PSYC 489D: Data Science for Psychology and Neuroscience Students I

Instructor:
Alec Solway, Ph.D.
asolway@umd.edu

Spring 2018

Class meeting time and location
Tuesday & Thursday, 3:30-4:45pm, BPS 0140

Office hours and location
Tuesday & Wednesday, 5pm until the last person leaves (5:30pm if no one shows up), BPS 1147G

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.

Required textbook
Free online reference for introductory statistics and probability theory

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 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:


This book also doubles as a second introduction to R, if you find something unclear in the main textbook.

Course description

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, a high level programming language for data analysis, and will also be briefly introduced to Python, another popular language. Common constructs shared by a variety of procedural programming languages will be emphasized. We will start from scratch, and no previous programming background is assumed. The second goal of this class is to explore basic statistics and probability theory from a computational perspective. Students will simulate toy data sets which they will then analyze, as well as work with real data. The class is highly hands-on with a large number of in-class and homework based projects. Expect to work a lot and move quickly. Examples will be drawn from psychology and neuroscience where required, but no domain specific knowledge is assumed. Because of the hands-on nature of the class, the overall focus is more on application and execution rather than strict theory.

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 times 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.

Some units will have a reading associated with them. It is advised that you read this
material at least twice, once before the scheduled class(es) and once after, if you
truly want to understand it. It is impossible to completely understand a brand new
technical subject in one reading. On your first pass, you don’t actually know what
you need to learn or how to read the material, so you take most of it in passively.
This allows you to get a glimpse of the “big picture”: what the broad concepts are
and how they might relate to one another. Subsequent readings are required to
start to fill in the details and to learn the material in a more active fashion, where
you can predict what might come next and what you have yet to learn.

**Special needs and disabilities**

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

**Grading**

*In-class assignments*

There will be a number of in-class 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 assignments  40% (divided evenly)  
Homeworks        25% (divided evenly, 5% each)  
Mid-term project  15%  
Final project     20%  

**Attendance and late homework**

This is a hands-on course, thus, it will be extremely difficult to succeed without regular attendance. Many classes will have in-class assignments that are due by the end of the class. You have two free “passes” that allow you to turn in an in-class assignment (not a homework) up to 48 hours late. Beyond these two passes, written documentation (e.g. from a physician) is required to justify handing in an assignment late. Unexcused late homeworks will not be graded and will receive an automatic zero. You are highly encouraged to discuss any missed material and get feedback during office hours, since almost all later topics in this course build on earlier ones.

**University-wide policies**

Please see [http://www.ugst.umd.edu/courserelatedpolicies.html](http://www.ugst.umd.edu/courserelatedpolicies.html).

**Tentative schedule of topics**

The following is an approximate schedule of topics. We are bound to break from this schedule (and perhaps even the topics themselves), thus, if you miss a class, you are encouraged to talk to a classmate or e-mail the instructor to see what you missed.
Jan 25

Introduction to the class, types of programming languages, interpreted vs compiled, starting with RStudio, Hello World

Reading: Cotton, ch 1

Jan 30

Variables and assigning values, primitive data types and basic operators, complex built-in data types: vectors, matrices, factors, lists, data frames

Reading: Cotton, ch 2-5

Feb 1

Algorithms as recipes, build a high level algorithm as a class, if/else and switch statements, for and while loops

Reading: Cotton, ch 8 (optional: ch 9)

**Homework 1:** implement basic summary statistics from scratch using loops and compare to built-in results. Due Feb 7th, 11:59pm.

Feb 6

Questions about homework, basic built-in summary statistics functions, user defined functions, functions as first class structures, functional programming constructs: map, filter, reduce

Reading: Cotton, ch 6
Feb 8

Review of everything so far, commenting and documentation, debugging

Reading: Cotton, section “Debugging” in ch 16

**Homework 2:** separate summary statistics code from homework 1 into functions, re-implement using functional programming constructs, comment both implementations. Due Feb 14th, 11:59pm.

---

Feb 13

Questions about homework, reading and writing CSV files, basic version control

Reading: Cotton, section “CSV and Tab-Delimited Files” in ch 12.

Feb 15

Review of basic probability theory, distributions, and moments. Generating random numbers in R and plotting histograms.

Reading: Navarro, ch 5, 9. Cotton, ch 15 up to “Formulae”.

**Homework 3:** generate samples from a number of different distributions and compute summary statistics using the functions made in homeworks 1 and 2. How do they compare to theoretical results? Due Feb 19th, 11:59pm.

---

Feb 20

Review of sampling distributions and t-tests, how to perform in R. Practice in class on simulated and real data.

Reading: Navarro, ch 10-11, 13.
Feb 22

Review of linear regression, how to perform in R. Plotting basic scatter plots. Practice in class on simulated and real data.

Reading: Navarro, ch 15, Cotton ch 15 from “Formulae” until the end.

Feb 27

Review of logistic regression, how to perform in R. Practice in class on simulated and real data.

**Project 1:** analyze accuracy and RT data for a simple cognitive task. Students will have to use the majority of concepts covered so far. Due Mar 15th, 11:59pm.

Mar 1

Work on project in class.

Mar 6

ggplot2 and guidelines for creating publication quality plots. In class, redo plots from regression assignments using ggplot2.

Mar 8

Work on project in class.

Mar 13

Work on project in class.
Mar 15

Packages and libraries. In class, create a package for the basic summary statistics functions developed in the first homeworks.

Mar 20, 22

Spring break

Mar 27

Many procedural programming languages are similar: programming redux in Python. Variables, data types, if/else and switch, for and while loops, functions, commenting, loading packages.

Mar 29

In class assignment: re-program the first homeworks in Python.

**Homework 4**: Code paired t-test in Python from scratch (almost). Due Apr 4, 11:59pm.

Apr 3

Displaying images, handling input, reading/writing files in Python.

Apr 5

In class assignment: Program simple binary choice decision experiment. Bare bones code will be provided, students will have to fill in the blanks.

Apr 10

Continue in class assignment from Apr 5.

**Homework 5**: Extend binary choice experiment programmed in class with a “distractor” condition. Bare bones code will be provided, students will have to fill in the blanks. Due Apr 18, 11:59pm.
Apr 12

Introduction to OOP in Python.

Apr 17

In class assignment: Code unpaired t-test in Python, combine with paired t-test code from homework 3 using OOP principles.

Apr 19

Hierarchical models and hierarchical linear and logistic regression. lme4 tutorial.

Apr 24

In class assignment: reanalyze accuracy data from project 1 using lme4.

Project 2 (final project): Unguided analysis and write-up of a more complex behavioral data set. Due May 14th, 11:59pm.

Apr 26

1D function optimization, conceptual high level math overview of optimization, conceptual overview of numerical optimization, black box optimization with optim in R, dealing with multimodality.

May 1

In class assignment: find the optima for a number of 1 and 2D functions. Implement multistart from scratch.

May 3

In class assignment: program linear regression from scratch using optim in R.

May 8

Work on final project in class.
May 10

Work on final project in class.