# MODERN DATA STRUCTURES

**Columbia University**
**GR5072, Spring 2025**
**Monday, 6:10-8:00PM**
**330 Uris Hall**

| | | |
|---|---|---|
| **Instructor:**<br>**Email:**<br>**Office Hours:** | Nick Anderson<br>nla2121@tc.columbia.edu<br>Held on Thursdays at 3:30 – 4:30 PM via this zoom link. I am also available by appointment if you can't make it then. | |
| **TAs:** | Min Zhuang | Vrinda Ahuja |
| **Email:** | mz3067@columbia.edu | vva2113@columbia.edu |
| **Office Hours:** | Wednesday, 1 – 2<br>Zoom link<br>ISERP (IAB) 270B | Monday, 4:30 – 5:30<br>Zoom link<br>Location TBA weekly on Ed Discussions |

## I. Overview

This course is intended to provide some foundational day-to-day tools for a data scientist in industry. Participants will learn how to access and clean data, organize and test code using functions and classes, pull data via web scraping, and use Github to keep code version-controlled and in sync with others. In addition, the course will enable participants to work with APIs, deal with data in a variety of formats, and access databases via SQL.

The goal is to prepare students for their role as a data scientist (and incidentally a flavor of what would be expected of them in a typical data science interview.) Each week will have simple, moderate, and complex examples in class, with code to follow. Students will then practice additional exercises at home. No analysis or visualization (beyond just basic tables and plots to make sure everything was correctly organized) will be taught; and this will free up substantial time for the "nitty-gritty" of the data science workflow.

Some familiarity with Python, in particular with regards to the base functions (i.e., lists, tuples, dictionaries etc.) is assumed. If you have no prior Python experience, you can still take the class but you may have to spend some additional time studying Python as the class starts (The first week assigns a full review of base Python).

To help me get to know you, please respond to this survey.
**Course Website:** Lecture materials, exercises, and HWs will be made available on the Course GitHub repository only (except for the first and second weeks, as you may not have an github account or know how to use it yet). All enrolled students need to add

their GitHub account information to this form to be added.

## II. Course Resources

**Textbooks:** There are no required textbooks for this course, but you will find these to be very useful in addition to the lectures and course readings:

- McKinney, Wes. (2017). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. O'Reilly Media: Good manual for understanding the capabilities of pandas as well as its strengths written by pandas' creator. Learn how to manipulate, process, clean, and crunch data in Python and learn Pandas for real work
- Sweigart, Al. (2019). Automate the boring stuff with Python: practical programming for total beginners. No Starch Press: Really useful worked examples on how to use Python to automate part of your workflow. For the 'total beginner' is probably an overstatement - well suited even for advanced Pythonistas who want to learn some additional tricks
- Reitz, K., Schlusser, T. (2016). The Hitchhiker's Guide to Python: Best Practices for Development (1st Edition). O'Reilly Media. - Not so much to learn Python, but a set of recommendations and best practices for making your code more "pythonic". A bit dated now.
- VanderPlas, Jake. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.
- Matthes, E. (2019). Python crash course: a hands-on, project-based introduction to programming. No Starch Press: Comprehensive guide to Python for beginners.

**Software:** The course will rely primarily on **Python** and **GitHub**.

- To install Python, I recommend downloading and installing the Anaconda Distribution, which will automatically install most of the Python libraries we'll be using
- Sign up for a GitHub account if you don't have one already
- Install GitHub Desktop, which is a desktop client that simplifies the use of git and nearly 100% of the functionality you will ever need
- Python resources
  - Python Cheat Sheet: A refresher and reference for basic Python commands
  - Awesome Python: A curated list of great Python packages and tools

**Cloud computing: AWS Academy, or Google Cloud and Databricks** allow you to leverage data at scale. We will discuss cloud computing near the end of the course, but we will not have resources to login and execute commands in these cloud environments.

**Data camp:** This class is supported by Datacamp, an intuitive learning platform for data science. Learn R, Python and SQL the way you learn best through a combination of short expert videos and hands-on-the keyboard exercises. This course will allow you

to access Datacamp for free.

The syllabus indicates Datacamp modules to complement the lecture material and graded assignments. Of course, feel free to try out other offerings you are interested in from Datacamp's high quality content or other comparable service providers.

**Communications:** An Ed Discussions workspace for this course will serve as the primary means of written communication before, during and after class, where students can communicate with each other and with instructors. You can also email me any questions you may have. When it comes to questions about the HW, ideally you would post these on Ed Discussions so that all students will have access to the same clarifications / expectations.

## III. Course Dynamics

**Expectation of Regular Participation and Utilization of Course tools**: We will be monitoring student participation and completion of assignments using the corresponding tools throughout the semester. We want to make sure that students are consistently engaged, and if that becomes difficult, that students alert us to their situations.

**In preparation for each class:** You should have (i) consumed all the curated materials listed in the syllabus for the week, including doing all Data Camp exercises before coming to class; (ii) posted any questions you have on **Ed Discussions** and contributed to answer other students' questions; and (iii) submitted homework assignments for that week using **Course Works**.

**During each live class:** We will engage in a combination of lecture and live-workshop where we will run through code examples, troubleshoot and answer questions. Starter code for in-class exercises will be made available through GitHub. You will need to **bring a laptop to class** to follow along the coding tutorials and examples.

## IV. Course requirements

The grade for this course will depend on the fulfillment of these main requirements:

**(i) Attendance and Class participation (10%):** Students are required to attend and actively participate in class exercises. Participation grades will depend on participation in class and in the online Ed Discussion workspace, where you can ask / answer questions about the topics we are learning.

**(ii) Data Camp exercises (20%):** The syllabus lists data camp exercises to complete throughout the course – these should be completed before coming to class on the week they are assigned. If a Data Camp assignment is not listed as optional below, then it is mandatory and should be completed **BEFORE** class each week. These are a completion grade and will generally take you 2-4 hours to complete each week.

a. You will be able to drop (or skip) **three** weeks of assigned Data Camp lessons

**(iii) Take-home exercises (30%):** There will be 6 or 7 homework assignments, which are expected to be turned in before class each week as instructed on GitHub.

**(iv) Group project (10%):** There will be one group project which has two or three questions which will require you to work together as a team. Each group will be 3-4 members. You can select your groups or let me pick for you (e.g., if you have a friend making a group of 2, I will find you 2 more members). Your group work must be linked by a github repo, with all group members making at least 2 meaningful commits.

**(v) Final Exam (30%)**: The final examination will require students to generate code to perform common data wrangling operations that are required for the successful completion of most data analysis projects and will use mostly pandas. The exam will be held onsite.

***Late Submission Policy:*** All class assignments are expected to be submitted on the due date. Please note that 10% of the maximum grade will be deducted from the score for every day the assignment is submitted late.

## V. Course Outline

**Week 1 (1/27)– Course Introduction & Setting up Project Workflow**

- I suggest coming to class with python installed already, as we will quickly jump in and review/introduce base python. Depending on your comfort with python, a lot of this may be review to you.
- **On your own:**
  - Install Python. I recommend you using the Anaconda Distribution as described above. The Anaconda Distribution includes most Python libraries we will need this semester. It also comes with Jupyter Notebook and Spyder, two IDEs (Integrated Development Environments) that we will use. Throughout the semester, we will mostly rely on Jupyter Notebooks to share code and complete HW assignments.
  - Sign up for GitHub (if you don't already have one) and respond to this Google survey with your account details be added to the course repo since it's private.
  - Install GitHub Desktop. We will be using this client to interact with GitHub. I've personally found it much easier to use than using command-line Git, and as you will see the learning curve is not bad. If you are confident using the command-line for Git, or have another software you like (i.e., Git Kraken), then feel free to skip this step.
- **Reading:**
  - McKinney, chapter 2 *Python Language Basics, IPython, and Jupyter Notebooks*
- **Datacamp:**
  - (Optional) To review base Python complete the course Introduction to Python

---

**PART I: Data Manipulation and Coding**

**Week 2 (2/3) – Git, GitHub, and Markdown in Jupyter Notebook**

- **Reading:**
  - GitHub Desktop Documentation: We will cover the basics in class of how to create/clone repositories, make commits etc. So don't go crazy diving into this, but browse through it. This is useful documentation if needed, we will learn everything in class
  - .gitignore
  - The Ultimate Markdown Guide (for Jupyter Notebook)
  - (Optional) Hello World - GitHub for the non-programming beginner, for those interested in command-line Git
- **Datacamp:**
  - (Optional) Introduction to Git: Feel free to check out this comprehensive introduction to Git, which emphasizes command-line Git.
- **Homework 1:** Using Jupyter Notebooks and GitHub
  - Only available on GitHub in the repository **QMSS-GR5072-Spring2025-HWs** in the **homework-1-main directory**
  - You must be invited to this repo, as it is private. Let me know your GitHub username by responding to the google survey posted above under week 1. Your assignment is saved as a README.md file, which I suggest you read directly on the GitHub website.
  - Submit your HW files on GitHub in the following directory:
    - ~/Submissions/LASTNAME-FIRSTNAME

**Week 3 (2/10) – numpy array objects**

- **Reading:**
  - McKinney, chapter 4 NumPy Basics: Arrays and Vectorized Computation
- **Datacamp:**
  - Introduction to numpy

**Week 4 (2/17) – Data Wrangling with pandas**

- **Reading:**
  - McKinney, chapter 5 *Getting Started with pandas;* chapter 7 *Data Cleaning and Preparation;* chapter 10 *Data Aggregation and Group Operations*
  - Look at how the speed of pandas compares to R here. Despite widespread claims made by Python users, pandas is actually slower than R
- **Datacamp:**
  - Data Manipulation with pandas
  - (Optional) Joining data with pandas - chapter 1 & 2 are the standard operations
- **Homework 2:** Data wrangling using Pandas

**Week 5 (2/24) – Functions I: Basic structure and logic**

- **Reading:**
  - Sweigart, *Automate the boring stuff with Python, chapter 3, Functions*
  - Some basics of code styling
- **Datacamp:**
  - Chapter 4 on *Loops* in Intermediate Python
  - Introduction to Functions in Python
- **Homework 3:** for loops and functions

**Week 6 (3/3) – Functions II: For loops, nested and complex operations**

- **Reading:**
  - McKinney, chapter 3 *Built-In Data Structures, Functions, and Files*
- **Datacamp:**
  - Python Toolbox
  - Context managers and decorators chapters from Writing functions in Python

**Week 7 (3/10) – Working with strings and classes in Python**

- **Reading:**
  - McKinney, chapter 7, *7.3 String Manipulation (p.211-218)*
  - Sweigart, chapter 7 *Pattern Matching with Regular Expressions*
  - Python RegEx
- **Datacamp:**
  - Regular expressions in Python
  - (Optional) Utilizing classes chapter from Software Engineering Principals in Python
- **Homework 4:** Functions II

**No Class! (3/17) – SPRING BREAK!**

**Week 8.1 (3/24) – Class cancelled!**

**Week 8.2 (3/31) – Testing**

- **Reading:**
  - Unit Testing for Data Scientists
  - Chapter 5 *Testing* from Hillard, Dane. (2020). "Practices of the Python Pro." Manning Publications (1st edition). Harvard. (PDF)
  - Reitz, K., Schlusser, T. The Hitchhiker's Guide to Python, *Testing* (p72-81)
  - Exception and Error Handling in Python
- **Datacamp:**
  - Introduction to Testing in Python

---

**PART II: Python and leveraging other software tools**

**Week 9 (4/7) – Introduction to Relational Databases and Advanced Excel**

- **Reading:**
  - Excel VLOOKUP Tutorial for Beginners: Step-by-Step Examples
  - Short cuts (Macs and Windows)
- **Datacamp:**
  - (Optional) Data Analysis in Excel: I strongly suggest you do this if you are not very confident with Excel. Based on the survey results at the beginning of the semester, many of you could benefit from parts or most of this course

**Week 10 (4/14) – Working with SQL**

- **Reading:**
  - There are lots of great SQL Tutorials to go further. Here are a few pointers:
    - Codeacademy's SQL Tutorial
    - SQLzoo's SQL Tutorial
    - W3School.com SQL reference (also interactive)
  - Practice using SQLZOO
- **Datacamp:**
  - (Optional) Introduction to SQL: Only do this if you are completely new to SQL, chapter 1 is a bit basic but chapter 2 is helpful
  - Joining Data in SQL: Joins in SQL are essential, especially given the usually relational nature of the data.
  - (Optional) Working with relational databases in Python Chapter from Introduction to Importing Data in Python

**Week 11 (4/21) – Working with APIs**

- **Reading:**
  - McKinney, *Interacting with Web APIs* (Section in Chapter 6)
  - API Integration in Python – Part 1
  - Getting started with APIs in Python to Gather Data
- **Datacamp:**
  - Intermediate Importing Data in Python
- **Homework 5:** APIs

**Week 12 (4/28) – Web scraping from HTML**

- **Reading:**
  - Sweigart, *chapter 12 Web Scraping*
- **Datacamp:**
  - Web scraping in Python
- **Homework 6:** Web Scraping

**Week 13 (5/5) – Cloud computing: Databricks**

- **Reading:**
  - TBD
- **Datacamp:**
  - (Optional) Introduction to Databricks
  - (Optional) Introduction to AWS Boto in Python

**Week 14 (5/5 - NO CLASS) – Review and explore other MDS topics**

- To be fully sufficient in your ability as a Python data scientists, understanding how to handle the following topics is important:
  - Duplicates
  - Missing data
  - Dates
  - JSON & XML data
- Here are some other topics related to MDS which we didn't cover, and are good places to continue this education as you have time:
  - Leveraging generative AI into your workflow
  - Machine learning, NLP and AI in Python
  - Multicore processing in Python
  - Airflow and Spark
  - pyenv and poetry
  - Databricks
  - Coding best practices: Read this article by Donald Knuth. This article is based on his book "The Art of Computer Programming." This book was "named among the best twelve physical-science monographs of the century by American Scientist, along with: Dirac on quantum mechanics, Einstein on relativity…" (Source)
- **Reading (optional):**
  - Managing Multiple Python Versions With pyenv

    - chapter 6 Shipping Great Code from Reitz, K., Schlusser, T. The Hitchhiker's Guide to Python
    - An Overview of Packaging for Python
- **Datacamp (optional):**
    - (Optional) Introduction to Airflow in Python
    - (Optional) The course *Introduction to PySpark* is beyond what we can cover but provides a good start on Spark using R. More info about the Big Data benefits and use of Spark is discussed in Big Data Fundamentals with PySpark

**Week 15 (5/12) – Final Exam**

**Statement on Academic Integrity**

Columbia's intellectual community relies on academic integrity and responsibility as the cornerstone of its work. Graduate students are expected to exhibit the highest level of personal and academic honesty as they engage in scholarly discourse and research. In practical terms, you must be responsible for the full and accurate attribution of the ideas of others in all of your research papers and projects; you must be honest when taking your examinations; you must always submit your own work and not that of another student, scholar, or internet source. Graduate students are responsible for knowing and correctly utilizing referencing and bibliographical guidelines. When in doubt, consult your professor. Citation and plagiarism-prevention resources can be found at the GSAS page on Academic Integrity and Responsible Conduct of Research.

Failure to observe these rules of conduct will have serious academic consequences, up to and including dismissal from the university. If a faculty member suspects a breach of academic honesty, appropriate investigative and disciplinary action will be taken following the Dean's Discipline procedures.

**Statement on Disability Accommodations**

If you have been certified by Disability Services (DS) to receive accommodations, please either bring your accommodation letter from DS to your professor's office hours to confirm your accommodation needs, or ask your liaison in GSAS to consult with your professor. If you believe that you may have a disability that requires accommodation, please contact **Disability Services** at 212-854-2388 or disability@columbia.edu.

***Important:*** To request and receive an accommodation you must be certified by DS.