ECON 21410: Computational Methods in Economics

University of Chicago, Spring 2014

ADMINISTRATIVE

Class: Tuesdays and Thursdays at 10:30am in Stuart 102

TA Session: Mondays at 5pm in Stuart 222.

Lecturer: John Eric Humphries   johneric@uchicago.edu

Office Hours: Mondays and Wednesdays 9:30-10:30am in the Stuart Cafe.

Teaching Assistant: Oliver Browne   obrowne@uchicago.edu

Office Hours: Thursday from 5-6 pm in Rosenwald 301 (this will serve as either office hours or a second informal T.A. session).

Website: The class will not use Chalk heavily. Homeworks will be posted on my website at http://home.uchicago.edu/johneric/teaching as well as on our github organization's page at: http://www.github.com/compecon .

ABOUT THE COURSE

This course aims to prepare students to: begin empirical research, be effective research assistants, and be more prepared for graduate school. This is roughly the same preparation needed for empirical work in the finance or consulting. Aiming to complement the theoretical training offered to top undergraduates at the University of Chicago, this class will focused on applying economic models and econometric methods.

The course is designed to equip students with the skills to:

1. take theoretical models and translate them into useful economic simulations and empirical tools.

2. use programming for solving economic problems.

3. apply numeric methods to solve economic problems.

4. develop, implement, and manage an empirical project.

5. produce professional output to clearly convey your results.

While these five goals are quite broad, they are the five foundational skills necessary for effective empirical research.

The course will focus on R, which is powerful, flexible, and relatively straight forward. You will be required to complete your exercises in R. Learning the basics in R will help you learn python, STATA, C++, Julia, or any other language you may prefer later.

Concrete Objectives

By the end of these two sessions, the goal is to equip you with the skills to:

- Read and understand R code.

- Write clear, documented, and reusable R code.

- Summarize empirical work using LaTeX

- Use packaged functions and libraries to do work for you.

- Implement your own econometric methods and simulations to solve problems others have not already solved for you.

- Solve optimization problems numerically.

To accomplish the list of objectives above, we will need to review the basics quickly. If you have never used R, I highly recommend spending a couple hours completing the online exercises available at www.codeschool.com/courses/try-r or www.datacamp.com. For additional review material, see the references below. If you have never programmed or

never used R, I strongly recommend spending a few hours over spring break taking these online courses or reading introductory material.

To get ahead of the curve, please install the following software:

- R: http://cran.r-project.org/

- R Studio: http://www.rstudio.com/ide/download/

- LaTeX: http://latex-project.org/ftp.html

- You may wish to install Lyx, a nice wrapper for LaTeXthat may be easier for people more familiar with MS Word: http://www.lyx.org/Download

- Alternatively Texmaker provides a nice graphical user interface: http://www.xm1math.net/texmaker

- If preferred, you can use an online latex editor such as sharelatex.com.

- Install the version control software "git": http://git-scm.com/book/en/Getting-Started-Installing-Git

LAYOUT

The course will consist of several small projects which involve simulation or estimation of economic models. On Tuesdays we will lay out the economic model we are working with that week and on Thursdays we will cover the requisite empirical and computational methods needed to complete the assignment. Additional support will be given in TA sessions. You do not need to bring a computer to class. I will do some live coding in class, but I will always email these files to the class afterwards.

COURSE TIMELINE

WEEK 1   A rapid introduction to R - solving linear regression multiple ways.

WEEK 2   Learning basic data management and control-flow in R - a simulation of Schelling's segregation model.

WEEK 3 Simulating Becker's marriage model and solving for optimal marriage matches using the Gale-Shapley algorithm.

WEEK 4 Understanding selection, a simulation of the Roy model.

WEEK 5 Correcting empirical estimates for selection bias (applying recently learned tools to real data).

WEEK 6-7 Further learning to work with data through replicating a paper on firm entry.

WEEK 7-10 To be determined based on the level and interests of the students in the course. May include simulating a simple full economy, job-search, parental investment, a simple game-theory model from industrial organization, or a trade application which requires basic linear programming.

FINAL The course will conclude with a final project. This will involve replicating a simplified version of a recent paper. Alternatively students may complete the final project with a research paper. We will also have a short written or computer based final to test the basic programming skills and concepts taught in the course.

RESOURCES

Here is a list of useful R resources:

- www.codeschool.com/courses/try-r

- www.datacamp.com

- http://cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf (slightly out of date, but a short useful guide

- http://link.springer.com/book/10.1007/978-0-387-77318-6 (also slightly out of date, free to download on the campus network)

- http://files.itslearning.com/data/ku/103018/teaching/lecturenotes.pdf

- search "with R" on springerlinks.com

GRADING

- 55% homework projects

- 10% final exam

- 15% final project

- 10% participation

- 10% side projects

As this is the first time this course has been taught, I reserve the right to change this breakdown if it becomes necessary. If we feel homework is not done independently enough to identify individual effort and undestanding, additional weight will be shifted to the final and final exam.

Suggestions for side-projects will be provided with each homework, but your own side-projects can be approved by the TA or myself. Side-projects are designed to let you explore additional methods or details that interest you. One point towards side projects earns one percentile of the 10 possible for side projects.

Participation will be based on attendance and engagement in class, but can also be fulfilled through coming to office hours, contributing to the class wiki on github, or answering and asking questions on the class github page (this will be explained further in a TA session).

PREREQUISITES

The prerequisites listed online were incorrect. You should have taken econ 200-201, econ 209 (or 210) and math 195-196 (or analysis). Exposure to some programming language will be very useful, but is not required.

POLICIES

We take cheating and plagiarism VERY seriously. Every class you have ever take probably states that cheating will not be tolerated, but we mean it.

Cheating or plagiarism will result in a 0 on that assignment with possible additional repercussions. You are welcome to work together in groups up to 3, but you are required to submit your own write-up and your own code. Please take precautions to avoid putting the TA or myself in a situation where we are forced to decide if two documents are "too similar". As future researchers, consultants, bankers, etc, learning to do honest work in a timely manner is more important than getting everything correct.

You are required to write the names of your other group members on your assignment. If you discuss the homework in detail with people not in your group, please note this in a footnote.

Homeworks submitted late, but prior to the TA session will receive half credit. We will not accept homework assignments submitted after the TA session where answers are discussed. If you have a conflict, please plan in advance. We will consider granting extensions requested more than 5 days in advance, as well as documented family emergencies.