PSCI 207: Applied Data Science

Fall 2019

Lecture: Mondays and Wednesdays, 2:00-3:30
Location: 101 Perelman Center (133 S. 36th St.)

Dr. John Lapinski

Email:	lapins@sas.upenn.edu
Office:	411 (inside suite 406) Perelman Center (133 S. 36th St.)
Office hours:	By appointment

Dr. Stephen Pettigrew

Email:	pettigr@sas.upenn.edu
Office:	404 (inside suite 406) Perelman Center (133 S. 36th St.)
Office hours:	Wednesdays, 10am-noon

Samantha Sangenito

Email:	ssange@sas.upenn.edu
Office:	36B Fox-Fells Hall (3814 Walnut St.)
Office hours:	Tuesdays, 10am-noon

Course description

Jobs in data science are quickly proliferating throughout nearly every industry in the American economy. The purpose of this class is to build the statistics, programming, and qualitative skills that are required to excel in data science. Students will learn the skills required to conduct research using surveys and experiments, and will further develop their programming abilities in R. The substantive focus of the class will largely be on topics related to politics and elections, although the technical skills can be applied to any subject matter. It is expected that students come in having some experience using R, which can be acquired by taking either PSCI 107, PSCI 338 or an equivalent course.

Data science jobs, particularly entry-level, junior analyst politics, will require statistics, programming, and qualitative skills in one or more of the following areas:

1. Implementation and interpretation of basic regression models
2. Running surveys and intepreting the results
3. Designing, implementing, and analyzing the results of experiments
4. Predictive modeling and machine learning

In this class, we will largely focus on the skills required to excel at (2) and (3), although we will touch on (1) and (4). One aim of the class is to equip each student with base set of skills to allow them to dive more deeply into any of the above topic areas in future coursework.

Expectations and policies

Attendance: Although we will post course materials on the class Canvas page, much of the instruction in this class will not be in the form of lecture slides or sample code. As such, it will not be possible to excel in this class without attending the class meetings. Although we will not take attendance, it will be obvious if you are regularly skipping lecture and it will impact the grade you receive in the class. If you have a valid excuse for multiple absences (such as a lengthy medical issue, but not multiple job/internship interviews scheduled during class meetings), please alert the teaching stuff so we can work with you to accommodate the situation.

Readings and weekly assignments: Throughout the semester, we will assign readings and homework assignments to be completed before coming to class. These will be a mixture of academic articles, readings from textbooks, popular press articles, and tutorial videos about R programming topics. We will expect that you have completed the assignment prior to arriving to class.

Computers and phones: Laptops, phones, tablets, and other electronics must be put away for the duration of the class meeting. This means your phones or computers must be completely out of sight throughout the class, not just sitting on top of your desk. The only exception will be on days in which we are working in R, which are noted in the class schedule. We may also make exceptions for some students to take notes on their computer, but you must receive prior approval from the teaching staff.

Academic integrity: We expect all students to abide by the rules of the University and to follow the Code of Academic Integrity.[1]. Self-plagiarism is still plagiarism. Turning in the same assignment or paper for two classes (or turning in an assignment from a previous class) without prior approval by the instructors of *both* classes is considered academic dishonesty and plagiarism.

Learning to collaborate is an important part of data science. We encourage you to help each other on problem sets and other class assignments. Ultimately, however, the write-up and code that you turn in must be your own creation. Similarly, you can discuss coding strategies or code debugging with your classmates, but you should not share code in any way. Please write the names of any students you worked with at the top of the problem set.

Re-grading of assignments: All student work will be assessed using fair criteria that are uniform across the class. If, however, you are unsatisfied with the grade you received on a particular assignment (beyond simple clerical errors), you can request a re-grade using the following protocol. First, you may not send any grade complaints or requests for re-grades until at least 24 hours after the graded assignment was returned to you. After that, you must document your specific grievances

[1] http://www.upenn.edu/academicintegrity/ai_codeofacademicintegrity.html

in writing by submitting a PDF or Word Document to the teaching staff. In this document you should explain exactly which parts of the assignment you believe were mis-graded, and provide documentation for why your answers were correct. If we approve your request for a re-grade, your assignment will be re-scored by a different grader than the original one. The new grader will re-score the entire assignment (including portions for which you did not have grievances), and their score will be the one you receive on the assignment (even if it is lower than your original score).

Assessment and grading

- Attendance, participation, and engagement (15%)

  As described in the expectation section

- Problem sets (40%)

  Six problem sets (roughly every two weeks)

  Graded on a ✓+, ✓, ✓−, 0 scale

  No penalty for your first late submission, as long as it is turned on within 5 days of the due date. Any late submissions after that will receive a zero, unless you have a valid (university designated) excuse

- Semester-long project (15% 'midterm' checkpoints; 30% end-of-semester final product)

  Most (or all) problem sets will include a section which will be graded separate of the problem set and, cumulatively through the semester, will make up the 20% 'midterm' checkpoints for the semester-long project. This will allow you build toward the final, end-of-semester product

Computing

We will use R in this class, which you can download for free at www.r-project.org. R is completely open source and has an almost endless set of resources online. Virtually any data science job you could apply nowadays to will require some background in R programming.

While R is the language we will use, RStudio is a free program that makes it considerably easier to work with R. After installing R, you should install RStudio (www.rstudio.com). Please have both R and RStudio installed by the end of the first week of classes.[2]

A significant number of the course meetings and lectures will be focused on learning R. For those class sessions, we would like everybody to bring their laptop so that you can follow along. If you do not have a laptop, or have any other constraint, please let us know and we will find a way to accommodate you.

[2]If you're having trouble installing either program, there are more detailed installation instructions on the course Canvas page.

Textbook

There is no assigned textbook for this class. Below is a list of recommended textbooks focused on R programming. We don't expect you to purchase any of them for this class, but we have found them to be useful resources in case you want to get a copy at some point.

- For programming with R and the Tidyverse:
  R for Data Science by Hadley Wickham and Garrett Grolemund.

- For doing social science statistics in R:
  Political Analysis Using R by James E. Monogan III
  Quantitative Social Science: An Introduction by Kosuke Imai

- For implementing basic to advanced machine learning techniques in R:
  An Introduction to Statistical Learning by James, Witten, Hastie, and Tibshirani. Download free at: http://www-bcf.usc.edu/∼gareth/ISL/

Course Schedule

Wednesday, Aug. 28

Lecture: Course overview - Goals for the course; What is data science?

Monday, Sept. 2

No class (Labor Day)

Wednesday, Sept. 4

Lecture: Writing survey questions - questionnaire design

Monday, Sept. 9

R: Principles of good programming; writing clean code; reading data into R; debugging code

Wednesday, Sept. 11

R: Conditional logic; subsetting data

Monday, Sept. 16

R: Cleaning and reshaping data

Wednesday, Sept. 18

Lecture: Sampling - types of samples (SRS, RBS, stratification); modes of sampling

Monday, Sept. 23

R: Recoding variables and other data cleaning

Wednesday, Sept. 25

Lecture: Survey biases - priming; ordering effects; non-response

Monday, Sept. 30

R: Data merging; loops; writing fast code

Wednesday, Oct. 2

R: User-defined functions

Monday, Oct. 7

R: Aggregating and summarizing data

Wednesday, Oct. 9

Lecture: Survey weighting

Monday, Oct. 14

R: Weighting survey data in R

Wednesday, Oct. 16

Lecture: Principles of writing with data

Monday, Oct. 21

R: RMarkdown

Wednesday, Oct. 23

Lecture: Principles of data visualization

Monday, Oct. 28

R: Data visualization

Wednesday, Oct. 30

R: Mapping data

Monday, Nov. 4

R: Basics of RShiny

Wednesday, Nov. 6

R: More RShiny

Monday, Nov. 11

R: Data visualization activity

Wednesday, Nov. 13

Lecture: experiments - basics of experiments; survey experiments

Monday, Nov. 18

Lecture: Experiments - confounding; randomization; balance

Wednesday, Nov. 20

R: Randomization; analyzing experimental results

Monday, Nov. 25

Lecture: understanding web scraping

Wednesday, Nov. 27

No class: Friday schedule

Monday, Dec. 2

R: basic web scraping

Wednesday, Dec. 4

R: advanced web scraping

Monday, Dec. 9

Wrapping up