

# **Oleson Lab Reseach Project Resource**

Alemarie Ceria

2024-01-30

## **Table of contents**

# Overview

This resource seeks to provide you with the following:

1. A project directory structure template which can be customized according to project needs
2. A list of tools

## **Interactive Code Annotations**

Throughout this resource, hover over the numbered annotations to the right of code chunks to reveal detailed explanations and comments about the code.

# Software and Platforms

## Download/Sign Up Links

Software/Platform/Purpose		Download/Sign Up Link
Slack	Enhances team communication and project coordination	<a href="#">Windows Link</a> <a href="#">Mac Link</a>
Google Drive for Desktop	Streamlines file storage, sharing, and collaboration	<a href="#">Link</a>
R	Provides tools for statistical analysis and data visualization	<a href="#">Link</a>
RStudio	Facilitates R coding, debugging, and project organization	<a href="#">Link</a>
Git	Manages version control for tracking and merging changes in code and documents	<a href="#">Link</a>
GitHub	Hosts and manages Git repositories, facilitating collaboration and code sharing	<a href="#">Link</a>
Zotero	Organizes and cites research sources consistently	<a href="#">Link</a>

## Setup Instructions

# Project Organization

## Naming Conventions

[https://www.youtube.com/watch?v=ES1LTlnpLMk&list=PLy\\_\\_EwS4oOnoQzWYaWDO\\_\\_x5t7O1RqHwuhw&index=5&pp=gAQBiAQB](https://www.youtube.com/watch?v=ES1LTlnpLMk&list=PLy__EwS4oOnoQzWYaWDO__x5t7O1RqHwuhw&index=5&pp=gAQBiAQB)

### TL-DR:

Best practices when naming directories and files:

1. Make it machine readable
  - Use globbing to isolate and find files that match a simple pattern using regular expressions in a scripting language
    - Eg., The following bash code will find all file names that start with `report-2020-` and ends with `.txt` `report-2020-*.txt`
  - No spaces or accented characters
  - No punctuation other than hyphens (-) and underscores (\_)
  - Use lowercase only
2. Make it human readable
  - Use highly informative slugs (e.g., `01_collect-cdec-snow-data.R`)
    - Should be descriptive and should accurately reflect the content
    - Should be concise
    - Use plain language, but still readable and understandable
    - Use hyphenation for spaces
    - Use underscores to separate metadata that needs to be parsed or processed in a scripting language using regular expressions
    - Avoid special characters
3. Sort it in a useful way
  - Put something numeric first
  - Logical sorting
    - Apply left padding (e.g., 01, 02, etc.) to single digit numbers in file names to maintain their sequential order

- Use when you have a predefined sequence or steps in a process (e.g., stages in a workflow)
- Chronological sorting
  - Follow the [ISO 8601](#) standard when formatting dates: YYYY-MM-DD (e.g., 2020-12-01\_draft.docx)
  - Use when the timeline of creation, modification, or relevance is important (e.g., In projects with regular updates or when you need to track progress or changes over time where the most recent date indicates the latest version)

Type	Naming Convention	Example
------	-------------------	---------

## Shared Google Drive

### Example Project Directory Structure

/g/path/to/example-project-directory	①
deliverables	②
papers	③
github-repo	④
example-project-github-repo	
meetings-and-events	⑤
meeting-notes	⑥
agendas	⑦
workshop-or-event-planning	⑧
presentations	⑨
project-materials	⑩
methods-drafts	⑪
lit-review	⑫
README.md	⑬

- ① Root directory housing all materials and documentation for the research project.
- ② Contains final outputs like reports, papers, or products resulting from the project.
- ③ Stores final versions and supplementary materials for academic papers or reports.
- ④ Specific GitHub repository directory for collaborative development and code sharing. *Only example-project-github-repo/ will be pushed to GitHub.*

- ⑤ Organizes documentation and planning materials for meetings and events related to the project.
- ⑥ Archives notes and decisions from project meetings.
- ⑦ Prepares and stores agendas for upcoming meetings to structure discussions.
- ⑧ Holds planning documents, schedules, and resources for workshops or project-related events.
- ⑨ Contains slides, speaker notes, and related materials for presentations about the project.
- ⑩ Stores various project-related documents not categorized elsewhere.
- ⑪ Keeps drafts and notes on methodological approaches and procedures.
- ⑫ Compiles literature reviews, reference materials, and bibliographies.
- ⑬ Provides an overview of the project directory, explaining the structure and contents of the folders/files.

## Create New Project Directory

The following code creates a directory for your project within the **current-projects/** directory in the **Oleson Lab/** Shared Google Drive by making a copy of an existing template.

To execute this code chunk, please do the following:

1. Ensure you have [Google Drive for Desktop](#) software downloaded, you are signed in, and are able to see the **Oleson Lab/** directory in your **Shared Drives/** directory
2. Change `#| eval: false` to `#| eval: true`
3. Input your desired project name in the code `project_name="example-project"`.

```
```{bash}
#| label: create-new-project-directory
#| code-overflow: wrap
#| eval: false # <1>

echo "Starting script..." # <2>

cd "/g/Shared drives/Oleson Lab/projects/project-management-resources" # <3>

new_project_template="/g/Shared drives/Oleson Lab/projects/project-management-resources/new-p
project_name="example-project" # <4>

new_current_project_directory="/g/Shared drives/Oleson Lab/projects/current-projects/$project

# Check if the directory already exists
if [ -d "$new_current_project_directory" ]; then # <6>
```

```

    echo "Error: A project with the name '$project_name' already exists. Please choose a dif
    exit 1 # <6>
fi # <6>

# Copy the new-project-template directory to the new project directory
cp -r "$new_project_template" "$new_current_project_directory" # <7>

echo "Completed copying." # <8>
```

```

- ① To run code chunk, set **false** to **true**.
- ② Confirms that the script is initiated.
- ③ Navigates to directory containing the new project template.
- ④ Defines a variable `new_project_template` which stores the `new-project-template/` directory file path.
- ⑤ Sets `project_name` to desired project name and constructs a path for the new project in the `current-projects` directory. **Replace [insert-project-name]**.
- ⑥ Checks to see whether or not the directory exists within the `current-projects/` directory and if so, prompts you to choose another name.
- ⑦ Copies everything within the `new-project-template/` directory into the `current-projects/` directory and renames it accordingly.
- ⑧ Confirms that the script ran successfully.

## GitHub Repository

### Example GitHub Repository Structure

|   |   |
|---|---|
| <code>/g/path/to/example-project</code> | ① |
| <code>outputs</code>                    | ② |
| <code>tables</code>                     | ③ |
| <code>plots</code>                      | ④ |
| <code>documentation</code>              | ⑤ |
| <code>code</code>                       | ⑥ |
| <code>functions</code>                  | ⑦ |
| <code>models</code>                     | ⑧ |
| <code>data</code>                       | ⑨ |
| <code>exploratory</code>                | ⑩ |
| <code>raw</code>                        | ⑪ |
| <code>tidied</code>                     | ⑫ |
| <code>output</code>                     | ⑬ |



|                       |    |
|-----------------------|----|
| LICENSE               | 14 |
| .gitignore            | 15 |
| example-project.Rproj | 16 |
| README.qmd            | 17 |
| README.md             | 18 |

- ① Serves as the root directory, encapsulating all project components for easy management and navigation.
- ② Contains generated files like tables and plots, separating results from the input data and code.
- ③ Stores tabular results and data summaries.
- ④ Holds visualizations and graphs generated by the analysis.
- ⑤ Stores project documentation, reports, and notes, centralizing information for reference and clarity.
- ⑥ Contains all the scripts and code used in the project, promoting modularity and code reuse.
- ⑦ Stores custom functions to ensure code modularity and readability.
- ⑧ Holds existing and output models.
- ⑨ Acts as the main repository for all datasets, organized to reflect different stages of data processing.
- ⑩ Contains initial explorations and analyses, fostering a sandbox environment for preliminary insights. (Will not be pushed to GitHub)
- ⑪ Stores unmodified raw data, preserving the original datasets for reproducibility and reference.
- ⑫ Holds processed and cleaned data, ready for analysis, ensuring consistency and reliability. (Will not be pushed to GitHub)
- ⑬ Stores the final dataset used for analysis.
- ⑭ Holds processed and cleaned data, ready for analysis, ensuring consistency and reliability.
- ⑮ Specifies the terms under which the project can be used or distributed, clarifying legal and usage aspects.
- ⑯ Lists files and directories to be ignored by version control, keeping the repository clean and relevant.
- ⑰ Provides a Quarto-rendered, detailed project overview and instructions, enhancing comprehension and usage.
- ⑱ Output of README.qmd used to be displayed on GitHub Repository.

## Data

**Code**

## **Reports and Publications**

# More Resources

DMP and SOP:

- [WA-Department-of-Agriculture / washi-dmp](#)
  - [Washington Soil Health Initiative: State of the Soils Assessment Data Management Plan](#)
- [emLab Standard Operating Procedures](#)
- [Data Management for Scientists by David LeBauer](#)

Documentation:

- [The Documentation System](#)
  - [What nobody tells you about documentation](#)
- [The University of Arizona Data Documentation Resource](#)
- [genophenoenvo/neon-datasets/README.md Example](#)

Videos:

- [Reproducible Research YouTube Playlist](#)