POLSCI 514 F 1:00 - 2:00 PM, Room: 7603 Haven Hall

Instructor: Kevin McAlister, 6756 Haven Hall

Office Hours: 1:30 - 2:30 PM Mondays, 1:30 - 2:30 PM Wednesdays, 4:00 - 5:00 PM Thursdays and by appointment.

Prerequisites:*

1. `R` programming experience at the level of POLSCI 514
2. Statistics at the level of POLSCI 599
3. Math at the level of POLSCI 598

Course Outline:

1. Computational Methods of Statistical Analysis
    - Numerical Differentiation and Integration
    - Monte Carlo Methods and Simulation
    - Markov Chains and MCMC
    - Optimization Procedures (light introduction to EM)
2. Big(-ish) Data in `R`
    - Introduction to `library(data.table)`
    - Memory Management and Chunking Algorithms
    - Parallel Implementation in `R`: `library(snowfall)` and `library(foreach)`
    - Handling Bigger Data: AWS and Flux
3. Data Collection, Management, and Cleaning
    - Scraping the Web for Data
    - Data Management: Databases and Intro to SQL
    - Data Processing Using UNIX Tools
    - Data Streaming
4. Other Stuff
    - Dynamic Programming and Game Theory
    - Evolutionary Game Theory

*If you are unsure about the math requirements, feel free to come chat with me and we can determine what you might need to do over the semester.

- EITM approaches
- Introduction to Machine Learning: Classification Approaches

Grading: The final course grade will be made up of three parts:

1. Attendance/Participation (20%): Regular attendance is expected. Similarly, I expect that you will come to class ready to work through the material and contribute some insight about the material at hand. Throughout the semester, I will post papers and slides on our Canvas site. These will be related to the topics that we are going to cover in the lecture. Though I do not expect you to have fully read these and completely understand the material prior to our lectures, a light skimming of the material is something that I find really helpful.

2. Homeworks (40%): Homework assignments will be given every one or two weeks. These assignments are intended to give you hands on experience with many of the topics that we will be covering in this course. In my view, the only way to learn to be an effective programmer is to have directed practice. These homework assignments are intended to be difficult and really challenge your overall ability in `R`. Collaboration with your peers is encouraged on these homework assignments! However, I do expect that each person will turn in their own individual answers. Homework should be turned in on Canvas in the corresponding assignment link. Each homework should include a PDF with your answers and a .txt file with your code. All coding should be done in `R` unless otherwise explicitly stated. Your `R` code should follow the Google `R` style guide. Variable names should be informative and the code should be thoroughly commented. Homework assignments will be graded on a pass/fail basis. If you turn in your homework assignments and it shows that you have made a concerted effort to complete the work, then you will pass. The goal of these homeworks are not to take up a lot of time, but, rather, give you directed practice with the materials covered in the course.

3. Project (40%): At the end of this course, each of you should complete a project that utilizes some of the topics that we have covered in this class. As with the homework assignments, I truly believe that the only way to really gain skills as a programmer is to complete tasks that interest you. For this reason, I want each of you to find some question that is relevant to your interests and research and create a short presentation of your findings. This project does not have to be a fully thought out research paper. In fact, the deliverable for this project should be a maximum of 5 pages. Your project can be a statistical analysis, an exercise in data collection, utilize topics in computational game theory, etc. The only real requirement is that your project utilizes something that we have covered in class. In the final one or two weeks of the semester, I'll ask that each of you make a few slides about your project and present your work to the class.

Office Hours and Emails: I will be available for homework and project help in my office hours. Over the course of the semester, I do ask that you come by my office hours at least once to let me know what you are thinking of doing for your final project. This way, I can at least know/let you know that you are on the right track.

One rule that I do have for this class is that I will not answer coding questions via email. The reason for this is twofold. The first is that it is often very difficult to diagnose code problems without

seeing and discussing your full code. In order to maximize my own usefulness to you, talking in person is better. Second, and probably more important, is that a large portion of being a successful programmer is learning to troubleshoot your own code. I can almost guarantee you that almost any question you may have about `R` has been asked before and the answer exists somewhere on the Internet. Being able to effectively find answers and mimic coding practices of others is an important skill. As your own research progresses and you branch into areas of programming that I may not even know about, being self-sufficient in troubleshooting is a key skill.

Troubleshooting Steps:

1. Read any corresponding documentation that comes along with any package in `R` that you may use.
2. Search for your problem on StackExchange.
3. Ask colleagues if they have had similar problems and what solutions they may have come up with.