Seascape Ecology

Lab 02 - Scripts, project management, and workflows

Ryan Reisinger

2023-10-10

Welcome to the second computer lab for Seascape Ecology.

Learning objectives

Part 1

1. Add and remove rows or columns.
2. Remove rows with NA values.
3. Append two data frames.
4. Understand what a factor is.
5. Convert a factor to a character vector and vice versa.
6. Display basic properties of data frames including size and class of the columns, names, and first few rows.
7. To be able to subset vectors, factors, matrices, lists, and data frames
8. To be able to extract individual and multiple elements: by index, by name, using comparison operations
9. To be able to skip and remove elements from various data structures.

Part 2

1. Create a new script, write code with comments, run the code, and save the script.
2. Create a project in RStudio.

Last week

In the previous lab we dusted off our knowledge of R and RStudio, using two lessons from Software Carpentry's R for Reproducible Scientific Analysis course:

1. Lesson 01 Introduction to R and RStudio https://swcarpentry.github.io/r-novice-gapminder/01-rstudio-intro.html
2. Lesson 04 Data Structures https://swcarpentry.github.io/r-novice-gapminder/04-data-structures-part1.html

We refamiliarised ourselves with the RStudio IDE and used it to:

1. Define a variable.
2. Assign data to a variable.

3. Manage a workspace in an interactive R session.
4. Use mathematical and comparison operators.
5. Call functions.
6. Manage packages.

We looked at the five main data types (numeric [double], integer, logical [Boolean], character and complex) and we began exploring dataframes and how they relate to vectors, frames and lists. We learned how to ask R about the type, class and structure of an object.

I also pointed you to a lesson on finding help in R:

Lesson 03 Seeking Help https://swcarpentry.github.io/r-novice-gapminder/03-seeking-help.html

This week

This week we'll continue refreshing a few concepts from your first year module SOES 1015 (Part 1), and we'll start to explore workflow and project management (Part 2):

Part 1

1. Working with data.

Part 2

2. Scripts.
3. Workflow and project management.

Resources

A useful resource, for this week and in general, is the free, open-source book R for Data Science by Hadley Wickham (chief scientist at RStudio), Mine Çetinkaya-Rundel and Garrett Grolemund. We'll dip into one of the chapters of the book this week. Be aware that the examples in the book use packages from the 'tidyverse', which has slightly different syntax for doing things compared to 'base' R (that is, the 'vanilla' R you're probably used to at the moment). We'll also use two lessons from Software Carpentry again.

But first: A Data Science Process

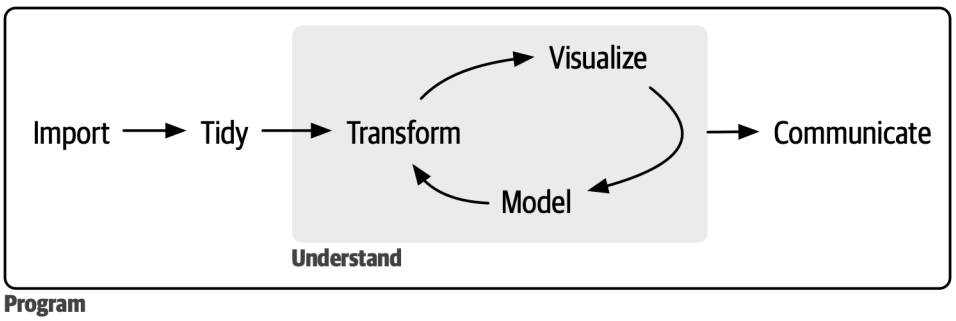

Figure 1: Wickham and Grolemund's model of the data science process. "In [our] model of the data science process you start with data import and tidying. Next you understand your data with an iterative cycle of transforming, visualizing, and modeling. You finish the process by communicating your results to other humans". From https://r4ds.hadley.nz/intro.

What's the point?

We want to contain include as much of this process in a contained set of files that we can move around and re-run easily. Remember from last week: we want our analyses to be **reproducible**, **transparent**, and we want to **save time**.

Part 1

Data tidying

Our first part of this lab tackles the **tidy** step of the process in Figure 1, and continues where we left off last week. Please work through lessons 5 and 6 of the Software Carpentry R for Reproducible Scientific Analysis course:

1. Lesson 05: Exploring Data Frames https://swcarpentry.github.io/r-novice-gapminder/05-data-structures-part2.html.
2. Lesson 06: Subsetting Data https://swcarpentry.github.io/r-novice-gapminder/06-data-subsetting.html

Mark and I will be walking around to help out and answer questions. We'll take 30 minutes to work through the two exercises, then we'll carry on with Part 2 of the lab together.

Part 2 - Workflows and Projects

Scripts

Using the R console (the bottom left pane in RStudio) to type and run code is fine for single lines or playing around with code, but once you want to run (and re-run) longer, more complicated code, and save it, you'll want to use a **script**. The top left pane in RStudio is a **script editor**, where you can write, edit, save and execute (send to R) your code.

You can open a new script either by clicking the File menu in RStudio, and selecting New File, then R script, or using the keyboard shortcut Cmd/Ctrl + Shift + N.

RStudio will automatically save the contents of the editor when you quit RStudio, and will automatically load it when you re-open. Nevertheless, it's a good idea to save your scripts regularly and to back them up somehow.

R will ignore anything on a line following the octothorpe symbol (pound/hash/number sign) - #. That means you can add all kinds of text to your scripts, and it's a good idea to do that! Be kind to your future self and others. Describe what you've done in a section or on a line, add headers for different sections, etc. But a good way to start, right at the top, is a title for the script, your name, the date, and some kind of short description of what it does.

For example:

```
# An Awesome Script
# Ryan Reisinger
# 2022-10-12
# This script loads a great dataset, does all the cleaning up automatically,
# generates some cool plots, and runs a basic statistical model.
```

It is recommended that you always start your script with the packages that you need. That way, if you share your code with others, they can easily see what packages they need to install.

Hadley and Grolemund say:

"The key to using the script editor effectively is to memorise one of the most important keyboard shortcuts: Cmd/Ctrl + Enter. This executes the current R expression in the console... It will also move the cursor to

the next statement (beginning with not_cancelled %>%). That makes it easy to run your complete script by repeatedly pressing Cmd/Ctrl + Enter."

You can also execute (run) a whole script, from the first to the last line, using Cmd/Ctrl + Shift + S. Doing this regularly is a great way to check that you've captured all the important parts of your code in the script.

R scripts are saved with a .R or .r extension.

Projects

Let's work through Wickham and Grolemund's book chapter on Workflow: scripts and projects (Chapter 7) together:

https://r4ds.hadley.nz/workflow-scripts