Journal of Data Science 0 (0), 1–19
??? 2026
DOI: 10.6339/26-JDS1248
Education in Data Science

The Data Extraction Plan: A Framework for Structured Decision-Making for Data Acquisition

Immanuel Williams[1,*]

[1]*Department of Statistics, California Polytechnic State University, San Luis Obispo, CA USA*

Abstract

Data acquisition from pre-existing secondary sources is a critical yet under-documented phase of applied data science. While frameworks exist for documenting finished datasets, teaching extraction techniques, and managing the broader data science lifecycle, no formal framework structures the planning and documentation decisions that guide secondary data acquisition. We introduce the Data Extraction Plan (DEP), a five-step framework for documenting decisions made before and during acquisition of pre-existing data: Purpose & Source, Legal & Ethical Review, Access & Technical Plan, Scope & Sampling, and Validation & Contingency. Through three illustrative case studies—Basketball Reference player statistics, Wikipedia population table scraping, and Spotify application programming interface (API) audio feature extraction—we demonstrate how the DEP captures decisions that are typically invisible in traditional workflows. We argue that structured extraction planning is increasingly essential in an era when artificial intelligence (AI)-assisted code generation makes writing extraction scripts trivial while the planning decisions upstream remain consequential. By elevating secondary data acquisition and extraction from an invisible prerequisite to a first-class methodological phase, the DEP provides a necessary foundation for trustworthy data cleaning, curation, and integration.

Keywords *API data; data science workflow; preregistration; reproducibility; secondary data; web scraping*

1 Introduction

Applied data science increasingly begins not with the collection of new data but with the acquisition of data that already exist: public websites, application programming interfaces (APIs), archives, repositories, and institutional systems. Lifecycle accounts of data science place acquisition at the head of the pipeline, where it conditions every later stage (Wirth and Hipp, 2000; Stodden, 2020), and the data acumen expected of graduates now includes the ability to obtain and manage data from diverse and often unstructured sources (Nolan and Temple Lang, 2010; National Academies of Sciences, Engineering, and Medicine, 2018). The decisions made at this stage are consequential. Which source is used, which records and fields are retrieved, what temporal and population boundaries are imposed, and under what legal terms the data are obtained together make up what we will call the provenance of the resulting dataset, extending the sense in which Buneman et al. (2001) use the term: the description of where a piece of data came from and the process by which it arrived, which they identify as essential to anyone concerned with the accuracy and currency of the data. What can legitimately be inferred from a dataset

* Email: imwillia@calpoly.edu.

© 2026 The Author(s). Published by the School of Statistics and the Center for Applied Statistics, Renmin University of China. Open access article under the CC BY license.
Received February 28, 2026; Accepted October 25, 2026

depends, in turn, on that record. The substantial effort that data wrangling and cleaning demand (Kandel et al., 2011; Chu et al., 2016) is spent on whatever the acquisition step delivers, so upstream choices constrain everything that follows, and reproducibility guidelines expect the path from source to result to be documented end to end (Sandve et al., 2013; Horton et al., 2022). Every dataset therefore has an origin story: before any statistical model is fit, before any visualization is rendered, before any data cleaning pipeline is executed, someone had to decide whether the needed data already existed, where they could be obtained, under what constraints, and through what mechanism. Yet in both pedagogy and practice, secondary data acquisition is often treated as a mechanical prerequisite rather than a decision-rich phase deserving its own documentation.

Some terminology is needed to make this argument precise, because the literature uses "collection," "acquisition," and "extraction" loosely. In this paper, data collection refers to generating new primary data from the world or from research participants, such as through surveys, experiments, interviews, or sensors. Data acquisition refers to obtaining data that already exist, often from secondary sources such as public websites, APIs, archives, repositories, or institutional systems. Data extraction refers to the technical mechanism by which acquisition occurs: the scrape, API call, download, database query, or manual retrieval that moves data from a source into an analyst's working environment. Acquisition and extraction decisions shape everything downstream—the variables available for analysis, the populations represented, the legal boundaries of use, and the reproducibility of the entire project—and it is these decisions, for pre-existing secondary data, that this paper addresses.

Existing frameworks and practices address parts of this territory, but each stops short of the acquisition decisions themselves. Dominant lifecycle models acknowledge the stage without structuring it. The "Data Understanding" phase in Wirth and Hipp (2000) recognizes that analysts must familiarize themselves with the data but provides no template for documenting how secondary data will be obtained or what decisions that process entails. The Data Science Life Cycle of Stodden (2020) traces a dataset from acquisition through cleaning, use, and publication, but treats acquisition as a single node with no internal structure. The FAIR principles (Wilkinson et al., 2016) establish standards for published data—Findable, Accessible, Interoperable, Reusable—but operate after data already exist in a shareable form. A parallel body of work documents datasets and models rather than their acquisition. Datasheets for Datasets (Gebru et al., 2021) provide a comprehensive documentation framework for finished datasets, addressing motivation, composition, collection process, and recommended uses; Hutchinson et al. (2021) draw on software engineering practices to propose accountability frameworks for dataset development; Data Cards (Pushkarna et al., 2022) summarize essential facts about machine learning (ML) datasets for responsible AI development; and Model Cards (Mitchell et al., 2019) document trained models. Pedagogical and workflow contributions come closer to the technical act of extraction. Dogucu and Çetinkaya-Rundel (2021) provide an excellent treatment of web scraping in the statistics and data science curriculum, Stoudt et al. (2024) offer ten rules for responsible data science workflows, and Sandve et al. (2013) set out rules for reproducible computational research. Empirical studies of data science practice similarly tend to begin once data are already available to the analyst: Quaranta et al. (2022), in their study of collaborative best practices across 1,380 Jupyter notebooks, focus on activities within notebooks—version control, documentation, code quality, and re-executability—rather than on the secondary-acquisition decisions that determine how the data entered those notebooks. This is not a flaw in that work; it marks the boundary of a parallel concern.

The specific gap that remains is the absence of any structured documentation for planning

and tracking secondary data acquisition: the space between "I know what secondary data I need" and "I wrote code to get it." The gap has widened as the technical landscape has shifted. The proliferation of public APIs, scrapable websites, and open data portals has expanded the sources available to data scientists, and tools like the R packages rvest (Wickham, 2024) and spotifyr (Thompson et al., 2024), Python libraries like Beautiful Soup (Richardson, 2026) and pandas (The pandas development team, 2020; McKinney, 2010), and AI-assisted coding tools such as ChatGPT and GitHub Copilot have made writing extraction code faster than ever (Chen et al., 2021). Speed of execution has outpaced quality of planning: a student can generate a web scraping script in thirty seconds using an AI assistant without ever considering rate limits, Terms of Service compliance, copyright implications, or what to do when the source changes. The decisions are still being made, but they are made ad hoc, live only in the analyst's head or in scattered code comments, and are lost when the dataset is shared with collaborators, in a publication, or in a classroom.

This paper introduces the Data Extraction Plan (DEP) as a response to that gap: a five-step framework for structuring and documenting the decisions that guide secondary data acquisition. The DEP is completed before systematic extraction begins—not before any interaction with the source, but before committing to full-scale acquisition of pre-existing data. It also records decisions that necessarily arise during acquisition, including discrepancies between documentation and source behavior, source changes, validation failures, or justified deviations from the original plan. Its five steps follow a deliberate logical sequence: what and where (Purpose & Source), may I (Legal & Ethical Review), how (Access & Technical Plan), how much (Scope & Sampling), and what if (Validation & Contingency). The artifact is the planning document itself—not an Extract, Transform, and Load (ETL) pipeline, not a software package, not a set of extraction scripts, but a structured record of the decisions that make those downstream artifacts meaningful. The DEP draws its rhetorical backbone from the preregistration movement in experimental science. Nosek et al. (2018) argued that documenting research intentions before observing results strengthens the credibility of empirical findings by constraining undisclosed analytic flexibility. The DEP makes the analogous argument for secondary data acquisition: documenting initial extraction decisions before full-scale execution, and recording later changes, constrains undisclosed data selection flexibility, strengthens provenance, and creates an auditable record of how pre-existing data came to be used.

Primary data collection raises its own planning, ethical, and documentation questions. The logic of the DEP—documenting purpose, permissions, access mechanisms, scope, validation, and contingency before data work begins—could extend naturally to surveys, experiments, interviews, sensors, and other forms of primary data collection. The present paper is deliberately bounded to secondary data sources because the three case studies and the motivating technical pressures concern data that already exist. A primary-collection version of the DEP is therefore a natural and potentially high-impact extension rather than a claim made here.

The remainder of this paper is organized as follows. Section 2 makes the case for structured secondary data extraction planning by examining real failures traceable to absent planning. Section 3 presents the five-step DEP in detail and positions it against adjacent frameworks. Section 4 demonstrates the framework through three case studies of varying complexity. Section 5 discusses when and how the DEP should be used, including scaling guidance and future directions. Section 6 highlights the importance of structured extraction planning for data integrity and reproducibility.

2 The Importance of Planning Extraction

2.1 Extraction Decisions Are Consequential

Secondary data extraction involves a series of decisions that are consequential, interconnected, and largely invisible in current practice. Which source to use? What variables to extract? What temporal and geographic boundaries to impose? How to handle rate limits? Whether Terms of Service permit the intended use? How to validate that extraction succeeded? Each of these decisions shapes the resulting dataset, yet they are rarely documented, almost never justified, and essentially impossible for a third party to audit.

Consider the chain of decisions in a seemingly straightforward secondary-acquisition task: obtaining National Basketball Association (NBA) player statistics from a sports statistics website. The analyst must choose a source (Basketball Reference vs. the NBA's official statistics portal), determine an access method (web scraping vs. API), review legal constraints (Terms of Service, robots.txt directives), decide which statistics to extract (per-game vs. advanced metrics), define temporal scope (which seasons), establish population boundaries (all players vs. a minimum games threshold), plan for technical constraints (crawl delays, pagination), determine storage format, and define validation criteria. Each decision has defensible alternatives, and each shapes the dataset in ways that may be invisible to downstream analysts.

Some acquisition decisions are currently made ad hoc because real sources behave unpredictably: an API returns different fields than expected, a page structure changes, a rate limit appears, or a planned field is unavailable. The problem is not that every ad hoc decision exists; the problem is that these decisions often live only in the analyst's head or in scattered code comments, rarely in a dedicated planning document. When a dataset is shared—with collaborators, in a publication, or in a classroom—the extraction decisions that produced it are lost.

2.2 Real Failures Trace to Absent Planning

The consequences of unplanned extraction are not hypothetical. In October 2020, nearly 16,000 COVID-19 cases went unreported in England because Public Health England's developers had chosen the legacy Excel XLS format for the templates that consolidated laboratory results; each template could hold only about 65,000 rows, and once that limit was reached further cases were simply left off (Kelion, 2020). Horton et al. (2022) cite the episode as an illustration of why robust workflows matter. A pre-extraction plan that estimated expected data volume would have identified this constraint before it corrupted public health reporting.

The Spotify Web API provides a more extended case study. In 2021, the API offered relatively open access to audio features—numerical descriptors like valence, danceability, and energy for any track in Spotify's catalog. Thousands of students, researchers, and developers built projects around these features, many using Charlie Thompson's spotifyr R package (Thompson et al., 2024). By November 2024, Spotify deprecated the Audio Features and Audio Analysis endpoints for new applications. By March 2025, extended access required 250,000 monthly active users and a registered business. By February 2026, Developer Mode required a Premium subscription and limited testing to five users. Developers who had no contingency plan lost months of work. Those who had identified alternative sources—MusicBrainz, AcousticBrainz, Last.fm—could pivot.

Terms of Service violations represent another class of planning failures. Indeed's Terms of Service, for example, prohibit using any automated system to access or data-mine the site without the company's express written permission, allowing only the crawling permitted by

its robots.txt file (Indeed, 2026). Pedagogical treatments of web scraping make the same point from the classroom side: Dogucu and Çetinkaya-Rundel (2021) teach students to check whether a server permits bots before scraping and stress that having the technical skill to scrape does not mean one is allowed to. Researchers who begin extraction without reviewing these terms risk not only the loss of acquired data but legal consequences. A plan that concluded "do not extract" before any code was written would have redirected effort toward legitimate alternatives.

2.3 AI-Assisted Coding Increases the Need for Planning

The rise of AI-assisted coding tools has made this problem more urgent, not less. Large language models trained on code can produce working programs from short natural-language prompts (Chen et al., 2021), so a student can prompt ChatGPT with "scrape all NBA player stats from Basketball Reference" and receive a functional Python script in seconds. In our experience, such a script includes no rate limiting, no consideration of robots.txt, no Terms of Service review, no validation checkpoints, and no contingency plan. This is consistent with systematic evaluations of AI coding assistants, which find that generated code frequently contains weaknesses and omissions that the prompt did not ask the tool to avoid (Pearce et al., 2022): the assistant delivers what was requested, and nothing that was not. The technical barrier to extraction has collapsed, but the planning barrier was never erected. The DEP therefore serves as a human-in-the-loop checkpoint for AI-generated extraction code. It is not anti-AI—indeed, AI tools can assist in drafting DEPs themselves—but it insists that planning precede full-scale execution and that any later deviations be documented regardless of who or what writes the code. In this respect the framework embodies what Boenig-Liptsin et al. (2022) call for in the Data Science Ethos Lifecycle: ethical reflection embedded at each stage, not appended as an afterthought.

2.4 The Preregistration Analogy

The closest precedent for the DEP is preregistration in experimental science. Nosek et al. (2018) argued that the credibility of empirical findings is undermined when researchers have undisclosed flexibility in their analytic choices—the ability to try multiple analyses and report only those that produce favorable results. Preregistration addresses this by requiring researchers to document their hypotheses, methods, and analysis plans before observing outcomes.

Secondary data acquisition involves an analogous form of undisclosed flexibility. Which records were included or excluded? Why this source and not that one? What temporal boundaries were imposed, and were they determined before or after seeing the available data? Without documentation, these decisions are indistinguishable from post-hoc rationalization. The DEP applies the preregistration principle to secondary data acquisition: document the initial decisions before execution and record justified changes as acquisition unfolds.

2.5 Adjacent Work and the Gap

Several frameworks and contributions approach this space without fully addressing it. CRISP-DM's Data Understanding phase (Wirth and Hipp, 2000) acknowledges the need to understand data before modeling but provides no structured template for planning secondary acquisition. The FAIR principles (Wilkinson et al., 2016) focus on the stewardship of data that already exists. Datasheets for Datasets (Gebru et al., 2021) document completed datasets and, in some contexts, primary collection processes; they do not provide a planning template for acquiring

secondary data before extraction begins. Dogucu and Çetinkaya-Rundel (2021) provide an excellent pedagogical treatment of web scraping techniques but focus on the how rather than the plan before the how. Stoudt et al. (2024) offer ten rules for responsible data science workflows, several of which implicate extraction planning, but do not provide a template for what such a plan should contain. Horton et al. (2022) emphasize reproducibility in the curriculum and document the consequences of planning failures, but no secondary data acquisition planning document is proposed.

The DEP fills the space between "I know what secondary data I need" and "I wrote code to get it." It is complementary to Datasheets, operational beneath FAIR, more specific than CRISP-DM, and the secondary data-acquisition instantiation of the Q1Q2Q3 workflow's insistence that qualitative context must precede quantitative work (Vance et al., 2026).

3 The Data Extraction Plan Framework

The DEP is a structured planning document initiated before secondary data extraction begins and maintained as acquisition proceeds. It consists of five steps, each representing a category of decisions that must be documented and justified. The steps follow a deliberate logical progression: identifying what is needed and where it lives, determining whether extraction is legally and ethically permissible, specifying how data will be technically obtained, defining the scope and boundaries of extraction, and establishing how success will be verified and failure addressed. The guiding philosophy is disciplined documentation for data acquisition: initial decisions should be made deliberately, and unavoidable changes should be recorded transparently. A critical design principle follows from this philosophy: the DEP is a document, not a tool. It is language-agnostic, platform-agnostic, and source-agnostic, so the same framework applies whether the analyst uses R or Python, whether the source is an API or a static website, and whether the project is a classroom assignment or a multi-year research study. The artifact is structured thinking, recorded in writing, and the five steps that follow define what that record must contain.

3.1 Step 1: Purpose & Source

The first step asks whether the extraction is necessary at all and whether the chosen source is the right one. It requires the analyst to state the specific research or analytic question the extraction serves and to justify why extraction is necessary. This mirrors the Q1 phase of the Q1Q2Q3 workflow (Vance et al., 2026), where qualitative context must precede quantitative work. The extraction must be justified by the question—not the other way around.

The analyst identifies the specific source or sources, documents why this source was selected over alternatives, and assesses what is known about the source's provenance and reliability. Has the source changed access policies recently? Is it a commercial platform whose incentives may shift? Is the source likely to persist through the project timeline? The Spotify API's progressive restriction from relatively open access in 2021 to requiring 250,000 monthly active users and a registered business in 2025 illustrates how quickly a source can become inaccessible.

Critically, this step also asks whether a suitable dataset already exists. A public repository, a prior extraction by another researcher, or a government data release may already contain the needed data. Extraction should be justified only when pre-existing data are insufficient—a principle consistent with FAIR's emphasis on reusing data before creating new copies (Wilkinson et al., 2016).

3.2 Step 2: Legal & Ethical Review

The second step asks whether the analyst is permitted to extract the data and, separately, whether the analyst should. It is the only step with veto power—it can terminate the entire plan. A DEP that concludes "do not extract" at this step is itself a valid and valuable outcome.

For APIs, the analyst must read and document the relevant Developer Terms of Service, acceptable use policies, and redistribution clauses. For websites, any posted Terms of Use must be reviewed. Traditionally, these agreements are accepted without reading; the DEP requires they be read, summarized, and cited. For web scraping, the analyst must document what the site's robots.txt file permits or prohibits, as ignoring these signals is ethically problematic and may be legally actionable under statutes like the Computer Fraud and Abuse Act (CFAA).

Copyright and intellectual property must be assessed: does extracting and storing this content create a derivative work? Is the data factual (generally not copyrightable) or expressive (protected)? If the data involves people, the analyst must determine whether Institutional Review Board (IRB) review is required and whether personally identifiable information (PII) will be acquired. When PII is present, the applicable legal framework must be identified. The Family Educational Rights and Privacy Act (FERPA) governs student education records, the Health Insurance Portability and Accountability Act (HIPAA) protects individual health information, and the General Data Protection Regulation (GDPR) applies to personal data of European Union residents regardless of where the data processor is located. State-level privacy laws may add another layer of complexity; for example, the California Consumer Privacy Act (CCPA) creates additional obligations for certain uses of California residents' personal information. Identifying which framework applies is not a formality: each imposes different requirements for consent, storage, de-identification, and breach notification, and these requirements must be understood before extraction begins, not discovered during analysis. This operationalizes the ethical reflection that Boenig-Liptsin et al. (2022) call for at each stage of the data science lifecycle.

3.3 Step 3: Access & Technical Plan

The third step asks how data will move from the source into the analyst's environment and under what constraints. It specifies the technical mechanism of extraction: the analyst documents whether extraction will use an API, web scraping, direct file download, database query, or manual retrieval—recognizing that this list is not exhaustive and that some projects may require hybrid approaches, browser automation for JavaScript-rendered content, or specialized tools not covered here. Each method carries different technical requirements, failure modes, and ethical implications. Dogucu and Çetinkaya-Rundel (2021) document the technical skills involved in web scraping; the DEP documents the decisions that precede deploying those skills.

Authentication and cost are documented separately: the credential type, the access tier, and any cost implications. The plan should also record the possibility that pricing structures may change mid-project, as Reddit's 2023 API pricing shift and Spotify's progressive access restrictions demonstrate. Rate limits and throttling strategies are specified: the maximum number of calls per time window, the delay between requests for scraping, and the consequences of exceeding limits, such as a Hypertext Transfer Protocol (HTTP) 429 response or temporary blocking.

The expected data format—JavaScript Object Notation (JSON), HyperText Markup Language (HTML), comma-separated values (CSV), Extensible Markup Language (XML), or another format—and parsing approach are specified, including any known challenges such as nested

JSON structures requiring flattening, inconsistent HTML layouts across pages, or encoding issues in non-English text.

Finally, the storage plan is documented with sufficient detail that a third party could locate, interpret, and verify the extracted data. This includes the storage destination (local disk, cloud storage, institutional server, or database), the file format and rationale for choosing it (e.g., CSV for portability, Parquet for columnar efficiency with large datasets, R Data Serialization (RDS) or pickle for preserving language-specific data types, or a relational database for normalized multi-table storage), file and directory naming conventions, and whether raw data will be preserved separately from any cleaned or transformed versions. The analyst must also assess whether constructing a local structured database from extracted content is permissible under the terms identified in Step 2—the EU Database Directive, for instance, can make the construction of a database from extracted content illegal even when individual extractions are not. For projects involving ongoing or repeated extraction, the storage plan should additionally address versioning: how successive extractions will be distinguished, whether earlier versions will be retained or overwritten, and how storage growth will be managed over time.

3.4 Step 4: Scope & Sampling

The fourth step asks how much data will be extracted and what justifies the boundaries. It is the analytical heart of the plan, where the research question meets data reality. The analyst specifies which fields, columns, or attributes will be extracted, justifying both inclusions and exclusions. Extracting everything available is not a plan—it is the absence of one. In practice, the decisions made in this step may require preliminary investigation: a small number of exploratory API calls or limited test scrapes may be necessary to assess what data is actually accessible, how it is structured, and whether the fields described in documentation match what the source returns. These preliminary calls should be treated as part of the planning phase, not as the extraction itself, and any discrepancies between documented and actual data availability should be recorded in the DEP.

Temporal scope is documented and justified: what date range will the extraction cover, and is it determined by the research question, by data availability, or by practical constraints? A decision to extract "the last five years" should have a stated rationale, not serve as an unexamined default. Population and geographic scope are specified: all available records or a subset, and if a subset, what sampling strategy—random, stratified, convenience, or exhaustive within defined boundaries? Expected volume is estimated before extraction: number of rows, file sizes, and storage requirements. The UK COVID-19 data loss described in Section 2.2 occurred because the volume of results exceeded the row capacity of the file format used to consolidate them (Kelion, 2020)—a constraint that an advance estimate of data volume would have exposed. Exclusion criteria are defined in advance: what data will not be extracted or will be filtered post-extraction, and why. This parallels preregistration's requirement to specify exclusion criteria before observing outcomes (Nosek et al., 2018). Just as undocumented analytic flexibility undermines the credibility of statistical results, undocumented extraction decisions undermine the credibility of the dataset those results depend on.

3.5 Step 5: Validation & Contingency

The final step asks how the analyst will know that the extraction worked and what happens if it does not. It addresses verification and resilience: validation checkpoints are defined before

Table 1: The DEP decision flow: what and where → may I → how → how much → what if.

Step	Core Question	Veto Power?	If Triggered
1. Purpose & Source	Is this extraction necessary?	No	May redirect
2. Legal & Ethical	Am I permitted?	Yes	Can stop plan
3. Access & Technical	How do I get it?	No	May reshape scope
4. Scope & Sampling	How much, and why?	No	Defines boundaries
5. Validation & Contingency	Did it work? What if not?	No	May trigger pivot

extraction, including expected row counts, expected schema (column names, data types), acceptable null rates, and value range checks. A validation checkpoint defined after extraction is indistinguishable from post-hoc rationalization.

Reproducibility is addressed explicitly. Can someone else, given this plan and the extraction code, reproduce the same dataset? If the source is dynamic—a live API returning real-time data, a Wikipedia article subject to ongoing edits—the analyst documents what "reproducibility" means in context, perhaps reproducing the process rather than the exact output. This connects to the broader reproducibility framework articulated by National Academies of Sciences, Engineering, and Medicine (2019) and operationalized by Sandve et al. (2013).

Contingency planning anticipates source failure: what if the API shuts down, the website restructures, access is revoked, or data quality is insufficient? Alternative sources are identified in advance. The DEP may conclude that extraction should not proceed—and that conclusion, documented and justified, is a successful plan, not a failed one. Once extraction begins, a decision changelog records every deviation from the plan, preserving transparency even when the plan proves imperfect (Nosek et al., 2018).

3.6 Positioning the DEP Against Adjacent Frameworks

The DEP occupies a specific and previously unfilled position in the data science workflow landscape. It is complementary to Datasheets for Datasets (Gebru et al., 2021): datasheets document the finished product, while the DEP documents the secondary-acquisition decisions that made that product possible. The DEP is an operational layer beneath FAIR (Wilkinson et al., 2016): FAIR establishes that data should be accessible and reusable, while the DEP specifies how to make secondary acquisition itself documented and reproducible. Table 1 summarizes the decision flow across all five steps. The DEP provides the planning specificity that CRISP-DM lacks (Wirth and Hipp, 2000): CRISP-DM's Data Understanding phase is a broad directive, not a structured template. And the DEP is the secondary data-acquisition instantiation of the Q1Q2Q3 philosophy (Vance et al., 2026): qualitative understanding of the source, its constraints, and its legal context must precede the quantitative act of extraction.

4 Illustrative Case Studies

To demonstrate how the DEP operates in practice, we present three extraction plans of varying complexity. Each case was selected to illustrate different decision challenges: low-complexity scraping with clear licensing, grey-zone scraping where legal signals conflict, and API access under terms that shift over time. The cases are ordered by increasing complexity to show how

the framework scales from minimal to comprehensive. Complete DEPs for all three cases are provided in the standalone supplementary document; here we highlight the key decisions each case makes visible and explain why those decisions matter. Throughout, we adopt an explanatory perspective—walking the reader through the reasoning at each step rather than presenting the DEPs as finished documentation—so that the decision-making process is legible and readers can apply the same reasoning to their own extraction contexts.

4.1 Case 1: Wikipedia Population Tables (January 2025)

We begin with a deliberately simple case: a statistics instructor wants to extract U.S. state-level population data from a Wikipedia article for use in a classroom exercise on data wrangling. The target is a single HTML table on the article "List of U.S. states and territories by historical population," which consolidates decennial Census counts from 1790 to 2020. This case counters the objection that a DEP is always burdensome by showing how the framework scales down when complexity is low.

The instructor's goal is pedagogical: students will practice reading HTML tables into R, cleaning column names, and computing decade-over-decade growth rates. The research question does not require authoritative precision; approximate population figures are sufficient for the exercise. Wikipedia's pre-consolidated table is therefore chosen over the Census Bureau API because the learning objective is extraction technique, not Census API navigation. The DEP documents this tradeoff explicitly: the source is selected for accessibility, not authority, and the instructor acknowledges that the Census Bureau remains the authoritative upstream source.

This source choice also clarifies why direct extraction is preferable to downloading a pre-existing CSV file. Numerous CSV files of state population data exist online, but most lack documentation of their extraction process, source date, and cleaning assumptions. Wikipedia is not treated as the authoritative origin of the population counts; rather, it is treated as a convenient secondary access layer that consolidates decennial Census counts. The DEP therefore requires the instructor to document the Census Bureau as the upstream source, save the Wikipedia snapshot actually used, and validate selected values against Census Bureau publications. If those provenance and validation checks failed, the DEP would redirect the instructor to the Census API or Census historical tables rather than treating Wikipedia as sufficient on its own.

As of January 2025, Wikipedia content is published under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0), which permits copying, redistribution, and adaptation for any purpose—including commercial use—provided that attribution is given and derivative works are shared under the same license. The site's robots.txt file permits scraping of article pages by all user agents, and the underlying population data is produced by the U.S. Census Bureau, a federal agency whose data are in the public domain. There is no Terms of Service conflict, no personally identifiable information, and no need for Institutional Review Board review. The legal review takes minutes rather than hours, but that brevity is itself documented evidence that the instructor assessed the landscape rather than skipping the step.

The access plan is similarly compact but still consequential. The extraction requires a single HTTP request to a static HTML page, parsed with a table-reading function such as `rvest::html_table()` in R (Wickham, 2024) or `pandas.read_html()` in Python (The pandas development team, 2026). No authentication, pagination, or rate limiting is required. The DEP nevertheless records two design choices that matter for reproducibility. First, a copy of the raw HTML page will be saved alongside the extracted CSV file because Wikipedia is a living

Table 2: Sample records from the Wikipedia population extraction.

Name	1960	2000	2020	Is state
Alabama	3,266,740	4,447,100	5,024,279	Yes
Alaska	226,167	626,932	733,391	Yes
American Samoa	20,051	57,291	49,710	No

document whose values, columns, or page structure may change after extraction. Second, the target table is identified by index position (e.g., the third table in the article), which creates a known fragility if the article is reorganized. Recording this fragility tells future users of the script to verify table contents rather than trust the index blindly.

The scope and validation decisions are deliberately modest. The classroom exercise uses 1950 through 2020, while all available years are extracted and retained in the raw data. All 50 states are included in the primary dataset; territories are extracted but flagged separately and excluded from the classroom growth-rate analysis. Table 2 provides a small sample of the extracted records and makes the planned separation visible: the selected decennial population columns support growth calculations, while `is_state` distinguishes states from territories such as American Samoa. The expected volume is small—approximately 56 rows and 25 columns, well under 50 KB—but the validation plan still specifies spot-checking selected values against Census Bureau publications. If any of these values fail, the extraction is investigated before cleaning or analysis proceeds.

The result is not merely an extracted table. It is a compact, auditable acquisition record that explains why Wikipedia was used, what legal and ethical checks were completed, how a source snapshot will be preserved, which records will be analyzed, and how the extraction will be verified. The entire DEP fits on a single page because the source is openly licensed, the access method is simple, and the legal landscape is clear. Its value lies not in length but in making even a low-complexity extraction transparent enough for another instructor or student to evaluate.

4.2 Case 2: Basketball Reference Player Statistics (October 2025)

The second case occupies a legal and ethical grey zone that many student and researcher projects encounter but rarely document. A researcher seeks to analyze the impact of the three-point revolution on player efficiency in the NBA from 2000 to 2025, using per-game and advanced statistics from Basketball Reference (basketball-reference.com), a Sports Reference LLC property. Basketball Reference is the de facto standard source for historical NBA statistics in both journalism and academic research, and its Uniform Resource Locator (URL) structure has remained consistent for over a decade.

The research question requires two types of data: shooting profile variables (three-point attempts, three-point percentage, field goal attempts) and efficiency metrics (Player Efficiency Rating, Win Shares, Box Plus/Minus). Basketball Reference provides both in pre-computed, season-level summary tables—one page per season for per-game statistics and one page per season for advanced statistics. The researcher selects Basketball Reference over the NBA's official statistics portal (stats.nba.com), whose API changes frequently and without notice, and over the nbastatR (Bresler, 2026) and hoopR (Gilani, 2026) R packages, which wrap the NBA API and inherit the same instability. For a 25-year longitudinal analysis, Basketball Reference's stable URL structure is part of the methodological rationale, not merely a convenience.

The legal and ethical review is the central contribution of this case. As of October 2025, the site's Terms of Use state that, without express written permission, users may not use automated means to access or use the site—including scripts, bots, scrapers, or data miners—"in a manner that adversely impacts site performance or access." Read literally, this language cautions against programmatic extraction, but the qualifying phrase about adverse impact matters. The site's robots.txt file, also reviewed in October 2025, provides a different signal: it permits crawling of player pages, team pages, league pages, and season summary pages, and it specifies a Crawl-delay of 3 seconds. In other words, the same site maintains broad legal language while publishing technical instructions for responsible automated access.

The DEP does not pretend to resolve this tension definitively. Instead, it makes the reasoning visible. The researcher records the Terms of Use language, the robots.txt directives, and the conditions under which extraction will proceed: the project is non-commercial and academic, the volume is low, requests will be spaced more slowly than the robots.txt minimum, data will not be republished in bulk or in a form substituting for the original site, and the researcher will stop immediately if contacted by Sports Reference. This reasoning may not satisfy every reader, but it is explicit and auditable. Without a DEP, the same judgment would likely be hidden in code or made informally without any record.

The technical plan then reduces both legal risk and server load. Rather than scraping individual player pages, which would require thousands of requests across 25 seasons, the researcher targets season-level summary pages where all players' statistics for a given season appear on a single page. This design reduces the total request count to exactly 52: one per-game page and one advanced-statistics page for each of 26 seasons, from 2000–2001 through 2024–2025. At a self-imposed delay of 5 seconds between requests, exceeding the 3-second robots.txt minimum, the full extraction completes in under 5 minutes while remaining deliberately conservative.

The remaining decisions connect the extraction design to the eventual analysis. The temporal scope is justified by the research question because 2000 through 2025 captures the pre-analytics era, the three-point inflection point around 2012–2015, and the current three-point-dominant era. Players with fewer than 20 games are retained in the raw data but excluded from regression analysis because small samples distort per-game averages. Table 3 samples the per-game component of the planned player-season acquisition and displays only the identifiers, participation measure, and shooting and scoring variables relevant to this case. In this sample, every player exceeds the 20-game threshold; the full acquisition retains the same games-played field so the exclusion can be applied reproducibly before analysis. These per-game records are subsequently joined to the advanced efficiency variables described above. The validation plan specifies expected row counts per season, spot-checks of prominent player-seasons, and a zero-tolerance check for unmatched rows when joining per-game and advanced tables. Together, these decisions transform a potentially sprawling scrape into a bounded, reviewable acquisition: approximately 13,000 merged player-season rows, gathered through a low-volume request plan, accompanied by a documented legal rationale and measurable validation criteria.

4.3 Case 3: Spotify Audio Features via the Web API (September 2021)

The final case is the most complex not because the code is difficult, but because the legal, ethical, and contingency dimensions are unusually unstable. It also illustrates something the other cases cannot: what happens when the extraction landscape shifts dramatically after a DEP is written. We present this case as it would have been planned in September 2021, then show how events between 2024 and 2026 vindicated the contingency planning that the framework requires.

Table 3: Sample per-game records from the Basketball Reference acquisition.

Player	Team	Games	FGA	3PA	3P%	PTS	Season
Allen Iverson	PHI	71	25.5	4.3	0.320	31.1	2001
Jerry Stackhouse	DET	80	24.1	5.9	0.351	29.8	2001
Shaquille O'Neal	LAL	74	19.2	0.0	0.000	28.7	2001
Kobe Bryant	LAL	68	22.2	2.9	0.305	28.5	2001
Vince Carter	TOR	75	22.1	5.3	0.408	27.6	2001

In September 2021, a researcher seeks to predict song popularity from Spotify's audio features—valence, danceability, speechiness, energy, acousticness, instrumentalness, liveness, and tempo—using multiple linear regression. The extraction will use the spotifyr R package (Thompson et al., 2024, Version 2.2.5), which wraps the Spotify Web API and enables bulk retrieval of artist discographies with associated audio features. Spotify is a singular source for this question because it is the only provider of these proprietary audio feature metrics, which are derived from the former Echo Nest audio analysis engine. No other platform computes or publishes equivalent measures, so the DEP records the dependency explicitly.

The researcher considers a pre-existing Kaggle dataset titled "Spotify Tracks Dataset," containing approximately 586,000 tracks uploaded in March 2021, but rejects it because it lacks documentation of its extraction parameters. There is no record of which API endpoints were used, what sampling strategy produced the track selection, when the data were pulled, or whether the extraction complied with Spotify's Developer Terms. The dataset is therefore not rejected because it is too small or irrelevant; it is rejected because its provenance is precisely the kind of ambiguity the DEP is designed to prevent.

As of September 2021, the Spotify Developer Terms of Service contained several provisions relevant to the extraction. One provision creates a direct tension with the research question: the Terms state that Spotify data may not be used to "train machine learning models." The researcher's intended use—fitting a multiple linear regression model to identify predictors of popularity—is not machine learning in the contemporary sense, but regression is technically a supervised learning technique. The DEP documents this ambiguity rather than smoothing it away. The analyst records the exact terms, explains the interpretation that academic regression analysis is distinct from building a competing recommendation service, and flags the risk that Spotify may not share that interpretation.

The access and sampling plan reflect the same emphasis on explicit tradeoffs. The Spotify Web API uses OAuth 2.0 authentication, and Spotify does not publish its rate limits. Through preliminary test calls, the researcher estimates a limit of approximately 100 requests per minute, while spotifyr handles pagination internally by batching up to 50 tracks per audio features call and up to 20 albums per artist discography call. The sampling plan retrieves complete discographies for the top 10 artists by Spotify monthly listeners in each of 6 genres: pop, rock, hip-hop, country, electronic, and R&B. This yields approximately 60 artists and roughly 7,200 tracks, while also introducing a documented limitation: because popularity is the response variable, selecting only the most-listened-to artists creates range restriction, and Spotify's Browse endpoint provides no mechanism for random artist retrieval. A small glimpse of the planned track-level data would include one row per track with fields such as artist, track name, popularity, valence, danceability, energy, tempo, acousticness, instrumentalness, liveness, and speechiness, making

clear what is being acquired without reproducing the full dataset.

The contingency section is, in retrospect, the most consequential part of the plan. The researcher identifies four scenarios: if rate limits tighten, increase delays; if audio feature endpoints are deprecated, fall back to the static Kaggle dataset with documented limitations; if the spotifyr package developed by Thompson et al. (2024) is removed from the Comprehensive R Archive Network (CRAN), pin the current version using renv (Ushey and Wickham, 2025); and if the Spotify developer account is suspended, pivot to ListenBrainz, Last.fm, or MusicBrainz. Between 2024 and 2026, every one of these scenarios materialized. Spotify deprecated the Audio Features and Audio Analysis endpoints for new developer applications in November 2024, restricted extended API access in March 2025, narrowed Developer Mode by February 2026, cited "advances in automation and AI" as part of the rationale for restriction, and spotifyr was removed from CRAN following the API deprecation.

A DEP written in September 2021 would not have prevented Spotify from changing its policies. Its value is that it would have preserved the terms under which the data were originally obtained, the rationale for proceeding, the dependency on a single proprietary source, and the alternative sources identified before the crisis made such planning urgent. The case therefore demonstrates a temporal property of the DEP framework: a 2021 plan could reasonably conclude "proceed with documented risks," while a 2025 or 2026 plan for the same research question would conclude "do not proceed with Spotify; use alternative sources." Both outcomes are valid. The plan is the contribution because it makes the acquisition decision auditable even when the source later becomes unavailable.

4.4 Cross-Case Comparison

Table 4 places the three cases side by side along the dimensions that the DEP makes explicit, and the comparison reveals three patterns. First, technical simplicity does not predict planning simplicity. The Wikipedia case requires only a single HTTP request, yet it still benefits from documenting the source rationale, reproducibility anchor, and validation checks. The Basketball Reference case involves only 52 requests, but its legal and ethical reasoning is more demanding because the Terms of Use and robots.txt signals must be interpreted together. The Spotify case is technically abstracted by an API wrapper, but it carries the highest contingency risk because access depends on a proprietary platform whose policies changed substantially over time.

Second, the cases show that a DEP can support different outcomes rather than simply authorize extraction. The Wikipedia plan proceeds without conditions because the legal landscape is clear and the technical burden is small. The Basketball Reference plan proceeds with documented conditions: low request volume, conservative delays, no bulk republication, and a willingness to stop if contacted. The Spotify plan proceeds under September 2021 conditions but would reach the opposite conclusion under 2025 or 2026 conditions. This range of outcomes is central to the framework: a DEP is not a permission slip for extraction, but a structured record of whether extraction is justified, under what conditions, and with what fallback options.

Finally, the contingency risk column emphasizes that extraction conditions are not static. An API that is open today may be restricted tomorrow; a permissive robots.txt file may be revised; a package that works today may be removed from its distribution channel. The DEP does not eliminate these uncertainties, but it ensures that the analyst has considered them, documented a plan for responding to them, and preserved a record of any necessary changes as extraction conditions shift.

Table 4: Cross-case comparison of DEP dimensions across three illustrative extraction scenarios.

Dimension	Wikipedia (2025)	Basketball Ref. (2025)	Spotify (2021)
Access method	Public HTML scrape	Public HTML scrape	Authenticated API
Legal complexity	Low	Medium	High
Rate limits	Not applicable (1 request)	Crawl-delay: 3 sec	~100/min (unpublished)
Total requests	1	52	~60
Data stability	Semi-stable	Stable	Dynamic
Contingency risk	Low	Moderate	Critical
DEP outcome	Proceed	Proceed with conditions	Proceed (2021); would not in 2025

5 When Should the DEP Be Used?

The DEP is designed to scale. A classroom assignment extracting a single public CSV file warrants a lightweight DEP—perhaps a paragraph addressing each step, completed in fifteen minutes. A multi-source research project acquiring secondary data from three APIs with different authentication requirements, Terms of Service, and rate limits warrants a comprehensive DEP that may span several pages and require legal consultation.

In our assessment, the framework is most valuable when any of the following conditions hold. The first is that secondary data acquisition uses an API with Terms of Service or that scraping targets a website the analyst does not own, situations in which legal and ethical review is a recognized prerequisite rather than an optional courtesy (Dogucu and Çetinkaya-Rundel, 2021). The second is that the pre-existing data involve personally identifiable information, which triggers the privacy and de-identification obligations discussed in Section 3.2 and reflected in journal guidelines for sharing data and code (Horton and Stoudt, 2024). The third is that the data are intended for publication or shared analysis, where reproducibility norms expect the full path from source to result to be documented (Sandve et al., 2013; Horton et al., 2022). The fourth is that the source has a history of changing access policies, as the Spotify case in Section 4.3 illustrates. The framework may be abbreviated when downloading a single public CSV from a government data portal, working with instructor-provided datasets, or conducting purely exploratory personal analysis that will not be shared. Even then, we recommend against omitting it entirely: every dataset carries a provenance that conditions its interpretation (Buneman et al., 2001), and a one-paragraph DEP preserves that provenance at negligible cost.

A DEP paired with version-controlled extraction code creates a substantially more auditable record of secondary data acquisition than code alone can provide. The DEP documents the decisions; the code documents the implementation; the version history documents how both evolved. Together, they enable a third party to understand not just what data was extracted, but why it was extracted that way, what alternatives were considered, and what would need to change if the source or context shifts. This connection to reproducibility extends the arguments made by Sandve et al. (2013), who proposed ten rules for reproducible computational research, and by

Horton et al. (2022), who emphasized the growing importance of reproducible workflow in the curriculum. The DEP contributes a missing piece: structured documentation of the acquisition phase that makes downstream reproducibility meaningful. If the data's origin is undocumented, reproducibility of the analysis is incomplete at best and illusory at worst.

The DEP also has natural applications as a pedagogical tool for secondary data work. Requiring students to complete a DEP before full-scale extraction teaches planning as a professional skill, surfaces legal and ethical considerations that pure coding exercises miss, forces scope decisions to be explicit and justified rather than implicit and habitual, and creates a written artifact that instructors can evaluate independently of code quality. In courses that teach web scraping or API access—such as those described by Dogucu and Çetinkaya-Rundel (2021) and motivated by the report of National Academies of Sciences, Engineering, and Medicine (2018) on Data Science for Undergraduates—the DEP provides the planning structure that complements technical instruction. The Wikipedia case study demonstrates that even a simple extraction provides enough material for a meaningful DEP exercise.

Several extensions of this work merit future investigation. Academic journals could require a DEP or comparable extraction documentation alongside data availability statements. Just as preregistration registries such as the Open Science Framework provide time-stamped records of research intentions, a DEP registry could provide time-stamped records of extraction decisions. The DEP could also be standardized into a machine-readable format that integrates with data management tools, extended to extraction from large language models where the "source" is a generative model and the "extraction" is prompt-based, or adapted into a parallel framework for primary data collection. These possibilities suggest that structured acquisition and extraction planning will become more necessary, not less, as data sources become more diverse and technically accessible.

6 Conclusion

This paper has introduced the Data Extraction Plan, a five-step framework for documenting the decisions that guide secondary data acquisition. The DEP's contribution is not a software tool or a data pipeline but a structured way of thinking about extraction—one that applies across programming languages, secondary data sources, and domains of application. By requiring analysts to document their purpose, legal standing, technical approach, scope decisions, contingency plans, and later deviations from those plans, the DEP elevates secondary data acquisition from an invisible prerequisite to a first-class methodological phase.

The three case studies presented here demonstrate that the DEP's value scales with complexity but is never zero. The Wikipedia case showed that even a single-page, CC-licensed extraction benefits from documenting the reproducibility strategy and validation approach. The Basketball Reference case showed that the DEP's greatest value may lie in documenting reasoning in legal grey zones, making the analyst's ethical considerations visible to others. The Spotify case showed that a DEP written in 2021 correctly identified contingency scenarios that materialized over the following four years—and that a DEP written in 2025 would reach the opposite conclusion, recommending against extraction entirely.

The preregistration movement succeeded not because documenting intentions is technically difficult, but because the scientific community recognized that undocumented flexibility undermines credibility. The same logic applies to secondary data extraction. Every secondary dataset in an analysis was acquired through a series of decisions. If those decisions are invisible, the

analysis built upon them cannot be fully evaluated, trusted, or reproduced. The DEP's closing demand is therefore discipline before and during data acquisition: analysts who cannot document how they planned to obtain the data, and how that plan changed when the source pushed back, cannot claim that the analysis is reproducible.

Disclosure of AI Usage

The ideas, research direction, interpretation, and final decisions in this manuscript originated with the author. Generative AI tools were used only as support tools to help organize the author's thought process, improve wording and structure, and generate or revise code for the supplementary materials. All AI-assisted text and code were reviewed, edited, and verified by the author. The author accepts full responsibility for the accuracy, integrity, and final content of the manuscript. This disclosure is provided in accordance with common scholarly transparency practices for reporting AI-assisted writing and coding support.

Supplementary Material

The supplementary material for this article is available on the official website of the Journal of Data Science at https://jds-online.org/. It may include additional results, appendices, tables, figures, code, and data.

References

Boenig-Liptsin M, Tanweer A, Edmundson A (2022). Data science ethos lifecycle: Interplay of ethical thinking and data science practice. *Journal of Statistics and Data Science Education*, 30(3): 228–240. https://doi.org/10.1080/26939169.2022.2089411

Bresler A (2026). nbastatR: R's interface to NBA data. R package version 0.1.153, commit ad75a30, https://github.com/abresler/nbastatR.

Buneman P, Khanna S, Tan WC (2001). Why and where: A characterization of data provenance. In: *Proceedings of the 8th International Conference on Database Theory* (Van den Bussche J, Vianu V, eds.). Lecture Notes in Computer Science, volume 1973, 316–330.

Chen M, Tworek J, Jun H, Yuan Q, Ponde de Oliveira Pinto H, . . . , Zaremba W (2021). Evaluating large language models trained on code. arXiv preprint: https://arxiv.org/abs/2107.03374.

Chu X, Ilyas IF, Krishnan S, Wang J (2016). Data cleaning: Overview and emerging challenges. In: *Proceedings of the 2016 International Conference on Management of Data* (Özcan F, Koutrika G, Madden S, eds.), 2201–2206.

Dogucu M, Çetinkaya-Rundel M (2021). Web scraping in the statistics and data science curriculum: Challenges and opportunities. *Journal of Statistics and Data Science Education*, 29(sup1): S112–S122. https://doi.org/10.1080/10691898.2020.1787116

Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, . . . , Crawford K (2021). Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Gilani S (2026). hoopR: Access Men's Basketball Play by Play Data. R package version 3.1.0.

Horton NJ, Alexander R, Parker M, Piekut A, Rundel C (2022). The growing importance of reproducibility and responsible workflow in the data science and statistics curriculum. *Journal of Statistics and Data Science Education*, 30(3): 207–208. https://doi.org/10.1080/26939169.2022.2141001

Horton NJ, Stoudt S (2024). Editorial: Guidelines and best practices to share deidentified data and code. *Journal of Statistics and Data Science Education*, 32(3): 227–231. https://doi.org/10.1080/26939169.2024.2364737

Hutchinson B, Smart A, Hanna A, Denton E, Greer C, . . . , Mitchell M (2021). Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Elish MC, Isaac W, Zemel RS, eds.), 560–575.

Indeed (2026). Terms of Service. Indeed.com, last updated 29 September 2026, https://www.indeed.com/legal. Accessed 11 October 2026.

Kandel S, Heer J, Plaisant C, Kennedy J, van Ham F, . . . , Buono P (2011). Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4): 271–288.

Kelion L (2020). Excel: Why using Microsoft's tool caused COVID-19 results to be lost. *BBC News*, 5 October 2020. https://www.bbc.com/news/technology-54423988. Accessed 11 October 2026.

McKinney W (2010). Data structures for statistical computing in Python. In: *Proceedings of the 9th Python in Science Conference* (S van der Walt, J Millman, eds.), 56–61.

Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, . . . , Gebru T (2019). Model cards for model reporting. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency* (boyd d, Morgenstern JH, eds.), 220–229.

National Academies of Sciences, Engineering, and Medicine (2018). *Data Science for Undergraduates: Opportunities and Options*. The National Academies Press, Washington, DC.

National Academies of Sciences, Engineering, and Medicine (2019). *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC.

Nolan D, Temple Lang D (2010). Computing in the statistics curricula. *American Statistician*, 64(2): 97–107. https://doi.org/10.1198/tast.2010.09132

Nosek BA, Ebersole CR, DeHaven AC, Mellor DT (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11): 2600–2606. https://doi.org/10.1073/pnas.1708274114

Pearce H, Ahmad B, Tan B, Dolan-Gavitt B, Karri R (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In: *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)*, 754–768.

Pushkarna M, Zaldivar A, Kjartansson O (2022). Data cards: Purposeful and transparent dataset documentation for responsible AI. In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1776–1826.

Quaranta L, Calefato F, Lanubile F (2022). Eliciting best practices for collaboration with computational notebooks. *Proceedings of the ACM on Human–Computer Interaction*, 6(CSCW1): Article 87. https://doi.org/10.1145/3512934

Richardson L (2026). Beautiful Soup Documentation. Beautiful Soup version 4.15.0, https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10): e1003285. https://doi.org/10.1371/journal.pcbi.1003285

Stodden V (2020). The data science life cycle: A disciplined approach to advancing data science as a science. *Communications of the ACM*, 63(7): 58–66. https://doi.org/10.1145/3360646

Stoudt S, Jernite Y, Marshall B, Marwick B, Sharan M, . . . , Danchev V (2024). Ten simple

rules for building and maintaining a responsible data science workflow. *PLOS Computational Biology*, 20(7): e1012232.

The pandas development team (2020). pandas-dev/pandas: Pandas. Zenodo. https://doi.org/10.5281/zenodo.3509134.

The pandas development team (2026). pandas.read_html. pandas documentation, https://pandas.pydata.org/docs/reference/api/pandas.read_html.html. Accessed 11 October 2026.

Thompson C, Antal D, Parry J, Phipps D, Wolff T (2024). spotifyr: R Wrapper for the 'Spotify' Web API. R package version 2.2.5, commit 68bc0ae, https://github.com/charlie86/spotifyr.

Ushey K, Wickham H (2025). renv: Project Environments. R package version 1.1.5.

Vance EA, Trumble IM, Alzen JL, House LL (2026). The Q1Q2Q3 workflow for statistics and data science collaborations. *Journal of Statistics and Data Science Education*, 34(3): 319–339. https://doi.org/10.1080/26939169.2025.2475775

Wickham H (2024). rvest: Easily Harvest (Scrape) Web Pages. R package version 1.0.4.

Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, . . . , Mons B (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3: 160018.

Wirth R, Hipp J (2000). CRISP-DM: Towards a standard process model for data mining. In: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (Mackin N, ed.), 29–39. The Practical Application Company.