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

This article focuses on a preliminary step in any retrospective (or "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 The Issue of Wrangling in 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 generality: they employ large quantities of data drawn from different sources (Cheng et al. 2024; Kizilova et al. 2024; Ruggles, Cleveland, and Sobek 2024; Wysmułek, Tomescu-Dubrow, and Kwak 2022). However, significant challenges remain in ensuring the quality of harmonized datasets, especially when it comes to evaluating whether the data are fit for their intended use and systematically monitoring the quality of the raw inputs (Dubrow and Tomescu-Dubrow 2016; **?**).

The wrangling step plays a critical role in determining data quality for harmonization, especially given the growing volume and diversity of available sources and the need for transparent, well-documented cleaning (Kołczyńska 2022). The conventional approach usually and notoriously involves a great deal of manual work on indicator identification, data merging, data scaling, and so on (see, e.g., Lohr 2014). Manual wrangling undermines transparency and makes full reproducibility of the research pipeline more difficult to achieve (**?**). Worse, this tiresome task makes it easy to introduce errors in the data. Finally, even meticulous documentation cannot eliminate the influence of human discretion embedded in manual processing. These discretionary decisions often leave little trace, making it challenging for collaborators or reviewers to verify the wrangling process or diagnose sources of error.

In short, incomplete/inaccurate data entry, the absence of reproducibility, and untrackable human discretion in manual janitor work have collectively become major obstacles on the way to data harmonization, yet these challenge has received surprisingly little attention. In this article, we provide a practical routine (a “TAO”) that takes advantage of automatic programming and teamwork to reduce data-entry errors and improve the reproducibility and transparency of the wrangling process for researchers and reviewers. This TAO covers the three phases of data wrangling: data selection and collection, data entry, and what we term second-order opening—the transparent documentation of the data generation process (DGP). We illustrate how researchers can use this routine on administrative statistics and survey data with examples of 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 study:

1. To reduce the manual entry errors in order to improve the accuracy of both harmonized and analytic datasets;
2. To incorporate as much available data as possible to provide a solid foundation for constructing comparable data and increasing the generalizability of inferences;[1] 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.

We use two data harmonization projects, SWIID and DCPO, to illustrate this routine. SWIID is a long-running project that seeks to provide harmonized income inequality statistics for the broadest possible coverage of countries and years (Solt 2020). As of its most recent update at the time of this writing, its source data consists of more than 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 survey items that intend to capture the identical concepts but with different question wordings or option scales (see e.g., Caughey, O’Grady, and Warshaw 2019; Claassen 2019). Advancing this line of work, DCPO not only provides latent variable measurements but also automated and reproducible data collection (Solt 2020), which has been applied in a complete pipeline for a variety of topics such as gender egalitarianism (**?**), political interest (**?**), and support for gay rights (**?**), among other aspects of public opinion and open it freely for global researchers (see the data available at https://dcpo.org/).[2]

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

Large-scale data selection and cleaning is often viewed as tiresome, as something to be delegated to research assistants, to someone—indeed anyone—else. Performing these tasks manually makes it easy to make mistakes and errors. Haegemans, Snoeck, and Lemahieu (2019, 1) lists examples of misrouted financial transactions and airline flights. In a more systematic examination, Barchard and Pace (2011) found that RAs 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 are pernicious, undermining our results and more broadly our confidence in the scientific enterprise.

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

A deliberative set requires the members in a research team—whether several coauthors or a primary author with one or more RAs—to have a clear and coherent understanding of the research 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 early years of the SWIID program, for example, RAs were told that the goal of the project is to generate comparable statistics of country-level economic inequality. They were provided a list of sources to start with, mainly from national statistic bureaus, but also told that updated statistics for some countries may come from academic papers, published documents, and other sources, and encouraged to add each of these new sources by recording a valid link.

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 (**?**). This process begins with a shared conceptual foundation established through literature review and the corresponding pre-defined potential dimensions of 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. (For a more detailed checklist, see OSM A.)

Before data selection begins, team members undergo hands-on training on how the method works and what types of data and metadata they need to collect, such as data format and weighting types, that are essential for enabling the automated data preparation process.

Following the initial round of item selection and collection, the dual-entry section begins. 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 data-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 may 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 the target variable (**?**).

Therefore, multiple lab meetings are held throughout the data selection phase to share insights from each member's coding work and ensure conceptual alignment across the team. The process concludes with a final cross-check of the selected items by all members.

In addition to reducing manual biases, teamwork also helps expand the data pool. Both SWIID and DCPO projects enrolled team members from outside the United States. These members draw on their linguistic and cultural expertise to detect extra sources in non-English languages and improve the precision of the data selection. To some extent, data from different sources helps mitigate the cultural biases embedded in survey instruments developed within particular national or linguistic contexts.

2.2 Step 2: Data Entry Automation

Putting the data into a format ready to merge is arguably the step most prone to manual errors and controversies. The best solution is to automate the entry process, taking advantage of scripting and any application programming interfaces (APIs) provided by the data source.

In the DCPO case, nce the research team manually codes the question numbers of relevant items, response scales, survey weights, project names, fieldwork waves, and dates from the raw survey files and accompanying documentation, data entry can be 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, applies consistent recoding and weighting procedures, and outputs structured country-year aggregates. To enable automation, raw data should be pre-processed to conform to `DCPOtools`' input format, despite inconsistencies in the original sources.[3]

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 (**?**). `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.

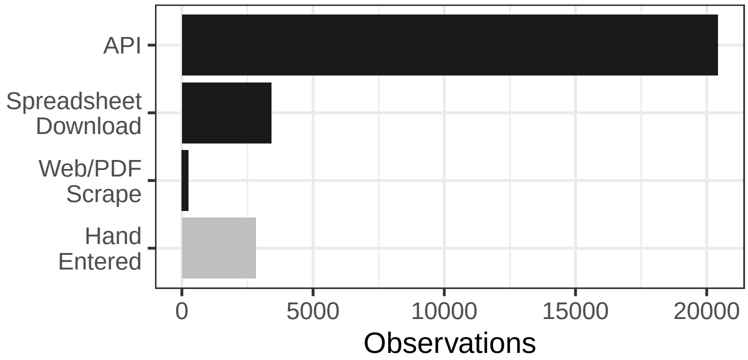

Figure 1: Income Inequality Observations by Method of Collection

While coding datasets and items into structured spreadsheets facilitates automation, an even better approach starts the automation from the data selection step via scripting and APIs. As shown in Figure 1, the current version of SWIID collects 76% of the observations through APIs. In OSM B, we provide an illustrative list of R packages that can assist with tasks such as collecting data via APIs and cleaning and transforming data. The list is long but far from complete. Readers are welcome to modify the arguments of the codes (such as the keywords used to select packages) in this article's replication file that we use to create the list, they will discover many times more packages that already exist to help collect and wrangle data.

Returning to the case of the SWIID: when no API is available, the automation script downloads and reads any available spreadsheets. In the absence of a spreadsheet, the process of scraping the data, either directly from the web or, preferably, from a pdf file, is automated (see Sepulveda 2024). Together the collection of 90% of the source data is scripted.

This process substantially reduces manual entry errors, though some risks may remain, particularly when scraping from unstructured PDFs. Nevertheless, it enables efficient integration of updates and revisions whenever data collection program is reran.

Even for data sources that have to be entered in hand, such as those from academic articles or books, there is still opportunity for partial automation. For the remaining 10% of the SWIID observations, for instance, many were collected using Sepulveda's `tabulapdf` R package to avoid data-entry errors. Optical Character Recognition (OCR) can be used to extend this method to even hard-copy data sources.

For data that one has to enter manually, the teamwork is crucial. Mirroring the approach for data selection described above, each hand-entered observation was independently entered twice into two separate spreadsheets. 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).

As the final point of Step 2, the "data entry automation" step indeed minimizes the errors by manual entry but *not* all mistakes or biases. Its effectiveness depends on the proper implementation of programming and software, both of which have inherent limitations. For example, OCR may misread characters or digits, and software bugs can affect quality and reproducibility of the process. Human supervision and validation therefore remain crucial. Researchers are encouraged to pilot the automation on a small, diverse sample or conduct spot checks to verify the quality of automated data entry.

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). This practice reflects the broader goals of open science. Nevertheless, issue of researcher degrees of freedom indicates that current standards of openness remain insufficient (see **?**). In the context of data harmonization, we advocate for what we call a "second-order" opening. The "first-order" openness refers to sharing code and outputs from the analysis stage, and second-order openness involves making transparent the data generation process (DGP)—including data collection, cleaning, wrangling, and harmonization decisions.

Empirical evidence has indicated the severe consequences of neglecting second-order opening. Recent research has found that the variation in the estimated effects caused by researchers may outweigh the population's variation (**?**). Within this researcher-choice variation, a substantial portion comes from the data-wrangling process (**?**). In a "many-analyst" analysis, (**?**) requested 146 research teams to complete the same research task, in which Group A decided how to accomplish the task all by themselves, Group B was given a specified research design, and Group C was given the same research design and a pre-cleaned data set. Group B, which used the same design but processed the data themselves, produced the highest variation in results. In contrast, Group C produced the most consistent results. These findings underscore the importance of openness about data preparation and processing decisions ("second-order" openness).

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, and thereby ensure that the full pipeline is documented and reproducible.[4]

With developed scientific and technical publishing system, such as Quarto or R Markdown, version control platforms like Github, and open collaboration platforms including the Open Science Framework, 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 trace a research project from the start in a Github repo and every wave of data update in the corresponding OSF project (see, for example, Tai, Hu, and Solt 2024).

3 Discussion

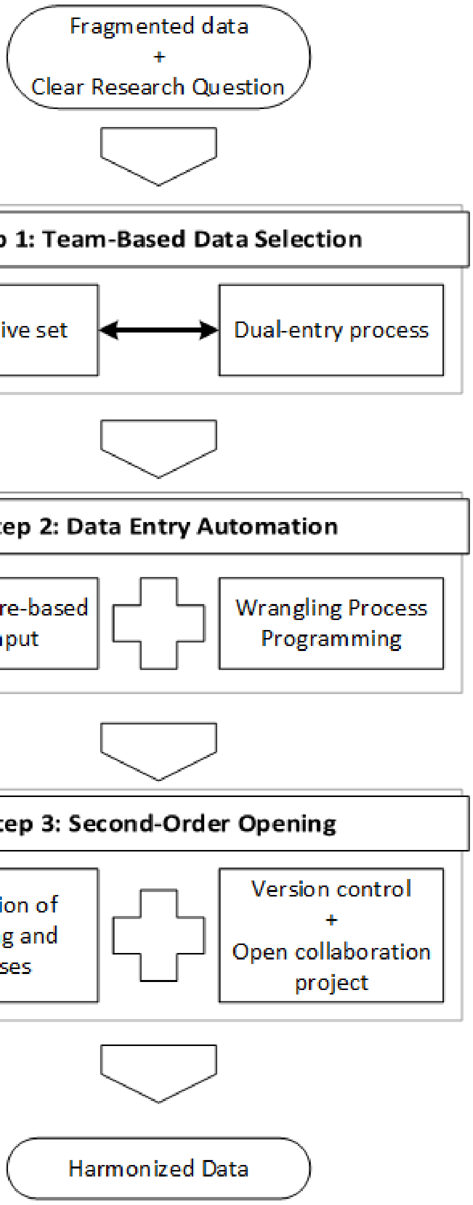
Fragmented data
+
Clear Research Question
Step 1: Team-Based Data Selection
Deliberative set
Dual-entry process
Step 2: Data Entry Automation
API/Software-based data input
Wrangling Process Programming
Step 3: Second-Order Opening
Replication of wrangling and analyses
Version control
+
Open collaboration project
Harmonized Data

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

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), which easily justifies the additional investment. Teamwork fosters conceptual alignment and construct refinement through collaborative discussion while also distributing tasks to reduce fatigue-related errors.

Social scientists now benefit from standardized harmonization workflows (?) and automated data processing (?). 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 (?).

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

For retrospective (or called "ex-post") harmonization projects, careful attention to pre-harmonization stages substantially contributes to overall data 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.

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. https://CRAN.R-project.org/package=countrycode.

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. https://www.sciencedirect.com/science/article/pii/S0747563211000707.

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. http://dx.doi.org/10.17016/FEDS.2015.083 (November 17, 2018).

Cheng, Cindy, Luca Messerschmidt, Isaac Bravo, Marco Waldbauer, Rohan Bhavikatti, Caress Schenk, Vanja Grujic, et al. 2024. “A General Primer for Data Harmonization.” *Scientific Data* 11(1): 152. doi:10.1038/s41597-024-02956-3.

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

Dubrow, Joshua Kjerulf, and Irina Tomescu-Dubrow. 2016. “The Rise of Cross-National Survey Data Harmonization in the Social Sciences: Emergence of an Interdisciplinary Methodological Field.” *Quality & Quantity* 50(4): 1449–67. doi:10.1007/s11135-015-0215-z.

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. https://www.sciencedirect.com/science/article/pii/S0378720617309801.

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. http://cps.sagepub.com/content/early/2016/01/11/0010414015621077.abstract.

Kizilova, Kseniya, Jaime Diez-Medrano, Christian Welzel, and Christian Haerpfer. 2024. "Harmonization in the World Values Survey." In *Survey Data Harmonization in the Social Sciences*, John Wiley & Sons, Ltd, 39–56. doi:10.1002/9781119712206.ch3.

Kołczyńska, Marta. 2022. "Combining Multiple Survey Sources: A Reproducible Workflow and Toolbox for Survey Data Harmonization." *Methodological Innovations* 15(1): 62–72. doi:10.1177/20597991221077923.

Lohr, Steve. 2014. "For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights." *The New York Times: Technology.* https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html (June 25, 2023).

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

Oser, Jennifer, and Barak Zur. 2025. "Guide for the (Soon-to-be) Perplexed: Considerations for Scholars Launching and Leading Survey Data Harmonization Journeys." *PS: Political Science & Politics* 0(0): Online.

Ruggles, Steven, Lara Cleveland, and Matthew Sobek. 2024. “Harmonization of Census Data: IPUMS – International.” In *Survey Data Harmonization in the Social Sciences*, John Wiley & Sons, Ltd, 207–26. doi:10.1002/9781119712206.ch12.

Sepulveda, Mauricio Vargas. 2024. “tabulapdf: Extract Tables from PDF Documents.” https://github.com/ropensci/tabulapdf.

Solt, Frederick. 2020. “Modeling Dynamic Comparative Public Opinion.” SocArXiv. https://doi.org/10.31235/osf.io/d5n9p (February 21, 2022).

Solt, Frederick, Yue Hu, Kevan Hudson, Jungmin Song, and Dong ”Erico” Yu. 2016. “Economic Inequality and Belief in Meritocracy in the United States.” *Research & Politics* 3(4, 4): 1–7. doi:10.1177/2053168016672101.

Solt, Frederick, Yue Hu, Kevan Hudson, Jungmin Song, and Dong ”Erico” Yu. 2017. “Economic Inequality and Class Consciousness.” *The Journal of Politics* 79(3, 3): 1079–83. doi:10.1086/690971.

Solt, Frederick, Yue Hu, and Yuehong Tai. 2018. “DCPOtools: Tools for Dynamic Comparative Public Opinion.” https://github.com/fsolt/DCPOtools (February 18, 2022).

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.

Wysmułek, Ilona, Irina Tomescu-Dubrow, and Joonghyun Kwak. 2022. “Ex-Post Harmonization of Cross-National Survey Data: Advances in Methodological and Substantive Inquiries.” *Quality & Quantity* 56(3): 1701–8. doi:10.1007/s11135-021-01187-7.

Table A.1: Checklist with Decision Rationale

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

Online Supplementary Materials

A Checklist for Deliberation Process

Table A.1 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 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.

B R packages for data wrangling

Here are exemplary R packages that researchers can use to collect, clean, and transform data. The following tables were generated by the `pkgsearch::pkg_search()` function with the keywords relating to data downloading, wrangling, and transforming. 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 explore 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 A.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 'CoinMarketCap' 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 Data Downloading and Importing	John Elizarraras
wdi2	Download World Development Indicators from the World Bank Indicators API	Christoph Scheuch
GDELTtools	Download, Slice, and Normalize GDELT V1 Event and Sentiment API Data	Stephen R. Haptonstahl
neonUtilities	Utilities for Working with NEON Data	Claire Lunch

Continued on next page

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

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
zen4R	Interface to 'Zenodo' REST API	Emmanuel Blondel
rstudioapi	Safely Access the RStudio API	Kevin Ushey
rdhs	API Client and Dataset Management for the Demographic and Health Survey (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 Bank API	Mauricio Vargas Sepulveda
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
riingo	An R Interface to the 'Tiingo' Stock Price API	Davis Vaughan
I14Y	Search and Get Data from the I14Y Interoperability Platform of Switzerland	Felix Luginbuhl
jsonlite	A Simple and Robust JSON Parser and Generator for R	Jeroen Ooms
kaigiroku	Programmatic Access to the API for Japanese Diet Proceedings	Akitaka Matsuo

Continued on next page

Table A.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 and Inequality Platform (PIP)' APIs	Maximilian Mücke
BFS	Get Data from the Swiss Federal Statistical Office	Felix Luginbuhl
rinat	Access 'iNaturalist' Data Through APIs	Stéphane Guillou
yfinancer	'Yahoo Finance' API Wrapper	Giovanni Colitti
opendotaR	Interface for OpenDota API	Kari Gunnarsson
PurpleAir	Query the 'PurpleAir' Application Programming Interface	Cole Brokamp
Visualize.CRAN.Downloads	Visualize Downloads from 'CRAN' Packages	Marcelo Ponce
PurpleAirAPI	Historical Data Retrieval from 'PurpleAir' Sensors via API	Heba Abdelrazzak
trud	Query the 'NHS TRUD API'	Alasdair Warwick
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 Scorecard Data	Benjamin Skinner
inegiR	Integrate INEGI's (Mexican Stats Office) API with R	Eduardo Flores
naptanr	Call the 'NaPTAN' API Through R	Francesca Bryden

Table A.3: Example packages for cleaning data with API

package	title	maintainer
discretization	Data Preprocessing, Discretization for Classification	HyunJi Kim
helda	Preprocess Data and Get Better Insights from Machine Learning Models	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
RobLoxBioC	Infinitesimally Robust Estimators for Preprocessing - Omics Data	Matthias Kohl
shinyrecipes	Gadget to Use the Data Preprocessing 'recipes' Package Interactively	Alberto Almuiña
RGCxGC	Preprocessing and Multivariate Analysis of Bidimensional Gas Chromatography Data	Cristian Quiroz-Moreno

Continued on next page

Table A.3: Example packages for cleaning data with API (Continued)

EEM	Read and Preprocess Fluorescence Excitation-Emission Matrix (EEM) Data	Vipavee Trivittayasil
cobalt	Covariate Balance Tables and Plots	Noah Greifer
clickR	Semi-Automatic Preprocessing of Messy Data with Change Tracking for Dataset Cleaning	David Hervas Marin
SerolyzeR	Reading, Quality Control and Preprocessing of MBA (Multiplex Bead Assay) Data	Jakub Grzywaczewski
PvSTATEM	Reading, Quality Control and Preprocessing of MBA (Multiplex Bead 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
prospectr	Miscellaneous Functions for Processing and Sample Selection of Spectroscopic Data	Leonardo Ramirez-Lopez
tidyr	Tidy Messy Data	Hadley Wickham
tibble	Simple Data Frames	Kirill Müller
biclust	BiCluster Algorithms	Sebastian Kaiser
pammtools	Piece-Wise Exponential Additive Mixed Modeling Tools for Survival Analysis	Andreas Bender
ebal	Entropy Reweighting to Create Balanced Samples	Jens Hainmueller
ordinalRR	Analysis of Repeatability and Reproducibility Studies with Ordinal Measurements	Ken Ryan
ff	Memory-Efficient Storage of Large Data on Disk and Fast Access Functions	Jens Oehlschlägel

Continued on next page

Table A.3: Example packages for cleaning data with API (Continued)

mlr3data	Collection of Machine Learning Data Sets for 'mlr3'	Marc Becker
lubridate	Make Dealing with Dates a Little Easier	Vitalie Spinu
mlrCPO	Composable Preprocessing Operators and Pipelines for Machine Learning	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
tosca	Tools for Statistical Content Analysis	Lars Koppers
preputils	Utilities for Preparation of Data Analysis	Josef Frank

Table A.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
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

Continued on next page

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

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
broom.mixed	Tidying Methods for Mixed Models	Ben Bolker
LAIr	Converting NDVI to LAI of Field, Proximal and Satellite Data	Francesco Chianucci
snirh.lab	Convert Laboratory Water-Quality Data to ’SNIRH’ Import Format	Luís Pereira
ILRCM	Convert Irregular Longitudinal Data to Regular Intervals and Perform Clustering	Atanu Bhattacharjee
intergraph	Coercion Routines for Network Data Objects	Michał Bojanowski
vcfR	Manipulate and Visualize VCF Data	Brian J. Knaus
gtfs2gps	Converting Transport Data from GTFS Format to GPS-Like Records	Pedro R. Andrade

Continued on next page

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

MissingHandle	Handles Missing Dates and Data and Converts into Weekly and Monthly from Daily	Mr. Sandip Garai
RJSONIO	Serialize R Objects to JSON, JavaScript Object Notation	Yaoxiang Li
sjlabelled	Labelled Data Utility Functions	Daniel Lüdecke
orsk	Converting Odds Ratio to Relative Risk in Cohort Studies with Partial Data Information	Zhu Wang
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
mergen	AI-Driven Code Generation, Explanation and Execution for Data Analysis	Altuna Akalin
unpivotr	Unpivot Complex and Irregular Data Layouts	Duncan Garmonsway
yyjsonr	Fast 'JSON', 'NDJSON' and 'GeoJSON' Parser and Generator	Mike Cheng
jsonld	JSON for Linking Data	Jeroen Ooms
mergenstudio	'Mergen' Studio: An 'RStudio' Addin Wrapper for the 'Mergen' Package	Jacqueline Jansen
play	Visualize Sports Data	Joe Chelladurai

Continued on next page

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

ctypesio	Read and Write Standard 'C' Types from Files, Connections and Raw Vectors	Mike Cheng