



Environmental Analysis Program

Pomona College
Biogeochemistry Laboratory

SOP-L803

Data Entry and Database Management

Standard Operating Procedure

SOP Number	SOP-L803
Revision	0.1
Author(s)	Marc Los Huertos
Date	2026-06-26
Organization	EA Program, Pomona College

FOR INTERNAL USE ONLY

This document is the property of the Environmental Analysis Program at Pomona College.
Unauthorized distribution is not permitted.

Table of contents

1 Purpose	2
2 Scope	2
3 Responsibilities	2
4 Definitions	2
5 Safety and Hazards	3
6 Equipment and Materials	3
6.1 Software	3
6.2 File Storage	3
7 Procedure	3
7.1 File Naming Conventions	3
7.2 Directory Structure	4
7.3 Data Entry from Lab Notebooks	4
7.4 Variable Naming Conventions	4
7.5 Data Types and Formatting	5
7.6 Data Validation in R	5
7.7 Data Dictionaries (Metadata)	6
7.8 Version Control with Git	6
7.9 Database Management for Multi-Year Projects	6
8 Quality Control and Acceptance Criteria	7
9 Waste Disposal	7
10 References	7
11 Training Record	7
12 Appendix A: CSV Template	8
13 Appendix B: Recommended R Packages for Data Management	8

1. Purpose

This SOP describes standardized procedures for entering, organizing, validating, and managing research data in the EA Program. It covers the full data lifecycle from initial recording through archival storage, ensuring that all datasets are consistent, traceable, and reproducible (Broman & Woo, 2018; Wickham, 2014).

2. Scope

- **Applies to:** All tabular data generated in the EA Program laboratory and field campaigns, including instrument outputs, field observations, and manually recorded measurements
- **Does not apply to:** Spatial/GIS data (see [SOP-L804](#)), bioinformatics sequence data (see [SOP-L705](#)), or instrument-specific export procedures (see [SOP-L807](#))
- **Location:** EA Program laboratory (SGM 133) and RStudio Server

3. Responsibilities

Role	Responsibility
Lab PI	Approve data management plan; ensure compliance with funder requirements
Lab Manager	Maintain database templates; perform quarterly backups
Analyst / Student	Enter data following this SOP; run validation checks; document metadata
Data Reviewer	Verify data entries against lab notebooks before analysis

4. Definitions

- **Data dictionary:** A companion metadata file that describes each variable in a dataset, including the column name, description, units, data type, expected range, and source SOP; required for every EA Program dataset.
- **ISO 8601:** An international standard for representing dates and times as YYYY-MM-DD; the required date format for all EA Program data files because it sorts chronologically and is unambiguous across locales.
- **Missing value:** A data cell for which no measurement was made or the measurement was invalid; represented as blank or NA in EA Program datasets; never coded as 0 (which is a valid measurement).
- **Primary key:** A unique identifier for each row in a data table (e.g., `sample_id`); enables unambiguous linking between related tables and detection of duplicate entries.
- **Tidy data:** A data structure in which each variable has its own column, each observation has its own row, and each type of observational unit forms its own table (Wickham, 2014); the standard format for EA Program analytical datasets.

- **Version control:** A system that records changes to files over time so that specific versions can be recalled; GitHub/Git is the EA Program version control platform for all data, scripts, and SOPs.

5. Safety and Hazards

Data entry is a computational activity with no physical hazards. Ergonomic best practices (regular breaks, proper screen height) are encouraged for extended data entry sessions.

6. Equipment and Materials

6.1. Software

Item	Access	Notes
RStudio Server	rstudio.pomona.edu	Primary data processing tool
R (version 4.x)	Pre-installed on server	Required for validation scripts
Git / GitHub	github.com/marclos	Version control for all data files
Microsoft Excel / Google Sheets	Campus license	Acceptable for initial entry only
Text editor (VS Code, nano)	Various	For editing CSV files directly

6.2. File Storage

Location	Purpose	Backup Frequency
GitHub repository	Active project data	Every commit
College network drive	Archival storage	Semester end
External hard drive (lab)	Physical backup	Semester end

7. Procedure

7.1. File Naming Conventions

All data files must follow this naming pattern:

[project]_[datatype]_[YYYY-MM-DD].[ext]

For example: claremont_soilpH_2026-05-21.csv

Rules for file names:

1. Use lowercase letters only
2. Separate words with underscores (not spaces or hyphens)

3. Include the date of collection or generation in ISO 8601 format (YYYY-MM-DD)
4. Use `.csv` as the default file format for tabular data

 Tip

Never use spaces, special characters, or accented letters in file names. These cause problems across operating systems and with R's file-reading functions.

7.2. Directory Structure

Organize project data using a consistent directory tree:

```
project_name/
  data/
    raw/          # Unmodified instrument or field data
    processed/    # Cleaned, validated datasets
    metadata/     # Data dictionaries, codebooks
  analysis/
    scripts/      # R scripts and Quarto documents
    output/       # Figures, tables, reports
  README.md      # Project overview and data description
```

Raw data files in `data/raw/` must never be modified. All cleaning and transformation steps should produce new files in `data/processed/` with scripts that document every change (Wilson et al., 2017).

7.3. Data Entry from Lab Notebooks

1. Open the lab notebook to the relevant page
2. Create a new CSV file (or open the project template) in `data/raw/`
3. Enter column headers in the first row following the variable naming conventions below
4. Enter each observation as a separate row
5. Record the notebook page number, date, and analyst initials in the metadata file
6. Save and commit to Git with a descriptive message

7.4. Variable Naming Conventions

Column headers (variable names) must be:

1. Lowercase, with underscores separating words (e.g., `soil_ph`, `no3_mg_l`)
2. Free of spaces, special characters, and units in the name itself
3. Documented in an accompanying data dictionary (see Section 6.7)

Use consistent naming across related datasets. For example, always use `site_id` (not `Site`, `site`, `SiteID`, or `location`) when referring to the same sampling locations.

7.5. Data Types and Formatting

Data Type	Convention	Example
Dates	ISO 8601 (YYYY-MM-DD)	2026-05-21
Times	24-hour (HH:MM)	14:30
Missing values	Leave cell empty or use NA	NA
Numeric	No commas; use period for decimal	1234.56
Categorical	Consistent spelling; lowercase	upstream
Below detection	Flag column; numeric = MDL/2	See QC SOP

Common Mistakes

Never use 0 to represent a missing value. A zero is a valid measurement. Use NA or leave the cell blank. Never merge cells in a spreadsheet, and never use color coding as the sole carrier of information (Broman & Woo, 2018).

7.6. Data Validation in R

After entry, run the validation script to check for common errors. A minimal validation template:

```
library(tidyverse)

# Read the raw data
dat <- read_csv("data/raw/project_variable_2026-05-21.csv")

# --- Structure checks ---
glimpse(dat)                                # Column types correct?
stopifnot(nrow(dat) > 0)                    # File is not empty

# --- Missing data summary ---
dat %>%
  summarize(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "variable",
               values_to = "n_missing") %>%
  filter(n_missing > 0)

# --- Range checks ---
dat %>%
  summarize(
    ph_min = min(soil_ph, na.rm = TRUE),
    ph_max = max(soil_ph, na.rm = TRUE)
  )
# Are values within physically plausible ranges?

# --- Duplicate check ---
```

```

dat %>%
  group_by(site_id, date) %>%
  filter(n() > 1)
# Flag unexpected duplicates for review

# --- Categorical consistency ---
dat %>% count(site_id, sort = TRUE)
# Look for typos (e.g., "upststream" vs "upstream")

```

7.7. Data Dictionaries (Metadata)

Every dataset must have a companion data dictionary saved as `data/metadata/[project]_datadictionary.csv` with the following columns:

Column	Description
<code>variable</code>	Column name in the data file
<code>description</code>	Plain-language description
<code>units</code>	Units of measurement (or “categorical”, “text”)
<code>type</code>	Data type: numeric, integer, character, date, logical
<code>range</code>	Expected range or valid categories
<code>source_sop</code>	SOP number for the method that generated the data

7.8. Version Control with Git

All data files and scripts must be tracked in a Git repository hosted on GitHub. Follow the procedures in [SOP-L801](#) for committing and pushing changes. Key rules:

1. Commit raw data files when they are first created (and never modify them after the initial commit)
2. Commit processed data files after each cleaning or transformation step
3. Write informative commit messages (e.g., “Add soil pH data from Apr 15 sampling” rather than “update”)
4. Use branches for experimental analyses; merge to `main` when finalized

7.9. Database Management for Multi-Year Projects

For projects spanning multiple semesters or involving multiple analysts, maintain a relational data structure:

1. Create a **sites table** (`data/metadata/sites.csv`) with location identifiers, coordinates, and site descriptions
2. Create a **samples table** (`data/metadata/samples.csv`) linking each sample to a site, date, and collector
3. Link analytical results to samples using unique `sample_id` values

This structure prevents duplication, supports merging across semesters, and enables relational joins in R (Wickham, 2014):

```
sites <- read_csv("data/metadata/sites.csv")
samples <- read_csv("data/metadata/samples.csv")
results <- read_csv("data/processed/chemistry_results.csv")

# Join all tables
full_data <- results %>%
  left_join(samples, by = "sample_id") %>%
  left_join(sites, by = "site_id")
```

8. Quality Control and Acceptance Criteria

- All data entries must be verified against the original lab notebook by a second person (or by the analyst after a 24-hour waiting period)
- Validation scripts must pass with zero unexpected errors before data are used in analysis
- The data dictionary must be complete for every variable in the dataset
- Refer to [SOP-L802](#) for analytical QC requirements that apply to the underlying measurements

9. Waste Disposal

Not applicable (computational procedure).

10. References

- Broman, K. W., & Woo, K. H. (2018). Data organization in spreadsheets. *The American Statistician*, 72(1), 2–10. <https://doi.org/10.1080/00031305.2017.1375989>
- Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10), 1–23. <https://doi.org/10.18637/jss.v059.i10>
- Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLOS Computational Biology*, 13(6), e1005510. <https://doi.org/10.1371/journal.pcbi.1005510>

11. Training Record

Training is documented in the EA Program training log. Before working independently with project data, personnel must:

1. Read this SOP in full
2. Complete the RStudio and GitHub orientation ([SOP-L801](#))
3. Successfully enter, validate, and commit a practice dataset under supervision
4. Be signed off by the PI or lab manager

12. Appendix A: CSV Template

A minimal CSV template for a typical water chemistry dataset:

```
sample_id,site_id,date,time,analyst,ph,conductivity_us_cm,temp_c,notes
W001,site_a,2026-05-21,09:30,MLH,7.2,345,18.5,
W002,site_b,2026-05-21,10:15,MLH,6.8,412,19.1,slight turbidity
```

13. Appendix B: Recommended R Packages for Data Management

Package	Purpose
<code>tidyverse</code>	Data import, cleaning, and transformation
<code>janitor</code>	Consistent column names, tabulation
<code>validate</code>	Rule-based data validation
<code>pointblank</code>	Automated data quality reporting
<code>here</code>	Portable file paths across systems
<code>skimr</code>	Quick summary statistics