

Code Curation Exercise

For the final part of the workshop, we'll apply a code curation checklist to a real dataset. No specialized software is required but you will need to inspect source code files with a text editor. The default text editor that comes with your operating system (e.g., Notepad or TextEdit) should work just fine. You aren't expected to run the code.

Choose a Dataset

The datasets listed below have gone through various degrees of curation, however, there are always opportunities for improvement.

Bump, Joseph K; Beyer, Dean; O'Neil, Shawn. (2019). *Code, data, and metadata document for the manuscript: Territorial landscapes: incorporating density-dependence into wolf resource selection study designs*. Retrieved from the Data Repository for the University of Minnesota, <https://doi.org/10.13020/s40h-fv72>.

Review "code.txt" file and relevant documentation in the "doc_metadata.txt"

Olivia Graham, Tiffany Stephens, Brendan Rappazzo, Corinne Klohmann, Sukanya Dayal, Emily Adamczyk, Angeleen Olson, Margot Hessing-Lewis, Morgan Eisenlord, Bo Yang, Colleen Burge, Carla Gomes, Drew Harvell. (2022) *Data and code from: Deeper habitats and cooler temperatures moderate a climate-driven disease in an essential marine habitat* [dataset] Cornell University eCommons Repository. <https://doi.org/10.7298/6ybh-w566>

Review .R files in each of the project subdirectories.

Desharnais, Robert et al. (2023). *Data and MATLAB files for: Timescale analyses of fluctuations in coexisting populations of a native and invasive tree squirrel [Dataset]*. Dryad. <https://doi.org/10.5061/dryad.w6m905qqv>

Source code files are deposited separately: <https://zenodo.org/record/5753478>

Evaluate the dataset using the following checklist

Code Curation Checklist

- Are source code files named appropriately, with an extension that matches the programming language?
- Are hardware and software dependencies adequately documented?
 - The hardware, operating system, and package versions used to run the code should be documented.
- Does the documentation describe how to use the code?
 - The README should give an example of how to use the code, including expected outputs or behaviors.
- Are important functions and components documented with comments in the source code?
 - Critical pathways should include inline comments describing what the code does (e.g., expected inputs and outputs).
- Are file paths in the source code portable? Would running the code on another computer or operating system cause the paths to “break”?
 - Relative file paths are preferred to absolute file paths. Ideally, paths are constructed using software tools that account for differences between operating system path delimiters.
- Are files referenced in the source code included in the submission?
 - If not, the documentation should include instructions on obtaining the missing files.
- Is there a README and is it appropriately descriptive. Contents may include:
 - Description: the code’s purpose and behavior, including any usage examples.
 - Contributors: name, institutional affiliations, and contact
 - Technical Requirements: notes on hardware and software dependencies
 - Known Limitation & Bugs
 - License
 - Preferred Citation
 - Additional Acknowledgements
- Does the code include a license and is it appropriate?
 - The code should be licensed with an open source software license, preferably a permissive one (Apache, BSD-3, MIT).
- Are files in the submission organized in a consistent and logical manner?
 - The directory structure should help to differentiate, for example, source code files and data files.