Intermediate R programming I

Objective

- Understand the utility of writing your own functions and managing control flow.
- Write functions and if/else statements to improve code readability and reusability
- Create pull requests updating a git repo with new code.

Lesson outline

- Review from last session
- Warm-up
  - Update working branch from upstream
- Slides/lecture: Why write functions?
  - Base on DC semester biology lesson
  - Understandable and reusable code
    - Human-understandable chunks
    - Designed to be reused
- Live-coding: functions
  - Function basics from DC semester lesson
  - Create new R script for functions content, and another new R script for control flow (`ifelse` and `for` loops) content. Put both scripts into `repro-DS-workshop` repo
  - Discuss ordering of an R script
- Slides/lecture: control flow
  - Examples of use of if/else statements (find some real-world examples)
  - DC semester on conditionals
  - Review conditional logic
  - Conditionals outside functions
  - Conditionals inside functions
- Live-coding: practice commit + PR
- Discussion: application to own work + showcase
- Homework:
  - Create a plan for applying these skills to their own research projects. Can apply different parts to different projects, or more integratively to one single project. Some possibilities:
    - Turn the project folder for one of your current research project into a git/GitHub repo and/or R project
    - Modify an R script to use functions, if/for loops, or format code according to best practices
    - Improve file and folder organization for project or your entire computer!
  - Find a section of own code to turn into a function

Installation & materials

1. Slides
2. Data Carpentry for Biologists functions lecture
3. Data Carpentry for Biologists conditionals lecture

Notes from before

Review

Ask to share cleaning up of example script, their own script, or their research project folder from lesson 5.

Modifications

- For functions lecture, had them do only the "Use and Modify" exercise
- From conditionals lecture, only did "if statements" section
- From latter, only did "Basic If Statements" #2 exercise
- Added on brief explanation of `ifelse`, using the example of `ifelse(length == 5, "correct", "incorrect")`
- Mention `case_when` from dplyr for more complicated if & else steps

Teaching notes

- Before doing functions and conditions, demonstrate updating local `cct-organization` repo from upstream after adding a commit to the upstream
- Create new R script for functions content, and another new R script for control flow (`ifelse` and `for` loops) content. Put both scripts into `repro-DS-workshop` repo
- Mention functions are often used with for loops or apply statements
- After functions, mention consistent order of sections in scripts; example is libraries, read in data, functions, executing functions
- Include real world examples of `if` and `ifelse` statements to motivate their use, especially for checking if files already exist
- Demonstrate making a new branch and opening up a pull request in their `repro-DS-workshop` repo at the end of each set of material; can merge pull request

Homework

Create a plan for applying these skills to their own research projects. Can apply different parts to different projects, or more integratively to one single project. Some possibilities:

- Turn the project folder for one of your current research project into a git/GitHub repo and/or R project
- Modify an R script to use functions, if/for loops, or format code according to best practices
- Improve file and folder organization for project or your entire computer!