Data Management and Manipulation

Lecture 02.1: Data Management

Lauren Sullivan

Module: Data Management, Visualization & Reproducibility

Readings

Required for class:

- NA

Optional:

- Lind, E.M. (2016) Unified Data Management for Distributed Experiments: A Model for Collaborative Grassroots Scientific Networks. *Ecological Informatics.* 23:231-236.
- Hart et al. (2016) Ten Simple Rules for Digital Data Storage. *Plos Computational Biology.*
- Borer, E.T. et al. (2009) Some Simple Guidelines for Effective Data Management. *Bulletin of the Ecological Society of America.* 90(2):205-214.

Why Manage Data?

- Allows you to **quality control** your data more easily.
- Helps you stay **organized** through the whole process of file management, script creation, version control, backups, etc.
- Enables **reproducibility**. You always want to be able to recreate figures and analyses from the data that produced them. Even 3 years later.
- Helps you **share** your data more easily for future meta-analyses, etc. This allows for larger understanding of your field.

Organization is Key to Data & Project Management

For each R project/manuscript, you will want to have a set of folders. Here is a suggestion, but there are many options.

- Data
- Data Wrangling
- Analysis
- Graphics
- Documents
- ReadMe

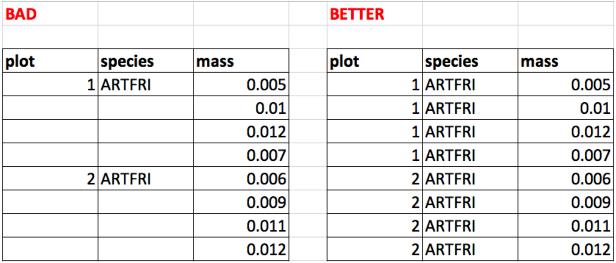
Folder Structure in Detail

- Data
    - raw data (read-only, pristine backup, not to be touched)
    - tidy data (intermediate and final R datasets)
    - Data Wrangling
        - DataAcquisition.R - script for compiling all data files into a single, usable dataset.
    - Analysis
    - Graphics
    - Documents
        - Manuscript folder
        - Literature folder
    - ReadMe
        - metadata
        - write down the driving questions and purposes of the project and other notes.

Note: Your code will stay cleaner if you use many smaller scripts, e.g. one for all analyses, one for all figures ***OR*** *one for each analysis and the associated figures*

Tips for Entering Raw Data for R Analysis

- No spaces in column headers
- Note units either in column header or in associated metadata
- R is case sensitive so keep column headers in a case structure (e.g. snake_case, dot.case)
- The difference between "0" and "NA" and a blank cell all tells you something
- Fill all columns



BAD

plot	species	mass
1	ARTFRI	0.005
		0.01
		0.012
		0.007
2	ARTFRI	0.006
		0.009
		0.011
		0.012

BETTER

plot	species	mass
1	ARTFRI	0.005
1	ARTFRI	0.01
1	ARTFRI	0.012
1	ARTFRI	0.007
2	ARTFRI	0.006
2	ARTFRI	0.009
2	ARTFRI	0.011
2	ARTFRI	0.012

Relational Data

When thinking about how to enter your data…

- Store data as smaller units (hierarchical and by type) and link through code.
- For example: store site level data in one file, and plot level data in another. Then link these datasets through code.
  - This helps avoid confusion and repetition
  - Great for large, interconnected datasets, especially those that can change
  - Data management systems play well with data in this format (e.g. Tidy, SQL)
  - Can be linked as 1:1, many to one (n:1), or one to many (1:n)

An Extreme Example of Relational Data

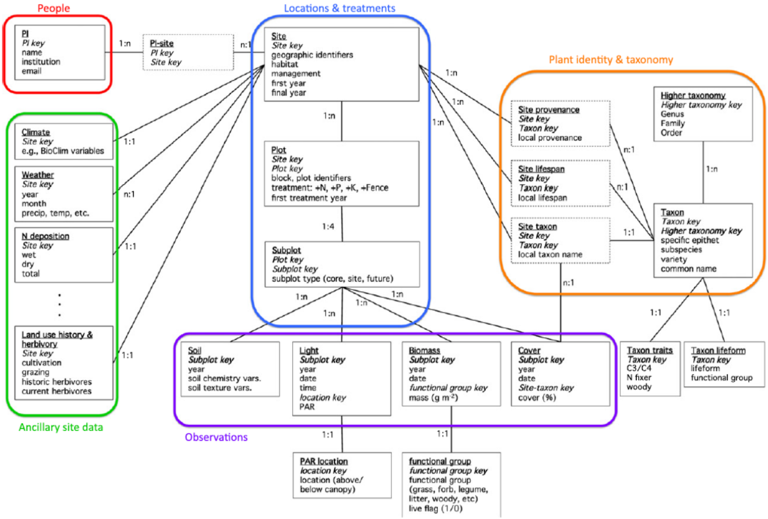


Figure 1: From Lind (2016) Ecological Informatics

Git and GitHub

GitHub is a great resource for managing data and code. If you are interested, there are lots of great resources out there. Here are a few.

- Perez-Riverol, Y. et al. (2016) Ten Simple Rules for Taking Advantage of Git and GitHub. *PLOS Computational Biology.*
- GitHub Guides