SHEAF Data Management Instructional Manual
Version 2.0 October 2019

Summary Overview

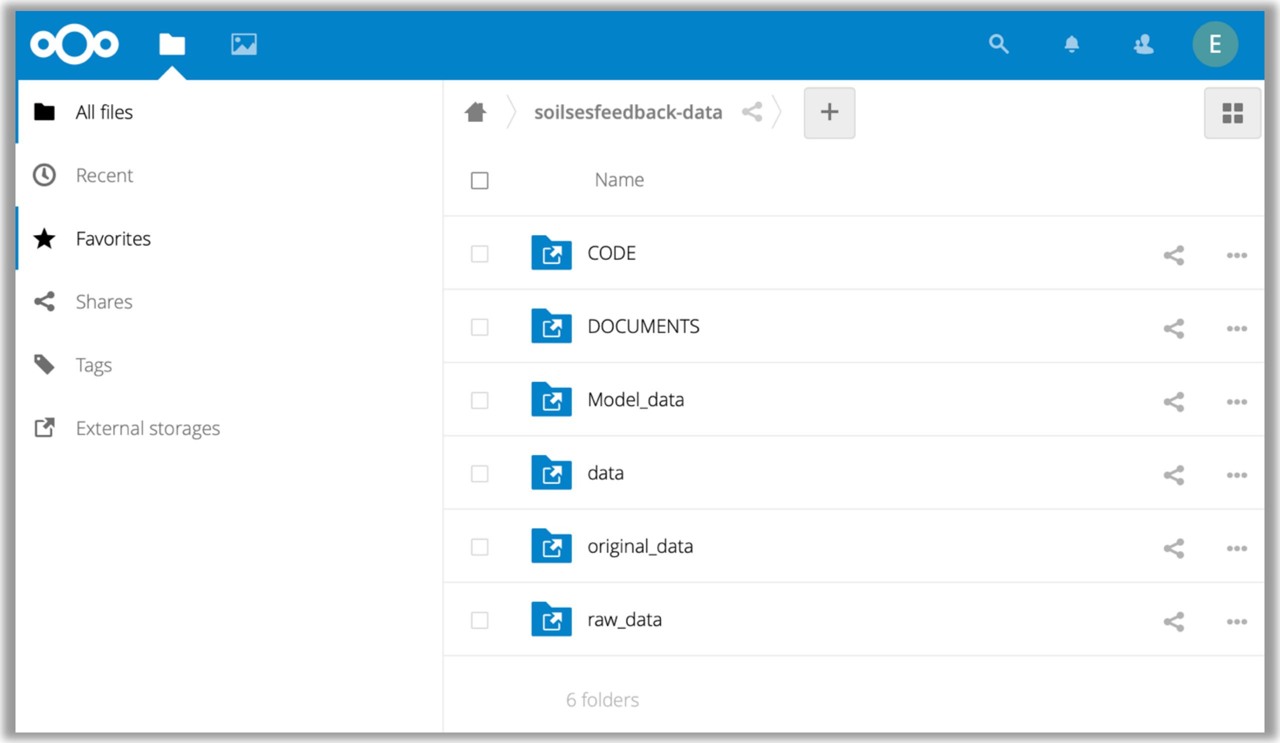
The SHEAF data management instructional manual is for defining the processes and steps for accessing, adding, analyzing, and modeling SHEAF and SHEAF-related data.

SHEAF web site and informational access

The SHEAF web site is the central location for accessing information related to the project as a whole (http://www.soilhealthfeedback.org). The web site has a basic structure that outlines the project premise, the team members, as well as links to access all of the data analysis and modeling components (file server, Rstudio server, gitlab code). The SHEAF web site will continue to be a central source for sharing information.

File Server Data Access

All data that is used as part of the SHEAF project is housed on SESYNC servers, at https://nextcloud.sesync.org. You must use your SESYNC credentials to access this server location (Figure 1).



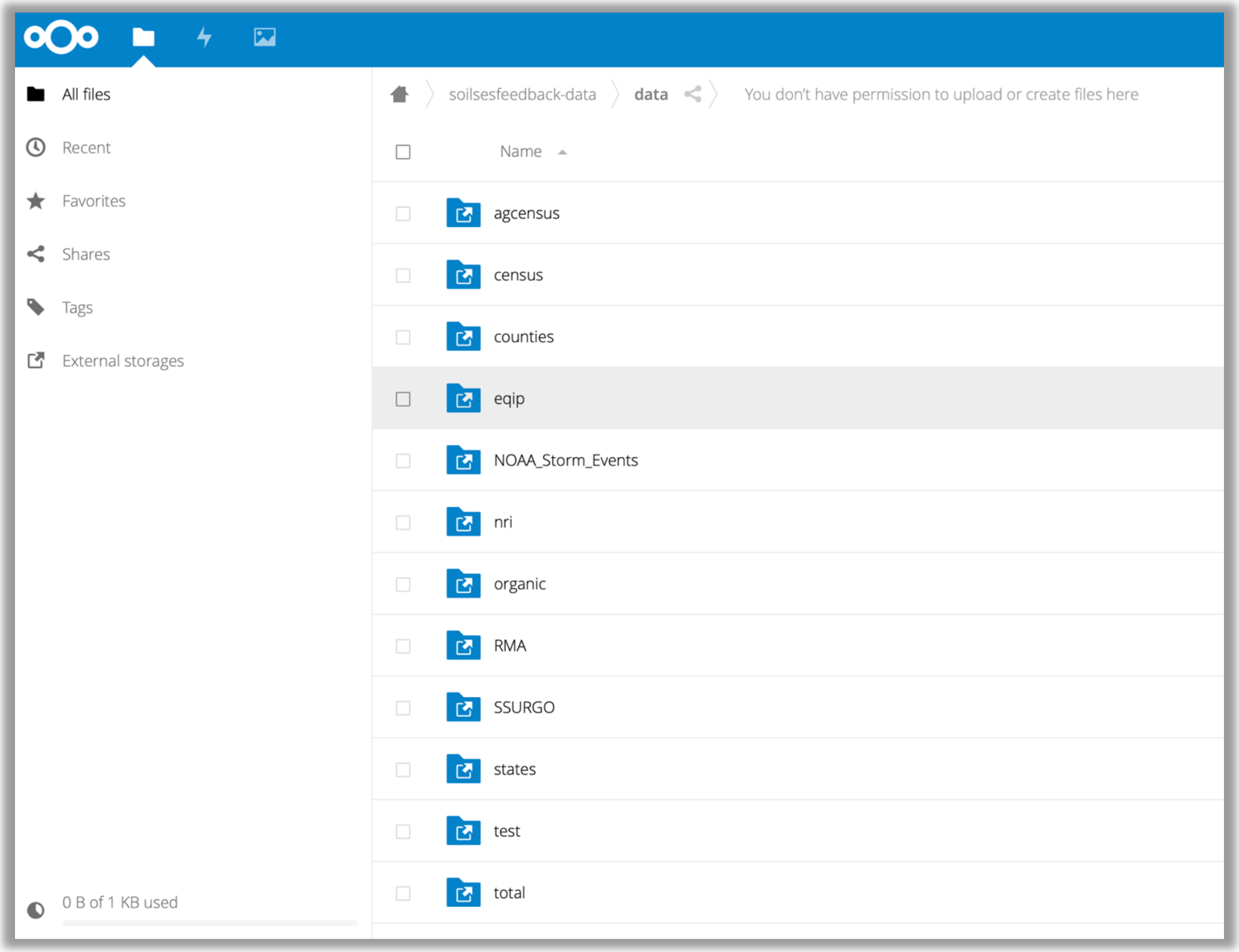
CODE
DOCUMENTS
Model_data
data
original_data
raw_data

6 folders

Figure 1. What you will generally see once you login to https://nextcloud.sesync.org

All data for use by the SHEAF team located in /soilsesfeedback-data. The data structure is as follows:

/data – contains our permanent data. This folder is READ ONLY. You cannot edit this folder (Figure 2).



soilsesfeedback-data › data

You don't have permission to upload or create files here

□ Name ▴

- agcensus
- census
- counties
- eqip
- NOAA_Storm_Events
- nri
- organic
- RMA
- SSURGO
- states
- test
- total

Figure 2. Current /data folder contents. Each dataset grouping has a folder with data within

/raw_data – folder for transitionary data. Data that is uploaded (see process below) will be moved to this folder.

/original_data – folder for original datasets that have not been altered. Data that is moved to the /raw data folder will eventually be placed here.

/model_data – folder that contains datasets generated by our data merging function. More on this topic under the Code section.

/CODE – folder where our code and RStudio projects are stored. THIS IS NOT THE LOCATION TO ACCESS THIS INFORMATION. We merely store our code here. The folders within our /CODE location are **Github Repositories**

/DOCUMENTS – this folder contains our foundational documents, such as our core mission, our initial grant document submittal, our logic model, and this Data Management Instructional Manual. This information is additionally located on our web site online (http://soilhealthfeedback.org).

Data Usage and Analysis

Data can be accessed by downloading directly from nextcloud, using the links provided within the nextcloud interface. However, if you want to access directly from R, use the download link, and add “download” to the end of the url.

Ex. Download EQIP data in R:

```
   eqip <- read.csv("https://nextcloud.sesync.org/index.php/s/os5ZxFXAAEgc2y4/download")
```

The result of this request provides you a data frame of the csv file eqip data for manipulation. You can do this with any of the datasets within the https://nextcloud.sesync.org file server.

Adding Data to SHEAF

If you have a dataset that you want to add to SHEAF, the process is as follows:

1. Access our Google data upload form. You will need to login to google to upload. https://docs.google.com/forms/d/1SnU_DS83yiSG3tORRbIHZPaBjhii4ypsNHhmdDPwBu4/
2. Fill out the form for each dataset to upload
3. Upload the dataset at the end of the form
4. After uploading the dataset, the file will go to google drive, and the metadata will be added to a google spreadsheet. The file will then be moved to the /raw data folder and altered if needed.
5. After alteration, the file will be moved to the permanent /data folder, with a copy of the original data to the /original data folder.

NOTE: If the above is too cumbersome, or if the file is too big, you may, as an alternative, upload the dataset directly to the /raw data folder, and communicate the metadata information by email to Erich.

Rstudio Server Access

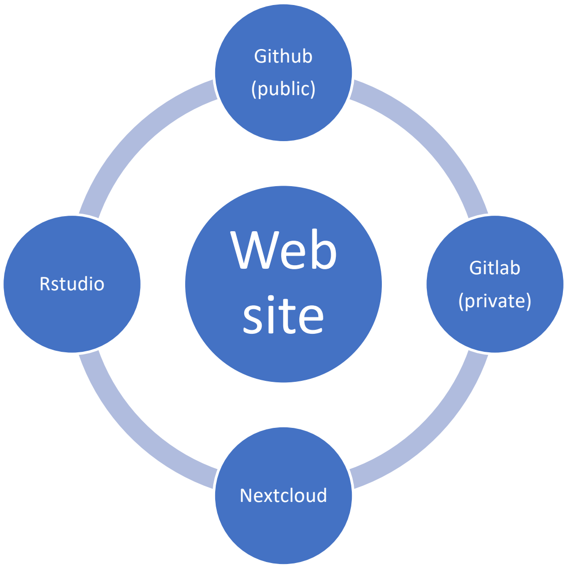
You can access the SESYNC Rstudio server from this url:

http://rstudio.sesync.org

You will need to use your SESYNC credentials. In order to use Rstudio with our code, you will need to pull code down from our Github repos first.

Code Usage: Github and Gitlab

As part of our SESYNC project, we use **Github** AND **Gitlab**.

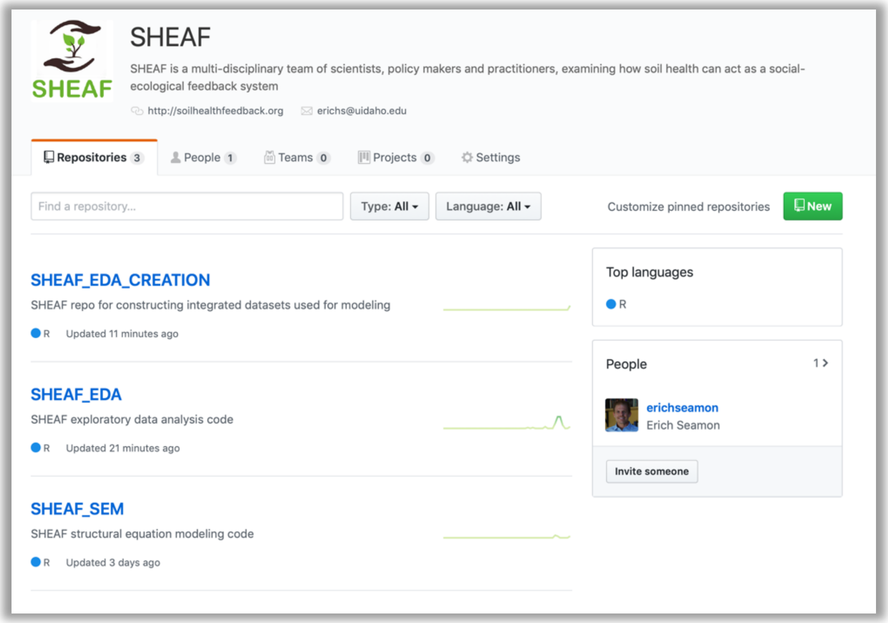


Github: a publically-accessible code repository at http://github.com. Many people store their code on Github, including teams. I have set up a team Github for use @ http://github.com/soilhealthfeedback.

Gitlab: Gitlab is software that allows SESYNC to have its' own private Github. Located at http://gitlab.sesync.org, and accessible using your SESYNC credentials (just like nextcloud), we can store code here and not expose it to the world.

One topic that needs more structure is WHAT code and data, if any, need to be securely and privately restricted to public use.

You will notice THREE repositories in our Github and Gitlab sites.



SHEAF

SHEAF is a multi-disciplinary team of scientists, policy makers and practitioners, examining how soil health can act as a social-ecological feedback system

http://soilhealthfeedback.org erichs@uidaho.edu

SHEAF_EDA_CREATION
SHEAF repo for constructing integrated datasets used for modeling
● R Updated 11 minutes ago

SHEAF_EDA
SHEAF exploratory data analysis code
● R Updated 21 minutes ago

SHEAF_SEM
SHEAF structural equation modeling code
● R Updated 3 days ago

Top languages
● R

People 1 >
erichseamon
Erich Seamon

1. **SHEAF_EDA**: This repo contains functions that individually examine a dataset. Want to make a graph or look at a map of data? You can use these functions, either on the SESYNC RStudio server, or locally on your own machine. They should work on your own computer, because they use a shared URL download of the data from SESYNC. You will need an internet connection. The **SHEAF-EDA** repo README has a description of how to use each function.

2. **SHEAF_EDA_CREATION**: This repo contains two functions, which MERGE data together into one data frame. Why do we do this? Well, if you want to run some analyses using all this data combined, based on spatial and temporal values (State, County, and Year), then you will ultimately need all that data combined. This function does that, and generates a file which is placed. The **SHEAF_EDA_CREATION** repo README has a description of how to use each function.

3. **SHEAF_SEM**: This repo contains functions which run a SEM model, using the output of the function from **SHEAF_EDA_CREATION**. Within the function, a user can choose to put the model together how they want to, using the available variables. The **SHEAF_SEM** repo README has a description of how to use each function.

The premise behind these three repositories are to allow for a researcher to:

1. Interrogate the data individually by dataset
2. Based on step one, to combine the data, and then
3. Model the data using SEM (or another modeling technique).

This approach will hopefully allow us to use these steps to iteratively, and in a collaborative fashion, explore the data and come up with the best modeling approach/path analysis that gets at the soil health question.

Operationally using the SHEAF code and data

Github works as a remote code respository (aka repo). It stores a base set of code and uses technology to sync code from multiple sources together. This is so a team can all work on the same code together. The way it works is – a user will initially 'clone' a repo to their local machine (a copy of the repository).

Then you work with the local repo copy. When you are finished, you can then 'push' your changes back to Github.

You can get the desktop version of github here: https://desktop.github.com

Or if you use Linux, Unix, or some derivation, you can use git from a command line.

If you are a total newbie to Github, then you will probably need some minimal training to be able to clone a copy of a repo and use it. During our second workshop, we will walk thru some of these steps.

Code Usage: Modeling

Our modeling approach uses two core packages: Lavaan and semPlot.