Case Studies in Reproducible Research: a spring seminar at UCSC

Eric C. Anderson, Kristen C. Ruegg, Tina Cheng, and the students of EEB 295

2017-04-03

Contents

Chapter 1

Course Overview

This is the home of the notes for a proposed course in data analysis and reproducible research using R, Rstudio, and GitHub.

The seminar is called, "Case Studies in Reproducible Research," but we utter that title with the caveat that, although the organizers have quite a few case studies they could spin up for this course, the case studies we will be studying in this course are going to be actual research projects that *you*—the participants—are working on. You're gonna bring 'em, and we are going to collectively help you wrassle them into a reasonable and reproducible data analysis. In the process we will touch on a number of elements of data analysis with R.

We will be working through a healthy chunk of the material in Garrett Grolemund and Hadley Wickham's book, R for Data Science, which is readable for free at the link above. We intend to use a handful of our own data sets each week to illustrate points from the book and show the material in action on real data sets.

This is not intended as a "first course in R". Students coming to the course should have at least a modicum of familiarity with using R, and we will launch more directly into using the tools of the tidyverse. EEB students with little or no experience in R might be interested in sitting in with Giacomo Bernardi's lab group on Mondays at 3PM in the COH library. They are conducting a Bio 295 seminar, working through "a super basic book that takes the very first steps into R."

For the interested, these materials were all prepared using RStudio's bookdown package. The RStudio project in which it lives is hosted on eriq's GitHub page here

1.1 Meeting Times, Location, Requirements

Intended to be Friday afternoons, 1:45–3:15 PM in the library/conference room at Long Marine Lab.

Students must bring a laptop to do examples during the seminar, and all students are expected to have a data set that they are in the midst of analyzing (or upon which they hope to commence analysis soon!) for a research project. We will

1.2 The origin of this seminar

The idea for this course was floated by Tina Cheng who was planning to lead a seminar in spring 2017 based in part on Eric C. Anderson's "Reproducible Research Course", taught at the Southwest Fisheries Science Center in the fall of 2014. Although going over those notes might have been a reasonable exercise, it turns out that a lot has changed in the world of data analysis since fall 2014, and the notes from that course are, today, a little bit dated.

We have been particularly excited by the ascendancy of Hadley Wickham's tidyverse approach to data analysis, and the tremendous development of a variety of tools developed by RStudio for integrating report generation and data analysis into reproducible workflows. In fact, Eric has been saying for the last year that if he were to teach another course on data analysis it would be structured completely differently than his "Reproducible Research Course". So, it was clearly time for him to stop talking and help put together an updated and different course.

At the same time, in working on our own projects and in helping others, we have consistently found that the most effective way for anyone to learn data analysis is to ensure that it is immediately relevant to whatever ongoing research project is currently consuming them. Therefore, in the current seminar, we are hoping to spend at least half of our time "workshopping" the data sets that seminar participants are actually involved in analyzing. Together we will help students wrestle their data, analyses, and ideas into a single, well-organized RStudio project under version control with git. Therefore, every student should come to this course with a data set and an associated analysis project.

1.3 Course Organizers

Kristen C. Ruegg Kristen is a conservation geneticist who specializes in the application of genome-wide data to understand population level processes and inform management, with a particular focus on migratory birds. She has has been enlightened to the powers of the "tidyverse" over the last couple of years (mostly through the constant insistence of her enthusiastic husband Eric Anderson) and is looking forward to becoming more fluid in its application over the course of the quarter. Her main role in this course will be to help with the course design and logistics and help reign Eric in when he has started to orbit into some obscure realm of statistical nuance.

Eric C. Anderson Eric trained as a statistician who specializes in genetic data. Since 2003 he has worked at the NMFS Southwest Fisheries Science Center in Santa Cruz. Although much of his statistical research involves the development of computationally intensive methods for specialized analyses of genetic data, he has been involved in a variety of data analysis projects at NMFS and with collaborators worldwide. Eric was an early adherent to reproducible research principles and continues, as such, performing most of his research and data analysis in the open and publicly available on GitHub (find his GitHub page here). In 2014, he taught the "Reproducible Research Course" at NMFS, and is excited to provide an updated version, focusing more, this time, on the recently developed "tidyverse".

Tina Cheng Tina is a graduate student in EEB. She is going to be leading the session during the first week of the course when Kristen and Eric are still on spring break, and then she is going to be joining in on the fun with us for the remainder of the quarter until she has to travel off to Baja, TA-ing the "supercourse" during the last four weeks of the quarter.

1.4 Course Goals

The goal of this course is for scientists, researchers, and students to learn to:

- properly store, manage, and distribute their data in a *tidy* format
- consolidate their digital research materials and analyses into well-organized RStudio projects.
- use the tools of the tidyverse to manipulate and analyze those data sets
- integrate data analysis with report generation and article preparation using the Rmarkdown format and using R Notebooks
- use git version control software and GitHub to effectively manage data and source code, collaborate efficiently with other researchers, and neatly package their research.

By the end of the course, the hope is that we will all have mastered strategies allowing us to use the above-listed, freely-available and open-source tools for conducting research in a reproducible fashion. The ideal we will be striving for is to be able to start from a raw data set and then write a computer program that

conducts all the cleaning, manipulation, and analysis of the data, and presentation of the results, in an automated fashion. Carrying out analysis and report-generation in this way carries a number of advantages to the researcher:

1. Newly-collected data can be integrated easily into your analysis.
2. If a mistake is found in one section of your analysis, it is not terribly onerous to correct it and then re-run all the downstream analyses.
3. Revising a manuscript to address referee comments can be done quickly.
4. Years after publication, the exact steps taken to analyze the data will still be available should anyone ask you how, exactly, you did an analysis!
5. If you have to conduct similar analyses and produce similar reports on a regular bias with new data each time, you might be able to do this readily by merely updating your data and then automatically producing the entire report.
6. If someone finds an error in your work, they can fix it and then easily show you exactly what they did to fix it.

Additionally, packaging one's research in a reproducible fashion is beneficial to the research community. Others that would like to confirm your results can do so easily. If someone has concerns about exactly how a particular analysis was carried out, they can find the precise details in the code that you wrote to do it. Someone wanting to apply your methods to their own data can easily do so, and, finally, if we are all transparent and open about the methods that we use, then everyone can learn more quickly from their colleagues.

In many fields today, publication of research requires the submission of the original data to a publicly-available data repository. Currently, several journals require that all analyses be packaged in a clear and transparent fashion for easy reproduction of the results, and I predict that trend will continue until most, if not all, journals will require that data analyses be available in easily reproduced formats. This course will help scientists prepare themselves for this eventuality. In the process, you will probably find that conducting your research in a reproducible fashion helps you work more efficiently (and perhaps even more enjoyably!)

1.5 Weekly Syllabus

1.5.1 Week 1 — Introduction and Getting Your Workspace Set Up

- At the end of this session we want to make sure that everyone has R, RStudio, and Git installed on their systems, and that they are working as expected.
- Additionally, everyoneshould have a free account on GitHub.
- And finally we need everyone's email address.

Some things to do:

- Get Rstudio cheat Sheets!
- Assemble data into a project
- Get private GitHub repos

Eric! You need to make an example project repo.

1.5.2 Week 2 — RStudio project organization; using git and GitHub; Quick RMarkdown

After this, students are going to have to put their own data into their own repositories and write a README.Rmd and make a README.md out of it.

1.5.3 Week 3 — Tibbles. Reading data in. Data rectangling

- Reading data into the data frames.
- read.table and read.csv
- tibbles
- The readr package
- Data types in the different columns and quick data sanity checks.
- A few different gotcha's
- Saving and reading data in R formats. `saveRDS` and `readRDS`.

1.5.4 Week 4 —

Chapter 2

Week One Meeting

Tina is going to be helping everyone get their systems all set up. After that we will have everyone clone an RStudio project from GitHub to see how easy that is.

2.1 Software Installation

1. **RStudio:** We want the latest "development" version of RStudio becuase it has features that we may want to use during this course. Get it from https://www.rstudio.com/products/rstudio/download/preview/ and install the appropriate one for your OS.
2. **R:** Let's make sure that we are all using the latest version of R. On March 7, 2017, version 3.3.3 was released. Go to https://cran.r-project.org/ and find the download link for your computer system. Download it and install it.
3. **bookdown:** This package is what I used to create these course notes. Getting it automatically installs a lot of other packages that are useful for authoring reproducible research. We want the latest development version, which can be obtained from GitHub by issuing the following commans at the R prompt (i.e. in the console window of RStudio:)

```
install.packages("devtools")
devtools::install_github("rstudio/bookdown")
```

4. Install **other packages** that we are going to be needing in the first few weeks. If you don't know how to install packages, ask Tina and she can show you. Install: `tidyverse`, and `stringr`.
5. Make sure that **git** is up and running on your system.
   - If you are using a Mac with a reasonably new OS, you should be able to just open the Terminal application (`/Applications/Utilities/Terminal`) and type "git" at the command line. If you have `git` it will say something that starts like:

```
    usage: git [--version] [--help] [-C <path>] [-c name=value]
   [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
   [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
   [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
   <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
```

```
   clone      Clone a repository into a new directory
   etc. etc. etc.
```

If you do not have `git` then it should pop up a little thing asking if you would like to install a reduced set of developer tools. You do. Click OK.

- If you are using a PC, I can't be as much help, but you can find links with instructions on how to download `git` for a PC here.
- If you are using Linux then we will assume you know how to get `git` or that you already have it.

2.2 Get an account on GitHub

If you don't already have an account on GitHub, go to github.com and click the "sign up" link near upper right of the page. It is pretty self-explanatory. Go ahead and get a **free** account. There is nothing to pay for here!

2.2.1 Private repositories

If you are a graduate student and you do not feel comfortable posting your data on a public site like GitHub, then you should request some private repositories from GitHub. GitHub has a great deal for academic users like students: free private repositories. Please go to https://education.github.com/pack to sign up for your free student pack.

2.3 Open an RStudio Project from GitHub

I am going to have everyone use RStudio and GitHub to clone and open an RStudio project that I prepared as a template so that people can see how I would like them to start putting together their own projects.

To open this project, from RStudio, go to the menu option "File->New Project…". Then from the resulting dialog, choose "Version Control". Then choose "Git". Then it asks for a "repository URL". Supply this: `https://github.com/eriqande/rep-res-coho-example` and leave the "Project Directory Name" empty. And then choose a directory in which to put it and click OK.

Bam! That will pull the RStudio project off of GitHub, make a local clone of it on your hard drive and open.

Once you have done that. Open `README.Rmd` within the project, and click the "knit" button which should be present near the top left of the editor window.

That is how you convert an R Markdown README to README.md which is easy to read and see on GitHub.

If you want to see what the project repository looks like on GitHub, have a look at https://github.com/eriqande/rep-res-coho-example.

2.4 Assignment for next week: Create an RStudio Project with Your Own Data

Your mission for the following week—i.e., please have this done (or as done as you can get it) by Friday, April 14, 2017—is to prepare an RStudio project with your own data set, and provide some background about the data and the ways that you would like to analyze it. The "rep-res-coho-example" is an example of what I have in mind for this. You should use the README.Rmd from that project as a template for

your own README.Rmd. (To do this you can just copy the README.Rmd file into the top level of your project directory and then edit it to reflect your own data and project.)

To do all this you are going to want to make your own project. Do that like this:

1. In RStudio, choose "File->New Project…"
2. Then choose "New Directory" and then choose "Empty Project"
3. In the next dialog, choose a name (*it is best to use only letters, numbers, dashes, and underscores, and include no spaces in the name*) for it **and be sure to click the "Create a git repository" button**.

4. Then click "Create Project".

That should give you a new project. Here are some guidelines for putting your own data in there

- Put all of your data in a directory named `data` in your project.
- CSV (comma separated values) is probably the best format to use. It is text-readable without proprietary software (unlike an Excel file); however if you need to look at it in a tabular way with Excel, (gasp), you can do that easily. Tab-delimited text works if you have that, but CSV is preferred.
- Use only letters, numbers, dashes, and underscores for the file names, (and periods for their extensions, i.e., `.csv`)
- Give a brief description of your data in the README.Rmd.

2.5 Reading for next week

This week (before Friday, April 14, 2017), please read the following sections of the R for Data Science book

- Workflow basics: super basic review on how R works.
- Workflow: projects: info about organizing RStudio projects.
- Workflow: scripts: how to evaluate code in scripts.
- tibbles: a streamlined data frame format.
- data import This is our key reading for the week.

When you are done with the *Data Import* reading, take a whack at writing some code to read the data files in your project into a variable (or several variables).

Chapter 3

Applications

Some *significant* applications are demonstrated in this chapter.

3.1 Example one

3.2 Example two

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter **??**. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure **??**. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table **??**.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (**?**) in this sample book, which was built on top of R Markdown and **knitr** (**?**).

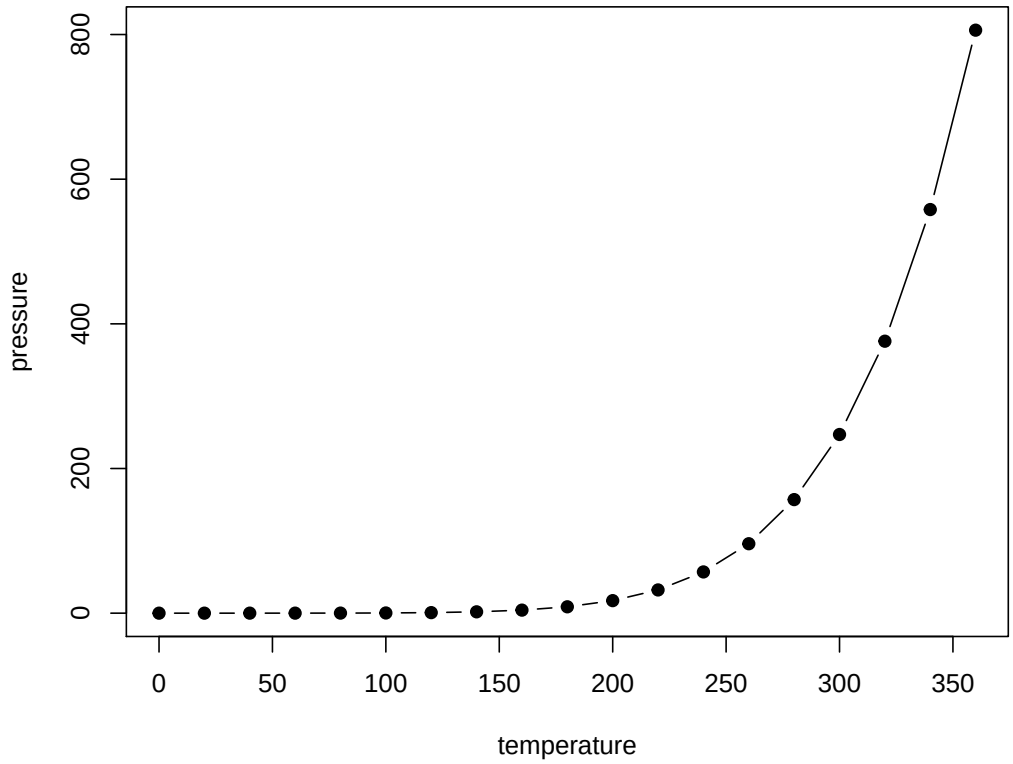

Figure 3.1: Here is a nice figure!

Table 3.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa