

# Economics 312: Data Science Tools for Social Scientists

## Fall 2022

### Syllabus

September 1, 2022

**Instructor:** Dr. Alexandr Moskalev  
(he/him/his)  
[moskalev@oberlin.edu](mailto:moskalev@oberlin.edu)

**Class:** Monday & Wednesday, 2:30 pm to 4:20 pm, Rice Hall 100B

**Office Hours:** Monday 5 - 6 pm, Rice Hall 210 *and by appointment.*  
Tuesday 12:30 - 1:30 pm, Rice Hall 210  
Wednesday 5 - 6 pm, Rice Hall 210

Please check <https://alexmoskalev.com/officehours> for any changes and to request an appointment.

**Course Description:** This course serves as an introduction to tools, environments and workflows that are frequently used by data scientists working in the industry. Students majoring in Economics and other social sciences often get mastery of econometrics and testing hypothesis with secondary (non-experimental) data, but they struggle to implement their analysis in modern data-oriented business environments. Common problems include unfamiliarity with general purpose scripting/programming languages, command line interfaces, shell scripting, remote and cloud computing; inability to automate and perform batch operations, to version and share work in progress, to package a solution for production; lack of knowledge and experience in querying databases, in programmatically accessing APIs, and in parsing unstructured datasets.

This course is not meant to teach statistics, econometrics, or data science methods. Instead, this course will focus on tools, implementation approaches, and hands-on experience that should improve student's chance for success at doing applied data science in a business-oriented environment.

### Learning Goals:





- Learn basics of coding in Python.
- Gain mastery of working with CLI environments.
- Obtain hands-on data engineering experience.
- Use formal reasoning/mathematical methods, tools, technology, and calculation to solve problems.
- Make judgments and draw appropriate conclusions based on the quantitative and/or formal analysis of data.
- Relay results in a manner appropriate to the audience using suitable terminology, symbols, and conventions.


**Prerequisites:** ECON 101 and (STAT 113 or ECON 255) or Instructor's permission.

Please note that prerequisites in this course are meant to select students studying social sciences and interested in improving their applied data science skills. If you think that this course might be beneficial to you, please do not hesitate to reach out.

**Readings:** The course does not have a required reading. I may publish links to online resources that you should explore to gain further knowledge on the topics covered.


### Grading:

50%		Data Analysis Project
20%		Python & SQL coding practice
20%		Class participation
10%		In-class presentation

 *Data Analysis Project:* In a group of 2 - 3 students you will propose and implement a data analysis project. You should be able to demonstrate mastery of skills related to one or more topics covered in this course (or of another related data analysis skill of a comparable level).

I will use the following breakdown of the 50% that data analysis project contributes to your course grade:

- (10%) *Description of the project ideas* **due September 20 at 5:00 pm.** Please describe the ideas you have for a data-analysis project. You should put more emphasis on ideas that you think are doable (obtaining/scraping data, selecting a framework/stack, coding, describing the results) within the rather short timeframe we have in this course. Please feel free to stop by my office hours to discuss your ideas and get feedback on what ideas seem to be the most doable.
- (10%) *A selected idea and an implementation plan* **due October 4 at 5:00 pm.** Based on the discussion of the ideas proposed on the previous step (i.e. you should visit me during the office hours and get feedback on your proposed ideas), please select a single idea and provide an implementation plan. In particular, please focus in detail on technologies involved, scripting/coding challenges to solve, timeline for data collection, and provide any proof-of-concept examples you may have.
- (10%) *A project showcase* **due October 31.** Please prepare a brief presentation that explains the essence of your project to your fellow classmates. Please make sure to highlight the technical challenges you encountered (mention the solutions you've found), describe a line from the dataset you've collected, explain what you want to do with the data and what question(s) you're trying to answer. You will present in class as a team.
- (10%) *A rough implementation* **due November 29 at 5:00 pm.** Please attempt to finish your project by this date. I will try to take a look at the projects in early December to give quick feedback. Make sure to include a good description of what's going on (i.e. I should not guess what you were trying to do by only looking at the code).
- (10%) *A final implementation* **due December 21 at 4:00 pm.** Please submit your final version of the data analysis project.

 *Python & SQL coding practice:* You will use an external learning-to-code platform to improve your SQL, Python coding, and problem-solving skills. The badges you earn at the

platform for these skills will be converted into points in this category. Since students likely came into this course with varying degrees of coding skills, I will use a non-linear transformation to shrink the distribution scores in this category. This will preserve the order, but will reduce the gap between the lowest and the highest performing students.

★ *Class participation:* To measure class participation, I will distribute attendance verification codes that you will credit to your learning portal account. You can also earn class participation points by doing additional in-class topic presentations.

📖 *In-class presentation:* Data science is a very fast-paced field. The approaches and technologies change quickly, and you should be able to adapt and to keep learning after completing this course. One of the best way to learn is to try to explain something to others. You will select one topic from the schedule and make an in-class presentation aimed to teach us all on how to use it in practice. First, please select a topic from the syllabus that you would like to present on. Second, please send me your presentation materials (not necessarily the presentation itself, but the materials you plan to use and share with other students) and a brief description of main talking points and examples at least one week in advance.







**Schedule:**

This is a very approximate schedule. The dates might not match the announced topics, and we likely won't be able to cover everything either. The additional topics (T+ days) are included in case we will progress faster than my expectation. You are also welcome to study ahead and ask questions during office hours.

	Introduction	Environment Setup
<b>Sep. 7</b>	<p style="text-align: center;">⚠️ <i>Learning Portal Registration</i></p> <p>Syllabus 📖 Syllabus</p>	<p>Python and Command Line (bash) 📖 See Anaconda setup instructions at <a href="#">official-website</a>. Linux/MacOS users already have a terminal, Windows users may look into WSL setup (and putty).</p>
	Python Fundamentals	Linux Fundamentals
<b>Sep. 12</b>	<p>Data types, variables, print statements 📖 Read <a href="#">about variables and types</a> and skim through <a href="#">print statements</a>.</p>	<p>Command Line Interface, Home Directory 📖 Explore “The Command Line” book chapter of Walsh’s <a href="#">book</a>. This is also a good introduction into Python.</p>
<b>Sep. 14</b>	<p>Functions 📖 Read <a href="#">about functions</a>.</p>	<p>File system navigation, Most common commands 📖 Read the “Command Line Cheatsheet” from the <a href="#">chapter above</a>.</p>
<b>Sep. 19</b>	<p>Lists, tuples, and dictionaries 📖 Look through information <a href="#">about lists and tuples</a>, and then read about dictionaries.</p>	<p>ssh, ssh keys 📖 Read about <a href="#">ssh and ssh keys</a>.</p>
<b>Sep. 21</b>	<p>Making choices 📖 Read <a href="#">about conditions</a>.</p>	<p>IPv4 networking basics 📖 Read about <a href="#">Internet Protocol (v4)</a> and IP addressing and about ports and protocols.</p>

<b>Sep. 26</b> Repeating actions  Read <a href="#">about loops</a> .	Traffic routing basics, Virtual Private Networks  Read about <a href="#">the use of VPNs for securing corporate networks</a> .		
<b>Sep. 28</b> Working with files  Read <a href="#">about reading and writing files</a> .	Redirection and piping  Read about <a href="#">redirection and pipes</a> .		
<b>Oct. 3</b> Output formatting  Read parts 1 and 2 of <a href="#">string formatting guide</a> .	SSH port forwarding, SCP, rsync  Read about <a href="#">ssh tunnelling</a> .		
<b>Oct. 10</b> Package management, modules, environments  Read <a href="#">about pip</a> and <a href="#">about conda virtual environments</a> .	User accounts, super user, ACL  Read about <a href="#">user management and ACL</a> .		
<b>Oct. 12</b> Exceptions, debugging  Read <a href="#">about raising exceptions and debugging</a> .	Block storage  Read about <a href="#">storage concepts and explore the guide on linux filesystems</a> .		
<b>Python: pandas</b>		<b>Linux Fundamentals</b>	
<b>Oct. 24</b> Series and DataFrame, data types  Consider reading <a href="#">Evans' pandas cookbook</a> ; Read <a href="#">about pandas objects</a> , you may also want to take a look at <a href="#">Schafer's pandas tutorial series</a> .	Bash scripting basics, Environment variables  Read about <a href="#">variables, conditionals, loops, and functions</a> .		
<b>Oct. 26</b> CSV files, IO methods  Read <a href="#">about reading data from a csv file</a> .	Package management, Linux distributions  Read an <a href="#">overview of package management in Linux</a> .		
<b>Python: pandas</b>		<b>regex</b>	
<b>Oct. 31</b> Basic methods and attributes  Read <a href="#">about the common data exploratory pandas methods</a> .	Matching alphanumeric characters  Read sections 2.1-2.6. Consider using <a href="https://regexr.com/">https://regexr.com/</a> (or a similar service) for practising.		
<b>Nov. 2</b> Indexing and selecting data, row iteration  Read <a href="#">about indexes</a> and <a href="#">about row iteration</a> .	Anchors, reserved and escaped characters, quantifiers and alternations  Read sections 2.7-2.11.		
<b>Nov. 7</b> Missing data, column operations  Read <a href="#">about missing data</a> .	Groups and references  Read section 2.12.		
<b>Nov. 9</b> Group by, groups  Read <a href="#">about pandas groupby</a> .	Advanced matches  Read section 2.13.		
<b>Nov. 14</b> Merges and joins  Read <a href="#">about merges and joins</a> .	grep/egrep and CLI integration  Read about <a href="#">egrep</a> .		

<b>Python: pandas</b>		<b>SQL</b>
<b>Nov. 16</b>	Cleaning data, applying functions  Read about applying a function to every row.	Select queries  Review <a href="https://selectstarsql.com/">https://selectstarsql.com/</a> to learn about SQL and practice by going through SELECT queries.
<b>Nov. 21</b>	SQL integration  Read about using SQL from pandas.	Aggregate functions  Read about aggregate functions.
<b>Python: matplotlib</b>		<b>SQL</b>
<b>Nov. 23</b>	Basic usage, pyplot  Read read section 1.5.2.	Group by queries  Read about GROUP BY queries.
<b>Nov. 28</b>	Axis, legend, layout  Read read section 1.5.3.	Nested queries  Read about nested queries.
<b>Nov. 30</b>	Colors, colormaps  Review section 1.5.4.	Joins  Read about joins.
<b>Dec. 5</b>	Text  Read about text in matplotlib plots.	Integration with Python  Explore pandas read_sql method.
<b>Python: API</b>		<b>git</b>
<b>Dec. 7</b>	Requests, request types, status codes  Read an introduction to python requests library.	Version Control Basics: init, add, commit, checkout, history  Read about version control and git basics.
<b>Dec. 12</b>	JSON data  Read about JSON data.	Branches, remote repositories  Read about git branches.
<b>T+1</b>	Query parameters and payload  Read more examples of python requests usage.	Staging, resetting  Read about interactive staging.
<b>T+2</b>	REST principles  Read REST and Python: Consuming APIs.	Merges, pulls, and pushes  Read about remote branches.
<b>Python: webscraping</b>		<b>git</b>
<b>T+3</b>	Requests, BeautifulSoup, HTML parsing  Read BeautifulSoup tutorial and this EFF article.	gitg, GitHub, and other tools  Explore <a href="https://github.com/">https://github.com/</a> .
<b>Python: webscraping</b>		<b>Docker</b>
<b>T+4</b>	Pandas for tables, non-table data parsing  Consider example of reading HTML table with pandas.	Basics of containerization  Review docker introduction.

<b>T+5</b>	Parsing dynamic webpages  Take a look at <a href="#">Selenium with Python</a> .	Layers and Dockerfiles  Read about building a container image.
<b>Python: multiprocessing</b>		<b>Docker</b>
<b>T+6</b>	Multiprocessing's Pool  Read about <a href="#">Multiprocessing Pools</a> .	Repositories  Read about <a href="#">repositories</a> .
<b>Python: multiprocessing</b>		<b>Kubernetes</b>
<b>T+7</b>	Random Number Generation, CPU/IO limitations  Read about <a href="#">issues</a> that may appear when you run multiple processes.	Intro to, if time permits  Explore <a href="#">Kubernetes basics</a> .

**My Role as Instructor:** As an Instructor, I am not only responsible for helping you understand economic concepts, I am also an advocate in place to protect and enhance your learning experience. If there are issues with any parts of the class (and especially with parts that may be changed quickly and easily), please let me know.

**Email Communication:** I will try to respond to emails within 48-hour period during work days. To ensure that your emails are going to be marked correctly and processed smoothly, please send those from your [@oberlin.edu](#) address. Be aware that during the days immediately before any midterm or exam you may not get a timely response from me due to peaking number of emails. Please plan and study ahead. Before sending an email to me, check the course syllabus thoroughly (use the latest online version to find TBD/TBA information). In a case of multiple emails from one person in a short period of time or a difficult question asked, I also reserve a right to transfer the conversation to office hours.

I assume that emails sent to your [@oberlin.edu](#) address are read in a timely fashion. You may receive class-wide notifications as well as individual messages related to class activities, assignments submitted, midterms/final exam arrangements, etc. I also assume that your email box is secure, since messages may contain details about your performance in the course and personal links to access course-related resources.

**Honor Code:** Academic Integrity is of utmost importance for maintaining a high-trust Academic Environment. I expect all students to be familiar with and follow Oberlin's [Honor Code](#).

**Religious Holidays:** I adhere to Oberlin's [Religious Holiday Observance Policy](#). Please let me know about any schedule conflicts that might affect your activity in this class as soon as possible.

**Students with Disabilities:** If you have a disability that requires an accommodation, please let me know as soon as possible. You will need to arrange for it through the [Student Accessibility Services](#). Please contact the [Student Accessibility Services](#) right away to start the documentation process. If you substantially delay your request, I may not be able to make necessary arrangements.

**Disclaimer:** I may adjust the syllabus if I believe it will serve the learning needs of the class. During the term, I may make statements about specific assets and asset classes, economic phenomena, behaviors of markets, firms and individuals, give opinion in relation to current/past events, and, among other things, discuss how certain situations will evolve or could have evolved under different sets of circumstances. Any information, idea, opinion, or other impression you get from this class should only be used for subject learning purposes and should not be considered an advice.