Syllabus - Spring 2025, CSE 6040/x: Intro to Computing for Data Analysis

> This syllabus was last edited on January 2, 2025, and is considered **a draft (not final)**. It should be finalized by the first official day of class, January 6. *Update history:*
>
> - (Jan 2) Initial release

TL;DR: (https://www.urbandictionary.com/define.php?term=TLDR)

1. Bookmark the website: cse6040.gatech.edu (https://cse6040.gatech.edu). Once class officially begins, the website will be a good place to get "at-a-glance" information on where we are in the course, e.g., what assignments or exams are due next.
2. Learn what UTC is — there will be no forgiveness for misunderstanding the due date/time convention!
3. Try the programming problems suggested under "prerequisites."
4. Communicate with us via the online discussion forum (Piazza) or office hours, **not** email! We make all announcements and updates via to the forum.
5. Read the schedule (https://docs.google.com/spreadsheets/d/e/2PACX-1vRJ5mbQRy0ExKFHXL4XGVdft6Mat4CSZQcsH_xtoGuJUjQ9C_nXjHkaMiSLdYLDx58ETktZnnX_oAPI/pubhtml?gid=1483993949&single=true). Bookmark it in your browser. Re-read it. Check it frequently.
6. Review your learning resources. There are several ways to "interact" with the teaching staff *depending on your needs*, so take a moment to learn about them and revisit them when you feel stuck.
7. Dig deeper on your own, as befits a graduate-level class, and get 100% on the assignments. The lab notebook assignments are **not** meant check what you know. Rather, they are designed to help us "deliver" key concepts; it's the exams that are the real assessments of where you stand.
8. Learn what an "MRE" is (minimal reproducible example, not meal ready-to-eat, although we'd never begrudge one the latter).
9. **Read this syllabus in its entirety!** While this TL;DR emphasizes the most important items, you need to be mindful of *all* points. So plan to read (and re-read) this syllabus.

Overview

This course is a hands-on introduction to programming techniques relevant to data analysis and machine learning. The main programming language is Python. The course materials **assume some prior programming experience** (but that does **not** have to be Python). The class will have you review and extend that experience in the context of data analysis and numerical computation.

- Co-instructors (co-CEOs): Professor Richard (Rich) Vuduc (https://vuduc.org) and Chris Kinkade (https://www.linkedin.com/in/chriskinkade)
  - OMSA Co-creators: Vaishnavi Eleti and Rachel Wiseley
- Chief Operating Officer:Jennifer Sherwood (https://www.linkedin.com/in/jensherwood53/)
- Chief Technology Officer: Albert Waldron (https://www.linkedin.com/in/albert-waldron)
- Co-Chief Information Officers: Larry Heckel (https://www.linkedin.com/in/larry-heckel/) and Katie Nelson (https://theorg.com/org/regulated-capital-consultants/org-chart/katie-nelson)

> This class is part of *three* different degree and certification programs: the residential (on-campus) Masters in Analytics program at Georgia Tech (MSA), the online Masters in Analytics (OMSA), and the edX Verified MicroMasters (VMM). For comments specific to one of these three "sections" of the course, please see the Section-specific notes.

What will I learn?

You will build "from scratch" the essential components of a data analysis pipeline: collection, preprocessing, storage, analysis, and visualization. You will see examples of a basic process that starts with high-level analysis questions, formalizes those questions into mathematical or computational problems, develops solutions for those problems, and lastly, translates solutions into working code. Beyond programming and best practices, you'll learn elementary data processing algorithms, notions of program correctness and efficiency, and numerical methods for linear algebra and mathematical optimization.

The underlying philosophy of this course is that you'll learn the material best by active practice, not passively watching videos. Therefore, we de-emphasize the videos and strongly encourage you to complete all assignments, including any ungraded ("optional") parts, and do the practice exams that we will release.

And since it is a graduate-level class, we expect you will go a bit beyond on your own (see *How much time and effort are expected of you?*, below).

How will I do all that?

(Assignments and grading.)

You'll learn the material through hands-on *lab notebook assignments* (the "homework"), which will be required and whose completion will count toward your final grade. We will test your learning through *exams*.

Overall, the breakdown of your final grade is as follows:

- Notebooks ("labs" or "homeworks"): 50%
- Exams ("assessments"): 50%, broken down as follows:
  - Midterm 1: 10%
  - Midterm 2: 15%
  - Final exam: 25%
- "DIY" extra credit: Max 3% bonus (but all final grades are capped at 100%) — **Details are to-be-determined until after Midterm 2.**

> *Important note:* We view notebooks as a vehicle for **introducing** the material that you need to learn. However, the notebooks are **not** a measure of what you have learned. For that, we use exams.
>
> This approach has two implications for you. First, you should view getting full credit on Notebooks as the *minimum* you should be doing. Beyond that, you should allow extra time to pore over the notebooks and really understand what's happening. The real check on what you've mastered of that material comes from the exams, which mimic real-world problems and are timed and proctored. We will provide past exams for practice.

Scoring "targets." For Georgia Tech students, we also assign "whole-letter" grades (A, B, C, D, and F) at the end of the semester. The threshold for the equivalent of an "A" grade is 90%, for a "B" 80%, a "C" 70%, a "D" 60%. Anything below 60% is "not passing" in both the Georgia Tech and edX programs.

Lab notebooks. As noted above, the main vehicle for learning new material is a collection of **Jupyter (https://jupyter.org) notebooks**, each of which introduces some concepts and reinforces them with programming exercises.

There are also accompanying videos. However, these are a guide, **not** an all-encompassing overview of everything you need to learn. *(For one take on why we de-emphasize videos in this course, see this missive (https://www.classcentral.com/report/why-my-mooc-is-not-built-on-video/).)* Instead, you need to master the content of the notebooks using a combination of the videos, references we suggest, and searching on your own.

The last sentence of the previous paragraph is essential! We make the notebooks a part of your grade to encourage you to go through the material. However, you should expect to spend **extra time** on your own and with your peers thinking about that material, looking things up to better understand something, and trying to master the material. See *How much time and effort do you expect of me?* for more discussion of this point.

Autograding of assignments. Your assignments are hosted on a cloud-based platform called Vocareum (https://www.vocareum.com/). You are expected to work on and submit your assignments through Vocareum, which autogrades them. While you are welcome to download the assignments and work on them on your local computer, the staff do not officially support this mode. If you go this route, **you** must ensure that whatever you do also works when entered or uploaded to Vocareum and submitted through its autograder by a given deadline.

If the autograder does not accept your assignment, you do not get any credit for it. Therefore, you must ensure your work passes the autograder. We will not accept any assignment because you claim it works on your local system but does not get through the autograder. Also, if you have a really bad bug in your code that causes the autograder to crash (e.g., an out of memory error or your solution takes too long to complete, causing a timeout), you will **not** get credit.

Exams. The way we assess how well you know the material is through two midterm exams and one final exam. Many prior students regard these as the toughest part of the course. We will provide old exam problems for practice. See also *How do the exams work?* for more information.

> There is approximately one notebook or exam due every week. The assignments vary in difficulty but are weighted roughly equally. Some students find this pace very demanding. We set it up this way is because we believe that learning to program bears similarities to learning to converse in a foreign language: doing so demands constant and consistent practice. Please plan accordingly.

How can I check my grade? Your score on each assignment or exam is always the most accurate on the Vocareum platform. So if you see your score in Vocareum, you are "good to go."

These scores are supposed to propagate automatically to your learning environment—the Canvas gradebook for GT MSA/OMSA students or the edX gradebook for VMM learners. However, there can be delays. At the very end of the semester, when we tabulate your final grade, we will make sure all Vocareum grades are transferred to and mirrored on Canvas or edX as appropriate.

What should I know already?

(Prerequisites.)

You should have at least an undergraduate-level understanding of the following topics. You'll have about a month at the beginning of the course to get up to speed on this material if you have gaps or haven't touched these subjects in a while.

- Programming proficiency in some programming language
- Basic calculus
- Probability and statistics
- Linear algebra

What does "programming proficiency" mean? We use Python in this course and assume you have some prior programming experience in an imperative language. That means we assume that you know what variables, assignments, functions, loops, and basic data structures like arrays or lists are, which should be enough to teach yourself basic Python. Armed with that basic Python, this course then aims to fill in gaps in your programming background that might keep you from succeeding in other programming-intensive courses of Georgia Tech’s MS Analytics program (http://analytics.gatech.edu/), most notably, CSE 6242 (http://poloclub.gatech.edu/cse6242/).

Checking your programming prerequisites

Here is a concrete way to check your programming background.

The site, `codewars.com`, provides a number of coding drills. These are small programming problems, at varying levels of difficulty, which you can try to solve in your web browser (no installation or pre-registration necessary). The site provides an in-browser editor and automatic testing environment, and for many problems, provides these for several programming languages. We recommend the following self-test:

- Try to do at least three of the "6 kyu" problems below. (The "kyu" number is an estimated difficulty level, where larger numbers are supposed to be "easier." For students *without* a computer science background, problems rated "6 kyu" would probably appear to have an "easy" to "medium" difficulty, and "5 kyu" problems would probably be rated "medium" to "hard.") Use any language you are comfortable with based on your prior experience.
- If you have no prior programming background, you can still succeed in our course, but you'll need to devote extra time. To see if you have at least the right basic "computational problem solving" skills, try to solve the 6-kyu problems *systematically* by hand (i.e., without programming). Here, "systematically" means you can develop a step-by-step procedure to solve the problem. A good way to determine if you can come up with such a procedure might be to design a spreadsheet (e.g., in Microsoft Excel or Google Sheets) that helps solve it.
- We've also listed a few "5 kyu" problems, below. These should seem easy to you by the *end* of our course. If they already seem easy, then you can probably justify placing out.

6 kyu problems: These should seem solvable.

- A banker's plan: https://www.codewars.com/kata/56445c4755d0e45b8c00010a (https://www.codewars.com/kata/56445c4755d0e45b8c00010a)
- Rectangles into squares: https://www.codewars.com/kata/55466989aeecab5aac00003e (https://www.codewars.com/kata/55466989aeecab5aac00003e)
- Multiples of 3 or 5: https://www.codewars.com/kata/514b92a657cdc65150000006 (https://www.codewars.com/kata/514b92a657cdc65150000006)
- Bouncing balls: https://www.codewars.com/kata/5544c7a5cb454edb3c000047 (https://www.codewars.com/kata/5544c7a5cb454edb3c000047)
- Find the odd int: https://www.codewars.com/kata/54da5a58ea159efa38000836 (https://www.codewars.com/kata/54da5a58ea159efa38000836)

5 kyu problems (harder!): These should seem solvable AFTER the course is over.

- Decimal to factorial and back: https://www.codewars.com/kata/54e320dcebe1e583250008fd (https://www.codewars.com/kata/54e320dcebe1e583250008fd)

- Common denominator: https://www.codewars.com/kata/54d7660d2daf68c619000d95 (https://www.codewars.com/kata/54d7660d2daf68c619000d95)
- Buddy pairs: https://www.codewars.com/kata/59ccf051dcc4050f7800008f (https://www.codewars.com/kata/59ccf051dcc4050f7800008f)
- Going to zero or infinity? https://www.codewars.com/kata/55a29405bc7d2efaff00007c (https://www.codewars.com/kata/55a29405bc7d2efaff00007c)

If you already have a significant programming background, consider placing out. If you have no programming background, please think carefully and realistically about how much time you can devote to getting caught up. See *How much time and effort do you expect of me?* for more specific guidance on what we assume are the two hardest gaps to fill, namely, programming proficiency and linear algebra.

When are things due?

(Deadlines and submission policies.)

Because students from all over the world take this course, we have standardized on all assignments being due at 11:59 UTC (https://en.wikipedia.org/wiki/Coordinated_Universal_Time) time of the designated date.

> If you are asking whether "11:59" above means AM or PM, you are already in trouble! Read below to get back on track.

What is UTC? UTC is an international standard time that uses a 24-hour convention (e.g., 11:59 UTC is distinct from 23:59 UTC) and has no notion of "AM" and "PM," nor does it observe "daylight savings time" changes. **You** need to figure out what all that means and how to translate that into your local time. Please make sure you are aware of the due date and time for your local area.

We will not grant extensions based on your misunderstanding of how to translate dates and times. You may wish to consult online tools, like the Time Zone Converter; here is an example of TZC for 11:59 UTC on January 13, 2025 (https://www.timeanddate.com/worldclock/converter.html?iso=20250113T115900&p1=1440&p2=25&p3=33&p4=195). We often post links to TZC with due dates as a reminder, so learn the correct conversion for where you live.

Late policy. Every lab notebook has an "official due date." You also get an automatic penalty-free 48-hour extension (the "extended due date") on **every** notebook; please refer to the course schedule (top of this page). After the extended deadline, you can still submit your notebook for half-credit (50% of whatever points you earn) up until the exam that covers it. (For example, if Notebook 7 is covered by Midterm Exam 2, you can submit Notebook 4 late for half-credit up until the opening date of Midterm Exam 2.)

Example:

- Notebook assignment 0 is *officially due* on Monday, January 13, 2025, at 11:59 UTC.
- The automatic *extended deadline* is Wednesday, January 15, 2025, at 11:59 UTC. There is no penalty for submitting up until this time. We will release sample solutions shortly thereafter.
- You can still submit Notebook 0 up until the date of the first exam that follows it. However, you will only get half-credit.

> Note: The late penalty applies **"per part"** of the notebook assignment. That is, suppose Notebook 0 has two parts (part 0 and part 1). If you turn part 0 in on-time and part 1 late, the penalty will only apply to part 1. However, that penalty applies to *all* exercises in the part. Therefore, if your late submission passed all exercises in Part 1, your Part 1 score would be 50% regardless of the number of exercises in that part submitted on time. **(You have to submit each part separately, which is why the penalties are applied in this fashion.)**

This extension does **not** apply to exams; see *How do the exams work?* for more information.

Unsolicited advice. There are many assignments, so any given assignment is only worth a small percentage of your final grade. If you miss one or can only submit it late, it won't necessarily hurt your final grade.

Sample solutions. About 48 hours after the official due date, i.e., right after the extended due date, we will release sample solutions. Therefore, it's generally possible to get a nonzero score for every assignment.

Strict enforcement. Because of the flexible automatic extension and partial credit policies, we enforce the late policy strictly. Therefore, we ask that you refrain from requesting exceptions unless you have a genuine hardship or extentuating circumstance.

How do the exams work?

For the exams, you will receive a window of several days in which to attempt the exam (the *exam window* or *exam period*), with a hard deadline to submit and **absolutely no early release or extensions.** Please carefully review the course schedule (top of this page) throughout the semester for these dates and plan accordingly.

Exams will be proctored. (For GT students, we use Honorlock, a Georgia Tech-sanctioned third-party proctoring tool.) The proctoring includes a room scan using a web camera. You are responsible for selecting a private location where you are comfortable with the space being videoed and audio recorded.

Exam format. The exam itself will typically consist of one or two "problems," which are structured similarly to the lab notebooks. Once you start the exam (any time during the exam window), you must submit your work within 3-4 hours or the end of the exam period, whichever comes first. Exams are open-book, open-note, and open-internet (meaning you can do searches).

However, **you are not allowed to ask for direct help** by, for instance, calling or messaging your peers, posting questions on sites like Stackoverflow, using solutions posted for exam questions, asking bots, such as ChatGPT or GitHub Copilot, to solve the problems, or paying others to do your work. The intent of the exam is to assess what **you can do on your own**, *with* access to information (so you don't have to memorize everything) but *without* access to the direct assistance of others.

> The policy restricting the use of bots is for exams. You may use them as a tool to help you **study**: completing homework problems, explaining and understanding sample solutions, and getting tutorial advice.

Exams are autograded. The same autograding policies that apply to the lab notebooks extends to exams.

> The phrase, *"structured similarly to the lab notebooks"* means that every time you do a notebook, you are practicing for the exam! However, we will also provide real problems from old exams for you to do additional practice.

Missed exams (MSA/OMSA only). If you cannot take the exam during the designated period, but you do have compelling and documented justification (e.g., medical condition, family emergency), we can give you an incomplete grade for the course. To document your situation, reach out to the Division of Student Life (https://studentlife.gatech.edu/) as soon as you can, and they will reach out to us confirming your issue. Then, to resolve the grade, you would take the equivalent exam in the next semester that the course is offered.

Per Georgia Tech's policies, incomplete grades are reserved for making up a "small" amount of work only. If your circumstances are more severe, you may need to drop and retake the course. Please refer to GT's information incompletes (https://registrar.gatech.edu/info/incomplete-grades) for more detail.

Proctoring. Exams will be proctored. The exact details will be provided during the first month of the course, but it is **critical** that you complete the following major steps well before the first exam; otherwise, you will not be able to take the exam and will receive a zero-score on it.

1. You must complete an exam onboarding step, which involves ensuring you have the right hardware and software.
2. You *complete* an identity verification step successfully. (edX/VMM students should also see this note about *two* separate ID verification steps.)
3. If you encounter issues, you need to resolve them as soon as possible and well *before* the exam periods begin if you want to avoid receiving a zero score.

How much time and effort do you expect of me?

*(From the instructors' perspective, this section of the syllabus is arguably the most important to accept **before** you sign up!)*

At Georgia Tech, this course is a three-credit-hour graduate-level (Masters degree) course. So what does that mean?

The "3 credit hours" part translates into an average amount of time of about 9-12 hours per week (or maybe 15 hours per week during the summer session). However, the actual amount of time you will spend depends heavily on your background and preparation. Past students who are very good at programming and math reporting spending much less time per week (maybe as few as 4-5 hours), and students who are rusty or novices at programming or math have reported spending more (say, 15 or more hours).

The "graduate-level" part means you are mature and independent enough to try to understand the material **at more than a superficial level.** That is, you don't just watch some videos, go through the assignments, and stop there; instead, you spend some extra time looking at the code and examples in detail, reviewing sample solutions, trying to cook up examples, and coming up with self-tests to check your understanding. Also, you will need to figure out quickly where your gaps are and make time to get caught up.

In past runs of this course, we've found the two hardest parts for many students are catching up on (a) basic programming proficiency and (b) linear algebra, which are both prerequisites to this course. We'll supply some refresher material, but expect that you can catch up. Here is some additional advice on these two areas.

Programming proficiency. Regarding programming proficiency, we expect that you have taken at least one introductory programming course in some language, though Python will save you the most time. You should be familiar with basic programming ideas at least at the level of the Python Bootcamp that most on-campus MS Analytics students take just before they start. A useful textbook reference for Python as you go through this course is Jake Vanderplas's A Whirlwind Tour of Python (https://jakevdp.github.io/WhirlwindTourOfPython/). There is also a nice interactive iPad app called tinkerstellar (https://tinkerstellar.com/) that "implements" the contents of this book.

We can also recommend several online resources, like CS 1301x (https://www.edx.org/course/computing-in-python-i-fundamentals-and-procedural-programming-2), which is Georgia Tech's undergraduate introduction to Python class. Students who struggled with this course in the past have reported success when taking CS 1301x and re-taking this class later. Beyond that, code drill sites, like CodeSignal (https://codesignal.com) and codewars.com (https://www.codewars.com) (the latter's absurdly combative name notwithstanding) can help improve your speed at general computational problem-solving. Please spend time looking at these or similar resources.

Part of developing and improving your programming proficiency is learning how to find answers. We can't give you every detail you might need; but, thankfully, you have access to the entire internet! Honing your skills at formulating queries, searching for helpful code snippets, and adapting those snippets into your solutions is time well-spent and, for better or worse, is common practice in the "real world" of modern software development. So, use this class to practice doing so. (During exams, you will be allowed to search for stuff on the internet!)

> It's also an excellent skill to have because whatever we teach now might not be state-of-the-art later on, so knowing how to pick up new things quickly will be a competitive advantage for you. Of course, the time to search may make the assignments harder and more time-consuming, but you'll find that you get better and faster at it as you go, which will save you the same learning curve when you're on the job.

Math proficiency. Regarding math, and more specifically, your linear algebra background, we do provide some refresher material within this course. However, it is ungraded self-study material. Therefore, you should be prepared to fill in any gaps you find when you encounter unfamiliar ideas. We strongly recommend looking at the notes from the edX course, Linear Algebra: Foundations to Frontiers (LAFF) (https://ulaff.net/). Its website includes a freely downloadable PDF with many helpful examples and exercises.

Can I work with others? Also: What can I ask of the teaching assistants (TAs)?

(Collaboration policy.)

You may collaborate with your peers on the lab notebooks at the "whiteboard" level. That is, you can discuss ideas and have technical conversations with other students in the class, which we especially encourage on the online forums. However, each student should write-up and submit his or her own notebooks. **Taking shortcuts here only hurts you on the exams.**

But what does "whiteboard level" mean? It's hard to define precisely, but here is what we have in mind.

The **spirit** of this policy is that we do not want someone posting their solution attempt (possibly with bugs) and then asking their peers, "Hey, can someone help me figure out why this doesn't work?" **That's asking others (including the instructors and TAs) to debug your work for you.** That's a no-no.

But what can I do instead of just posting my code to ask for debugging help? (Answer: MREs!)

In such situations, try to reduce the problem to the simplest possible example that also fails. Please see Stackoverflow's guidelines on "MREs" – minimal, reproducible examples (https://stackoverflow.com/help/minimal-reproducible-example). In that case, posting code on the class discussion site (see below) would be OK. Indeed, the mere process of distilling an example often reveals the bug!

In other words, it's okay and encouraged to post and discuss code examples as a way of learning. But you want to avoid doing so in a way that might reveal the solution to an assignment.

When posting questions in the online forums, the same policy outlined above applies. Indeed, your peers will answer questions much faster if you pare it down to an MRE, and the instructors will advise the TAs to prioritize answering "well-formed" MREs over large code snippets.

You must do all exams entirely on your own, **without any assistance from others.** You can do internet searches. But, you cannot post questions or actively communicate with others during the exam period. Doing so may be considered a violation of the honor code (see below) and, depending on the nature of what you did, will result in no credit on the assignment or exam and a formal report to Georgia Tech or edX.

Honor code. All course participants—you and we—are expected and required to abide by the letter and the spirit of the Georgia Tech and edX Honor Codes. In particular, always keep the following in mind:

- Ethical behavior is critical in all facets of life. We expect honest and ethical conduct at all times.
- You are responsible for completing your work.
- Any learner found in violation of the edX Honor Code will be subject to any or all of the actions listed in the edX Honor Code.

Important note: No GitHub repos! Students often ask if they can post their work on GitHub or in any other public source code repository. The answer is, "no!" Doing so is, essentially, publishing your homework solutions. If we determine that you have done so, we will consider it a violation of the honor code and take action accordingly.

Of course, you might want to start or expand your online portfolio of sample code. However, it is better to do so with your *own* projects, that you develop from scratch, rather than with our homeworks.

What resources are available to me?

(Learning resources, books, materials, equipment.)

Resources. This course is typically taken by many students with varying backgrounds, so we provide several ways to interact with the teaching staff. You do not need to use them all! Pick and choose based on your

- **Online discussion forum** (*for all students*): All announcements go here, and the discussion forum should be your default stop for help. See this explanation for more information.

- **Notebook Office Hours** (*for all students, held virtually and recorded; GT on-campus have an additional in-person office hour, to be announced in class*): If you need help with the current week's homework (lab notebook), you can ask your questions in this live office hour.
- **Data Analysis in Python Bootcamp** (*held virtually; live interaction for GT students only, but with session recordings made available to both GT+VMM students*): For students who come into the class without a rigorous programming background, the Bootcamp sessions cover Python data analysis techniques, approaches, and methodologies to help succeed in the course. Some of the topics are specific to this class and its tools, but all of the topics can be generalized to other scenarios and toolsets. A self assessment notebook is posted in Canvas/edX, for students to determine if attending (or viewing recordings of) the sessions would be useful for themselves.
- **Python Student Success Service** (*held virtually for online or on-campus GT students only; no recordings*): These sessions cover elementary Python in a small group "drop-in" format (no advance sign-up required). They are led by GT undergrads ("shepherds") who have **not** taken this class but *are* fluent in Python. The intended audience are students who need more help than what the preceding resources provide, especially if they feel shaky on the programming prerequisites; the small-group format allows more opportunities to ask the shepherds specific questions about those concepts. These sessions are held virtually 3-4 evenings per week (US East Coast time) depending on staffing levels. These are *not* recorded.

Books & "equipment."" The main pieces of equipment you will need are a pen or pencil, paper, an internet-enabled device, and your brain!

We highly recommend the following three references for this course.

- Jake Vanderplas. A Whirlwind Tour of Python (https://jakevdp.github.io/WhirlwindTourOfPython/). The full "source" of this book and a PDF version are freely available online (see the previous link).
  - The iPad app, tinkerstellar (https://tinkerstellar.com/), replicates the main contents of this book in an interactive format.

- William McKinney. Python for Data Analysis: Data wrangling with Pandas, NumPy, and IPython, 2nd edition (http://shop.oreilly.com/product/0636920050896.do). O'Reilly Media, September 2017. ISBN-13: 978-1449319793. The e-book version is freely available through the Georgia Tech Library for GT students (https://galileo-gatech.primo.exlibrisgroup.com/discovery/fulldisplay?

docid=alma9914344441902947&context=L&vid=01GALI_GIT:GT&lang=en&search_scope=MyInst_and_CI&adaptor=Local%20Search%20Engine&isFrbr=true&query=any,contains,9915166251102931&sortby=date_d&facet=frbrgroupid,include,9064268955907375883&offset=0). Otherwise, you can also buy it on Amazon (https://www.amazon.com/Python-Data-Analysis-Wrangling-IPython/dp/1491957662/ref=sr_1_4?s=books&ie=UTF8&qid=1513180913&sr=1-4&keywords=python+for+data+analysis)

- Jake Vanderplas. Python for Data Science (https://jakevdp.github.io/PythonDataScienceHandbook/). The full "source" of this book and a PDF version are freely available online (see the previous link).

AI assistants like ChatGPT can be a great resource while you are learning the material. However, for exams, we disallow its use—see the explanation of exams for more information.

I have more questions. Where do I go for help?

(Course discussion forum and office hours.)

The primary way for us to communicate is through the online discussion forum, Piazza (https://piazza.com).

We will post instructions on how to reach this site when the course opens. We will make all course announcements and host all course discussions there. Therefore, it is imperative that you access and refer to this forum when you have questions, issues, or want to know what is going on as we progress. You can post your questions or issues anonymously if you wish. You can also opt-in to receive email notification of new posts or follow-up discussions to your posts.

Here are some pro-tips to improve the response time for your questions:

1. Make your post public, rather than private to the instructors. By doing so, anyone in the class can see and respond to your post.
2. Adhere to the "Collaboration Policy," above (see *Can I work with others?*). If you create a post that violates this policy, the instructors may ignore your post or even delete it.
3. Post during the week rather than the weekend. The staff is also trying to maintain some semblance of work-life balance. You can expect slower responses over the weekend.
4. Be sure to tag your post with the relevant notebook assignment so we can better triage issues. (In Piazza, a "tag" is also a "folder," though unlike desktop folders, you can place a post in more than one folder.)

5. For private questions (things you do not want posted publicly to your peers), **avoid** emailing the professor or members of the teaching staff directly. Instead, see below.

What if my question is private? In that case, you can make your post private to the instructors. (After pressing `new post` to create the post, look for the `Post to` field and select `Individual student(s)/instructor(s)`. Then type `Instructors` to make the post visible only to all instructors—it's vital to include all instructors so that all of them will see and have a chance to address your post, which will be faster than sending only one person.)

Office hours. We will have live office hours, to-be-scheduled. Watch Piazza for an announcement and logistical details. For the different types of sessions and availability to different course sections (GT on-campus, GT online, edX/VMM), please see "Resources".

Letters of completion. Some of you require letters of completion to give to your employer to verify your final grade and the fact that you finished the course. We follow the Georgia Tech calendar (https://registrar.gatech.edu/calendar) for submitting and releasing grades. Therefore, we will not be able to provide any such letter before the official date when grades are available to students (May 5, 2025). If your employer requires something sooner, please ask for an extension **now** or contact Georgia Tech's MSA office by email at omsanalytics@gatech.edu (mailto:omsanalytics@gatech.edu).

Section-specific notes

On-campus MSA and online MSA (MSA + OMSA) at Georgia Tech

On-campus class sessions, in-person office hours, and masking. Classes will be held in-person. Office hours will be available in both in-person and virtual formats. *(Updated for Spring 2025.)*

For in-person classes and office hours, we will be monitoring the public health situation and, as conditions warrant, we may strongly advise the wearing of masks. Otherwise, masking should be considered optional.

Accommodations for individuals with disabilities (GT students only). If you have learning needs that require special accommodation, please contact the Office of Disability Services at (404) 894-2563 or http://disabilityservices.gatech.edu/ (http://disabilityservices.gatech.edu/), as soon as possible, to make an appointment to discuss your individual needs and to obtain an accommodations letter.

What about COVID-19 accommodations? For all students, we hope our late policy on assignments can accommodate many circumstances. In particular, remember that you can submit after the 48-hour grace period for half-credit up until the exam that covers that assignment. And since every lab notebook is only a few percent of your final grade, submitting a few assignments late can still leave you in the range to get an A-equivalent grade.

However, if your illness is severe enough that you must miss a substantial part of the semester, consider dropping the class or, if would only affect one or two assignments, taking an incomplete grade. You should obviously prioritize your health, and we will be happy to welcome you back in a future semester. Again, the Division of Student Life (https://studentlife.gatech.edu/) can advise you.

edX VMM notes

A confusing aspect of the edX VMM concerns identity verification. You will need to verify your identity **twice**:

1. The first ID verification is for edX. The process appears here (https://support.edx.org/hc/en-us/articles/206503858-How-do-I-verify-my-identity-). This process is part of how you get the VMM degree. It may take 5-7 days to complete.
2. The second ID verification is for exam proctoring. That is, before you can take any exam, you must verify your identity with the exam proctoring service. **You cannot take any exam** unless you've done this process.

Because these processes can be confusing and result in unexpected delays, you are **strongly encouraged** to complete both as soon as possible after the class begins. More details will appear in edX and in the Piazza discussion forums.

Planned Topics for Spring 2025

The topics are divided into roughly three units, as outlined below. A more detailed schedule will be posted when the class begins, but the typical pace is 2 topics and notebooks per week.

Module 0: Fundamentals.

- Topic 0: Course and co-developer intros
- Topic 1: Python bootcamp review + intro to Jupyter
- Topic 2: Pairwise association mining

    - Default dictionaries, asymptotic running time
- Topic 3: Mathematical preliminaries
    - Probability, calculus, linear algebra
- Topic 4: Representing numbers
    - Floating-point arithmetic, numerical analysis

Module 1: Representing, transforming, and visualizing data.

- Topic 5: Preprocessing unstructured data
    - Strings and regular expressions
- Topic 6: Mining the web
    - (Ungraded; notebook only) HTML processing, web APIs
- Topic 7: Tidying data
    - Pandas, merge/join, tibbles and bits, melting and casting
- Topic 8: Visualizing data and results
    - Seaborn, Bokeh
- Topic 9: Relational data (SQL)

Module 2: The analysis of data.

- Topic 10: Intro to numerical computing
    - NumPy / SciPy
- Topic 11: Ranking relational objects
    - Graphs as (sparse) matrices, PageRank
- Topic 12: Linear regression
    - Direct (e.g., QR) and online (e.g., LMS) methods
- Topic 13: Classification
    - Logistic regression, numerical optimization
- Topic 14: Clustering
    - The k-means algorithm
- Topic 15: Compression

- Principal components analysis (PCA), singular value decomposition (SVD)

> *Wiki rev:* `85a7e85`