University of San Francisco
Department of Economics
ECON 611: Computation for Economic Analysis
Fall 2019
M: 6:30 PM - 9:15 PM, Lo Schiavo 103 (LS 103)

Instructor: Mario Javier Carrillo
Office: MCL 114
Office Hours: Monday 3:00 PM to 5:00 PM. Use this link to book your office hour with me.
Email: mjcarrillo@usfca.edu

Teaching Assistant: Lilla Szini
Email: lkszini@dons.usfca.edu
Office: MCL 114
Office Hours: Wednesday 6:30 PM to 8:00 PM & Thursday 4:30 PM to 6:00 PM

Teaching Assistant: Stephen Embry
Office: Monday - MCL 119, Tuesday and Thursday - MCL 114
Email: sbembry@dons.usfca.edu
Office Hours: Monday 4:00 PM to 6:00 PM & Tuesday 2:00 PM to 3:00 PM
EXTRA* Thursday 3:30 PM to 4:30 PM
* Stephen TAs for other classes during this time

COURSE OVERVIEW:
The ECON 611 course is designed to offer master students exposure to the fundamental concepts of programming and data manipulation with an application to Social Science. Students will learn to use Python and SQL to program algorithms and manage datasets. The class will cover topics related to the command line, databases, functions, control structures, data types, strings, data structures, debugging, version control, APIs and creation of Python packages. Students who take this class should be comfortable with statistics and linear algebra.

This course will expose students to the theory and practical implementation of Python, thus it will move through a large body of material and code. It is strongly recommended that students keep up with readings, assigned projects, and complete or replicate any in-class labs or code-along notebooks used during lectures. Besides Python and SQL, students will also use Git and GitHub to share their projects online. This will be the foundation for their professional portfolio; a way to showcase their analytical skills.

I highly suggest you attend office hours as much as possible. If you cannot attend the hours listed, please contact me to set up a time that works best. There are tons of topics we will cover, so please ask questions!

COURSE RESOURCES:
- **Python**. With limited exceptions, the entire course will be conducted in Python 3.6 (If you have a higher version is ok). Students need to install anaconda and those students who are familiar with Python are encouraged to set up custom environments for the course and/or project(s).
- **Jupyter.** With limited exceptions, instruction and assignments will utilize Jupyter Notebooks. Jupyter comes with the Anaconda Python installation.

- **GitHub**. With limited exceptions, all official copies of course materials will be kept in GitHub. Students are expected to have a GitHub account (free accounts are fine) as the use of GitHub will be mandatory for projects.
- **Text Editor.** Students are required to use a text editor of their preference. I will be using Visual Studio Code.
- **Slack.** This will be our primary way of communication or sharing code during class. Please download the app.
    - **NOTE: Besides Slack, you are responsible for checking your USF email for any class updates.**

- **Materials.** All students **must** bring their laptops to every class. All student laptops must be closed during class unless we are doing a lab or I specifically ask you to follow along as I type into my computer. During class, students need to **refrain** from visiting other websites that are not related to the class. This disturbs the learning experience of the students. If necessary, you will be asked to exit from a site that is not related to the class and ultimately leave the classroom.
- **Computer Maintenance**. You are responsible for maintaining or updating your computer software. We will not provide any assistance related to software updates or troubleshooting your computer unless it has to do with the software we use in the course. If you own a PC our assistance to your computer needs will be very limited.

TEXTS AND ONLINE RESOURCES:

Being an open-source language, Python updates take place regularly and thus a book on the subject becomes obsolete fast. However, the fundamentals of programming remain the same independently of updates. In this course we will be using the following:

- How to Think Like a Computer Scientist: Learning with Python (available online) by Allen B. Downey.
    - We will use this book in a large portion of the course.
- Effective Computing in Quantitative Research by H.J. Paarsch and K. Golyaev, MIT Press, 2016.
    - We will use this book in a large portion of the course.
- Python for Data Analysis by Wes McKinney.
    - We will use this book towards the end of the course.
- The SQL Tutorial for Data Analysis by Mode Analytics (available online)
    - We will use this online tutorial for the SQL part of our course.

NOTE: Additional required readings and online trainings will be made available to students in advance.

COURSE REQUIREMENTS:

Class Attendance and Participation: Students are expected to attend every class, including workshops. The class will move at a medium to fast pace and will cover additional material that may not be part of the readings but it is related to the topic. Therefore, your attendance and participation in class is expected. If you don't feel comfortable asking questions in public, I ask that you come to office hours.

Projects and HWs: Projects and HWs are due at or before the beginning of class on the day they are due. There are no excuses! Feel free to talk to other students about the projects, review online resources (there are plenty of them). However, students should not submit code which has, in whole or in part, been copied from *any* other source (including another student, a web page, or another text). Students caught violating the academic honesty policy will face a severe penalty (read academic honesty below).

Exams: Both the midterm and the final will be cumulative, and will echo the projects and labs in content and structure. There will be absolutely no make-up exams. If you miss the midterm due to a documentable legitimate absence, I will reweight your grade so as to exclude the midterm. If you miss the final, you will be given an incomplete for the course. Exams start promptly at the beginning of class, so to ensure that you and your classmates have enough time to complete the exam I ask that you arrive at the classroom no later than 5 minutes before the start of class on the day of the exam. No extra time will be given to students who arrive late to class on the day of an exam.

Grading:

- HWs 35%
- Midterm 25% (pseudo code + perfect syntax)
- Final Exam: 40% (20% Final Project* + 20% Exam) *We will discuss the final project in class, but it will entitle the creation of your own Python package.

Percentage grade will be mapped to letter grades in the following manner (percentages will be rounded up to the unit digit):

- 100-94: A
- 93-90: A-
- 89-87: B+
- 86-83: B
- 82-80: B-
- 79-77: C+
- 76-73: C
- 72-70: C-
- 69-67: D+
- 66-63: D
- 62-60: D-
- 0-59: F

HWs submitted after the deadline will be marked down by 10% for each 24 hours it is late. Assignments more than 2 days late will not be accepted.

ACADEMIC HONESTY: Cheating on exams, copying homework answers from other students, reproducing online or other material, and similar offenses can result in penalties ranging from a zero grade for the assignment to a failing grade for the class and referral to the Dean's office. If you are in doubt, please read USF's Honor Code.

GENERAL CLASSROOM ETIQUETTE:

- **Cell phones** must remain out of sight and out of mind.
- **Laptops** have only the websites needed for the class open. You will be asked to close websites that are not related to the class.
- **Slack,** you will need to have Slack on during class.
- **Food and drinks,** you and your classmates will have their laptops open during class, make sure your drinks and food have a way to re-seal.
- **Be on time**, class starts at 6:30. If you are consistently late to class I will speak with you about the matter; repeat offenders may see their grades penalized.
- **During and after class**, get in the habit of using Stack Overflow to troubleshoot your code errors or simply find a more efficient way to program, and share it with the class via Slack.
- **I expect** everyone in the class to be polite and respectful of everyone else. If you feel that anyone in the

class (including myself or the TA) is not holding his or herself to this rule, please feel free to approach me.

COURSE SCHEDULE

The course will follow USF's academic calendar. I will try to follow the schedule below as much as possible, but I reserve the right to alter the timing.

CLASS DAY	TOPIC	WHY	RESOURCES-READINGS
August 26th	Navigating your computer via the Linux command line. Absolute and relative paths (part 1).	You need to know what is behind the hood of your computer.	The Linux Command Line by William E. Shotts, Jr. **Readings**: from page 9 to page 47 (DO NOT memorize commands) Linux Pocket Guide by Daniel J. Barret **Readings:** from page 13 to page 21, (DO NOT memorize commands)
September 2nd	Labor day Holiday (no class)	Complete this command line training.	
September 9th	Absolute and relative paths (part 2). A gentle introduction to version control with Git.	You need to know how to share your code and collaborate with other developers/analysts..	Linux Pocket Guide by Daniel J. Barret **Readings:** from page 13 to page 21, (DO NOT memorize commands) ProGit by Chacon and Straub **Readings:** Chapters 1, 2, and 6.
September 16th	Programs - what the heck are they. Pythonic Thinking: ● PEP 8 style guide	You need to understand what a program is and what is the content of a program.	How to Think Like a Computer Scientist:

	• Strings • Iteration & loops		Learning with Python Effective Computing in Quantitative Research
September 23th	Pythonic Thinking: • Lists • Tuples • Iteration & loops	We need to cover our basic Python concepts before we move on.	How to Think Like a Computer Scientist: Learning with Python Effective Computing in Quantitative Research **HW_1 due**
September 30th	Pythonic Thinking: • Dictionaries • Sets • Booleans • Control Flow	We need to cover our basic Python concepts before we move on.	How to Think Like a Computer Scientist: Learning with Python Effective Computing in Quantitative Research **HW_2 due**
October 7th	Pythonic Thinking: • List comprehensions • Modules • Functions • Mathematics and Statistics (Descriptive) from scratch.	A Python code is more than one line. It is a set of logic statements..	How to Think Like a Computer Scientist: Learning with Python Effective Computing in Quantitative Research **HW_3 due**

October 14th	Fall Break (No Classes)		
October 21st	**MIDTERM**		
October 28th	Pythonic Thinking: ● Regular Expressions (basics) ● Object-Oriented Programming (Introduction)	Once the basics are covered we are ready to start creating our own functions.	How to Think Like a Computer Scientist: Learning with Python Effective Computing in Quantitative Research
November 4th	Relational Database: ● SQL (basics)	Before you bring in data for analysis you need to connect multiple tables and perform descriptive stats.	Mode Analytics **HW_4 due**
November 11th	Relational Database: ● SQL (Intermediate)	Before you bring in data for analysis you need to connect multiple tables and perform descriptive stats..	Mode Analytics
November 18th	Arrays and Vectorized Computation: ● Linear Algebra (basics in Python) ● Using NumPy (basic commands) Pandas ● Essential functionalities (part 1)	Applying Python to perform data analysis.	Python for Data Analysis by Wes McKinney
November 25th	Pandas: ● Essential functionalities (part 2) Application Programming Interface (APIs - basics)	Applying Python to perform data analysis and learn how to pull data from external resources.	Python for Data Analysis by Wes McKinney
December 2nd	Wrap-up	Is there something not clear yet, we will use this session for review.	**HW_5 due**
December 9th	**Final Exam (DATE TBD)**	Did you learn something, if so prove it!	