Foundational Compuational and Statistical Tools for the Natural Sciences

Clayton M. Small, William A. Cresko, Andrew Muehleisen, Hope Healey and Sabrina Mostoufi

2022-12-07

Contents

Chapter 1

Book Overview

This is the *associated book* for the *Foundational Computational and Statistical Tools for the Natural Sciences Course* at the University of Oregon. It is written in **Markdown** so that it can be easily updated.

In this book you will find nearly all the information you will need to complete the course.

Chapter 2

Introduction to the course

This is the complete set of *course materials* for the *Foundational Statistics Course* at the University of Oregon for the Spring of 2020. It is written in **Markdown** so that it can be easily updated.

In this book you will find nearly all the information you will need to complete the course.

2.1 Instructors

Dr. Bill Cresko, wcresko@uoregon.edu

Hope Healey, hhealy@uoregon.edu

Sabrina Moustoufi, smostouf@uoregon.edu

2.2 Course Information

Class Time: T-R 8:30 to 9:50

Working laboratory: xxxx

Cresko Office Hours: xxxx

2.3 Software

- Latest version of R
- Latest version of RStudio

2.4 Inclusion and Accessibility

Please tell us your preferred pronouns and/or name, especially if it differs from the class roster. We take seriously our responsibility to create inclusive learning environments. Please notify us if there are aspects of the instruction or design of this course that result in barriers to your participation! You are also encouraged to contact the Accessible Education Center in 164 Oregon Hall at 541-346-1155 or uoaec@uoregon.edu.

We are committed to making this course an inclusive and respectful learning space. Being respectful includes using preferred pronouns for your classmates. Your classmates come from a diverse set of backgrounds and experiences; please avoid assumptions or stereotypes, and aim for inclusivity. Let us know if there are classroom dynamics that impede your (or someone else's) full engagement.

Because of the COVID-19 pandemic, this course is being delivered entirely remotely. We realize that this situation makes it difficult for some students to interact with the material, for a variety of reasons. We are committed to flexibility during this stressful time and emphasize that we will work with students to overcome difficult barriers as they arise.

Please see this page for more information on campus resources, academic integrity, discrimination, and harassment (and reporting of it).

Chapter 3

Course Schedule

3.1 Weeks 1-2

1. Data organization and management
    - best practices, reproducibility, etc.
2. Basic programming fundamentals for data curation
    - The Unix environment and fundamental commands
    - Formatting and manipulating tabular text files from the terminal
3. Introduction to R and Rstudio
    - Installation/Updates
    - R object types and assignment
4. Practice with R objects
    - vectors, matrices, data frames, etc.
5. Applying core programming fundamentals in R
    - vectorized operations
    - replicate, apply family, ifelse, for loops, etc.

3.2 Week 3

1. Plotting/visualizing data as a means of exploration
    - Different plot types
    - Scale, transformations, etc.
2. Fundamentals of plotting in base R
    - par
    - using palettes, points, sizes, etc. to convey information
    - axes and labels
3. R markdown

3.3 Week 4

1. Population parameters, samples, and sampling distributions
    - Central Limit Theorem and the normal dist.
    - Mean and st. dev.
2. Probability and probability distributions
3. Calculating summary statistics
    - Other common summary statistics (quantiles, etc.)

3.4 Week 5

1. Parameter estimation
    - Simulating data sets with known parameters
    - Revisit probability distributions
2. Uncertainty in estimation
    - Parametric and nonparametric approaches to uncertainty

3.5 Week 6

1. Experimental design
    - lexicon
    - considering sources of variance
    - types of variables (categorical, ordinal, rational)
    - confounding variables
2. Frequentist hypothesis testing
    - error types
    - p-values
    - degrees of freedom
    - statistical power
    - multiple testing problem

3.6 Week 7

1. Comparing means between groups
    - Student's t-test
2. Bootstrapping and randomization to compare means

3.7 Week 8

1. Relationships between quantitative variables
    - correlation and covariance

2. Simple linear regression
    - residuals and least squares
    - fitting linear regression models

3.8 Week 9

1. Analysis of variance
    - Table components and test statistics
2. General linear models in R
    - Model formulae
    - Interpretation of summary output
3. More complex ANOVA frameworks
    - Nested models
    - Factorial models

3.9 Week 10

1. Frequency-based statistical tests
    - Chi-squared tests
    - Contingency tables and tests of independence
2. Brief introduction to generalized linear models (time permitting)
    - logistic regression

Chapter 4

Background material for the course

4.1 Description of the course

This course in an introduction to data management, data visualization, and statistical inference. It is intended for early-stage graduate students with no background in statistics. No prior coursework (undergraduate or graduate) in statistics or programming is assumed. The primary objective of the course is to get students up to speed with respect to organization, manipulation, visualization, and analysis of data, using the R statistical language. The emphasis on application is strong, with the goal of enabling students (after the course) to analyze their own data sets with confidence using reasonable approaches, and, when faced with more difficult analyses, to be able to communicate their inference objectives clearly to expert analysts. Students will learn to organize and analyze data sets in the form of RStudio projects, using R Markdown files to reproducibly capture and render code, visualizations, and analyses. In-class exercises will be delivered in the form of pre-formatted R Notebooks, which can be interactively executed by students without having to write all code from scratch.

The course is designed to acquaint students primarily with univariate (single response variable) analysis. Multivariate analysis will be covered in the Advanced Biostatics 2-course series offered during the Fall and Winter terms. Examples and assignments in class will include data sets primarily from the biological sciences, including studies of morphological and molecular traits, behaviors, ecological questions, and clinical studies. For specific statistical topics covered in class, please see the course goals and tentative schedule below.

4.2 Course goals:

- Properly organize and format primary data and metadata files for analysis
- Learn programming fundamentals of the R statistical language, including objects, functions, iteration, and simulation.
- Make publication-quality data visualizations, including scatterplots, boxplots, frequency distributions, mosaic plots, etc.
- Understand Type I and Type II statistical error, including p-values and power analysis.
- Understand ordinary least-squares regression and linear models in general
- Learn the fundamentals of strong experimental design
- Learn to apply general linear models to basic univariate analysis problems, including Analysis of Variance (ANOVA)
- Learn nonparametric approaches to parameter estimate and statistical inference, including resampling (bootstrapping), permutation, and rank-based analysis.
- Understand how to analyze binary response variables and frequency-based (e.g. contingency table) data sets.

4.3 Introduction to R and RStudio

R is a core computational platform for statistical analysis. It was developed a number of years ago to create an open source environment for advanced computing in statistics and has since become the standard for statistical analysis in the field, replacing commercial packages like SAS and SPSS for the most part. Learning R is an essential part of becoming a scientist who is able to work at the cutting edge of statistical analysis – or even to perform conventional statistical tests (e.g. a t-test) in a standard way. An important part of R is that it is script-based, which makes it easy to create reproducible analysis pipelines, which is an emerging feature of the open data/open analysis movement in science. This is becoming an important component of publication and sharing of research results, so being able to engage fully with this effort is something that all young scientists should do.

RMarkdown is an extra layer placed on top of R that makes it easy to integrate text explanations of what is going on, native R code/scripts, and R output all in one document. The final result can be put into a variety of forms, including webpages, pdf documents, Word documents, etc. Entire books are now written in RMarkdown and its relatives. It is a great way to make quick webpages, like this document, for instance. It is very easy to use and will be the format that I use to distribute your assignments to you and that you will use to turn in your assignments.

R Projects are a simple way of designating a working directory in which to house files related to a given, well, project. Those files might include primary data and metadata files ready for reading into R, `.R` scripts, Rmarkdown files, and

output such as Rmarkdown-rendered .html files or individual plots, for example. The nice thing about organizing your work with R Projects is that you can keep everything needed to reproduce an analysis in a single directory on your computer. You can open an R Project in RStudio by opening the project's index (`.RProj`) file, which will automatically set your working directory to that of the project and facilitate loading any saved environments, etc.

In Chapter 6 we will begin working in R and RStudio, but you can get them installed now (in that order) on your computer, if you haven't already. Get the most recent *released* R version by following this link: https://www.r-project.org/

We will do our work using Rstudio, which is a powerful and convenient user interface for R, and can be downloaded from here for installation: https://rstudio.com/products/rstudio/

4.3.1 Learning resources

There are tons of resources for learning R and RMarkdown on the internet. Here are just a few, but you will no doubt find your own favorites as you become routine R users.

There is an organized group that is dedicated to training in R called DataCamp (https://www.datacamp.com/). They provide all of the basics for free. They actually have training for most data science platforms. RStudio provides links for training directly related to R and RMarkdown here: https://education.rstudio.com/

There are also many, many R training videos on YouTube. Most of them are very well meaning but may not be as in-depth as you want.

You can also go the old "paper" manual route by reading the materials provided by R itself: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

In reality, if you want to do almost anything in R, simply type in what you are interested in doing into Google and include "in R" and a whole bunch of links telling you exactly what to do will magically appear. Most of them appear as discussions on websites like StackOverflow and Stats.StackExchange. In that case, the first thing that you see is the question–usually someone doing it just a bit wrong–so you should scroll down to see the right way to do it in the answers. It is really an amazing resource that will speed you along in nearly every form of analysis that you are interested in.

Please do not hesitate to contact us if you have questions or run into obstacles. The point of this class is to learn by doing, but our aim is that the doing should involve reasonable first efforts supplemented with help if needed. Also, many of your classmates have some experience with R, writing code, or statistics in general, so they are an excellent resource as well!

Chapter 5

Organizing and manipulating data files

5.1 Introduction

Many of you will already be familiar with data file organizaiton, editing, and formatting for analysis. If so, much of the following material may be review. If not, some of the following guidelines and tools should prove to be quite useful. In biology, and many other fields, primary data are routinely stored as "flat" text files. The exact formatting depends on the type of data, of course, but often we are working with text files organized into rows and columns. Rows can naturally be defined by lines in a file, and columns can be defined by separators (also called delimiters) such as spaces, tabs, or commas, to name a few commonly used ones. Fortunately there are some very powerful and simple-to-use (with a little practice) tools that can be invoked directly from a computer's command line, or included in written "scripts" that your computer's operating system can interpret upon you running them. These command line tools are now nearly ubiquitous on all personal computer platforms. Computers running a LINUX operating system allow direct access to these tools via the command line, as does the macOS operating system of Apple computers via the Terminal. Computers running Microsoft Windows 10 now also facilitate use of these conventional "UNIX tools" through a Windows Subsystem for Linux.

In the following sections, we provide a *very brief* introduction to using some of these tools in order to organize your data files, parse them for information, and perform some basic text manipulations. Mastering these activities is not necessary for this course (in fact, many of the text manipulation tasks can be done in R!), but if you learn to adopt at least some of these skills you will become a better, more organized analyst, and it will help you become comfortable with the command line and programming in general.