Data Management Plan

Washington Soil Health Initiative: State of the Soils Assessment

Jadey Ryan
Dani Gelardi

Last Update: 2024-06-14

Table of contents

Overview

View this data management plan as an online book at https://wa-department-of-agriculture.github.io/washi-dmp/.

The Washington Soil Health Initiative (WaSHI) is a partnership between the Washington State Department of Agriculture (WSDA), Washington State University (WSU), and the State Conservation Commission. WaSHI establishes a coordinated approach to healthy soil in Washington.

To date, nearly 1,000 soil samples and management surveys across 50 different cropping systems have been collected as a part of the State of the Soils Assessment (SOS). WSDA and WSU lead this project with support from staff, students, conservation districts, and agricultural professionals throughout Washington.

The State of the Soils Assessment has four primary goals:

Assess baseline soil health in Washington

Understand how climate, crop type, and management impact soil health

Develop cost-effective ways for producers to assess their own soil health

Develop crop-specific decision-support tools

Chapter outline

This Data Management Plan (DMP) is a living document to be continually reviewed and improved based on lessons learned, new information, and collaborator feedback. The header on the first page displays the date of the last update.

Chapter 1 describes what data management is, why it is crucial to achieve our data-driven goals, and how our data move through the data life cycle.

Chapter 2 describes the various data formats we collect and manage. ISO standards are also described for date and geospatial data.

Chapter 3 describes naming conventions, best practices, and examples for how we name folders and files.

Chapter 4 describes how we organize our folders into a hierarchical structure.

Chapter 5 describes where we store data, what our backup policies are, how we protect our raw data, and how we use version control.

Chapter 6 describes how we record each element of the data life cycle with project-level, dataset-level, and variable-level documents such as standard operating procedures, readme files, data dictionaries, etc.

Chapter 7 describes how data are generated, processed, and moved from start to finish. Processes and tasks are grouped by pre, during, and post field season.

Chapter 8 describes how we protect producer privacy; how our data fits into WaTech data categories; requirements and processes for maintaining confidentiality; and our data share agreement, public access policies, and our preferred acknowledgements.

Chapter 9 describes our recommended project structures, code-specific naming conventions, script structures, and code style.

> Links to shared drive folders and files
>
> This DMP includes many links to folders and files on the shared drive, which are only accessible to WSDA staff on the state network or remotely connected using VPN (Virtual Private Networking).

> GitHub links
>
> If you aren’t logged into a GitHub account that is part of the WSDA organization or has access to the `washi-sos` repository, GitHub links to specific scripts will take you to a `404 not found` error page.

Roles and responsibilities

To maximize the benefits of effective data management, all WaSHI personnel who interact with SOS data must familiarize themselves with this DMP.

The WSDA Data Scientist, supported by the Co-Principal Investigators (CoPIs), is responsible for providing guidance to WaSHI staff working with SOS data and ensuring the implementation of this DMP. The Data Scientist is also responsible for reviewing and updating this document annually, and as needed. Upon updates, the Data Scientist will distribute this document to WaSHI staff and commit the source code to the GitHub repository.

Current roles

Role	Affiliation	Name	Title
CoPI	WSDA	Dani Gelardi	Senior Soil Scientist
CoPI	WSU	Deirdre Griffin LaHue	Assistant Professor
Data Manager	WSDA	Jadey Ryan	Data Scientist
Data Stewards	WaSHI staff		

Staff turnover

When staff leave, they take their skills, institutional knowledge, and personal understanding of their file management with them. Proper offboarding is essential to ensure knowledge isn't lost, time isn't wasted trying to recreate workflows, and projects keep moving.

Before the employee leaves, the Senior Soil Scientist and Data Scientist ensure that:

- Folders and files are moved from the employee's personal drive to the shared drive. They are named and organized according to Chapter 3.
- Workflows and specific processes the employee was responsible for are well documented.
- Permission and ownership for the following are transferred to the appropriate remaining staff:
    - GitHub WSDA organization
    - ArcGIS data products and online groups
    - Database credentials
    - Box.com folders

More resources and offboarding checklists from Harvard Research Data Management can be found in our data-management shared drive.

Acknowledgements

This DMP was adapted from the R.J. Cook Agronomy Farm Long-Term Agroecological Research Site DMP (Carlson 2021), U.S. Fish and Wildlife Service data management life cycle (U.S. Fish & Wildlife Service 2023), Harvard Medical School Longwood Research Data Management DMP guidelines (Harvard Medical School 2023), and the Data Management in Large-Scale Education Research book (Lewis 2023).

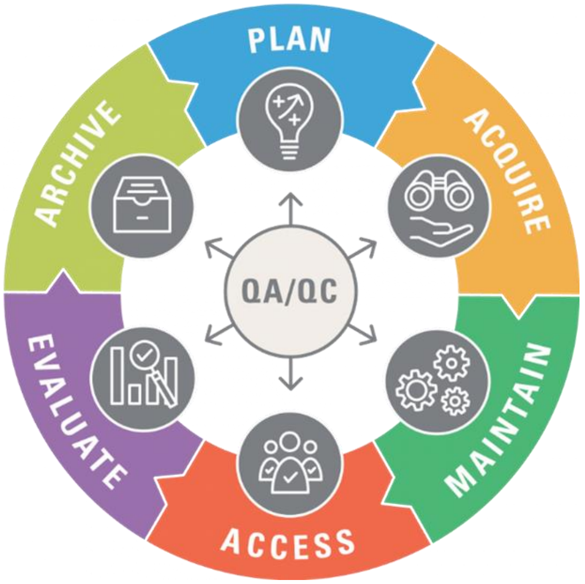
1. What is data management?

Effective data management involves properly documenting, storing, and sharing data and information derived from the data. If the data aren't usable by researchers, policymakers, or growers, then all the time, energy, and effort spent collecting and analyzing the samples may be wasted.

The guidelines detailed in this DMP help us achieve our data-driven goals, while also optimizing the value of the data by supporting information sharing and innovation. Our data management policies aim to implement **FAIR** (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable) principles while also maintaining data privacy (Wilkinson et al. 2016).

1.1 Data life cycle

This graphic explains the data life cycle (U.S. Fish & Wildlife Service 2023), in which each step requires care to ensure transparency, quality, and integrity.



Our adaptation is outlined below and the following chapters detail our internal processes and standards to follow throughout each step in the data life cycle.

Plan

Planning includes decisions about data acquisition, management, and quality control, as well as regular examinations of ways to improve. For example, each year we provide an updated spreadsheet template to Soiltest lab to ensure that measurements are reported with correct units and in the correct format. Special projects that deviate from our standard operating procedures require additional planning.

Acquire

We acquire data by collecting and analyzing new samples, deriving new insights from existing samples, or accepting datasets from collaborators.

Maintain

Maintenance involves processing data for aggregation, analyses, and reporting. We create metadata that facilitates interpretation of the data. We also store a copy of our data in a format that is accessible to our collaborators and future selves.

Access

Access refers to data storage, publication, and security. Raw and processed data with accompanying metadata should be stored, backed up, and available for information sharing with our partners. With PI approval, anonymized and aggregated data that does not compromise growers' personally identifiable information can be made publicly available in a data repository or data product/decision-support tool.

Evaluate

We evaluate data while processing and analyzing it to maximize accuracy and productivity, while minimizing costs associated with errors or tedious data cleaning labor. Evaluation workflows should be efficient, well-documented, and reproducible. Our evaluated data help us better understand how environmental factors and management decisions impact soil health.

Archive

Properly archiving our results supports the long-term storage and usefulness of our data. While similar to the Access element of the life cycle, archiving focuses on preserving data for historical and long-term access. For example, we archive each year's raw data for long-term storage and set those files to Read-Only.

Quality Assurance / Quality Control (QA/QC)

Data quality management prevents data defects that hinder our ability to apply data towards our science-based conservation efforts. Defects include incorrectly entered data, invalid data, and missing or lost data. QA/QC processes should be incorporated in every element of the data life cycle.

2. Formats & standards

2.1 Data formats

Data generated from or integrated into WaSHI can be non-digital or digital.

Non-digital data

Non-digital data, such as field forms, management surveys, and chain of custody forms, are manually recorded on paper forms. Paper forms must be transcribed or converted to digital file formats and then stored in the WaSHI filing cabinet in the Natural Resources Building in Olympia.

Digital data

Digital data include tabular, spatial, and binary data, such as lab results, sample locations, and field photos. Non-conventional data also include code, algorithms, tools, and workflows.

Tabular data include comma separated values (csv), tab separated values (tsv), Microsoft Excel open XML spreadsheet (xlsx), and portable document format (pdf).

Spatial data include file geodatabases (gdb), vector shapefiles (zipped folder containing multiple file extensions), keyhole markup language (kml or kmz). Tabular data may also contain spatial data such as longitude and latitude.

Binary data include photos (jpeg, png, gif, tiff), videos (mp4), code (R, py, js), and object-oriented data files (RDS, Rdata, parquet, arrow).

Proprietary data formats include Microsoft Excel, Word, and Powerpoint files (xlsx, docx, pptx). RDS and RData files are examples of application-specific data formats that can only be opened using the R programming language or RStudio IDE. These types of files should be saved in conjunction with a copy of the data in a non-proprietary and open-standard format, such as csv, to maintain accessibility for those who do not have Microsoft Office or do not use R.

Written documents and presentations are in formats including Microsoft Word and PowerPoint (docx and pptx), hypertext markup language (HTML), and pdf.

Notebooks combine text with executable code to generate written documents and presentations in docx, pptx, html, or pdf formats. These notebooks are stored in formats depending on the programming language: a few examples include R markdown (rmd), Quarto (qmd), and Jupyter notebook (ipynb).

The list below is not exhaustive and will continue to grow as additional data sources are discovered.

Type	Source	Formats
Lab results	Provided by an analytical lab, study PI, or grower	csv, xlsx, pdf, xml, json, RDS, RData
Management surveys	Collected through interviews with grower	csv, xlsx, RDS, RData, scanned paper form
Field forms	Completed in the field during/immediately after sampling	pdf, scanned paper form, csv, xlsx
Sample locations	Identified prior to sampling using ArcGIS Online and updated while sampling using ArcGIS Field Maps	ArcGIS feature layer, shp, kmz, csv, xlsx
Chain of custody forms	Completed prior to shipping or dropping off samples	pdf, scanned paper form
Climate data	OSU PRISM, NOAA, Esri Living Atlas	csv, shp, netCDF, tiff, gdb
Soil data	NRCS Web Soil Survey, NRCS WA gSSURGO	gdb, accdb
Images	Logos, icons, photos taken in the field	jpeg, png, gif, tiff, svg
Videos	Recordings of meetings, training videos	mp4
Documents	Reports, manuscripts, SOP, QAPP, factsheets, brochures	docx, txt, html, pdf
Presentations	PowerPoints, slide decks	pptx, html, pdf
Code	Scripts for wrangling and analyzing data; markdown for documents and presentations; style sheets for html	R, py, ipynb, js, yml, rmd, qmd, css, scss

2.2 Data standards

Date will be expressed as **YYYY-MM-DD** according to ISO 8601 standard.

Date with time will be expressed as YYYY-MM-DD**T**HH:MM:SS**Z**.

- **T** separates date from time.
- **Z** designates the time zone (Z or -HH:MM).
  - **Z** if using Universal Time Coordinated (UTC) with no offset.
  - Pacific Standard Time (PST) offset is **-8:00**. YYYY-MM-DD**T**HH:MM:SS**-8:00**
  - Pacific Daylight Time (PDT) offset is **-7:00**. YYYY-MM-DD**T**HH:MM:SS**-7:00**

Geospatial data will be accompanied by metadata that abides by the ISO 19115 standard and follows Esri's documentation when using ArcGIS Pro. Metadata contains information about the identification, extent, quality, spatial and temporal schema, spatial reference, and distribution of digital geographic data.

Code will follow the style guide in Chapter 9.

PUBLIC SERVICE ANNOUNCEMENT:
OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.
THIS IS THE CORRECT WAY TO WRITE NUMERIC DATES:
2013-02-27
THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:
02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. 27/2-13 2013.158904109
MMXIII-II-XXVII MMXIII LVII/CCCLXV 1330300800
((3+3)×(111+1)-1)×3/3-1/3³ 2013
10/11011/1101 02/27/20/13 01237
2-27-13
HISSsss

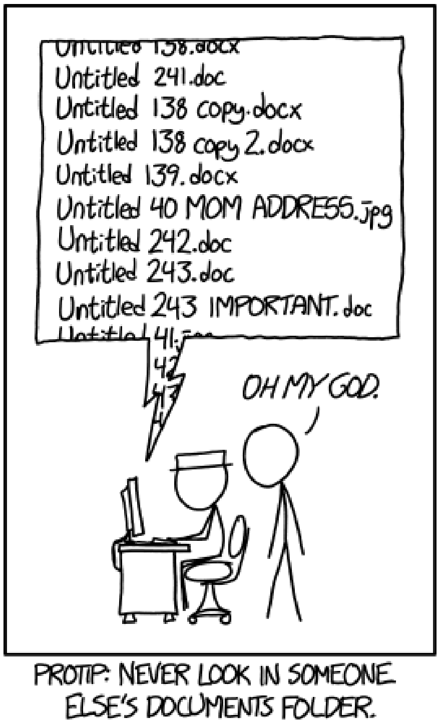
ISO 8601, Randall Munroe's xkcd

3. Naming conventions

When naming folders and files, use consistent and clear names that are findable and understandable by both humans and computers. A file name should convey what it contains and which file is the most recent version.

3.1 Why are conventions important?

- Improves consistency and predictability, making it easier to browse folders and know what they contain.
- Enables sorting files by date, conservation district, or another theme.
- Facilitates collaboration so all team members can find the information they need.
- Standardizes file paths and URLs for efficient programming and website hosting.
  - URLs and programming languages are case-sensitive. `WaSHI-data.csv` and `washi-data.csv` are completely different files.
  - URLs cannot have spaces in them. They must be escaped with this character entity `%20`. For example, `wasoilhealth.org/producer spotlights` would need to be `wasoilhealth.org/producer%20spotlights`.



Documents, Randall Munroe's xkcd

For more web-specific naming conventions, see this Learn the Web webpage.

3.2 Best practices

Some files and folders in our shared drive do not follow these best practices or naming conventions. We are learning and improving as we go.

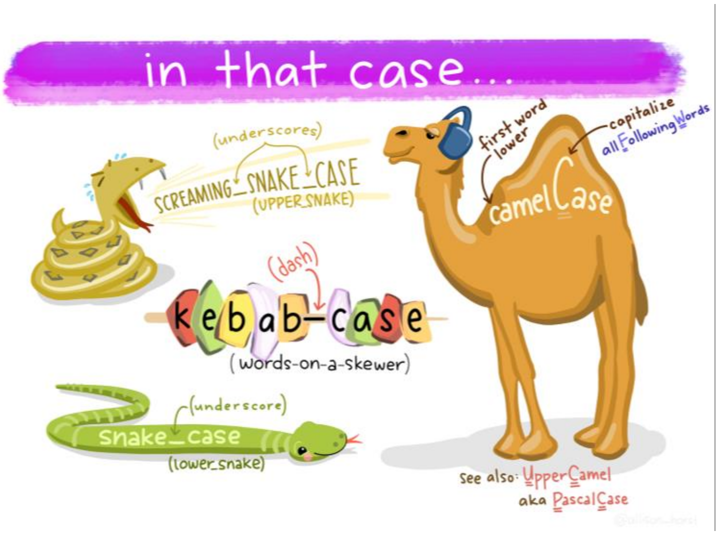
These are just guidelines. Because naming things is hard, we only ask that you try your best. If you're unsure about names or adding externally named files, the Data Scientist can support you.

See Section 3.3 for a table of examples of folder and file names following these best practices.

Meaningful name casing

Different conventions work for different purposes (folders and files versus programming objects).

- **kebab-case**: all lowercase with hyphens separating words. Use for **folders** and **files**.
- **snake_case**: all lowercase with underscores separating words. Only use for **column names** in spreadsheets and **code**, such as functions and variables in R. See Section 9.2.3.1 for example R errors when including hyphens in object names.



Artwork by Allison Horst

Delimiters convey meaning

Deliberately use underscores and hyphens so we can easily understand the contents and programmatically parse file and folder names.

- Use underscores to delineate metadata elements (i.e. date from name from version `date_name_version`).
- Use hyphens to separate parts of one metadata element (i.e. date `YYYY-MM-DD` or name `wsda-washi-presentation`).

No spaces or special characters

Avoid spaces and special characters (only use underscores and hyphens). Characters like `/` `()` `!` `?` `%` `+` `"` `'` have special meaning to computers and can break file paths and URLs.

Character length matters

Computers are unable to read file paths and file names that surpass a certain character length. Be concise AND descriptive. Omit prepositions and articles when possible. Abbreviate long words. The path limit on Windows is 260 characters.

‘Back to front’ date

Express date ‘back to front’ like `YYYY-MM-DD` according to the ISO 8601 standard. Left pad single-digit months and days with zeros to maintain chronological order of records when sorting alphanumerically.

✅ **Do this**	❌ **Don’t do this**
2020-05-28_agenda.pdf	2-14-2023_Agenda.pdf
2023-01-01_agenda.pdf	2023-Jan-1_Agenda.pdf
2023-02-14_agenda.pdf	Dec052020_Agenda.pdf
2023-12-05_agenda.pdf	May_28_2020_Agenda.pdf

Group & sort files by name

Consider how folders and files should be grouped and sorted, and include the appropriate metadata at the beginning of the file name. See examples below.

Sort by district	**Sort by date**
cowlitz_coc_2023-05-01.pdf	2023-05-01_cowlitz_coc.pdf
cowlitz_coc_2023-05-23.pdf	2023-05-01_cowlitz_tracking.pdf
cowlitz_tracking_2023-05-01.pdf	2023-05-09_ferry-cd_tracking.pdf
ferry-cd_coc_2023-05-10.pdf	2023-05-10_ferry-cd_coc.pdf
ferry-cd_coc_2023-05-17.pdf	2023-05-17_ferry-cd_coc.pdf
ferry-cd_coc_2023-06-06.pdf	2023-05-23_cowlitz_coc.pdf
ferry-cd_tracking_2023-05-09.pdf	2023-06-06_ferry-cd_coc.pdf

Version numbers

Including the date in the file name is one way to version a file. Alternatively, or in addition to, append a number. Consider how many possible versions there could be. If more than 10, use leading zeros so the numbers have the same length. v1 through v15 will not sort the same way as v01 through v15.

✅ **Do this**	❌ **Don’t do this**
sop_v01.pdf	SOP_v1.pdf
sop_v02.pdf	SOP_v10.pdf
… [v03 - v09]	SOP_v11.pdf
sop_v10.pdf	SOP_v2.pdf
sop_v11.pdf	… [v3 - v9]

Collaboration

Add your initials to the end of the file name when “saving as” a file that multiple people are working on (i.e., `2023_sop-soil-health-monitoring_lm-jr.docx`). This ensures a version is kept as a backup. Alternatively, use Track Changes if working in a MS Word document.

Literature and references

When saving journal articles, user guides, and other reference materials, use the convention `author_year_abbreviated-title`. Use underscores to separate different metadata.

✅ **Do this**	❌ **Don't do this**
lal_2004_soil-c-to-mitigate-cc.pdf	lal-2004-soil-c-to-mitigate-cc.pdf
clark-et-al_2020_pmn-sampling	clark-et-al_2020_pmn-sampling

Column names and code

Naming conventions for data column headers differ from folders and files. The hyphens in kebab-case cause errors in R and SQL code. Additionally, hyphens, spaces, and other special characters are invalid for ArcGIS table and field names.

Use **snake_case** for column names in spreadsheets and code objects (R vectors, lists, dataframes, and functions).

In variable or parameter names, include the measurement with the unit. This prevents unit confusion and reduces the risk of misinterpreting or inappropriately using the data.

Do not use special characters. Instead of `toc_%`, use `toc_percent`.

> ⓘ Note
>
> The code naming convention here applies primarily to R. Python and other programming languages have different conventions.

See Chapter 9 for more details in the code style guide.

3.3 Naming examples

	Naming convention	**Examples**
Folders	kebab-case	2024_sampling data-management
Files	kebab-case	2023-11-15_survey-perennial.xlsx 2024-03_washi-newsletter-wsda-sos.docx washi-logo-color.png 01_load-metadata.R 2024_producer-report.qmd
Column Names & Code	snake_case	sample_id pmn_lb_ac crop_summary assign_quality_codes()

4. Organization

Folders are organized into a hierarchical structure to clearly delineate project segments, improve searchability, and ensure reproducibility across time.

4.1 Folder structure

There is a delicate balance between deep and shallow folder structures. If too shallow, too many files in one folder are difficult to search. If too deep, too many clicks are required to find a specific file.

`Y:/NRAS/soil-health-initiative` is the parent folder for all WaSHI content.

The `state-of-the-soils` subfolder uses **date-** (each year has its own subfolder) and **categorical-** based (dataset and documentation that span across years) folder structures.

```
Y:/NRAS/soil-health-initiative/state-of-the-soils/
├── _complete-dataset
├── 2019_scbg
├── 2021_sampling
├── 2022_sampling
├── 2023_sampling
├── 2024_sampling
├── data-management
├── data-sharing
├── data-sources
├── maps
├── projects
├── qapp
├── sop
├── training-videos
├── equipment-inventory.xlsx
├── archived-sample-inventory.xlsx
└── sos-impacts.xlsx
```

Within each year subfolder, use sub-subfolders for planning, forms, data, and processes to maintain a reproducible workflow each year. See the `2023_sampling` folder tree for an example:

```
Y:/NRAS/soil-health-initiative/state-of-the-soils/2023_sampling/
├── applications
├── coc
├── equipment
├── field-forms
├── forms
├── gis
├── lab-data
├── labels
├── management-surveys
├── public-docs
├── purchases
├── reports
├── sample-id-assignments
└── scripts
```

4.2 Archive folders

When too many drafts or versions clutter a subfolder, create a new folder with the naming convention of `archive-folder-description`. Place the old drafts there. Leave the most current, accurate file in the main folder.

For example, the most recent sample labels for each conservation district are listed in the top level `completed-labels` folder, and previous working drafts were moved to the `archive-labels` folder.

```
Y:/NRAS/soil-health-initiative/state-of-the-soils/2023_sampling/labels/completed-labels
├── archive-labels
│   ├── cowlitz-county_labels.docx
│   ├── ferry-cd_labels.docx
│   ├── lewis-cd_labels.docx
│   └── stevens-cd_labels.docx
├── cowlitz-county_labels_v2.docx
├── ferry-cd_labels_v2.docx
├── ...
├── south-yakima-cd_labels.docx
├── stevens-cd_labels_v2.docx
└── walla-walla-cd_labels.docx
```

4.3 Code-based project organization

Code-based projects should be organized according to Section 9.1.1 in the code style guide.

5. Storage & version control

Non-digital data, such as paper forms, must be transcribed or converted to digital file formats and then stored in the WaSHI filing cabinet in the Natural Resources Building in Olympia.

All digital data are stored in the WSDA shared drives, and other locations listed below.

WSDA shared drives:

- Agency files: Y:/NRAS/soil-health-initiative
- GIS: K:/NRAS/Arc_Data/soil-health (access requires permissions from IT)

Esri products and services:

- ArcGIS Online Soil Health - WSDA Internal Group
- WSDA GIS on-premise ArcGIS REST Services Directory (only Jadey, Perry, and Joel can publish to this server; Ed Thompson is the contact for getting access)

Database for lab results and management data:

- WISKI, but very likely will migrate to SQL Server or a less water-focused database

GitHub organizations for code-based projects:

- WSDA
- WaSHI

Microsoft Teams for data sharing between WSDA and WSU:

- WSDA and WSU Teams WaSHI channels

Box.com for external file sharing:

- WSDA has a box.com account. The WSDA Senior Soil Scientist has an account and can add editors as needed.

Individual devices (laptop, tablet, phone):

- Must NOT be the only place data are stored!

5.1 Backup

Data must be stored in multiple locations. At minimum, data on an individual computer must also be saved on the WSDA shared drive. Backing up data using version control (GitHub) or a cloud service (Microsoft OneDrive or Box.com) is strongly recommended.

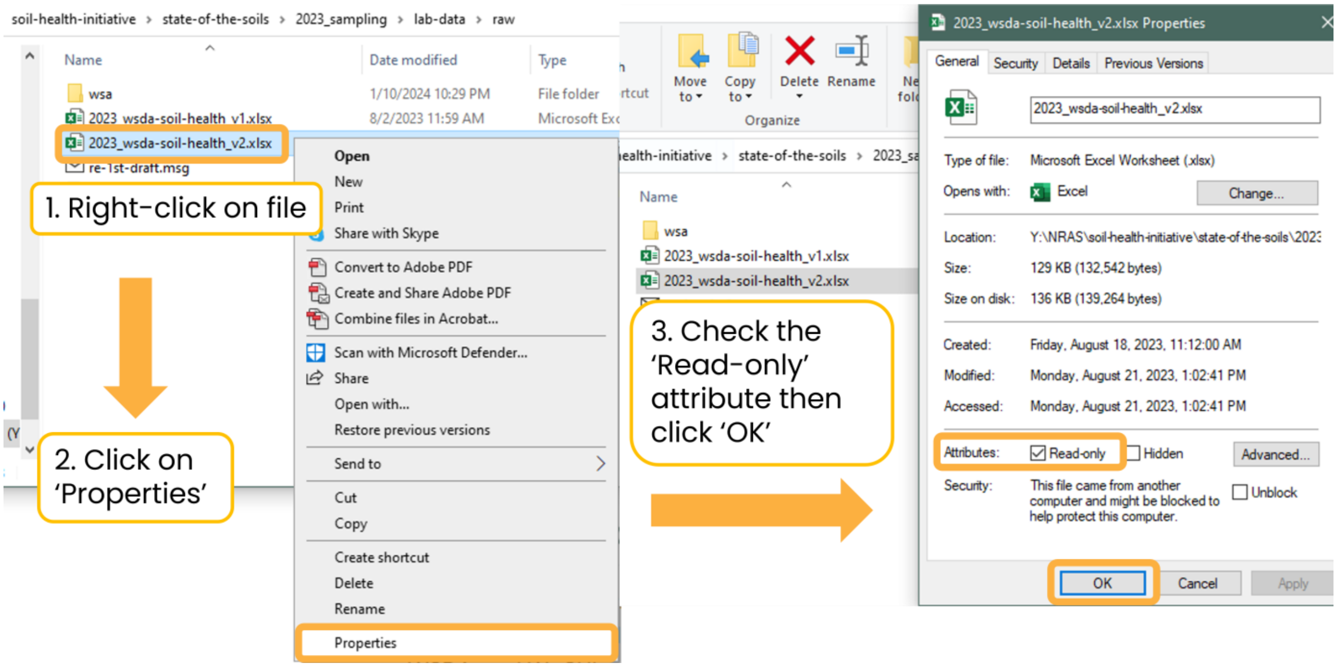
5.2 Read-only raw data

Always set raw data files, such as lab results or ArcGIS Online exports, as `Read-Only` to avoid accidental corruption or overwriting. For example, in the `lab-data` folder, all original data files are set to `Read-Only` and saved in the `raw` folder.

Copy the raw data file to the `working` folder for processing and analyses. Then save the final dataset in the separate `clean` folder with a descriptive title. Keeping a `readme.txt` to document processing steps is good practice, as discussed in Section 6.2.1.

```
Y:/NRAS/soil-health-initiative/state-of-the-soils/2023_sampling/lab-data
├── 2023_data-template-soiltest.xlsx
├── clean
├── qc
├── raw
└── working
```

To set a file as `Read-Only`: right-click the file > `Properties` > check the `Read-only` attribute box > `OK`.


soil-health-initiative > state-of-the-soils > 2023_sampling > lab-data > raw
Name
Date modified
Type
wsa
1/10/2024 10:29 PM
File folder
2023_wsda-soil-health_v1.xlsx
8/2/2023 11:59 AM
Microsoft Exc
2023_wsda-soil-health_v2.xlsx
re-1st-draft.msg
Open
New
Print
Share with Skype
Convert to Adobe PDF
Create and Share Adobe PDF
Combine files in Acrobat...
Scan with Microsoft Defender...
Share
Open with...
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Move to
Copy to
Delete
Rename
Organize
ealth-initiative > state-of-the-soils > 2023_sa
Name
wsa
2023_wsda-soil-health_v1.xlsx
2023_wsda-soil-health_v2.xlsx
2023_wsda-soil-health_v2.xlsx Properties
General
Security
Details
Previous Versions
Change...
Advanced...
OK
Cancel
Apply

1. Right-click on file

2. Click on 'Properties'

3. Check the 'Read-only' attribute then click 'OK'

2023_wsda-soil-health_v2.xlsx

Type of file: Microsoft Excel Worksheet (.xlsx)

Opens with: Excel

Location: Y:\NRAS\soil-health-initiative\state-of-the-soils\2023

Size: 129 KB (132,542 bytes)

Size on disk: 136 KB (139,264 bytes)

Created: Friday, August 18, 2023, 11:12:00 AM

Modified: Monday, August 21, 2023, 1:02:41 PM

Accessed: Monday, August 21, 2023, 1:02:41 PM

Attributes: ☑ Read-only ☐ Hidden

Security: This file came from another computer and might be blocked to help protect this computer. ☐ Unblock

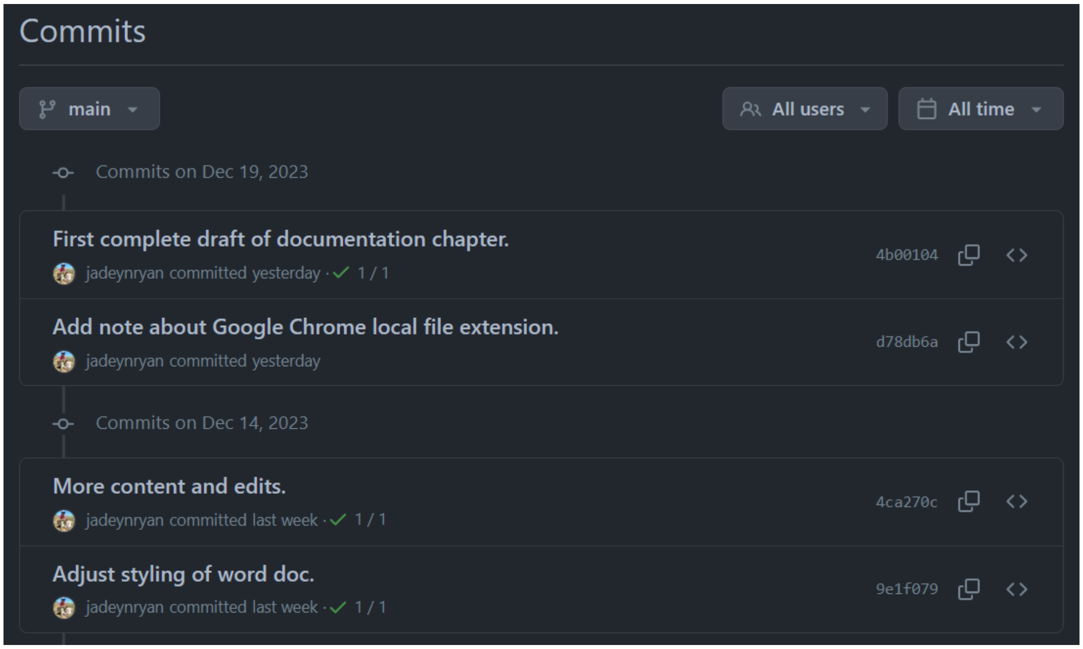
5.3 Version control with Git and GitHub

A version control system records changes to files over time. Git is a free and open-source distributed version control system. GitHub is the hosting site we use to interface with Git. Git and GitHub are fundamental to reproducible statistical and data scientific workflows (Bryan 2018).

Version control ensures changes are documented and previous versions are accessible if changes must be recalled. Additionally, version control enables robust collaboration across projects.

It's useful for not only code projects, but also for documents, presentations, and books (like this DMP!). Git and GitHub automatically save the revision history of each file, so there is only a single name for each file (e.g., `report.docx`) instead of `report_v01.docx` and `report_v02.docx`. For a reminder on version naming, see Section 3.2.7).

The screenshot below shows who made commits (i.e., named version histories) and when they were made. From this screen, a user can click on the commit message to view all files that were changed.



Commits

Commits on Dec 19, 2023

First complete draft of documentation chapter.
jadeynryan committed yesterday · ✓ 1 / 1 — 4b00104

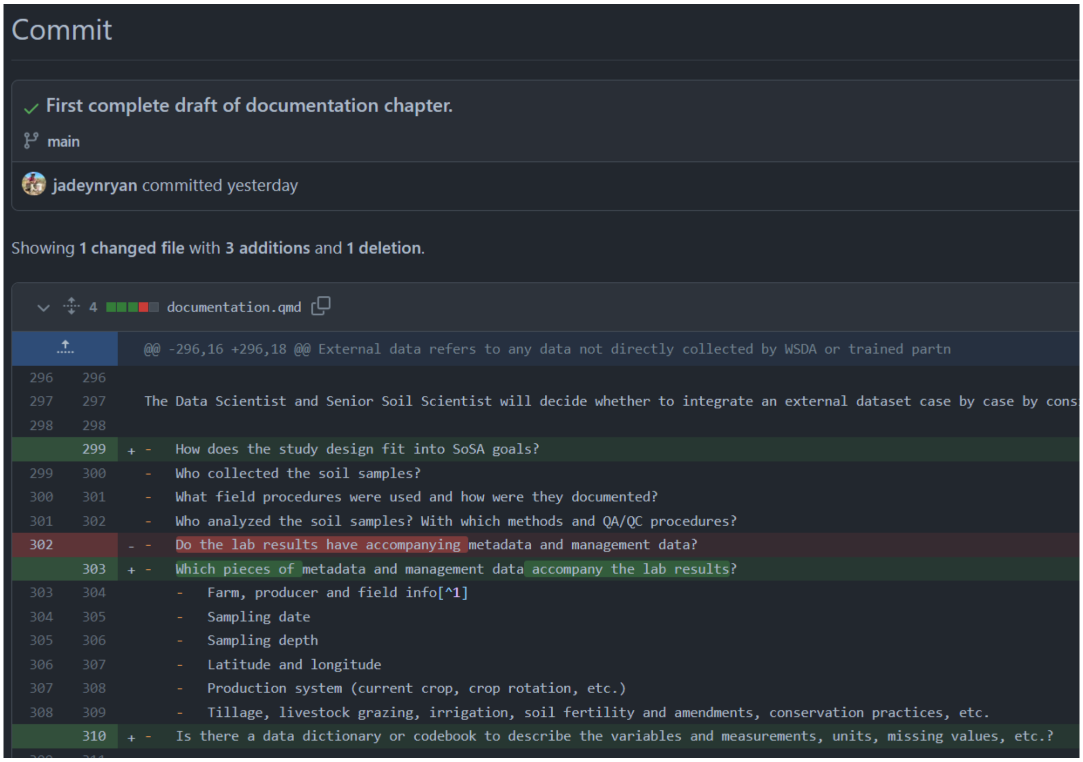
Add note about Google Chrome local file extension.
jadeynryan committed yesterday — d78db6a

Commits on Dec 14, 2023

More content and edits.
jadeynryan committed last week · ✓ 1 / 1 — 4ca270c

Adjust styling of word doc.
jadeynryan committed last week · ✓ 1 / 1 — 9e1f079

After clicking the first commit message, a `diff` (i.e., a visual of what changed) displays the additions to `documentation.qmd` highlighted in green and deletions highlighted in red.



Commit

✓ First complete draft of documentation chapter.

jadeynryan committed yesterday

Showing **1 changed file** with **3 additions** and **1 deletion**.

```
@@ -296,16 +296,18 @@ External data refers to any data not directly collected by WSDA or trained partn

   The Data Scientist and Senior Soil Scientist will decide whether to integrate an external dataset case by cons

+ -   How does the study design fit into SoSA goals?
  -   Who collected the soil samples?
  -   What field procedures were used and how were they documented?
  -   Who analyzed the soil samples? With which methods and QA/QC procedures?
- -   Do the lab results have accompanying metadata and management data?
+ -   Which pieces of metadata and management data accompany the lab results?
      -   Farm, producer and field info[^1]
      -   Sampling date
      -   Sampling depth
      -   Latitude and longitude
      -   Production system (current crop, crop rotation, etc.)
      -   Tillage, livestock grazing, irrigation, soil fertility and amendments, conservation practices, etc.
+ -   Is there a data dictionary or codebook to describe the variables and measurements, units, missing values, etc.?
```

Privacy considerations

Review Chapter 8 to categorize the data included in the repository to protect grower privacy. If the data are not anonymized and aggregated, either 1) the repository must be set to private or 2) data files and any scripts containing Category 3 data as described in Section 8.1.0.2 must be added to the .gitignore file.

Git and GitHub resources

Read Jenny Bryan's article *Excuse Me, Do You Have a Moment to Talk About Version Control* (open-access pre-print; full article on WSDA shared drive) for a background on Git and GitHub, why we should use it, and how to get started. For detailed instructions, follow along with her free online book *Happy Git and GitHub for the useR*.

GitHub: A Beginner's Guide is a helpful resource created by Birds Canada for less advanced programmers. If you prefer to look through slides, see Byron C. Jaeger's presentation *Happier version control with Git and GitHub (and RStudio)*.

6. Documentation

Documentation is the process of recording all aspects of project design; sampling; lab analyses; data cleaning; data analyses; data quality control and assurance procedures; and development of decision-support tools. Seem familiar? These are the steps of the data life cycle. Documentation helps to:

- standardize procedures
- enable reproducibility
- establish credibility
- ensure others (including our future selves) use and interpret data correctly
- provide searchability

All documentation (including this document) should be updated (and versioned) as procedures change and lessons are learned.

Samples collected by WSDA for the SOS must follow the procedures and standards described in the below documentation.

External data must have, at minimum, the documentation outlined in Section 6.4 to be integrated into the SOS dataset.

6.1 Project-level

Project-level documentation includes all descriptive information about the SOS dataset, as well as planning decisions and process documentation. Documentation includes **quality assurance project plans**, **standard operating procedures**, and other high-level documents (e.g.., request for proposals, applications, meeting agendas/notes).

Quality assurance project plan (QAPP)

The QAPP is the highest level of project documentation and covers everything from the project description; personnel roles and responsibilities; project timelines; data and measurement quality objectives; study design; and overviews of field, laboratory, and quality control.

Ours can be found in Y:/NRAS/soil-health-initiative/state-of-the-soils/qapp.

Standard operating procedures (SOP)

SOPs provide detailed instructions for field, lab, or data processing procedures and decision-making processes.

Ours can be found in Y:/NRAS/soil-health-initiative/state-of-the-soils/sop.

SOS sampling

The purpose of this SOP is to detail the procedures for a typical site visit in which soil samples are collected for physical, chemical, and biological soil health indicator analyses. Procedures include equipment preparation prior to sampling; best practices for filling out field forms; the selection of sampling locations; sampling protocols; sample handling and storage; and submitting samples to the lab. Following this SOP ensures data quality by creating audit trails and reminders to check that data are present, complete, and accurate. Additionally, this SOP will be used to maintain consistent sample collection procedures throughout the state for WSDA employees and partners.

Quality control / quality assurance (QA/QC)

This SOP outlines the process for screening sample metadata and lab results for completeness, consistency, and quality. Procedures involve subject matter expertise, investigation, communication with sampling teams and labs, algorithmic quality control, and tagging sample results with quality codes (listed in the below table). Data are then integrated into the statewide database.

Code	Tag	Description	Inclusion in analyses
0	Excellent	Met lab's and WSDA's QC criteria	Yes
100	Estimate	Interpolated missing value	Yes
110	Derived	Derived from an estimated value	Yes
120	Suspect	Z-score is ≥ |3|	Yes
130	Calculated ND	Calculated value using at least one ND	Yes
140	Non-detect	Below the method detection limit	No
160	Poor	Did not meet lab's QC criteria	No
180	Outlier	Outlier, designated by soil scientist	No
200	Unknown	External dataset	Yes
ND = non-detect			

6.2 Dataset-level

Dataset-level documentation applies to lab results, sample locations, grower information, and management data. **Readmes** and **changelogs** document what each dataset contains, how they are related, potential issues to be aware of, and any alterations made to the data.

Readme

`readme` files are plain text documents that contain information about the files in a folder, explanation of versioning, and instructions/metadata for data packages. These files are saved as `.txt`, instead of MS Word documents that take longer to open and can only be opened on computers with Microsoft installed. See below for examples of what to include.

Describe contents of folder

The readme.txt in the `complete-dataset` folder describes each files' structure, contents, and other pertinent information such as data sources.

Explain versions

The readme.txt in the `2023_sampling` > `lab-data` > `raw` folder explains why there are two different versions of the lab results and where to find additional information.

Provide instructions

Another readme.txt instructs how to use the files in the ArcGIS soil sample points box.com folder. When this folder is shared with partners, the readme helps orient them to the contents of the folder and modify the files as needed for their own project.

Changelog

Changelogs are also simple and concise plain text documents saved in a folder alongside data files that document changes to the dataset. For more information, see keepachangelog.com.

At the bare minimum, the `changelog.txt` contains:

- date of modification
- initials of who made the changes
- description of the changes

See the example changelog.txt in the `_complete-dataset` folder.

6.3 Variable-level

Variable-level documentation includes **data dictionaries**, which are tabular collections of names, definitions, and attributes about the variables in a dataset. Data dictionaries are ideally created in the planning phase of the project before data are collected.

Data dictionary

Each row is a different variable, and each column is a different attribute of that variable. With a data dictionary, a user should be able to properly interpret each variable in the data.

Our data-dictiosnary.xlsx in the _complete-dataset folder contains three tabs (`lab-results`, `sample-locations`, and `qc-codes`) that describe the attributes of each variable.

6.4 External data

External data refers to any data not directly collected by WSDA or trained partners (e.g., WSU or conservation districts) that follow our SOPs. These can include other studies pre-dating WaSHI, special soil health surveys, and publicly available datasets.

On a case-by-case basis, the Senior Soil Scientist and Data Scientist consider the following questions when deciding whether to integrate an external dataset:

- How does the study design fit into SOS goals?
- What field procedures were used and how were they documented?
- Who analyzed the soil samples? With which methods and QA/QC procedures?
- Are the following required metadata and management data available along with the lab results?
    - Farm, producer, and field info[1]
    - Sampling date
    - Sampling depth
    - Latitude and longitude
    - Production system (current crop, crop rotation, etc.)
    - Information concerning tillage, livestock grazing, irrigation, soil fertility and amendments, land use history, and/or conservation practices
- Is there a data dictionary or codebook describing the measurements, units, missing values, etc.?

Generally, external data should **1)** be well documented, **2)** be collected and analyzed by well-trained scientists and labs; and **3)** have adequate accompanying metadata and management data to facilitate interpretation of the results.

Some publicly available datasets to consider are in Y:/NRAS/soil-health-initiative/state-of-the-soils/external-data.

Intake form

External data may be provided in the External Data Intake spreadsheet, alongside related documents such as SOPs, management surveys, raw data files, etc.

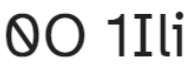
[1] Enough farm, producer and field info to distinguish unique farmers and fields for assigning unique IDs. They don't need to include personally identifiable information.

7. Data flow

This chapter outlines how data are generated, processed, and moved from start to finish.

7.1 Pre field season

When preparing sample ID assignments, labels, chain of custodies, and other materials, use an accessible font to reduce transcription errors. Atkinson Hyperlegible has very distinct alphanumeric characters, which improves legibility. Download it from Google Fonts.



0O 1Ili

"0O" and "1Ili" in Atkinson Hyperlegible

Assign unique identifiers

Before sample IDs can be assigned, collect the following information for each proposed sample:

- County
- Organization of sampling team
- Farm name (optional)
- Producer name
- Producer contact information (optional)
- Field name
- Crop
- General management practice (i.e., conventional, cover crop, reduced tillage)

View examples of the 2024 Sample Request Form sent to conservation districts and the Berries Sample Request Form used for a WSDA/WSU special project.

Once producers and fields have been identified, assign a unique ID for the producer, field, and sample with the following convention:

- **Producer ID**: first three letters of county + three-digit landowner number
  - `WHA001`
- **Field ID**: two-digit field number
  - `01` and `02`
- **Pair ID** (optional): letter extension added to paired fields
  - `A`
- **Sample ID**: last two digits of year + Producer ID + Field ID + Pair ID
  - `24-WHA001-01-A` and `24-WHA001-02-A`

The following counties have different abbreviations than their first three letters:

- Clallam → CLL
- Grays Harbor → GRY
- Kitsap → KIS
- Skamania → SKM

Match producer and field IDs to previous participants. Continue the sequence for new producers and fields. Producer IDs and sample IDs must not be duplicated.

For an example R script to automate this process, see assign-sample-ids.R.

Create sample labels

Sample label creation is automated using R and Microsoft Word's mail merge tool. Generate a spreadsheet with the information to be printed on the labels using labels.R. Then open labels-template-mail-merge.docx, select the spreadsheet as the recipient list, and run the mail merge to generate a word document with all labels to be printed (as shown in the completed-labels folder).

Create a data tracking sheet

Create a spreadsheet to track which data have been submitted for each sample, including:

- GPS points through the ArcGIS Field Maps field form
- Scanned paper field forms (for those without ArcGIS Field Maps)
- Management surveys through ArcGIS Survey123
- Scanned chain of custodies with shipping tracking numbers
- Location of archival falcon tubes (once retrieved by WSDA staff)
- Notes for if a sample will no longer be sampled, a sample ID was changed, etc.

See the 2023 spreadsheet for an example.

Develop ArcGIS web tools

Use ArcGIS to build tools for managing spatial data and collecting management survey data. In ArcGIS Pro, create a **sample selection feature layer** with domains for point numbers, bulk density, and crop types. Publish this feature layer to ArcGIS Online as a web map with a soil series layer. Then publish a second copy without the soil series layer and enable offline use. On ArcGIS Online, use Field Maps to configure the **field form** for the feature layer. **Management surveys** are created and hosted with Survey123 and Experience Builder. Schedule the ArcGIS Notebook with Python that backs up all data to run as a task every Monday, Wednesday, and Friday during the field season.

This template ArcGIS Pro project includes a readme.txt that describes this process.

View the code from the ArcGIS Notebook on the website version of this DMP.

7.2 During field season

Data collection in the field is detailed in the sampling SOP. Here, we focus on the behind-the scenes tasks for managing data.

Update data tracking spreadsheet

Throughout the season, update the data tracking spreadsheet as various forms, surveys, and correspondence are received, as described in Create a data tracking sheet.

Modify IDs when samples change

Sometimes a producer can no longer participate, or they need to change which field is sampled. Update, version, and archive the sample request form (`sample-request-form-ferry.xlsx` → `sample-request-form-ferry_v2.xlsx`). Run the `assign-sample-ids.R` script again to update the sample IDs. Lines 362 - 386 should be commented out as shown in the highlighted lines of the script on GitHub.

See 01_returned-sample-requests and 02_completed-sample-ids for an example of this flow.

Add a concise, explanatory note to the data tracking spreadsheet.

7.3 Post field season

Organize multiple sources of data

To unify the information from multiple data sources (e.g., sample request forms, ArcGIS Field Maps forms, and management surveys), cross-reference each source and reach out to the sampling teams to resolve conflicting information as needed. This is especially important for verifying the crop planted at the time of sampling.

See how to *mostly* automate this in: 01_load-metadata.R and 02_check-crops.R.

Process lab data

Follow the QA/QC SOP for processing lab data.

See the 2023 processing scripts and QA/QC report on GitHub:

- 03_process-spatial-data.R
- 04_load-lab-data.R
- 05_calculate-z-scores.R
- 2020-2023_qc-results-summary.qmd

Generate reports

Use the {soils} package to create a new project for each year. To avoid email attachment size limitations, save reports to Box.com for distribution to the sampling partners who send the reports to the participants. Access to this folder requires a share link provided by WSDA staff.

Save data to shared drive and WSU Teams channel

Copy the output data files and reports from Process lab data and Generate reports to the state-of-the-soils folder in its respective `year_sampling` folder. See Chapter 4 to review folder structure and organization.

Save the final datasets (in wide and long formats) and documentation (data dictionary, changelog, readme) to the WSU SCBG Soil Health Assessment Teams channel.

Archive jars and falcon tubes

Store the archival subsamples in glass jars in the Yakima WSDA storage room and the cryogenic archive subsamples in falcon tubes in the -80 °C freezer at the WSU Mount Vernon Northwestern Washington Research & Extension Center.

Tape the labels on the falcon tubes with a generous amount of packing tape to avoid falling off when they freeze.

Update the archive spreadsheet with the additional sample IDs, number of falcon tubes, and box number of the glass jar.

8. Data sharing

Our data sharing policies promote FAIR principles so our data are "**as open as possible, as closed as necessary**" (European Commission 2016). Data should be **open** enough to facilitate efficient re-use; avoid duplicating data collection efforts; enhance scholarly rigor; and promote engagement across the research and public communities (Whyte and Pryor 2011). However, data must also be **closed** enough to protect grower privacy and to honor agreements with growers and other researchers (Czarnecki and Jones 2022; Korzekwa 2023).

The SOS relies on growers' willingness to volunteer their fields for sampling and participate in the required management survey. Their willingness depends on their trust in WaSHI to protect their privacy. Only aggregated and anonymized results will be publicly available or shared. The below data privacy statement may be shared with potential participants.

Data privacy statement

> All survey responses will be combined across all respondents. Results will not be reported in a way that makes individuals identifiable. Information collected in this survey are subject to release in accordance with RCW 42.56 (Public Records Act).

Procedures for anonymizing data are detailed in Section 8.2.

8.1 WaTech data categorization

Under Washington State Policy 141.10 (Securing Information Technology Assets), state agencies must classify data into categories based on the sensitivity of the data. WaTech provides guidance on the four categories of data.

Category 4: "Confidential information requiring special handling"

WaTech: Data requires strict handling requirements applied by statues or regulations.

SOS: Not applicable.

Category 3: "Confidential information"

WaTech: Data includes "personal information" as defined in RCW 42.56.590 (Security Breaches) and RCW 19.255.010 (Personal Information Disclosure). An individual's first name or first initial and last name *in combination* with at least one of the following elements: social security number, driver's license or Washington identification card number, or any account numbers that permit access to their financial account.

SOS: We do not collect the above elements in combination with grower names. However, we treat individual names, farm names, and latitude and longitude coordinates as confidential information.

Category 2: “Sensitive information”

WaTech: Data are intended for official use only and withheld unless specifically requested.

SOS: This category includes lab results and management surveys. Access to this data requires a data share agreement.

Category 1: “Public information”

WaTech: Data is not covered in any of the above categories or is already released to the public.

SOS: De-identified and aggregated data, such as the number of soil samples and from which counties and crops they were collected, fall under this category. For example, the SOS dashboard is publicly available and the map zoom is disabled at the 1:1,600,000 scale (counties level).

8.2 Maintain confidentiality

Only under special circumstances and with proper justification in the data share agreement will the following Category 3 data be released to external collaborators. Under no circumstances will these data be made publicly available.

- Farm name
- Grower first and last name
- Field names that contain street names or other identifying information
- Latitude and longitude coordinates or other geospatial identifiers
- Any information identifying the individual farm or grower

Anonymize and aggregate Category 2 data to honor our data privacy statement by either removing or replacing Category 3 confidential information with dummy data. The {randomNames} R package can be used to replace real names with fake names. Round latitude and longitude to a precision that does not identify the farm or fields sampled.

See an example R script that anonymizes data on the website version of this DMP.

8.3 Data share agreement

SOS CoPIs created a Data Sharing and Scope of Work Agreement detailing the type of data to be shared, the scope of work in which the data may be used, and terms for using SOS data.

Once both CoPIs and the “Partnering Scientists” sign the agreement, save it in its own subfolder within Y:/NRAS/soil-health-initiative/state-of-the-soils/data-sharing. If this agreement is part of a grant, place a copy in its corresponding grant subfolder within Y:/NRAS/soil-health-initiative/contracts-grants/grants. Include relevant correspondence, code to subset the data, and final dataset in the agreement’s subfolder in the `data-sharing` folder. This documentation allows us to track publications, attributions, and the broader impact of the SOS dataset.

8.4 Public access

Currently, the only publicly available SOS data are the counts of samples across the project, counties, and crop types displayed in the ArcGIS Online Dashboard.

A small, anonymized subset is included as example data in the {washi} and {soils} R packages for demonstration purposes.

When the data are more mature and hosted in a proper database, we may publish an anonymized subset in a public repository such as:

- GitHub via an R package or Shiny app
- Zenodo: integrates with GitHub and is citable with a DOI
- Data.WA.gov: open data portal for the State of Washington

More inspiration for enhancing data discoverability and sharing across the agricultural and soil health communities include:

- USDA LTAR data dashboards
- CAF LTAR metadata tool

Chapter 16 of *Data Management in Large-Scale Education Research* discusses more considerations for data sharing and choosing public repositories (Lewis 2023).

8.5 Acknowledgments

All research and data partially or completely funded by WaSHI must include acknowledgements to the State of Washington. The following text should be included in all publications resulting from this funding:

> Data were in part provided by the Washington Soil Initiative, which is supported by the State of Washington and administered by the Washington State Department of Agriculture, Washington State Conservation Commission, and Washington State University.

If WaSHI staff make substantial scientific contributions to the manuscript, discuss the possibility of co-authorship credit.

9. Code style guide

This style guide adapts the *Tidyverse Style Guide* and incorporates best practices from *R for Data Science (2e)* (*R4DS*), *Data Management in Large-Scale Education Research*, and other resources.

All WaSHI staff who code in R should thoroughly read and consistently implement this style guide.

Using consistent project structures, naming conventions, script structures, and code style will improve code readability, analysis reproducibility, and ease of collaboration.

> Good coding style is like correct punctuation: you can manage without it, butitsuremakesthingseasiertoread…
>
> All style guides are fundamentally opinionated. Some decisions genuinely do make code easier to use (especially matching indenting to programming structure), but many decisions are arbitrary. The most important thing about a style guide is that it provides consistency, making code easier to write because you need to make fewer decisions.
>
> - Hadley Wickham in the Tidyverse Style Guide

9.1 Projects

Keep all files associated with a given project (input data, R scripts, analytical results, figures, reports) together in one directory. RStudio has built-in support for this through projects, which bundle all files in a portable, self-contained folder that can be moved around on your computer or on to other collaborators' computers without breaking file paths.

Create a GitHub repository and commit the project folder for version control as discussed in Section 5.3. If not in a GitHub repository, the folder must be copied onto the shared drive.

Learn more about projects in the Workflow: scripts and projects chapter of *R4DS*, in Jenny Bryan's article *Project-oriented workflow*, and Shannon Pileggi's workshop slides.

Project folder structure

A consistent and logical folder structure makes it easier for you (especially future you) and collaborators to make sense of the files and work you've done. Well documented projects also make it easier to resume a project after time away.

The below structure works most of the time and should be used as a starting point. However, different projects have different needs, so add and remove subfolders as needed.

- **root**: top-level project folder containing the `.Rproj` file
- **data**: contains raw and processed data files in subfolders. Raw data should be made *read-only* and not changed in any way. Review Section 5.2 for how to make a file read-only
- **output**: outputs from R scripts such as figures or tables
- **R**: R scripts containing data processing or function definitions

- **reports**: Quarto or RMarkdown files with the resulting reports
- **README**: markdown file (can be generated from Quarto or RMarkdown) explaining the project

R packages, such as {washi} and {soils}, contain additional subfolders and files:

- **inst**: additional files to be included with package installation such as `CITATION`, fonts, and Quarto templates.
- **man**: `.Rd` ("R documentation") files for each function generated from {roxygen2}.
- **vignettes**: long-form guides that go beyond function documentation and demonstrate a workflow to solve a particular problem
- **tests**: test files, usually using {testthat}
- **pkgdown** and **docs**: files and output if using {pkgdown} to build a website for the package
- **DESCRIPTION**: file containing package metadata (authors, current version, dependencies)
- **LICENSE**: file describing the package usage agreement
- **NAMESPACE**: file generated by {roxygen2} listing functions imported from other packages and functions exported from your package
- **NEWS.md**: file documenting user-facing changes

Learn more about other R package components in *R Packages (2e)*.

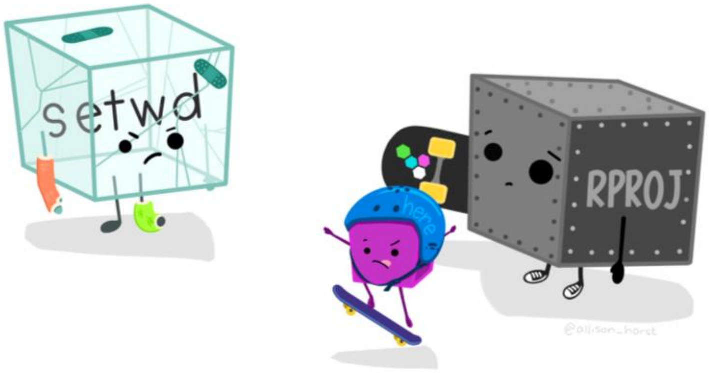
Absolute vs relative paths

> Note
>
> Directories and folders are used interchangeably here. If you're interested in the technical differences, **directories** contain folders and files to organize data at *different levels* while **folders** hold subfolders and files in a *single level.*

❌ Absolute paths start with the root directory and provide the full path to a specific file or folder like `C:\\Users\\jryan\\Documents\\R\\projects\\project-demo\\data\\processed`.[2] Run `getwd()` to see where the current working directory is and `setwd()` to set it a specific folder. However, a working directory set to an absolute folder path will break the code if the folder is moved or renamed.

✅ Instead, always use relative paths, which are *relative* to the working directory (i.e. the project's home) like `data/processed/data-clean.csv`. When working in an RStudio project, the default working directory is the **root** project directory (i.e.., where the `.Rproj` file is).

[2] Note the two backslashes. Windows paths use backslashes, which mean something specific in R. To make Windows paths with a backsplash work, replace them with two backslashes or use forward slashes.



Artwork by Allison Horst

{here} package

In combination with R projects, the {here} package builds *relative* file paths. This is especially important when rendering Quarto files because the default working directory is where the `.qmd` file lives. Using the above example project structure, running `read.csv("data/processed/data-clean.csv")` in `soil-health-report.qmd` errors because it looks for a data subfolder in the reports folder. Instead, use `here` to build a relative path from the project root with `read.csv(here::here("data", "processed", "data-clean.csv"))`. This takes care of the backslashes or forward slashes, so the relative path works with any operating system.

Artwork by Allison Horst

9.2 Naming conventions

> "There are only two hard things in Computer Science: cache invalidation and naming things."
>
> — Phil Karlton

Based on this quote, Indrajeet Patil developed a slide deck with detailed language-agnostic advice on naming things in computer science.

R code specific naming conventions are listed below. Python and other programming languages have different conventions.

Project folder, .RProj and GitHub repository

Name the project folder, .RProj file, and GitHub repository name the same. Be concise and descriptive. Use kebab-case.

Example: `washi-dmp` and `washi-dmp.RProj`.

Files

Be concise and descriptive. Avoid using special characters. Use kebab-case with underscores to separate different metadata groups (e.g., `date_good-name`).

Examples: `2024_producer-report.qmd`, `tables.R`, `create-soils.R`.

If files should be run in a particular order, prefix them with numbers. Left pad with zeros if there may be more than 10 files.

Example:

```
01_import.R
02_tidy.R
03_transform.R
04_visualize.R
```

Variables, objects, and functions

Variables are column headers in spreadsheets (that become column names in R dataframes), **objects** are data structures in R and ArcGIS (vectors, lists, dataframes, fields, tables), and **functions** are self-contained modules of code that accomplish a specific task.

Variable examples:

```
# Good
clay_percent
min_c_96hr_mg_c_kg_day
pmn_mg_kg

# Bad

# Uses special character
clay_%

# Less human readable, inconsistent with style guide,
# starts with number and will error in R
96hrminc_mgckgday

# Hyphen will need to be escaped in R code to avoid error
pmn-mgkg
```

Objects and functions

Objects names should be nouns, while function names should be verbs (Wickham 2022). Use lowercase letters, numbers, and underscores. Do not put a number as the first character of the name. Do not use hyphens. Do not use names of common functions or variables.

Object examples:

```
# Good
primary_color
data_2023

# Bad

# Less human readable, inconsistent with style guide
primarycolor

# Using a hyphen in an object name causes error
data-2023 <- read.csv("2023_data-clean.csv")
Error in data - 2023 <- read.csv("2023_data-clean.csv") :
could not find function "-<-"

# Starting an object name with a number also causes error
2023_data <- read.csv("2023_data-clean.csv")
Error: unexpected input in "2023_"

# Overwrites R shortcut for TRUE
T <- FALSE

# Overwrites c() R function
c <- 10
```

Function examples:

```
# Good
add_row()
assign_quality_codes()

# Bad

# Uses noun instead of verb
row_adder()

# Inconsistent with style guide
assignQualityCodes()

# Overwrites common base R function
mean()
```

9.3 R scripts

Header template

Headers in R scripts standardize the metadata elements at the beginning of your code and document its purpose. The following template and instructions are adapted from Dr. Timothy S Farewell's blog post (Farewell 2018).

1. **Script name**: meaningful and concise
2. **Purpose**: brief description of what the script aims to accomplish
3. **Author(s) and email**: name and contact if there are any questions
4. **Date created**: automatically filled in from the template
5. **Notes**: space for thoughts or to-do's

```r
## Header ======================================================================
## Script name: check-crops.R
##
## Purpose: Cross reference sample requests, Field Maps forms, and management
## surveys to get the correct crop planted at the time of sampling.
##
## Author: Jadey Ryan
##
## Email: jryan@agr.wa.gov
##
## Date created: 2024-01-02
##
## Notes:
##

# Attach packages ==============================================================

library(readxl)
library(writexl)
library(janitor)
library(dplyr)
library(tidyr)

# Load data ====================================================================
```

Add this template to RStudio using snippets:

1. Modify the below code with your name and preferred packages.
2. In RStudio, go to `Tools` > `Edit Code Snippets`.
3. Scroll to the bottom of the R code snippets and paste your modified code (the indent and tabs are important!).
4. Click `Save` and close the window.
5. Try opening a new blank `.R` script, typing "header", and then pressing `Shift` + `Tab`.

```
snippet header
    ## Header ======================================================================
    ##
    ## Script name:
    ##
    ## Purpose:
    ##
    ## Author: Jadey Ryan
    ##
    ## Email: jryan@agr.wa.gov
    ##
    ## Date created: `r paste(Sys.Date())`
    ##
    ## Notes:
    ##

    # Attach packages ==============================================================

    library(readxl)
    library(writexl)
    library(janitor)
    library(dplyr)
    library(tidyr)

    # Load data ====================================================================
```

Section template

The above header template also uses section breaks (e.g., commented lines with `=` that break up the script into easily readable chunks). Section breaks are a fantastic tool in RStudio because they allow you to easily show or hide blocks of code, see an outline of your script, and navigate through the source file. Read more about code folding and sections in this Posit article.

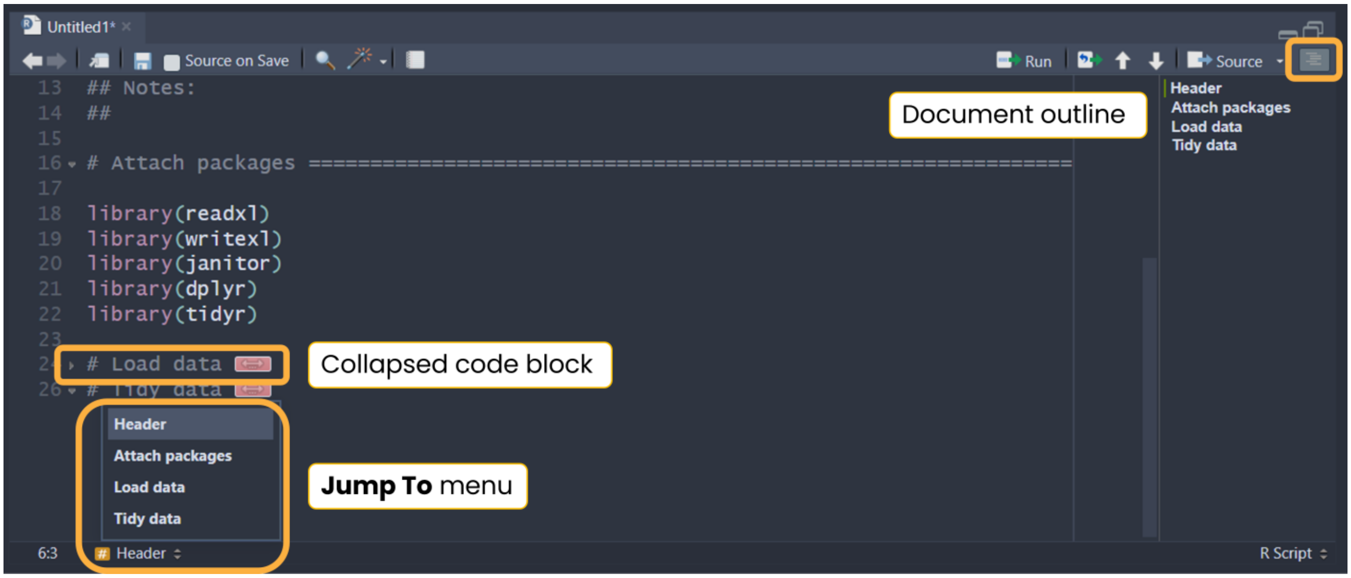
The snippet to create this section template that fills in the rest of the line with `=` was adapted from this stack overflow answer.

```
snippet end
    `r strrep("=", 84 - rstudioapi::primary_selection(rstudioapi::getActiveDocumentContext())$range$start[2])`
```

After adding the above code to your snippets, try creating a new section by typing “# Tidy data end” then pressing `Shift` + `Tab`.

`# Tidy data end<Shift+Tab>` results in:

```
# Tidy data ====================================================================
```



```
## Notes:
##

# Attach packages ===============================================================

library(readxl)
library(writexl)
library(janitor)
library(dplyr)
library(tidyr)

# Load data
# Tidy data
```

Document outline

Collapsed code block

Jump To menu

9.4 Code styling

Review the Syntax chapter of the *Tidyverse Style Guide* for details on spacing, function calls, long lines, semicolons, assignments, comments, and more. For the opinionated "most important parts of the Tidyverse Style Guide," skim through Chapter 4 Workflow: code style in *R4DS*. Instead of including each detail in this style guide and memorizing the content, use the {styler} package (as advised in *R4DS* Chapter 4).

{styler} includes an RStudio Addin that automatically formats code, making the style consistent across projects. We deviate slightly from the Tidyverse Style Guide and instead use {grkstyle}, an extension package developed by Garrick Aden-Buie, that handles line breaks inside function calls and indentation of function arguments differently. See the readme for examples.

Set up {styler} and {grkstyle}

Install {styler} and {grkstyle} with:

```
install.packages("styler")

options(repos = c(
    gadenbuie = "https://gadenbuie.r-universe.dev",
    getOption("repos")
))

# Download and install grkstyle in R
install.packages("grkstyle")
```

Set `grkstyle` as the default in {styler} functions and addins with:

```r
# Set default code style for {styler} functions
grkstyle::use_grk_style()
```

or add the following to your `~/.Rprofile`:

```r
options(styler.addins_style_transformer = "grkstyle::grk_style_transformer()")
```

Access your `.Rprofile` with `usethis::edit_r_profile()` to open the file in RStudio for editing. You may need to install the [{usethis}] package.

Use {styler} and {grkstyle}

Once installed, apply the style to `.R`, `.qmd`, and `.Rmd` files using the command palette, keyboard shortcut, or addins menu.

Command palette

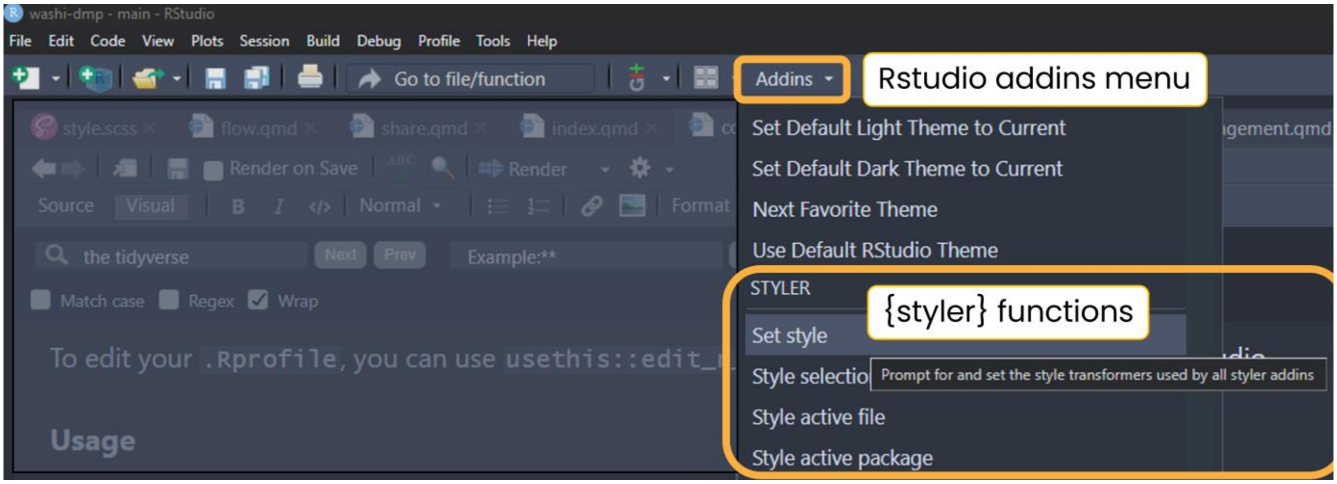
Use RStudio's command palette to quickly and easily access any RStudio command and keyboard shortcuts. Open the command palette with `Cmd/Ctrl + Shift + P` then type "styler" to see its available commands and shortcuts.

Keyboard shortcuts

Use `Cmd/Ctrl + Shift + A` to style the entire active file. We recommend styling the active file after finishing each code block or section. To style just a selection, use `Cmd/Ctrl + Alt + Shift + A`.

Addins menu

Use the addins menu in RStudio to style code by clicking a button to run the command.



Rstudio addins menu

{styler} functions

Prompt for and set the style transformers used by all styler addins

To edit your .Rprofile, you can use usethis::edit_r

Usage

References

Bryan, Jennifer. 2018. “Excuse Me, Do You Have a Moment to Talk about Version Control?” *The American Statistician* 72 (1): 20–27. https://doi.org/10.1080/00031305.2017.1399928.

Carlson, Bryan. 2021. “Data Management Plan for the R.J. Cook Agronomy Farm Long-Term Agroecological Research Site.”

Czarnecki, Joby M. Prince, and Mary Ann Jones. 2022. “The Problem with Open Geospatial Data for on-Farm Research.” *Agricultural & Environmental Letters* 7 (1): e20062. https://doi.org/10.1002/ael2.20062.

European Commission. 2016. “H2020 Programme Guidelines on FAIR Data Management in Horizon 2020.” https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf.

Farewell, Dr Timothy S. 2018. “My Easy r Script Header Template Tim Farewell.” https://timfarewell.co.uk/my-r-script-header-template/.

Harvard Medical School. 2023. “Data Management Plans.” https://datamanagement.hms.harvard.edu/plan-design/data-management-plans.

Korzekwa, Kaine. 2023. “Protecting Privacy While Making Data Open in Agricultural Research.” *CSA News* 68 (3): 6–10. https://doi.org/10.1002/csan.20979.

Lewis, Crystal. 2023. *Data Management in Large-Scale Education Research [in Preparation]*. https://datamgmtinedresearch.com/.

U.S. Fish & Wildlife Service. 2023. “Data Management Life Cycle.” https://www.fws.gov/data/life-cycle.

Whyte, Angus, and Graham Pryor. 2011. “Open Science in Practice: Researcher Perspectives and Participation.” *International Journal of Digital Curation* 6 (March): 199–213. https://doi.org/10.2218/ijdc.v6i1.182.

Wickham, Hadley. 2022. *The Tidyverse Style Guide*. https://style.tidyverse.org/index.html.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. “The FAIR Guiding Principles for Scientific Data Management and Stewardship.” *Scientific Data* 3 (1): 160018. https://doi.org/10.1038/sdata.2016.18.