**Report on Data Collection and Management Practical Experience**

**1. The Goal and Mode of Collection**

The goal of this data collection exercise was to **retrieve historical weather data** for **Idaho, USA**, using an **automated observation-based method**. The collected data included:

* **Daily maximum temperature (TMAX)**
* **Minimum temperature (TMIN)**
* **Precipitation (PRCP)**
* **Wind speed (AWND)**

The data was intended for **analyzing long-term climate trends** and ensuring structured data management for further analysis. I did the data analysis for a year.

The **mode of data collection was observation**, utilizing **publicly available weather data APIs** to fetch real-time and historical weather records. Initially, I thought of using **OpenWeatherMap API**, however, it required a **paid subscription for historical data**. So, I decided to **switch to NOAA’s Climate Data Online (CDO) API**, which provides **free access to historical weather records**.

This method allowed for the **automatic retrieval of weather data using a Python script**. The script made **API requests to NOAA’s database and stored the results in CSV format** for further analysis.

**Details of the Mode and Purpose:**

* **Observation Method:** Weather data was collected via an **API-based approach**, fetching daily weather records for a chosen location (Idaho).
* **Purpose:** The goal was to **analyze weather trends**, such as **temperature fluctuations and precipitation patterns** over time.
* **Driven by:**
  + **A research question**: *What are the daily weather trends in a specific location, and how do they correlate with typical seasonal patterns?*
  + **A need for structured weather data** for potential future applications such as climate modeling or environmental impact studies.

**2. Data Acquisition, Conversion, and Lessons Learned**

* **How Data Was Acquired:**
  + The Python script used the **NOAA API** to fetch historical weather data from the **nearest NOAA weather station in Idaho**.
  + API calls were made using the **station ID**, and relevant weather metrics (**TMAX, TMIN, PRCP, AWND**) were extracted.
  + Data was structured and stored in **CSV format**, while metadata was saved in a **JSON file**.
* **Problems Faced:**
  + The initial approach using **OpenWeatherMap API** failed due to a **paid subscription requirement** for historical data.
  + NOAA API imposed **rate limits**, requiring adjustments to avoid excessive requests.
  + Some NOAA stations had **missing data points**, causing **gaps in collected data**.
* **Lessons Learned:**
  + **Choosing the right API** is crucial to avoid unexpected costs.
  + **Automating API calls efficiently** (with error handling and retry mechanisms) prevents data loss.
  + **Data cleaning** is necessary to handle missing or incomplete values.

**3️. Organization of Data and Metadata – Successes and Failures**

* **Physical Organization:**
  + Data was stored in a structured directory (WeatherData/).
  + Files were named dynamically (noaa\_weather\_data\_YYYY-MM-DD.csv) to prevent overwriting.
  + **Backups were made** on Github to ensure data was not lost.
    - **Github:** [**https://github.com/Bibek246**](https://github.com/Bibek246)
* **Logical Organization:**
  + Data was stored in **CSV format** for easy processing and sharing.
  + Metadata was stored separately in **JSON format**, following **Dublin Core and ISO 19115 standards**.
* **Successes:**
  + **Automated retrieval worked well**, reducing manual effort.
  + **Standardized metadata ensured** proper documentation.
* **Failures:**
  + **Some stations lacked complete datasets**, requiring manual checking and alternative station selection.
  + **Formatting inconsistencies** required extra processing before analysis.

**4️. Metadata and Documentation Collected / Provenance Identified**

* **Metadata Elements Recorded:**
  + Dataset **Title, Creator, Date Created, Description**
  + **Location & Station ID**
  + **Timeframe of Data Collection**
  + **Columns and Data Types (e.g., Temperature in °C, Precipitation in mm)**
  + **Collection Method and API Details**
  + **Standards Used:** Dublin Core & ISO 19115
* **Provenance Identified:**
  + Data source: **NOAA Climate Data Online (CDO) API**
  + API calls and timestamps were recorded in logs for reproducibility.

**5️. Link to Data and Metadata**

* **Data File:** [noaa\_weather\_data\_YYYY-MM-DD.csv]
* 
* **Metadata File:** [noaa\_metadata.json]
* 

**Reflection on Data Collection and Management Plan Execution.**

**How Well the Plan Worked in Practice**

The data collection and management plan **mostly worked well**, but there were **unexpected challenges** that required modifications. Initially, the **OpenWeatherMap API was considered**, but it required a **paid subscription** for historical data. This led to a **switch to NOAA’s API**, which provided **free access** but introduced new **challenges** such as **station availability issues, rate limits, and missing data**.

**🔹 What Worked Well and What Did Not**

| **Category** | **What Worked ✅** | **What Did Not Worked ⚠** | **Why?** |
| --- | --- | --- | --- |
| **API Selection** | Switching from **OpenWeatherMap** to **NOAA** allowed free data retrieval. | NOAA **did not always have complete data** for every station. | Some NOAA stations had **gaps in daily records**, causing missing data issues. |
| **Automated Data Retrieval** | The script **successfully pulled data** from the NOAA API. | Some requests **failed due to rate limits**, requiring retries. | NOAA **limits API requests per minute**, which caused failures. |
| **Data Storage** | CSV files were **well-structured** and dynamically named. | Some records contained **missing values** for certain days. | Certain NOAA stations **did not report data consistently**. |
| **Metadata Management** | Metadata was **structured correctly** and followed **Dublin Core** standards. | No automatic **change logs** were recorded. | Would be useful to track **dataset updates** over time. |
| **Data Formatting for Excel** | Enforcing the MM-DD-YYYY format **resolved date issues**. | Excel **initially displayed # symbols** instead of dates. Now, to see all the results, click the cell double to reveal it. | Excel **misinterpreted some columns** until properly formatted. |

**🔹 What Would I Do Differently Next Time?**

1. **Verify API Data Availability First**

* Before **choosing an API**, I would check **data availability for a given location** to avoid switching APIs mid-project.
* This would have saved **time spent debugging the OpenWeatherMap and NOAA API limitations**.

1. **Test Multiple NOAA Stations Before Data Collection**

* Instead of assuming **one station is sufficient**, I would first **run a check on multiple stations** to find the **one with the most complete dataset**.
* This would reduce **missing data issues**.

1. **Implement a Data Quality Check**

* Add a **pre-processing step** to check for **missing data** and **fill gaps** using interpolation or previous values.
* This would ensure a **continuous dataset** for better analysis.

1. **Improve Metadata Versioning**

* Store **change logs** when metadata updates occur, making it easier to track **dataset modifications over time**.

**🔹 Effects on Data and Metadata Quality**

* **Data Completeness:**
  + Some **days were missing** weather data due to NOAA station gaps.
  + This **reduces the reliability** of the dataset for trend analysis.
* **Data Consistency:**
  + The dataset structure was **consistent** (date, temperature, precipitation, wind speed),
  + However, **occasional missing values** required post-processing.
* **Metadata Quality:**
  + The metadata file **accurately described** the dataset and collection method.
  + However, **automatic versioning was missing**, meaning changes weren’t tracked over time.

**REFERENCES**

**National Centers for Environmental Information (NCEI).** (n.d.). *Climate Data Online (CDO) Web Services API v2.* National Oceanic and Atmospheric Administration (NOAA). Retrieved February 10, 2025, from <https://www.ncdc.noaa.gov/cdo-web/webservices/v2>​

**National Centers for Environmental Information (NCEI).** (n.d.). *Find a weather station: NOAA Climate Data Online.* National Oceanic and Atmospheric Administration (NOAA). Retrieved February 02, 2025, from <https://www.ncdc.noaa.gov/cdo-web/datatools/findstation>​

**Dublin Core Metadata Initiative (DCMI).** (n.d.). *Dublin Core Metadata Element Set, Version 1.1.* Retrieved February 10, 2025, from <https://www.dublincore.org/>​

**Reitz, K., & Banz, C.** (2023). *Requests: HTTP for Humans.* Python Software Foundation. Retrieved February 06, 2025, from <https://docs.python-requests.org/en/latest/>​

**The Pandas Development Team.** (2023). *pandas: Powerful Python Data Analysis Toolkit.* Python Software Foundation. Retrieved February 07, 2025, from <https://pandas.pydata.org/docs/>​

**The Python Software Foundation.** (2023). *Python JSON module documentation.* Retrieved February 10, 2025, from <https://docs.python.org/3/library/json.html>​

**The Matplotlib Development Team.** (2023). *Matplotlib: Visualization with Python.* Retrieved February 10, 2025, from <https://matplotlib.org/stable/index.html>​

**University of California, Berkeley Library.** (n.d.). *Best practices for data collection and storage.* Retrieved February 04, 2025, from <https://guides.lib.berkeley.edu/data-management>​

**DataONE.** (n.d.). *Guide to metadata for data repositories.* Retrieved February 10, 2025, from <https://www.dataone.org/metadata>​