**CSS2: Data and Model Programming for Computational Social Science**

**Instructor: Garrett Swan ()**

Office Hours: By appt and Tuesday 9:00 to 10:00am

Office hours zoom link:

**TA: Sunyoung Park ()**

Office Hours: By appt and Thursday 2-3pm

Office hours zoom link:

Discussion Section: Wednesday 2-2:50pm and 3-3:50 pm

**UGIA: Brendon Hahm ()**

Office Hours: By appt and Friday 5 to 6 pm

Office Hours zoom link:

Discussion Section: Wednesday 3:00 to 3:50 pm

**Lecture time: Monday and Wednesday 5:00pm to 6:20pm**

**Lecture Zoom Link**:

**Section A01: Wednesday 2:00pm to 2:50pm**

**Section A01 Zoom Link**:

**Section A02: Wednesday 3:00pm to 3:50pm**

**Section A02 Zoom Link**:

**Piazza Link:**

**Piazza access code:**

**Course description**

This course explores the use of computational methods across the social sciences. Topics include thinking like a computational social scientist; research design for big data; legal and ethical dimensions of Computational Social Science (CSS). Students will implement demonstrations of these methods in Python through data visualization, selection, and modeling.

**Goal of the course**

At the end of the course you should have a good understanding of how to visualize, clean, and organize data for data analysis and modeling. Throughout the course, we will utilize NumPy and Pandas and will cover Matplotlib, Seaborn, and Scikit packages as well, which all commonly used in data science. By the end of the course, you will be able to do basic data science from a variety of data sources.

**Class time**

Each lecture will consist of coding via Python and you are encouraged to follow along. In sections you will cover problem sets that reinforce the topics covered in lecture.

The initial plan is that each lecture will take place during regular course time over Zoom. The lectures are non-mandatory and will be recorded and uploaded online for you to access throughout the semester.

**Material and Textbook**

Lecture material (videos and code sheets) will be provided on Canvas.

There is no required textbook for this course. However, there are many intro books out there that you might find helpful, such as “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython” by Wes McKinney and “Python Data Science Handbook: Essential Tools for Working with Data” by Jake VanderPlas

**Assignments**

**Short Quizzes**

Every week, there will be a quiz with multiple-choice questions on material discussed that week. Each of the quizzes will be worth 5 points (i.e., 5 questions). There will be 9 short quizzes starting week 1. Short quizzes will be released after class on Wednesday and due Sunday evening.

**Problem sets**

Every week, there will be a problem-set using material discussed that week. These assignments are meant to give you the hands-on practice that you need to develop fluency in the language. Each problem set will be worth 10 points. There will be 9 problem sets starting week 1. Problem sets will be released on Monday and due Sunday evening.

**Discussion board posts**

There will be a bi-weekly prompt on Canvas whereby you submit and respond on the Canvas discussion board. Per prompt, you will need to submit a post and respond to a post to get full credit. In total, there will be 4 discussion board posts with each post worth 2.5 points. Starting week 2, discussion board posts will be released on Monday and due Sunday.

**Final Project**

Instead of a final exam you will have a final project that will require you to generate code to extract, visualize, engineer, and analyze data from potentially 1 of 5 datasets. More details will be provided during 2nd week of class. The final project will be worth 50 points.

**Extra Credit**

You can earn extra credit by creating a github repository for this class and having at least 9 commits by the end of the quarter (basically, at least 1 commit per problem set). You will earn 0.44 bonus points per commit for a maximum of 4 bonus points. Follow the guide here for the easiest way to maintain your github: <https://desktop.github.com/> If you would prefer to use some other repository (e.g., <https://osf.io/> or <https://about.gitlab.com/> or maybe your own website), that is fine too.

**General debugging advice**

Like learning any skill, practice makes perfect. Fortunately, programming is really good at letting you know when you’ve done something wrong! When that happens, try the following steps in this order:

1. Double check the spelling/syntax
2. Print and check the variables
3. If it’s a function or operation, check the help menu to see that you are using the correct syntax
4. Create a new workbook, copy and paste the variables and code specifically related to the error, then try it
5. If the error is still occurring, try googling the error message
6. If there is no obvious solution on google, try rewording the problem you are having
7. If the problem is still occurring, email your TA
8. If the problem is still occurring, contact your instructor

Learning how to debug is an important part of programming and critical when dealing with long scripts. I encourage each of you to try debugging your errors before asking for the solution!

**Letter grade in course**

Grades will be based on the percentage of points earned on the midterm, final project, attendance/participation, and problem sets.

Letter grades will be assigned based on the following table:

A+ 97-100,                   A 93-96.99999,             A- 90-92.99999

B+ 87-89.99999,           B 83-86.99999,             B- 80-82.99999

C+ 77-79.99999,           C 73-76.99999,             C- 70-72.99999

D+ 67-69.99999,           D 63-66.99999,             D- 60-62.99999

F 0-59.99999

Grades will not be rounded.

**Academic Integrity**

[From UCSD Academic Integrity office](https://academicintegrity.ucsd.edu/take-action/promote-integrity/faculty/syllabus-statements.html#General-statement-on-academic-i)

"Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind."

[Please read the full UCSD policy](http://senate.ucsd.edu/Operating-Procedures/Senate-Manual/Appendices/2)

This course will also make use of online quizzes and exams via Google Forms. This means that each of you will be sent an email and will take quizzes/exams online with your [user\_name@ucsd.edu](mailto:user_name@ucsd.edu) email linked to your grade. Taking a quiz or exam logged in as another student will be treated as a violation and you will be referred for disciplinary action. Similarly, emailing with or otherwise communicating with other students or anyone else during a quiz or exam will be treated as a violation and also referred for disciplinary action.

**Class Conduct**

I expect respect and consideration for both your classmates and myself. Please contact me if you feel as if any problems arise were you feel as if the classroom is no longer an exclusive environment.

**Changes to Syllabus**

Changes may have to be made to the syllabus and/or schedule. Any changes will be posted online and announced in class.

**Course Topics**

Week 1: Google colab, pandas, numpy,

Week 2: matplotlib, and seaborn

Week 3: obtaining data, basics of web scraping,

Week 4: web scraping continued, then what is feature engineering, missing data

Week 5: duplicates, outlier removal, then what is machine learning,

Week 6: linear regression

Week 7: polynomial regression, binary classification

Week 8: binary classification (decision tree, logistic regression, and support vector machine)

Week 9: Cross validation, k-means clustering,

Week 10: PCA, review course