined dimensions and/or revise the codebook accordingly to add new dimensions—an iterative process aimed at improving construct validity, intercoder reliability, and reducing oversimplification of target variable (Slomczynsi, Tomescu-Dubrow, and Wysmulek 2025).

Therefore, we broke down the cross-check step into several lab meetings interspersed during the data selection to collect new insights from each members' selection works and make sure everyone were on the same page through the whole process. The process ends with a systemic cross-check of the final selected data among members.

In addition to reducing manual biases, teamwork also helps expand the data pool. Both SWIID and DCPO projects enrolled team members from countries other than the U.S. These members have well used their language and cultural advantages to discover more data recorded in non-English languages and improve the precision of the data selections. To some extent, data from different sources also help correct the biases caused by the designers' cultural backgrounds.

## 2.2 Step 2: Data Entry Automation

Formatting data is arguably the easiest step to involve manual errors and controversies. The best solution is to automate the entry process taken the advantages of the programming languages and application programming interfaces (APIs) of the data source.

In the DCPO case, data entry is fully automated through the R-based software, `DCPOtools` (Solt, Hu, and Tai 2018). This software processes raw survey files directly, ensuring reproducible data entry. It converts various file formats to R-readable objects, extracts variables of interest, reorders response values, applies survey weights, and

aggregates weighted respondents by country and year based on actual fieldwork dates.

To address theoretical comparability concerns, DCPO employs conservative filtering, removing items appearing in fewer than five country-years in countries surveyed at least three times, minimizing the risk of sacrificing comparability for coverage (Koc and Kołczyńska 2025). `DCPOtools` standardizes country names using Arel-Bundock, Enevoldsen, and Yetman (2018)'s `countrycode` and ensures years reflect actual fieldwork dates, creating aggregated respondent data for the latent variable model.

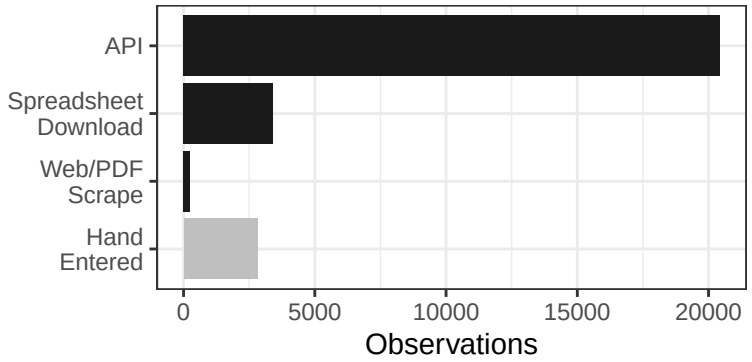


Figure 1: Income Inequality Observations by Method of Collection

While coding datasets and items into structured spreadsheets facilitates automation, an even better version starts the automation since the data selection step via programming and APIs. As shown in Figure 1, the current version of SWIID grapes 76% of the observations through API. In OSM 4.2, we also provide an examplary list of R packages that can ease the processes to, for example, collect data with APIs and clean and transform data. The list has been long, but far from complete. If readers modify the arguments of the codes in this article's replication file that we use to create the list, they will discover times more packages that already exist to help collect and wrangle data. The keyword-based searching function that most these packages equip also ensure researchers to conduct data harmonization analysis with most updated data pools the sources have.

When no API is available, the automation script downloads and reads any available spreadsheets (see Wickham 2016). In the absence of a spreadsheet, the process of scraping the data either directly from the web or, preferably, from a pdf file (see Sepulveda 2024) is automated. Together the collection of 90% of the source data is scripted. This means not only that the possibility of errors introduced by hand entry for a vast majority of observations is eliminated but also that the updates and revisions that are frequent in

these data are automatically incorporated as they become available.[1]

For data sources, such as those from academic articles or books, that have to be entered in hand, there is still rooms for automation. For the remaining 10% of the SWIID observations, for instance, we collected them using Sepulveda's `tabulapdf` R package to avoid data-entry errors as long as they are in pdf (Sepulveda 2024). The advanced Optical Character Recognition (OCR) can extend this method on data sources even in hard copies.

And finally, for data that one has to enter manually, the team-based working flow can be supplementary. One source of SWIID contains crucial information encoded in the typeface of its tables (see Mitra and Yemtsiv 2006, 6); this information would be lost if the tables were read directly into R. We reapplied the approach from the data selection here to enter them twice into separate spreadsheets.[2] The dual-entry process allows for automated cross-checks of the newly entered data that increase the chances that errors are identified and corrected (see Barchard and Pace 2011).

## 2.3 Step 3: "Second-Order" Opening

Since the replication crisis, replication files for analytical results in academic articles has become a standard requirement for top-tier journals in political science (Chang and Li 2015; Open Science Collaboration 2015). Nevertheless, the continual raising controversies on the researcher degrees of freedom indicated that current open is still not adequate.[3] Especially in relation with data harmonization, we eager researchers to conduct a, what we called, the "second-order" opening. That is, not only opening analytical steps (the "first-order") but also the data generation process (the "second-order"), including data collection, cleaning, and wrangling.

Empirical evidence has indicated the severe consequences without the second-order opening. A recent research has found that the variation in the estimated effects caused by researchers may outweigh the population's variation (**?**). Within these researcher-choice variations, a substantial portion comes from the data-wrangling process (**?**). In a "many-analyst" analysis, (**?**) requested 146 research teams to complete the same research task. The study found that the teams who were given the same research design but no pre-cleaned data set generated the highest outcome variation—even higher than those teams who were only given the research task. The teams who were given the pre-cleaned

data set generated the lowest outcome variation. These findings indicate that the research replicability cannot be guaranteed if only with the first- but not the second-order opening.

If researchers apply our suggestions of team-based construct building, systematic data selection, and automated data entry, the second-order opening will be both feasible and efficient. Along with a clearly conceptualized theoretical framework, researchers can simply share their programming scripts for data downloading, formatting, and wrangling, ensuring that the full pipeline is documented and reproducible.

With developed scientific and technical publishing system, such as Quarto or R markdown, and version control platforms (e.g., Github) and open collaboration platforms (e.g., Open Science Framework, OSF), researchers can integrate the entire workflow—from raw data collection to final analysis—within a single, publicly trackable archive. We reached at this step for all the DCPO projects so far. Readers can find a Github repo for the research from scratch, and every wave of data update in the corresponding OSF project.[4]

# 3 Discussion

Figure 2 presents the whole process of the 3-step routine of data wrangling for later harmonization phase. Implementing these practices requires effort, just as in many open-science endeavors (see Engzell and Rohrer 2021). Though labor-intensive, the double-entry method reduces error rates thirty-fold (Barchard and Pace 2011, 1837), justifying the investment. Teamwork distributes tasks, reducing fatigue-related errors, while allowing discrepancies to be resolved through discussion.

Social scientists now benefit from standardized harmonization workflows (Slomczynsi, Tomescu-Dubrow, and Wysmulek 2025) and automated data processing (Kritzinger, Lutz, and Boomgaarden 2025). Researchers can reuse high-quality harmonized datasets, enhancing efficiency and comparability. Open-source software packages like those used by the SWIID and `DCPOtools` have already automated many data preparation tasks. With large language models emerging, intelligent agents may soon handle parts of these routines, potentially advancing automation to new levels (Kritzinger, Lutz, and Boomgaarden 2025).

A final point we would like to clarify is that, in our three-step routine, researchers remain central to harmonization. As illustrated in the SWIID and DCPO examples, re-

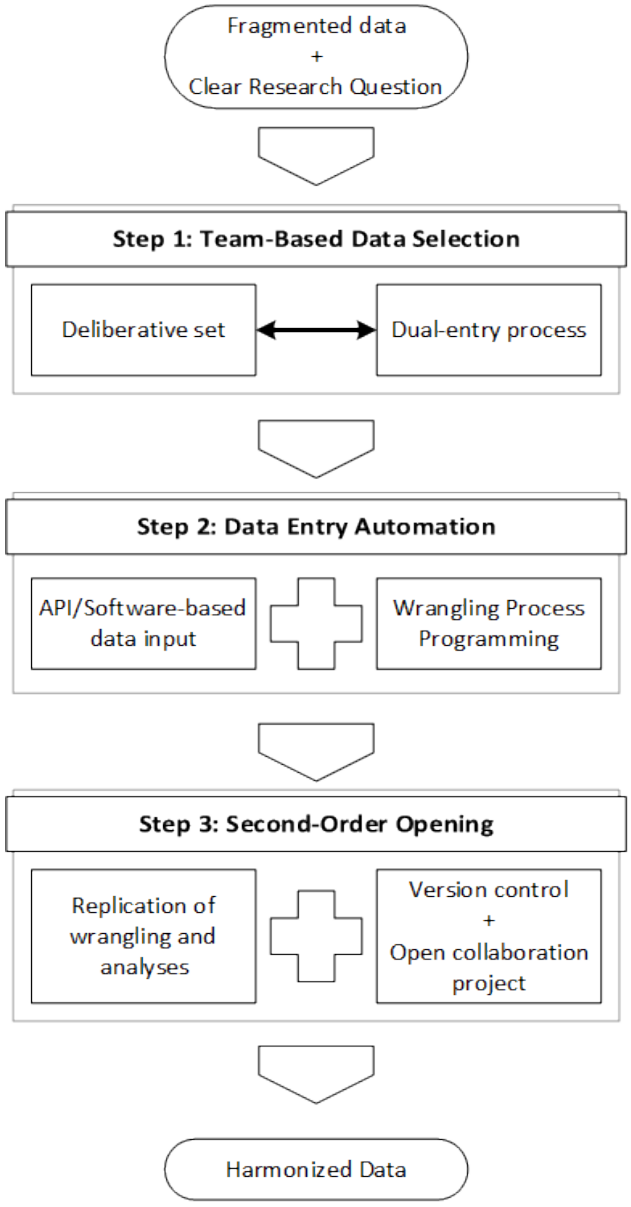


Figure 2: The TAO of Data Wrangling Before Data Harmonization. Source: Self generated.

searchers are responsible for all critical decisions from clarifying research questions and building theoretical constructs to conducting version control and developing replication materials. Early and critical steps, such as construct development and codebook refinement, must be conducted iteratively to achieve high intercoder reliability. Even with automated data entry, human validation remains essential for verifying variable formats and value ranges. Computing environments should be documented to minimize system-related discrepancies (Liu and Salganik 2019).

For ex-post harmonization projects, careful attention to pre-harmonization stages substantially contributes to overall dataset quality. While some error is inevitable, with responsible researcher oversight, data-entry errors can be minimized while transparency, openness, and research credibility continue to grow.

# Notes

[1] The R community has often built software to ease the access of APIs and make the batch work for multiple waves of data in a more comfortable and efficient way (see Blondel 2018; Lahti et al. 2017; Lugo 2017; Magnusson, Lahti, and Hansson 2014; Wickham, Hester, and Ooms 2018).

[2]Most often this has been done by two different investigators, but sometimes sequentially by a single researcher.

[3]See a summary of the "researcher degrees of freedom" discussion in Hu, Tai, and Solt (2024).

[4]See a comprehensive example applied the second-order opening strategy in Tai, Hu, and Solt (2024).

# References


Arel-Bundock, Vincent, Nils Enevoldsen, and C. J. Yetman. 2018. "countrycode: Convert Country Names and Country Codes." *Journal of Open Source Software* 3(28): 848–49.

Barchard, Kimberly A., and Larry A. Pace. 2011. "Preventing Human Error: The Impact of Data Entry Methods on Data Accuracy and Statistical Results." *Computers in Human Behavior* 27(5): 1834–39.

Blondel, Emmanuel. 2018. "rsdmx: Tools for Reading SDMX Data and Metadata."

Caughey, Devin, Tom O'Grady, and Christopher Warshaw. 2019. "Policy Ideology in European Mass Publics, 1981–2016." *American Political Science Review*: 1–20. doi:10.1017/S0003055419000157.

Chang, Andrew, and Phillip Li. 2015. "Is Economics Research Replicable? Sixty Published Papers from Thirteen Journals Say 'Usually Not'." *Finance and Economics Discussion Series* 7: 1–25.

Claassen, Christopher. 2019. "Estimating Smooth Country–Year Panels of Public Opinion." *Political Analysis* 27(1): 1–20.

Engzell, Per, and Julia M. Rohrer. 2021. "Improving Social Science: Lessons from the Open Science Movement." *PS: Political Science & Politics* 54(2): 297–300.

doi:10.1017/S1049096520000967.

Haegemans, Tom, Monique Snoeck, and Wilfried Lemahieu. 2019. "A Theoretical Framework to Improve the Quality of Manually Acquired Data." *Information & Management* 56(1): 1–14.

Hagemann, Sara, Sara B. Hobolt, and Christopher Wratil. 2017. "Government Responsiveness in the European Union: Evidence from Council Voting." *Comparative Political Studies* 50(6): 850–76.

Hu, Yue, and Fredrick Solt. 2024. "Macrointerest Across Countries." In Shenzhen.

Hu, Yue, Yuehong Cassandra Tai, and Frederick Solt. 2024. "Revisiting the Evidence on Thermostatic Response to Democratic Change: Degrees of Democratic Support or Researcher Degrees of Freedom?" *Political Science Research and Methods*: 1–7. doi:10.1017/psrm.2024.16.

Koc, Piotr, and Marta Kołczyńska. 2025. "Modeling Trends in Public Opinion: An Overview of Approaches, Assumptions and Trade-Offs." *Working Paper*.

Kołczyńska, Marta, Paul-Christian Bürkner, Lauren Kennedy, and Aki Vehtari. 2024. "Modeling Public Opinion over Time and Space: Trust in State Institutions in Europe, 1989-2019." *Survey Research Methods* 18(1): 1–19.

Kritzinger, Sylvia, Georg Lutz, and Hajo Boomgaarden. 2025. "Visions for the Future: Challenges and Opportunities for Creating Sustainable Scholarly Infrastructures for Data Harmonization." *Working Paper*.

Lahti, Leo, Janne Huovari, Markus Kainu, and Przemysław Biecek. 2017. "Retrieval and Analysis of Eurostat Open Data with the Eurostat Package." *The R Journal* 9(1): 385–92.

Liu, David M, and Matthew J Salganik. 2019. "Successes and Struggles with Computational Reproducibility: Lessons from the Fragile Families Challenge." *Socius* 5: 2378023119849803.

Lohr, Steve. 2014. "For Data Scientists, 'Janitor Work' Is Hurdle to Insights." *New York Times*: B4.

Lugo, Marco. 2017. "CANSIM2R: Directly Extracts Complete CANSIM Data Tables."

Magnusson, Mans, Leo Lahti, and Love Hansson. 2014. "pxweb: R Tools for PX-WEB API."

Mitra, Pradeep, and Ruslan Yemtsiv. 2006. "Increasing Inequality in Transition Economies: Is There More to Come?"

Open Science Collaboration. 2015. "Estimating the Reproducibility of Psychological Science." *Science* 349(6251): aac4716.

Sepulveda, Mauricio Vargas. 2024. "tabulapdf: Extract Tables from PDF Documents."

Slomczynsi, Kazmierz M., Irina Tomescu-Dubrow, and Ilona Wysmulek. 2025. "Navigating Complexities of Ex-Post Harmonization of Cross-National Survey Data: Insights from the Survey Data Recycling, SDR, Project." *Working Paper*.

Solt, Frederick. 2009. "Standardizing the World Income Inequality Database." *Social Science Quarterly* 90(2): 231–42.

Solt, Frederick. 2015. "On the Assessment and Use of Cross-National Income Inequality Datasets." *Journal of Economic Inequality* 13(4): 683–91.

Solt, Frederick. 2016. "The Standardized World Income Inequality Database." *Social Science Quarterly* 97(5): 1267–81.

Solt, Frederick. 2020. "Measuring Income Inequality Across Countries and over Time: The Standardized World Income Inequality Database." *Social Science Quarterly* 101(3, 3): 1183–99. doi:10.1111/ssqu.12795.

Solt, Frederick, Yue Hu, and Yuehong Tai. 2018. "DCPOtools: Tools for Dynamic Comparative Public Opinion."

Tai, Yuehong 'Cassandra', Yue Hu, and Frederick Solt. 2024. "Democracy, Public Support, and Measurement Uncertainty." *American Political Science Review* 118(1): 512–18. doi:10.1017/S0003055422000429.

Torres, Rachel. 2017. "Me: Shouldn't There Be Someone in a Basement That We Just Pay to Do All This Awful Data Cleaning? Advisor: That's Who You Are."

Wickham, Hadley. 2016. "rvest: Easily Harvest (Scrape) Web Pages."

Wickham, Hadley, James Hester, and Jeroen Ooms. 2018. "xml2: Parse XML."

Woo, Byung-Deuk, Lindsey A. Goldberg, and Frederick Solt. 2023. "Public Gender Egalitarianism: A Dataset of Dynamic Comparative Public Opinion Toward Egalitarian Gender Roles in the Public Sphere." *British Journal of Political Science* 53(2): 766–75. doi:10.1017/S0007123422000436.

Woo, Byung-Deuk, Hyein Ko, Yuehong Cassandra Tai, Yue Hu, and Frederick Solt. 2024. "Public Support for Gay Rights Across Countries and over Time." *Social Science Quarterly* 106(1): e13478. doi:10.1111/ssqu.13478.

# 4 Supplementary Materials

## 4.1 Checklist for Deliberation Process

Below is a checklist with notes or rationales for key decisions made during the deliberation process. The focus on each step may vary depending on the research purpose. For

Table 1: Checklist with Decision Rationale

| Step | Checklist | Notes |
|---|---|---|
| 1.Clarify Conceptual Construct | Literature review and shared across the team. | Notes from team discussion |
| | Confirm shared understanding of theoretical construct. | |
| | Relevant theoretical dimensions are discussed and documented. | |
| 2.Document Research Goals | Instructions on key variable formats and downstream analytical needs. | Update when necessary |
| | Review initial codebook. | |
| | Data input training. | |
| 3.Assign Data Collection | Assign datasets to team members by geography or source. | Document ambiguous items |
| | Each team member maintains a seperate sheet for raw data collection and a log of decisions. | |
| 4.Dual-Entry and Cross-Check | Conduct dual entry by second team member. | Record discrepancies found |
| | Discrepancies flagged and logged for group discussion. | |
| 5.Deliberation on Discrepancies | Team discussion on discrepancies. | Provide examples of key disputes and how they were resolved |
| | Items with unclear mapping to conceptual dimensions are categorized or excluded. | |
| | Update codebook/documentation | |
| 6.Log Data and Decision | Log all decisions and changes in version-controlled repository (e.g., OSF, GitHub). | Mention major updates. |

example, in public opinion harmonization projects like DCPO, more time is typically devoted to conceptualization and construct development compared to administrative data projects such as SWIID. However, this general checklist can serve as a useful guide across a range of harmonization efforts.

## 4.2 R packages for data wrangling

Here are the tables of example R packages that researchers can use to collect, clean, and transform data. The tables were generated by the `pkgsearch::pkg_search()` function with the keywords relating to data downloading (**?@tbl-download**), wrangling (**?@tbl-clean**), and transforming (**?@tbl-transform**). The packages are ranked based on the 'score' metric that reflects both textual relevances with the keyword and package popularity in the last month. Only the top twenty packages and only the maintainers' names are shown. We encourage readers to use the codes in this paper's replication file to ex-

plore more useful packages. We also recommend readers to refer to the "CRAN Task View: Reproducible Research" page for more useful tools to achieve the first-order and second-order opening.

Table 2: Example packages for downloading data with API

| package | title | maintainer |
|---|---|---|
| giscoR | Download Map Data from GISCO API - Eurostat | Diego Hernangómez |
| rwebstat | Download Data from the Webstat API | Vincent Guegan |
| crypto2 | Download Crypto Currency Data from 'CoinMarket-Cap' without 'API' | Sebastian Stoeckl |
| RKaggle | 'Kaggle' Dataset Downloader 'API' | Benjamin Smith |
| csodata | Download Data from the CSO 'PxStat' API | Conor Crowley |
| hansard | Provides Easy Downloading Capabilities for the UK Parliament API | Evan Odell |
| cranlogs | Download Logs from the 'RStudio' 'CRAN' Mirror | Gábor Csárdi |
| clinicalomicsdbR | Interface with the 'ClinicalOmicsDB' API, Allowing for Easy DataDownloading and Importing | John Elizarraras |
| wdi2 | Download World Development Indicators from the World BankIndicators API | Christoph Scheuch |
| GDELTtools | Download, Slice, and Normalize GDELT V1 Event and Sentiment APIData | Stephen R. Haptonstahl |
| neonUtilities | Utilities for Working with NEON Data | Claire Lunch |
| piggyback | Managing Larger Data on a GitHub Repository | Carl Boettiger |
| nasapower | NASA POWER API Client | Adam H. Sparks |
| Quandl | API Wrapper for Quandl.com | Dave Dotson |
| rstudioapi | Safely Access the RStudio API | Kevin Ushey |
| rdhs | API Client and Dataset Management for the Demographic and HealthSurvey (DHS) Data | OJ Watson |
| fishtree | Interface to the Fish Tree of Life API | Jonathan Chang |
| ridigbio | Interface to the iDigBio Data API | Jesse Bennett |
| tradestatistics | Open Trade Statistics API Wrapper and Utility Program | Mauricio Vargas |
| FlickrAPI | Access to Flickr API | Koki Ando |
| easycensus | Quickly Find, Extract, and Marginalize U.S. Census Tables | Cory McCartan |
| shutterstock | Access 'Shutterstock' REST API | Metin Yazici |
| wbstats | Programmatic Access to Data and Statistics from the World BankAPI | Jesse Piburn |
| gwasrapidd | 'REST' 'API' Client for the 'NHGRI'-'EBI' 'GWAS' Catalog | Ramiro Magno |
| ecos | Economic Statistics System of the Bank of Korea | Seokhoon Joo |
| rscopus | Scopus Database 'API' Interface | John Muschelli |
| I14Y | Search and Get Data from the I14Y Interoperability Platform ofSwitzerland | Felix Luginbuhl |
| riingo | An R Interface to the 'Tiingo' Stock Price API | Davis Vaughan |
| PurpleAir | Query the 'PurpleAir' Application Programming Interface | Cole Brokamp |
| kaigiroku | Programmatic Access to the API for Japanese Diet Proceedings | Akitaka Matsuo |

Continued on next page

Table 2: Example packages for downloading data with API (Continued)

| | | |
|---|---|---|
| mgpStreamingSDK | Interact with the Maxar MGP Streaming API | Nathan Carr |
| GetLattesData | Reading Bibliometric Data from Lattes Platform | Marcelo Perlin |
| worldbank | Client for World Banks's 'Indicators' and 'Poverty andInequality Platform (PIP)' APIs | Maximilian Mücke |
| BFS | Get Data from the Swiss Federal Statistical Office | Felix Luginbuhl |
| trud | Query the 'NHS TRUD API' | Alasdair Warwick |
| jsonlite | A Simple and Robust JSON Parser and Generator for R | Jeroen Ooms |
| rinat | Access 'iNaturalist' Data Through APIs | Stéphane Guillou |
| yfinancer | 'Yahoo Finance' API Wrapper | Giovanni Colitti |
| zen4R | Interface to 'Zenodo' REST API | Emmanuel Blondel |
| opendotaR | Interface for OpenDota API | Kari Gunnarsson |
| Visualize.CRAN.Downloads | Visualize Downloads from 'CRAN' Packages | Marcelo Ponce |
| PurpleAirAPI | Historical Data Retrieval from 'PurpleAir' Sensors via API | Heba Abdelrazzak |
| pacu | Precision Agriculture Computational Utilities | dos Santos Caio |
| cbsodataR | Statistics Netherlands (CBS) Open Data API Client | Edwin de Jonge |
| kosis | Korean Statistical Information Service (KOSIS) | Seokhoon Joo |
| MetaculR | Analyze Metaculus Predictions and Questions | Joseph de la Torre Dwyer |
| trelloR | Access the Trello API | Jakub Chromec |
| rscorecard | A Method to Download Department of Education College ScorecardData | Benjamin Skinner |
| naptanr | Call the 'NaPTAN' API Through R | Francesca Bryden |
| inegiR | Integrate INEGI's (Mexican Stats Office) API with R | Eduardo Flores |

Table 3: Example packages for wrangling data with API

| package | title | maintainer |
|---|---|---|
| discretization | Data Preprocessing, Discretization for Classification | HyunJi Kim |
| helda | Preprocess Data and Get Better Insights from Machine LearningModels | Simon Corde |
| recipes | Preprocessing and Feature Engineering Steps for Modeling | Max Kuhn |
| dunlin | Preprocessing Tools for Clinical Trial Data | Joe Zhu |
| dataprep | Efficient and Flexible Data Preprocessing Tools | Chun-Sheng Liang |
| smallsets | Visual Documentation for Data Preprocessing | Lydia R. Lucchesi |
| rtry | Preprocessing Plant Trait Data | Olee Hoi Ying Lam |
| PupilPre | Preprocessing Pupil Size Data | Aki-Juhani Kyröläinen |
| mpactr | Correction of Preprocessed MS Data | Patrick Schloss |
| bdpar | Big Data Preprocessing Architecture | Miguel Ferreiro-Díaz |
| webtrackR | Preprocessing and Analyzing Web Tracking Data | David Schoch |
| tsrobprep | Robust Preprocessing of Time Series Data | Michał Narajewski |
| VWPre | Tools for Preprocessing Visual World Data | Vincent Porretta |
| binst | Data Preprocessing, Binning for Classification and Regression | Chapman Siu |
| PreProcessing | Various Preprocessing Transformations of Numeric Data Matrices | Swamiji Pravedson |
| esmtools | Preprocessing Experience Sampling Method (ESM) Data | Jordan Revol |

Continued on next page

Table 3: Example packages for wrangling data with API (Continued)

| | | |
|---|---|---|
| RobLoxBioC | Infinitesimally Robust Estimators for Preprocessing - Omics Data | Matthias Kohl |
| shinyrecipes | Gadget to Use the Data Preprocessing 'recipes' PackageInteractively | Alberto Almuiña |
| RGCxGC | Preprocessing and Multivariate Analysis of Bidimensional GasChromatography Data | Cristian Quiroz-Moreno |
| cobalt | Covariate Balance Tables and Plots | Noah Greifer |
| EEM | Read and Preprocess Fluorescence Excitation-Emission Matrix(EEM) Data | Vipavee Trivittayasil |
| clickR | Semi-Automatic Preprocessing of Messy Data with Change Trackingfor Dataset Cleaning | David Hervas Marin |
| SerolyzeR | Reading, Quality Control and Preprocessing of MBA (MultiplexBead Assay) Data | Tymoteusz Kwiecinski |
| PvSTATEM | Reading, Quality Control and Preprocessing of MBA (MultiplexBead Assay) Data | Tymoteusz Kwiecinski |
| huge | High-Dimensional Undirected Graph Estimation | Haoming Jiang |
| klaR | Classification and Visualization | Uwe Ligges |
| datawizard | Easy Data Wrangling and Statistical Transformations | Etienne Bacher |
| dplyr | A Grammar of Data Manipulation | Hadley Wickham |
| pagoda2 | Single Cell Analysis and Differential Expression | Evan Biederstedt |
| ggplot2 | Create Elegant Data Visualisations Using the Grammar of Graphics | Thomas Lin Pedersen |
| biclust | BiCluster Algorithms | Sebastian Kaiser |
| tidyr | Tidy Messy Data | Hadley Wickham |
| tibble | Simple Data Frames | Kirill Müller |
| prospectr | Miscellaneous Functions for Processing and Sample Selection ofSpectroscopic Data | Leonardo Ramirez-Lopez |
| microeco | Microbial Community Ecology Data Analysis | Chi Liu |
| pammtools | Piece-Wise Exponential Additive Mixed Modeling Tools forSurvival Analysis | Andreas Bender |
| ebal | Entropy Reweighting to Create Balanced Samples | Jens Hainmueller |
| ordinalRR | Analysis of Repeatability and Reproducibility Studies withOrdinal Measurements | Ken Ryan |
| ff | Memory-Efficient Storage of Large Data on Disk and Fast AccessFunctions | Jens Oehlschlägel |
| mlr3data | Collection of Machine Learning Data Sets for 'mlr3' | Marc Becker |
| lubridate | Make Dealing with Dates a Little Easier | Vitalie Spinu |
| daltoolbox | Leveraging Experiment Lines to Data Analytics | Eduardo Ogasawara |
| mlrCPO | Composable Preprocessing Operators and Pipelines for MachineLearning | Martin Binder |
| readr | Read Rectangular Text Data | Jennifer Bryan |
| CRMetrics | Cell Ranger Output Filtering and Metrics Visualization | Rasmus Rydbirk |
| JointAI | Joint Analysis and Imputation of Incomplete Data | Nicole S. Erler |
| HiClimR | Hierarchical Climate Regionalization | Hamada S. Badr |
| gcxgclab | GCxGC Preprocessing and Analysis | Stephanie Gamble |
| simulariatools | Simularia Tools for the Analysis of Air Pollution Data | Giuseppe Carlino |
| powerjoin | Extensions of 'dplyr' and 'fuzzyjoin' Join Functions | Antoine Fabri |

Table 4: Example packages for transforming data with API

| package | title | maintainer |
|---|---|---|
| yaml | Methods to Convert R Data to YAML and Back | Shawn Garbett |
| geojsonio | Convert Data from and to 'GeoJSON' or 'TopoJSON' | Michael Mahoney |
| jsonlite | A Simple and Robust JSON Parser and Generator for R | Jeroen Ooms |
| reticulate | Interface to 'Python' | Tomasz Kalinowski |
| keyToEnglish | Convert Data to Memorable Phrases | Max Candocia |
| qtl2convert | Convert Data among QTL Mapping Packages | Karl W Broman |
| gtools | Various R Programming Tools | Ben Bolker |
| rmarkdown | Dynamic Documents for R | Yihui Xie |
| interleave | Converts Tabular Data to Interleaved Vectors | David Cooley |
| do | Data Operator | Jing Zhang |
| rio | A Swiss-Army Knife for Data I/O | Chung-hong Chan |
| data.tree | General Purpose Hierarchical Data Structure | Christoph Glur |
| wktmo | Converting Weekly Data to Monthly Data | You Li |
| GDPuc | Easily Convert GDP Data | Johannes Koch |
| nuts | Convert European Regional Data | Moritz Hennicke |
| wearables | Tools to Read and Convert Wearables Data | Peter de Looff |
| xml2relational | Converting XML Documents into Relational Data Models | Joachim Zuckarelli |
| TidyMultiqc | Converts 'MultiQC' Reports into Tidy Data Frames | Michael Milton |
| odk | Convert 'ODK' or 'XLSForm' to 'SPSS' Data Frame | Muntashir-Al-Arefin |
| spbabel | Convert Spatial Data Using Tidy Tables | Michael D. Sumner |
| exp2flux | Convert Gene EXPression Data to FBA FLUXes | Daniel Osorio |
| ecocomDP | Tools to Create, Use, and Convert ecocomDP Data | Colin Smith |
| tbl2xts | Convert Tibbles or Data Frames to Xts Easily | Nico Katzke |
| LAIr | Converting NDVI to LAI of Field, Proximal and Satellite Data | Francesco Chianucci |
| broom.mixed | Tidying Methods for Mixed Models | Ben Bolker |
| ILRCM | Convert Irregular Longitudinal Data to Regular Intervals andPerform Clustering | Atanu Bhattacharjee |
| intergraph | Coercion Routines for Network Data Objects | Michał Bojanowski |
| gtfs2gps | Converting Transport Data from GTFS Format to GPS-Like Records | Pedro R. Andrade |
| vcfR | Manipulate and Visualize VCF Data | Brian J. Knaus |
| RJSONIO | Serialize R Objects to JSON, JavaScript Object Notation | Yaoxiang Li |
| MissingHandle | Handles Missing Dates and Data and Converts into Weekly andMonthly from Daily | Mr. Sandip Garai |
| orsk | Converting Odds Ratio to Relative Risk in Cohort Studies withPartial Data Information | Zhu Wang |
| sjlabelled | Labelled Data Utility Functions | Daniel Lüdecke |
| dplyr | A Grammar of Data Manipulation | Hadley Wickham |
| tidytree | A Tidy Tool for Phylogenetic Tree Data Manipulation | Guangchuang Yu |
| pack | Convert Values to/from Raw Vectors | Joshua M. Ulrich |
| ggplot2 | Create Elegant Data Visualisations Using the Grammar of Graphics | Thomas Lin Pedersen |
| string2path | Rendering Font into 'data.frame' | Hiroaki Yutani |
| tdata | Prepare Your Time-Series Data for Further Analysis | Ramin Mojab |
| DDIwR | DDI with R | Adrian Dusa |
| CADF | Customer Analytics Data Formatting | Ludwig Steven |
| tidyr | Tidy Messy Data | Hadley Wickham |
| tibble | Simple Data Frames | Kirill Müller |

Continued on next page

Table 4: Example packages for transforming data with API (Continued)

| | | |
|---|---|---|
| redquack | Transfer 'REDCap' Data to Database | Dylan Pieper |
| tinytable | Simple and Configurable Tables in 'HTML', 'LaTeX', 'Markdown','Word', 'PNG', 'PDF', and 'Typst' Formats | Vincent Arel-Bundock |
| mergen | AI-Driven Code Generation, Explanation and Execution for DataAnalysis | Altuna Akalin |
| unpivotr | Unpivot Complex and Irregular Data Layouts | Duncan Garmonsway |
| jsonld | JSON for Linking Data | Jeroen Ooms |
| mltools | Machine Learning Tools | Ben Gorman |
| mergenstudio | 'Mergen' Studio: An 'RStudio' Addin Wrapper for the 'Mergen'Package | Jacqueline Jansen |