GBA464 – Programming for Analytics

Course Syllabus (last modified 07/20/2024)

Prefall 2024

Course Logistics:

Lectures: Mon, Tue, and Thu, 10am-12pm, or 1pm-3pm, or 3:20pm-5:20pm
Labs: Fri, 10am-11am, or 11:15am-12:15pm, or 1:30pm-2:30pm
Location: Schlegel Room 407
Class schedule: full weeks for Mon, Jul 22 to Friday, Aug 16 and one more lecture on Tue, Aug 20
Final exam: Wed, Aug 23
Please mark your calendar for these dates and times

Instructor:

Yufeng Huang, Associate Professor of Marketing
yufeng.huang@simon.rochester.edu

Lab instructor:

Kang Huang, PhD student in Marketing
khuang32@simon.rochester.edu

Teaching Assistants:

Babli Dey (bdey2@ur.rochester.edu)
Dawn (Zezhen) He (zezhen.he@simon.rochester.edu)
Qihao Lin (qlin12@u.rochester.edu)
Muyang Zhou (mzhou36@simon.rochester.edu)

Office Hours:

Yufeng – Wednesdays (starting July 24), 4-6 pm, Carol Simon 3-221
Babli – Thursdays (starting July 25), 4-6 pm, Zoom
Kang – Fridays (starting July 26), 3-5 pm, Schlegel 407
Muyang – Saturdays (starting July 27), 3-5 pm, Zoom
Dawn – Sundays (starting July 26), 3-5 pm, Zoom
Qihao – Tuesdays (starting July 30), 4-6 pm, Zoom
Zoom link: https://rochester.zoom.us/j/95667861861
Side note: avoid sharing your completed assignments on Zoom as much as possible

Course Information

1. Course Description and Learning Objectives

GBA464 provides a foundation for programming in the Python environment, specifically tailored for data analysis and business analytics. The course covers fundamental programming concepts such as operators, data structures, control structures, loops, and user-defined functions. These concepts will be taught in the context of real-world business problems related to data management and visualization.

Throughout the course, students will:

- Learn the basics of Python programming, including installing and setting up the Python environment, working with basic data types, and understanding flow control structures.
- Gain proficiency in using powerful libraries such as NumPy for numerical operations, Pandas and Polars for data manipulation, and Matplotlib and Seaborn for data visualization.
- Develop skills in handling and processing data, including reading from and writing to various data formats, and performing data cleaning and transformation tasks.
- Understand how to apply regular expressions for text processing and data extraction.
- Practice writing efficient and readable code to solve complex business problems and generate insightful visualizations.

The course includes a combination of lectures, hands-on coding sessions, and assignments designed to reinforce learning. Students are expected to complete individual assignments and a group project, which will involve applying Python programming skills to real-world data analysis tasks.

By the end of the course, students will have a strong foundation in Python programming and be well-equipped to tackle data analysis challenges in their future careers.

2. Textbooks and Required Material

There is no required textbook. The course materials largely include lectures and notes provided directly by the instructor. In addition, mirroring the real world of programming, students are encouraged to use Google (or other search engines) and the numerous freely available online resources to search for relevant materials and help for specific issues they encounter.

All class sessions are recorded, and students can view the recording after the class.

3. Evaluation and Grading

The course grade is based on the following:

Environment set-up	1%
Replication assignments	5*5=25%
Homework assignments	3*8=24%
Final Exam	50%
Total	100%

(0) Class participation
Students should attend classes in person because it is a very effective mode of learning. I will elaborate on this in the first lecture.

(1) Environment set-up
By Wednesday of the first week (July 24), students should upload a screenshot showing that they have successfully set up the environment. Submit a screenshot to get the full mark.

(2) Replication assignments
There are five replication assignments: two in week 1, and one in each of week 2, 3, and 4. Each assignment asks students to type and run the codes shown and discussed in class. Grading is participation-based. Submit any code file to Blackboard and get full marks.

(3) Homework assignments
The three homework assignments all involve writing code that uses the language concepts introduced in the course to solve real-world problems. The assignments will be more demanding than some of the in-class examples and will require students to integrate the programming knowledge into a structured way of using code to produce desired datasets or data visualizations.

Grading is completion-based. Submit the code file onto Blackboard, the TA will check whether the assignment is completed, and if so, the full mark will be given. But, different from the replication assignments, we do check whether the code runs bug-free (so please do check this) and whether there is a reasonable attempt to answer the questions.

The due dates for all assignments are at 11:59pm Eastern Time on the specified date (i.e., by the end of day at local Rochester time). Late submissions within six hours have a grade cap at 80% of the original grade.

(4) Exams
The exam is comprehensive and involves writing and snippets of code to solve specific problems. The entire exam is closed book, but one has access to a “cheat-sheet” – a one-

page A4 (letter-sized) paper with anything printed or written on it prior to the exam. The exam will be held synchronously in the classroom.

(5) Grading policy
Each student's numeric grades, according to the above-described weights, will be ranked to create an alphabetical grade. All MSBA students will be curved together. All MSMA students will be curved together.

4. AI Policy

GBA 464: Generative AI Use Policy		
COURSE	**EXAM**	**DISCLOSURE POLICY**
Only When I say Yes Students should not use generative AI unless the instructor gives explicit permission for particular student work.	***Never*** All work related to completing the exam must be your own. Generative AI may not be used.	***Citation Only*** Include a citation of the generative AI and version number used and a shareable link that contains the prompt and chat for each chat used in the student work.

One should not ignore ChatGPT's (and other large language models') existence. But one should not over-rely on it. Here are my suggestions:

- ChatGPT is not only good at coding but also good at explaining how the code works. Use it to your advantage. Good examples:
    - What is the syntax that takes mean of a column "var" in pd.DataFrame df?
    - I have columns "x" and "y" in data frame df, plot y against x using matplotlib, make it a blue dashed line.
- You will not get help in the exam, so over-using ChatGPT during the learning phase will weaken your position in the exam. Please keep this in mind. Bad examples:
    - I have an assignment, see uploaded pdf, write code to answer it.

- A rule of thumb is never ask ChatGPT to complete a full task for you. More discussions on this in class.

5. Accommodations and Accessibility

Both Simon and the University of Rochester respect and welcome students of all backgrounds and abilities. In the event you encounter any barriers to full participation in this course due to the impact of a disability, please contact both your instructor and the Office of Disability Resources as soon as possible. The access coordinators in the Office of Disability Resources will meet with you to discuss the barriers you are experiencing and explain the eligibility process for establishing academic accommodations. You can reach the Office of Disability Resources at (585)276-5075 or at disability@rochester.edu . More information can be found at https://www.rochester.edu/college/disability/ .

6. Credit-Hour Policy Adherence

This course follows the Simon credit-hour policy for 2.5-credit courses. The course meets three times per week for 2 hours each meeting. In addition to these 6 hours of synchronous class sessions per week, students are required to work on homework assignments and complete other asynchronous learning activities (such as exercises that are in part finished at home). For each hour in class, students should expect twice the time outside of the class on supplemental work, including reviewing the class material, completing in-class exercises, and completing the homework assignments.

7. Academic Integrity

Simon's Code of Academic Integrity (see the section Academic Integrity Policy in the Simon School Student Handbook) states: "*Every Simon student is expected to be completely honest in all academic matters. Simon students will not in any way misrepresent their academic work or attempt to advance their academic position through fraudulent or unauthorized means. No Simon student will be involved knowingly, or unknowingly yet passively within a team, with another student's violation of this standard of honest behavior.*"

In addition to refraining from obvious forms of cheating and plagiarism:

- On assignments, do not copy or paraphrase work from each other, from students who have taken the class previously, from materials of mine distributed in a previous class, or from outside sources. Any written work should be entirely your own (or your team's, as applicable).
- Exams or assignments that include an Academic Integrity/Honesty Pledge must have the pledge signed. Submissions without the pledge signed will receive a score of zero.
- Do not obtain advice, notes, solutions, or other material from students who took the class previously in ways that would give you an unfair advantage or would undermine the learning experience for you and the class (such as, notes from past case discussions). Similarly, do not use others' case analyses posted on-line.
- Use quotation marks when quoting any text directly. Changing a few words of a sentence or longer section does not make the work your own. Independently written texts rarely have even five consecutive words in common.
- Students may not audio or video record class lectures or other classroom or laboratory activities without the instructor's permission.
- Students may not publish, distribute, or sell—electronically or otherwise—any course

materials that the instructor has developed in any course of instruction in the University (e.g., presentation slides, lecture aids, video or audio recordings of lectures, and exams) without the explicit permission of the instructor.

Most forms of disallowed shortcuts are easy to detect and will be referred to the school's Academic Integrity Committee. To help prevent other students from violating academic integrity, do not pass on notes or give advice on assignments to any students who are taking the course in a later term or are taking it at the same time in a different section. Please refer to the Student Handbook for any questions regarding the Code of Academic Integrity.

If a situation in your professional or personal life prevents you from finishing assigned work in a timely manner, please contact me before the deadline to discuss how to proceed. You should also contact your OSE advisors, if appropriate. Do not violate the Academic Integrity Code to manage a difficult situation.

Tentative Course Schedule

Week 1 (July 22, 23, 25)

Class 1:

- Installing Anaconda, setting up the environment, installing packages
- Optional: Installing VSCode
- First few lines of code: Arithmetic operations between numbers and booleans

Class 2:

- Lists
- Strings
- Concatenation, indexing, and slicing
- Quick mention: dictionary, tuple, set

Class 3:

- Flow control: the natural flow of code
- Conditionals (if/else)
- Loops (while/for)
- Nested conditionals and loops
- List comprehension
- Functions, classes, methods

Replication assignments:

1. Replicate all class codes from Class 2 (due Wednesday)
2. Replicate all class codes from Class 3 (due Sunday)

Week 2 (July 29, 30, Aug 1)

Class 4: NumPy

- ndarray properties
- Query of dimensions
- Broadcasting

Class 5: Pandas

- Series, DataFrames

- Query of rows and columns, operations between columns
- Read/write data, folder structure

Class 6:

- Visualization basics
- Basic Matplotlib and Seaborn syntaxes
- Examples

Replication assignment:

1. Replicate all class codes from Class 4 and 5 (due Wednesday)

Assignment:

1. Belgium ATM (due Sunday)

Week 3 (Aug 5, 6, 8)

Class 7:

- Advanced pandas: merge, group by, and pivot table
- Polars - fast and efficient version of pandas

Class 9:

- Regular expressions and examples

Class 8:

- Advanced data frame (polars) and numpy examples

Replication assignment:

1. Replicate all class codes from Class 7 and 8 (due Wednesday)

Assignment:

1. Twitch viewership and Steam player count (due Sunday)

Week 4 (Aug 12, 13, 15)

Class 10:

- Review of loops and functions
- Loop and function examples

Class 11:

- The map function and its applications

Class 12:

- Advanced examples

Replication assignment:

1. Replicate all class codes from Class 9, 10, and 11 (due Wednesday)

Assignment:

1. Recency, frequency, and monetary value (due Sunday)

Week 5 (Aug 20)

Class 13 (Aug 20):

- Review class