Data science for economists
(EC 607)

WINTER 2021 SYLLABUS

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

Summary

When:	Tue & Thu, 10:15–11:45
Where:	Remote! A Zoom link will be sent to you.
Web:	https://github.com/uo-ec607
Who:	Grant McDermott 🎓 Assistant Professor of Economics ✉ grantmcd@uoregon.edu ⌛ Mon & Wed, 09:00–10:30

Course description

This seminar is targeted at economics PhD students and will introduce you to the modern data science toolkit. While some material will likely overlap with your other quantitative and empirical methods courses, this is not just another econometrics course. Rather, my goal is bring you up to speed on the practical tools and techniques that I feel will most benefit your dissertation work and future research career. This includes many of the seemingly forgotten skills — like where to find interesting data sets in the "wild" and how to actually clean them — that are crucial to any successful scientific project, but are typically excluded from core econometrics and statistics classes. We will cover topics like version control and effective project management; programming; data acquisition (e.g. web-scraping), cleaning and visualization; GIS and remote sensing products; and tools for big data analysis (e.g. relational databases, cloud computation and machine learning). In short, we will cover things that I wish someone had taught me when I was starting out in graduate school.

Practical matters

Class rules

~~Please bring your laptops to class. This will be a very hands-on course, with relatively little in the way of formal theory. Instead, we'll be working through lecture notes together in class and you'll be running code on your own machines.~~ **Update:** With COVID-19 pushing us to remote classes, I'll be changing how I teach this course. The most important change is that I'll be delivering lectures *asynchronously*, essentially flipping the classroom. My expectation is that you'll watch and work through the pre-recorded lecture videos before class. We'll reserve actual class time for two things: 1) student presentations and 2) troubleshooting and follow-up from any of the lecture material. We may need to adapt as the quarter develops, but that's what we'll start out with and stay flexible.

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 here). Please make sure that you install the **RStudio IDE** too (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 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.

[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.

Other

You are ready to start this course once you have installed R, RStudio, and Git (as well as created an account on GitHub). 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. I also recommend that you make sure your C++ toolchain is configured/open. Don't worry, it's simpler than it sounds. Just download the macOS Rtools installer and follow the instructions.
- **Windows:** Install Rtools. While its not essential, I also recommend that you install the Chocolatey package manager for Windows.

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) 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.

Textbook and other readings

There's no set textbook for this course (Ed Rubin and I are working on one). The lecture notes are pretty detailed and are thus "self-contained". However, I've drawn inspiration from various sources; a few of which are listed below. You don't *need* to buy or read any of these (excellent) books to complete the course. But I can eagerly recommend leafing through at least one or two of them. Each of these books is freely available online if you can't afford a hard copy:

- "**Data Visualization: A practical introduction**" (Kieran Healy)
- "***R* for Data Science**" (Garrett Grolemund and Hadley Wickham)[2]
- "**Advanced *R***" (Hadley Wickham)
- "**Geocomputation with *R***" (Robin Lovelace, Jakub Nowosad and Jannes Muenchow)
- "**Spatial Data Science**" (Edzer Pebesma and Roger Bivand)
- "**An Introduction to Statistical Learning**" (Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani)

[2] FWIW, Jake VanderPlas's "**Python Data Science Handbook**" is excellent option for anyone looking for a Python equivalent.

- Etc.

Taking a step back, one of the goals of this course is to make you aware of the incredible array of instruction material that is freely available online. I also want to encourage you to be entrepreneurial. In that spirit, many of the lectures will follow a tutorial on someone's blog tutorial, or involve reproducing an existing study with open source tools. Each lecture will come with a set of recommended readings, which I expect you to at least look over before class.

Evaluation and grading

Grade determination

Grades will be determined as follows:

4 × homework assignments (20% each)	80%
1 × short presentations	10%
1 × OSS contribution	10%

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

This breakdown should (hopefully) be pretty self-explanatory. Specific requirements will be made clear as we proceed through the course. Here are some additional details, though:

Homework assignments (and/or final presentation)

Homework assignments are to be completed individually. Late submissions will not be graded. There is no final exam or project for this course. However, you have the option of swapping out one of the individual homework assignments for a final (20 min) presentation of your own research. Think of this as an opportunity to develop and refine one of your PhD projects using the tools that we will cover in this course. In particular, some of you may wish to present your second-year field paper, or a dissertation chapter idea. You are allowed to do this individually or in pairs. However, please note the following caveats: 1) You need to get prior approval from me and let me know which HW assignment you are dropping. 2) These final presentations will only be graded on content relevant to this course. (Don't present a theory paper!)

Short presentations

Almost every lecture will begin with a short student presentation. These should last 5–10 minutes and will cover a prescribed topic (i.e. that I have either allocated or approved ahead of time). Some presentations will involve summarising a key reading or topic of relevance to the main lecture for that day. Other presentations are a chance for you to describe a software package or analytical method of your choice — again, subject to my approval. I will provide a list of slots, as well as prescribed and suggested topics via GitHub Classroom. Topics will be assigned on a first-come-first-go basis. But don't be surprised if I volunteer you for something.

OSS contribution

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. 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.

Lecture outline

Data science basics

1. Introduction: Motivation, software installation, and data visualization
2. Version control with Git(Hub)
3. Learning to love the shell
4. *R* language basics
5. Data cleaning and wrangling: 1) tidyverse and 2) data.table
6. Webscraping: (1) Server-side and CSS
7. Webscraping: (2) Client-side and APIs

Analysis and programming

8. Regression analysis in *R*
9. Spatial analysis in *R*
10. Functions in *R*: (1) Introductory concepts
11. Functions in *R*: (2) Advanced concepts
12. Parallel programming

Scaling up: Big data and cloud computation

13. Docker
14. Virtual machines / cloud servers (Google Compute Engine)
15. High performance computing (UO Talapas cluster)
16. Databases: SQL(ite) and BigQuery
17. Spark
18. Machine learning: (1)
19. Machine learning: (2)... Or, student project presentations (demand dependent)
20. Peer-review and student project presentations (demand dependent)

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.

The other data science courses that I've seen all have at least one whole lecture dedicated to data visualization. Where's yours?

Every lecture in this course is dedicated to data visualization! Okay, seriously, we'll cover the basics of ggplot2 in the opening lecture (and first assignment) and consistently build upon that in subsequent weeks. Much as I'm tempted to have a standalone lecture on the topic, I have to triage because of the time constraints of a 10-week course. I don't want to run out of road before we can get to some of the big data stuff towards the end of the course. Trust me, though. There will be a *lot* of data visualization in this course.

What about regular expressions? I hear those are super important too.

100% agree and, much like data visualization, I've tried to include examples throughout the course rather than in a standalone lecture. I'm confident that you will have a solid grip of the basics by the time we get to the end of the quarter.

I hear that data scientists use Bayesian methods a lot. Will you be covering those in depth?

Sadly, no. I'm a Bayes fanboy (as my research interests will attest), but again have to think about time constraints. The good news is that running Bayesian models in *R* is super easy thanks to a multitude of packages, and you will be very well positioned to jump right into these after finishing this course. We might even get to an example or two in the lecture on regression analysis. The even better news is that Jeremy Piger teaches an excellent Bayesian course here at the UO that you should attend.

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/etc. Can I use that instead?**

Short answer: No. Longer answer: Look, I like and use 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.