PSYC 417: Data Science for Psychology and Neuroscience Majors (lab course, 4 credits)

Instructor:

Alec Solway, Ph.D.

asolway@umd.edu

Graduate Assistant:

Yuqing Lei

ylei1@terpmail.umd.edu

Spring 2021

Class meeting time and location

Mon, Wed, 1-3pm. The class will be entirely online, but we will meet **synchronously** at the scheduled time. Please see ELMS for instructions on how to connect.

Course description and objectives

The target audience for this class is psychology, neuroscience, and others majors that are interested in data science, but are outside the typical data science track. The class starts from scratch, with very minimal background assumed (described below). A large number of industry and academic jobs require basic programming and data analysis skills. This class represents an introduction to both. Students will learn to program in R and will briefly be introduced to Python, the two most popular programming languages for data science. Common constructs shared by a variety of procedural programming languages will be emphasized. Basic statistics and probability theory will be reviewed from a computational perspective, and more advanced topics introduced. During the course, students will simulate toy data sets which they will then analyze knowing how the data came about, as well as work with real data. The class is highly hands-on with a large number of in-class lab and homework projects. Expect to work a lot and move quickly. Because of the hands-on nature of the class, the overall focus is more on application and execution rather than theory. However, some theory is covered at a high level so that students are aware of *why* they are doing something, rather than mindlessly writing code.

There are many different valid approaches to teaching a broad introductory class such as this. One dimension of relevance is whether to emphasize more fundamental topics, or to dive head first into more “advanced” analyses using third-party programming libraries (packages). The advantage of the package approach is that you are able to work with real world data much sooner. The disadvantage is that what you learn is highly situation dependent – on the the type of data you analyze, and on the specific packages you learn about. In this class, we take a fundamentals first approach, emphasizing foundational skills. If you are debating between this course and a more package oriented course (in another department), the courses may be complementary, and it may even make sense to take both. See the instructor.

Upon successfully completing this course, students will be able to:

1. Write programs in R and Python.
2. Understand, at a high level, the theoretical foundation of frequentist statistics and how it applies to real data.
3. Understand when and how to (using R) apply t-tests, standard linear and logistic regression, and hierarchical linear and logistic regression.
4. Understand what a model is, how models may be fit to data, and how to perform numerical optimization in R.
5. Create publication quality plots of data.
6. Given a new data set, independently think through and implement a basic data analysis plan.

Office hours and location

TBD via when2meet based on student votes after the first week of class. Please check ELMS for updates on this.

Prerequisites

Understanding of introductory statistics, probability theory, and research methods at the level of PSYC 200 & PSYC 300, and one semester of Calculus (we will not rely on the latter, but this requirement is used as a measure of quantitative maturity). No background in programming is required.

Recommended textbooks

There are a number of perspectives from which to teach an introductory class such as this one. We have created a unique combination of learning units specifically tailored for psychology and other traditionally ’non-quantitative’ majors. As such, there is not a good textbook on the market that covers all of the material in this course. All of the material required to succeed will be discussed during lectures and presented in slides.

However, having some written reference can be beneficial as a backup and an alternative source from which to learn the material. There are two books that together cover roughly 60% of the material. (Don’t worry, this doesn’t mean that the class covers more than two textbooks worth of material, or more than one textbook. It covers bits and pieces from different sources.) The following is a good introduction to programming in R:

Cotton, R. (2013). *Learning R: A Step-by-Step Function Guide to Data Analysis*. O’Reilly Media.

It is relatively cheap as far as textbooks go, and will be available in the bookstore.

As a prerequisite for this course, you are expected to have a basic understanding of introductory statistics and probability theory. However, if you’ve only taken a couple of classes and haven’t applied the techniques you’ve learned beyond the classroom, you may be rusty. We will review everything we need in class, but it would be highly beneficial to have a written reference you can read before and after class. If you still have and like your introductory stats/probability theory textbook, feel free to use that. Assuming most people don’t, a good reference is:

Navarro, D. *Learning Statistics with R: A tutorial for psychology students and other beginners*. https://learningstatisticswithr.com/.

This book is available free online. It also doubles as a second introduction to R, if you find something unclear in the first textbook above.

Each section in the course outline below, through the mid-term, lists recommended chapters from these two books as reference. After the mid-term, we will rely purely on in-class material.

Study strategies and academic integrity

The teaching philosophy in this class is centered around “learning by doing.” Most assignments and assessments are based on hands-on projects completed either in the classroom or at home. Putting learning into practice is a much more powerful way of understanding and remembering things than relying purely on one-shot tests (although often the latter cannot be avoided for practical reasons), because 1) you have to retrieve what you’ve learned in a context dependent manner, and 2) you are repeatedly made aware of what you don’t know, and are forced to go back and learn it. With this philosophy in mind, you are allowed to help each other *through high-level discussions* on the assignments. However, each individual is responsible for learning all of the material, and you have to code/write and turn in your own individual answers.

Accessibility and Disability Service

If you require special accommodations, please present current documentation from the Accessibility and Disability Service (ADS) before the schedule adjustment deadline. More information on University policies can be found at
https://www.counseling.umd.edu/ads/.

Grading

Lab assignments

There will be a number of in-class lab assignments during the semester. These will not be graded, but you have to hand them in to show that you did them and get credit.

Homeworks

There will be five homeworks during the course of the semester.

Projects

There will be two larger projects, in lieu of a mid-term and final exam. You will largely work on these at home, but also have ample class time to work on them and ask questions.

The relative weight of these is as follows:

In-class lab assignments	40% (divided evenly)
Homeworks	25% (divided evenly, 5% each)
Mid-term project	15%
Final project	20%

Homeworks and lab assignments will generally be programming problems. You will hand in your code and short answers to questions, provided inline together with your code, on ELMS/Canvas. The projects are more involved and will come with detailed instructions. See the course outline below for more information.

Attendance and late homework

This is a hands-on course, thus, it will be extremely difficult to succeed without regular attendance. Note that although the class will be entirely online, we will meet **synchronously** at the scheduled time. Many classes will have in-class lab assignments that are due the same day. **However, it goes without saying that we are in a very unique situation**

this year. I am very happy to work with you with regards to assignment deadlines if issues arise. ***Please feel free to discuss any concerns with me.*** You are highly encouraged to discuss missed material with other students and with the instructors during office hours or in one-on-one meetings (all over Zoom or similar). Almost all later topics in this course build on earlier ones.

Inclusive Learning Environment

Students will be invited to share their thoughts in class and a diversity of opinions is welcome. Respectful communication is expected, even when expressing differing perspectives. Supporting one's statements with research findings is encouraged. In accordance with free speech statues, speech that contains threats of violence is prohibited.

University-wide policies

Please see http://www.ugst.umd.edu/courserelatedpolicies.html.

Course Outline

The following outline is subject to change. The pace of the course may be altered based on the technical background of each cohort of students. Topics marked with a * are more likely to be cut if we find that we have to slow down. Each * topic can be independently removed without altering the rest of the course. We will use the R programming language for the majority of the course, except where explicitly noted.

1. Introductory class: What is data science? Scope of this class. Software setup.

Recommended reading: Cotton, chapter 1

2. Variables and data types

Recommended reading: Cotton, chapters 2-5

3. Algorithms, loops, and conditional statements

Recommended reading: Cotton, chapter 8 (optional: chapter 9)

Lab assignment 1: Finding the maximum of an array of numbers

Homework 1: Implementing summary statistics from scratch

4. Functions

Recommended reading: Cotton, chapter 6

Lab assignment 2: Practice with functions

Homework 2: More practice with functions

5. Probability

Recommended reading: Navarro, chapters 5 & 9; Cotton, chapter 15 up to “Formulae”

Lab assignment 3: Probability, generating random numbers in R

6. A generative perspective on sampling distributions and t-tests

Recommended reading: Navarro, chapters 10-11, 13

Lab assignment 4: Simulating t distributions and confidence intervals

Homework 3: More on sampling distributions

7. Linear and logistic regression

Recommended reading: Navarro, chapter 15; Cotton, chapter 15 from “Formulae” until the end

Lab assignment 5: Regression

Midterm project

You will analyze a data set from a real psychology study from scratch, starting with the ‘raw’ data. You will be provided a step-by-step guide that will help take you through the process. In the end, you will hand in your analysis code and a brief report detailing the experiment, how you analyzed it, and what you learned.

8. Introduction to Python

Recommended reading: There are no readings for the remaining units. The class slides will contain all of the information necessary to learn the material.

Lab assignment 6: Python

Homework 4: Revisit Homework 1 using Python

9. Object oriented programming in Python

Lab assignment 7: OOP in Python

Homework 5: OOP in Python

10. Generating publication quality plots

Lab assignment 8: Revisit the plots generated for the mid-term project using the more advanced plotting tools covered in this unit.

11. Hierarchical models*

Lab assignment 9: Hierarchical linear and logistic regression

12. Numerical function optimization*

Lab assignment 10: Practice with optimization, implement simple linear regression from scratch using optimization methods.

Final project

The final project is open-ended. You will be provided a rubric with a set of general criteria to meet, e.g. ‘demonstrate that you know how to use a “for” loop’. You can use any data set and programming language to meet the criteria described in the rubric, including data from a lab you work or volunteer in. If you do not have your own data, you will be pointed to a large online data bank where you can choose a third-party data set to work on. You will hand in your analysis code and a report, more detailed than the one for the mid-term project, describing the data, how you analyzed it, and what you learned.