

# **Data Science Assignment 2: Data Collection in Practice**

DataScience\_Assignment2

Aoife Wang, Xinchao Chen, Jim Lu  
Rensselaer Polytechnic Institute

October 7, 2025

# **1. Report on your data management practical experience**

## **1.1. The goal (what it was driven by), and the mode(s) of collection**

The primary goal of this data collection exercise was to create a comprehensive dataset for analyzing the relationship between urban climate patterns, economic indicators, and energy systems in three major U.S. cities: New York City, Los Angeles, and Seattle. This was driven by the hypothesis that local weather phenomena (like heatwaves or cold snaps), economic conditions (like household income and energy prices), population size, education level significantly influence energy consumption and production. The primary mode of collection was a programmatic pipeline built in Python, which utilized public APIs to automatically fetch data. Specifically, we used the Open-Meteo API for historical weather data and the U.S. Energy Information Administration (EIA) v2 API for state-level electricity generation data. For economic indicators, I used a hybrid approach: data on energy prices (electricity, gasoline, natural gas) was programmatically collected from the Bureau of Labor Statistics (BLS) database, while annual per capita income data was manually downloaded as CSV files from the U.S. Census Bureau and Bureau of Economic Analysis (BEA) websites, as they lacked a straightforward monthly API for this specific granularity.

## **1.2. How data was really acquired, any problems, lessons learned, etc., converted.**

The data acquisition process was largely successful but presented several practical challenges. The Open-Meteo API was very straightforward and required no authentication, allowing my script to efficiently pull daily weather data from 2014 to 2024. In contrast, the EIA API required registering for an API key and carefully navigating its documentation to find the correct data series for state-level generation. A significant problem I encountered was a mismatch in data frequency and geographic granularity. My goal required monthly city-level data, but I found that critical economic indicators like per capita income were only available annually and at the Metropolitan Statistical Area (MSA) level. Similarly, detailed energy generation data from the EIA was at the state level. This was a crucial lesson learned: the ideal dataset for a research question often doesn't exist, forcing a compromise. I had to adapt by using annual income figures for each month of a given year and mapping state-level energy data to the corresponding city, noting these assumptions clearly in my documentation.

### **1.3. Physical and logical organization of data/ metadata, success, failures.**

The logical organization of the data was a major success. I adopted a standard 'raw' and 'curated' directory structure. All data as it was originally acquired from the APIs or downloaded was stored in `data/raw`, with subfolders for each city or data source. This preserved the original, untouched data. Then, processed, cleaned, and aggregated data was stored in `data/curated`. For example, daily weather data was aggregated into monthly summaries, and all monthly data sources (weather, energy prices, income) were eventually joined into a single file. This separation proved invaluable for debugging and ensuring reproducibility. Physically, all data is stored on my local file system. The only minor failure was an initial lack of a consistent file naming convention, which I corrected early on to the format `source_frequency_geo.format` (e.g., `weather_monthly_by_city.parquet`).

### **1.4. Metadata and documentation collected/ stored, what provenance was identified.**

Comprehensive metadata and provenance were documented in a `README.md` file at the root of the project repository. This file serves as the central documentation hub. For each dataset, I recorded its source (e.g., a direct link to the EIA API documentation), the time coverage, the spatial coverage, and collection method. The provenance of all data is clearly identified; for instance, the weather data's origin is explicitly stated as the Open-Meteo API, and economic data is attributed to the BLS and Census Bureau. I also created detailed data dictionaries within the `README` for each curated file, listing every field, its data type, and a description in both English and Chinese. This practice ensures that any future user (including myself) can understand the dataset's contents and origins without ambiguity.

### **1.5. A link to the data/ metadata (or a copy of it).**

All collected data and metadata are stored within my project repository. The raw and curated data files are located in the `/data/` subdirectory, organized as described above. The complete metadata, including data dictionaries, source information, and collection methodology, is available in the `README.md` file located in the project's root directory.

## 2. Describe your experience in how well (or not) the data management plan turned out in practice

My data management plan was a critical guide, but in practice, it required significant adaptation. It served as an effective roadmap for goals like using a scripted pipeline and maintaining a logical data structure, but it was too rigid in its initial assumptions about data availability. The experience highlighted that a data management plan should be a living document, flexible enough to accommodate the real-world complexities of data acquisition.

### 2.1. Highlight what worked and what did not and why.

- **Data Acquisition (Worked):** The plan to use Python scripts and APIs for collection was highly effective for sources like Open-Meteo and BLS. It made the process reproducible and scalable.
- **Data Storage Format (Worked):** The decision to use Parquet for structured data and JSON for metadata/dictionaries worked very well. Parquet is efficient for numerical analysis in pandas, and JSON is human-readable and universally compatible.
- **Data Granularity (Did Not Work):** My initial plan called for monthly, city-level data for all variables. This failed because government sources like the Census Bureau only provide annual income data at the MSA level. This discrepancy between the plan and reality was the single biggest challenge.
- **Geographic Scope (Partially Worked):** The plan to map state-level energy generation data to cities was a necessary compromise, but it introduces a layer of abstraction. It worked in the sense that it allowed the data to be joined, but it's a noted limitation in the dataset's accuracy.

### 2.2. What would you do differently next time and why?

Next time, I would build a dedicated "data discovery" phase into the very beginning of the project, before finalizing the data management plan. I would write small, exploratory scripts to test the availability, frequency, and granularity of each potential data source. This would have identified the annual limitation of income data much earlier. This upfront investment of time would create a more realistic plan and prevent the need for major methodological changes midway through the data collection process.

### **2.3. Were there effects on the data or metadata collected, its quality, etc.?**

Yes, the practical realities of collection directly affected the data's quality and characteristics. The most significant effect was on temporal resolution. By having to repeat annual income values for twelve consecutive months, the dataset's ability to capture short-term economic shifts is diminished. This is a compromise in data quality that had to be made. This limitation was then explicitly documented in the metadata (`README.md`) to ensure any subsequent analysis acknowledges it, which in turn improved the quality and integrity of the documentation itself.

## **3. 6000-Level student question**

### **3.1. What standards (data format, metadata, etc.) did you use (whether planned or not) and/or what ones couldn't you use and why?**

I used several de facto standards in practice. For data formats, I standardized on Apache Parquet for its efficiency with columnar data and JSON for its readability and web compatibility. For dates and times, I used the ISO 8601 standard (YYYY-MM-DD) to ensure consistency. I could not use a formal, overarching metadata standard like Dublin Core or ISO 19115 because my data sources were too diverse. Each API and government website had its own unique data structure and terminology, making it impractical to map everything to a single external standard. Instead, I created a custom but thoroughly documented metadata schema in the project's `README.md` file.

### **3.2. What standards/conventions would have been helpful ('none' is not an acceptable answer), i.e. what area?**

A standardized convention for geographic identifiers across different U.S. government agencies would have been immensely helpful. While FIPS codes exist, navigating between city names, county definitions, and Metropolitan Statistical Areas (MSAs) across the Census Bureau, BLS, and EIA datasets required manual mapping and introduced potential for error. A single, unified geographic ontology would streamline the integration of such disparate datasets.

### **3.3. Were you aware of a best practice for the type of collection you carried out? Did you use it? If not, why not? - describe details.**

Yes, I was aware of several best practices for this type of data collection. The primary one I used was building a reproducible, scripted pipeline to acquire and process the data, which is a cornerstone of modern data science. Another best practice I followed was the separation of raw,

immutable data from processed, curated data. I did not, however, use a formal data versioning tool like DVC (Data Version Control). While I was aware of it as a best practice for tracking changes in datasets, it felt like an unnecessary complexity for a single-person project of this scope, though I would certainly adopt it for a larger, collaborative effort.