Big data in economics
(EC 410/510)

SPRING 2020 SYLLABUS

Grant R. McDermott
Dept. of Economics, University of Oregon

Summary

When:	Tue & Thu, 14:00–15:20	
Where:	Remote! (A Zoom link will be provided.)	
Web:	https://github.com/uo-ec510-2020-spring	
Who:	Grant McDermott (instructor) 🎓 Assistant Professor of Economics ✉ grantmcd@uoregon.edu ⌛ Tue & Thu, 15:30–16:30	Garrett Stanford (GE) 🎓 Doctoral student in economics ✉ gos@uoregon.edu ⌛ Wed 14-00–15-00

Course description

Data are getting bigger. As data get big (i.e. they cannot fit in your computer's memory) the conventional empirical tools of the applied economist's toolbox often become inefficient or even ineffective. In this course, we will introduce, discuss, and implement some of the key tools that have been developed to overcome these challenges. We will also see how these "big data" tools be profitably repurposed for "medium data" settings too. We will start by laying the foundations for effective data science work, covering topics like version control and the shell (i.e. command line). From there, we will learn how to handle data and become efficient programmers in *R* (our primary computational environment). While our immediate focus will be on making the most of the local resources at our disposal, the skills that we master here will serve us well once we scale up to dedicated big data environments in the final section of the course. By the end of the quarter, you will have connected to cloud-based based services and high-performance computing clusters, queried petabyte-sized databases, and run distributed code across a network of computers. More importantly, you will have a better understanding of how computers work, what tools are at your disposal for tackling big data problems, and how to meaningfully integrate them into your everyday workflow.

Practical matters

Class rules

Have your laptops ready. This will be a very hands-on course, with relatively little in the way of formal theory. Instead, we'll be working through the lecture notes together in real-time, and you'll be running code on your own machines during class.

Software requirements

All of the software requirements for this course are open-source and/or free. Please aim to have everything installed by the start of our first lecture. I will be available for installation troubleshooting during the first week of the quarter. If you want a detailed tutorial on how to achieve a perfect working setup, I can think of no finer guide than Jenny Bryan *et al.*'s `http://happygitwithr.com/` (see esp. sections 4 – 15).

R and RStudio

We will mainly be using the statistical programming language ***R*** (download/update here). Please make sure that you install the **RStudio IDE** too (I recommend the preview version, which you can download here).

Git and GitHub Classroom

We will also make extensive use of the **Git** version control system (follow the OS-specific installation instructions here). Once you have installed Git, please create an account on **GitHub** (here) and register for an education discount to get unlimited private repos (here).[1] Now is probably a good time to tell you that I am going to run the entire course through GitHub Classroom. You will receive an email invitation to the course repo with instructions in due time, but suffice it to say that this is how we'll submit assignments, provide feedback, receive grades, etc.

Other

You are ready to start this course once you have installed R, RStudio, and Git (as well as created an account on GitHub). I will provide instructions for any further software requirements as the need arises; i.e. when we get to the relevant lecture. On that note, the lectures have all been posted ahead of time on the course website. Each lecture lists all of the *R* packages and external libraries (if relevant)

[1] GitHub recently announced unlimited free private repos for everyone. However, you are limited to three collaborators per private repo, so the education discount still makes sense.

required for a particular class. I'll try to remind you, but my expectation is that you will look at these requirements and ensure that you have them installed *before* we start class. The last thing I want you to do for now is make sure that your system is configured to handle some additional packages that we will be using down the line. This varies by operating system:

- **Linux:** You should be good to go.
- **Mac:** Install the *Homebrew* package manager (see here). I also recommend that you make sure your C++ toolchain is configured/open (see here; don't worry, it's simpler than it sounds).
- **Windows:** Install *Rtools* (see here). While it's not essential, I also recommend that you install the *Chocolatey* package manager for Windows (see here).

Textbook and other readings

There's no set textbook for this course (Ed Rubin and I are working on one). However, I can eagerly recommend the following, which are available for free online if you don't feel like buying a hard copy:

- "***R* for Data Science**" (Grolemund and Wickham)
- "**Advanced *R***" (Wickham)

Evaluation and grading

Grade determination

Grades will be determined very simply as follows:

EC 410		EC 510	
4 × HW assignments (25% each)	100%	4 × HW assignments (20% each)	80%
		OSS contribution	20%

Note: A class participation bonus worth an additional 2.5% will be awarded at my discretion.

Any changes or specific requirements will be made clear as we proceed through the course. In the meantime, here are some additional details.

Homework assignments

Homework assignments are to be completed in **teams of two**. Late submissions will not be graded. There is no final exam or project for this course.

OSS contribution (EC 510 only)

You are going to contribute to open-source software (OSS) in some way, shape, or form. This could be by identifying and correcting bugs in a package that you use. Or, it could be by contributing material (e.g. documentation) to an open-source project. This year, I particularly want to encourage you to contribute to the Library of Statistical Techniques (https://lost-stats.github.io/). There's clearly quite a bit of leeway here and I'll need to sign off on whatever you propose. Similarly, depending on the scope and size, you may need to make several different contributions to fulfill the requirement.

Honesty and academic integrity

Students caught cheating or plagiarizing will automatically be assigned a zero grade. Please acquaint yourself with the Student Conduct Code at http://studentlife.uoregon.edu.

Accessibility

If you have a documented disability and anticipate needing accommodations in this course, please make arrangements with me during the first week of the term. Please also request that the Accessible Education Center send me a letter verifying your disability. Students with infants or young children that need ongoing care should similarly come and see to me. We'll have to take it on a case-by-case basis, but I'll do my utmost to accommodate you.

Tentative lecture outline

Note: We only have 80 minutes allocated for each lecture. I expect that several individual topics will run over two or more lecture slots. Please bear that in mind as you look over this tentative outline.

Foundations

Expected no. of lectures slots: 5 (6)

- Introduction: Motivation, software installation, and data visualization
- Version control with Git(Hub)
- Learning to love the shell
- *R* language basics (*optional*)

Data wrangling, I/O, and acquisition

Expected no. of lectures slots: 5

- Data cleaning and wrangling: (1) Tidyverse
- Data cleaning and wrangling: (2) data.table
- Big data I/O
- Webscraping: (1) Server-side and CSS
- Webscraping: (2) Client-side and APIs

Programming

Expected no. of lectures slots: 4

- Functions in *R*: (1) Introductory concepts
- Functions in *R*: (2) Advanced concepts
- Parallel programming

Cloud resources and distributed computation

Expected no. of lectures slots: 6

- Docker
- Cloud computation (Google Compute Engine)
- High performance computing (UO Talapas cluster)
- Databases: SQL(ite) and BigQuery
- Spark

Other potential topics (time permitting)

- Regression tools for big data problems
- Google Earth Engine
- Networks
- Deep learning
- Automation and workflow
- Rcpp (i.e. integrating C++ with R)

FAQ

This course looks interesting! Can I use/adapt your lecture notes for a similar course that I'm teaching at XYZ?

Sure. I've benefited greatly from other people making their teaching materials publicly available (and have tried my best to acknowledge them directly in the relevant sections of this course). Say nothing of the incredible open-source software that powers everything. I'm more than happy to pay it forward. I only ask two favours. 1) Please let me know (email/Twitter) if you do use material from this course, or have found it useful in other ways. 2) A minor acknowledgment somewhere in your own syllabus or notes would be much appreciated.

How can you teach a "big data" class without a dedicated machine learning section??

Normally, you can't. But given that this class forms part of an MSc sequence that already contains Ed Rubin's fantastic machine learning and prediction course (EC 524), I'm going to use our limited time to focus on other things. We will, however, encounter various ML methods and algorithms in the course of discussing other topics.

Okay, is there anything else that you aren't covering that I should know about?

The obvious thing that springs to mind is workflow automation and analysis pipelines (make files, etc.). Again, triage rules the day. We will, however, be working extensively with R Markdown documents, which is at least a big step in the direction of self-contained analysis. And I'm more than happy to point students in the right direction if anyone wants to learn more. (Here, here, and here are great places to start.) Another thing we won't have time for is package development and maintenance, although I don't see this class as the primary audience for that. OTOH, students will be rewarded for package contributions if they choose to do so in the peer-review section of the course.

***R* looks cool, but I'm more familiar with Python/Julia/MatLab/etc. Can I use that instead?**

Short answer: No. Longer answer: Look, I like and use a lot of those languages too, but I'm not changing my lecture notes or assignment templates for you. Plus, I really do think that *R* makes the most sense for applied economists looking to develop their data science skills. It already has all of the statistics and econometrics support, and is amazingly adaptable as a "glue" language to other programming languages and APIs. Learning multiple languages is never a bad idea in the long run, though.

I already have a BitBucket/GitLab/etc. account. Do I still have to use GitHub?

Since I'm running this course through GitHub Classroom, yes. But good for you! (Seriously... those are great platforms too and as an open-source advocate, I fully support a plurality of tools and software options.)

On that note, do you have any advice for running a course on GitHub Classroom?

I mostly followed this excellent tutorial by Jacob Fiksel.