R tools for a code-based data workflow

Contents

Webinar Information

Description

After a brief review of the advantages of a code-based workflow for ecological survey data, we introduce participants to some useful tools available via the R programming language for moving data along the data life cycle. We suggest some accessible tools in R for each step of the life cycle, and conclude with a walk through of how the functionality available in R can increase the reliability, efficiency, and transparency of scientific data management.

Presenters

- McCrea Cobb (Refuge Inventory and Monitoring Program, Alaska) and Adam Smith (Refuge Inventory and Monitoring Program, IR2/4)

When

June 24, 2020 (3:00-4:30 EST)

Location

Webinar link

Additional resources

GitHub repository

Outline

Introduction (***McCrea, 10 min***)

- Data life cycle review
- Manual versus code-based workflow
    - The manual data workflow
        - Example
        - Limitations
    - The code-based data workflow
        - Advantages
            - Documented
            - Reproducible
            - Replicable
            - More efficient
            - Less error-prone

Planning (***McCrea, 10 min***)

- Make an R project self contained and portable
    - File directory structure
    - Relative paths
- Dependency management
    - packrat
    - containers (docker)
- Standardize file naming convention
- Organizing R files (Numeric preface in the names of ordered scripts)
- Recommended RStudio settings
    - E.g., Uncheck “restore .RData into workspace at startup”
- Version control
    - Storing versions
    - Collaboration

Documenting (***Adam, 10 min***)

- Commenting
- Informative naming
- Explicit dependencies
- Modular code/functions
- Automated documentation

Acquiring (***Adam, 10 min***)

- Getting data into R
    - flat files
    - relational databases
    - web APIs
        - AGOL example

Processing (*Adam, 10 min*)

- Data manipulation
- Tidy data
- QA/QC
  - Data profiling
  - Assertive programming
- EDA/Visualization

Analyzing (*Adam, 5 min*)

- CRAN task views

Sharing (*McCrea, 10 min*)

- Reporting
  - RMarkdown
    - Bat reporting for mobile aucistics
  - Dashboards
  - COVID 19 example
  - Shiny apps
    - collarviewer
    - power analysis for butterfly surveys

Archiving (*McCrea, 5 min*)

- Saving results to ServCat or some other data repository

An example R project / Live demo (*10 min*)

Questions (*10 min*)

Resources (Links)

Introduction to R

- An Introduction to R book
- R for Excel Users

Resources for Teaching R

- DataCamp's tidyverse course
- learnr package
- RStudio teaching resources
- Data Wrangling, Exploration and Analysis with R "STAT 545"

R Resources

- Why learn R
- What they forgot to teach you about R
- R cheatsheets
- Project-oriented workflow

Style Guides - Tidyverse style guide - DataNovia R style guide

R Packages

- Packaging your reproducible analysis
- R packages

- Packaging data analytical work reproducibly using R (and friends)

Project management

- Stop working directory insanity!
- A minimal project tree in R
- Organizing the project directory
- Designing projects
- Project management with RStudio
- File structure for data management
- Organizing files for data analysis
- A meaningful file structure for R projects
- An introduction to Docker for R users

Project Directory Templates

- MakeProject package
- rrtools package
- prodigenr package

General Coding Best Practices

- What's in a name? The concepts and language of replication and reproducibility
- Best practices for scientific computing
- Good enough practices in scientific computing
- Ten simple rules for documenting scientific software
- Art of README - see examples and checklist
- Introduction to `roxygen2` vignette

Version Control

- Happy Git with R

Other

- How to share your data with a statistician
- Tools for reproducible research
- Reproducibility vs. replicability: a brief history of a confused terminology