QTM 350 - Data Science Computing

Danilo Freire

2025-01-01

Contents

Course Description

Welcome to QTM 350! This course introduces key tools in modern data science, focusing on three essential aspects: reliability, reproducibility, and robustness. We will cover the command-line interface, version control with Git and GitHub, and literate programming using Quarto and Jupyter Notebooks. You will also learn about data storage and manipulation with SQL and Pandas, and parallel computing with Dask. We will explore artificial intelligence-assisted programming with GitHub Copilot and finish with Docker and containerisation.

By working with real-world datasets and problems, students will gain hands-on experience using these tools and methods to extract insights from data. This course will develop technical skills and critical thinking needed to solve complex data challenges. Upon completion, students will be prepared to apply these tools to their own research and professional work.

Learning Objectives

By the end of this course, students will be able to:

- Use the command line interface to manage files and directories.
- Work with version control systems to track changes in code and collaborate with others.
- Create reproducible reports and presentations.
- Use AI tools to assist with programming tasks.
- Apply advanced techniques for data storage, manipulation, and querying.
- Understand the basics of containerisation and parallel computing.

Course Requirements

Some knowledge of programming is recommended, and familiarity with basic data manipulation and visualisation techniques is helpful. However, no prior experience with the tools covered in the course is required.

In terms of software, you will need to install the following tools: Anaconda distribution of Python 3.x, VS Code, PostgreSQL, GitHub Desktop, Git, Docker, Quarto, Dask, GitHub Copilot.

Please feel free to reach out if you have any questions about the course content or your readiness to take the class.

Materials

This course is designed to be self-contained, providing all the necessary resources and materials to succeed in mastering the core concepts. However, students are encouraged to explore the following suggested books and online courses to deepen their understanding of the topics covered in the course.

Suggested Books

- Python for Data Analysis by Wes McKinney
- Elements of Data Science by Allen Downey
- SQL for Data Scientists by Renee M. P. Teate
- Data Science on the Command Line by Jeroen Janssens
- Docker for Data Science by Joshua Cook
- Pro Git by Scott Chacon and Ben Straub
- Free programming books

Online Courses

- Coursera: Python for Everybody Specialisation
- edX: Python Basics for Data Science
- Codecademy: Learn Python
- DataCamp: Introduction to SQL
- Coursera: SQL for Data Science
- Coursera: Introduction to Git and GitHub
- Microsoft Learn: GitHub Copilot Fundamentals

Documentation

- Official Python Documentation
- NumPy Documentation
- Pandas Documentation
- Matplotlib Documentation
- PostgreSQL Documentation
- Git Documentation
- GitHub Documentation
- Dask Documentation
- GitHub Co-Pilot Documentation
- Docker Documentation

Course Information

We will meet every Monday and Wednesday from 16:00 to 17:15 in the Psychology Building 230. It is important that you read the materials before class. All information about the course is available on the course's GitHub repository at `https://github.com/danilofreire/qtm350-summer`. While I will try to adhere to the course schedule as much as possible, I also want to adapt to your learning pace and style. The syllabus and course plan may change in the semester. Again, please check the course repository regularly to check for updates. I will also announce any changes in class and via email.

Software

We will mainly use Python in this course. Python is a free and powerful programming language that is widely used in data science, machine learning, and scientific computing. I recommend using the Anaconda distribution as it comes with many necessary Python libraries for data analysis, such as Pandas, NumPy, and Jupyter.

You can write your Python code in any text editor, but I recommend VS Code with the Python extension. Pycharm is also well-regarded by developers. If you are feeling adventurous, you can also use

Neovim with the coc-pyright plugin. That is, if you can exit the editor. :)

We will use SQLite for database management. SQLite is a lightweight, serverless, and self-contained database engine that is widely used in data science and web development. You can also easily integrate it with Python using the sqlite3 library. You are also free to use other databases, such as PostgreSQL, MySQL, or MongoDB, if you prefer.

We will also use Jupyter Notebooks and Quarto in class. Jupyter itself comes pre-installed with Anaconda, but please install the Jupyter extension for VS Code as well. To install Quarto, please follow the instructions on the official website. We will have a hands-on session to learn how to use both of them (but I assume you are already familiar with Jupyter).

Please also install Docker to work with containers. Docker is a platform for developing, shipping, and running applications in containers. Containers allow you to package your application and its dependencies together into a single unit. This makes it easy to ensure that your application will run on any other machine, regardless of any custom settings that machine might have that could differ from the machine that was used for writing and testing the code.

Finally, we will use GitHub for version control. Please create a free account on GitHub and install GitHub Desktop to manage your repositories. We will also use Git in the course. Git is a distributed version control system that allows you to track changes in your codebase and collaborate with others. You can install Git from the official website.

To help you get started, I have prepared a series of tutorials on how to install Anaconda, Jupyter, SQLite, VS Code, GitHub Copilot, and open a free educational account on GitHub. Please follow these tutorials to ensure that you have all the necessary tools for the course.

Office Hours

I am very flexible with office hours, and we can schedule an online meeting at any time that works for you. Feel free to send me a message at `danilo.freire@emory.edu`, and I will likely reply within a few hours. If you prefer, you can meet me in the afternoon at my office. My office address is in the Psychology and Interdisciplinary Sciences Building, 36 Eagle Row, room 480. If possible, please email me before coming to ensure that no two students book the same time slot.

Academic Integrity

Upon every individual who is a part of Emory University falls the responsibility for maintaining in the life of Emory a standard of unimpeachable honour in all academic work. The Honour Code of Emory College is based on the fundamental assumption that every loyal person of the University not only will conduct his or her own life according to the dictates of the highest honor, but will also refuse to tolerate in others action which would sully the good name of the institution. Academic misconduct is an offense generally defined as any action or inaction which is offensive to the integrity and honesty of the members of the academic community. Any suspected case of academic misconduct will be referred to the Emory Honour Council.

Artificial Intelligence

Students have to submit ten problem sets and complete five in-class quizzes. You are allowed to use AI to assist with your assignments. I recommend using GitHub Copilot to generate code snippets, as it is free for students and provides good suggestions and explanations. Claude, ChatGPT, and Perplexity AI are also good tools. I am available to provide support and assistance with these tools during office hours or by appointment. However, please note that any errors or omissions resulting from the use of AI tools are your responsibility. Do not rely solely on AI to complete your assignments; you must always double-check your work. Remember to cite all sources used in your problem sets and projects, including AI tools. Please include a note at the end of any document indicating that AI was used in its development.

Special Needs and Accessibility Services

I am committed to providing necessary accommodations to ensure all students have an equal opportunity to succeed in this course. Students with medical or health conditions that may impact their academic performance should visit the Department of Accessibility Services (DAS) to determine eligibility for appropriate accommodations. Those who receive accommodations should provide me with an Accom-

modation Letter from DAS at the beginning of the semester or as soon as the accommodation is granted. Please note that DAS accommodations, such as extra time or quiet spaces, will apply only to quizzes, not assignments. This is because assignments are released in advance, allowing students to work at their own pace. Athletes and students with other commitments should also inform me of any scheduling conflicts at the beginning of the semester. I will do my best to accommodate these students, but I cannot guarantee that all requests will be granted. If you have any questions or concerns, please contact me.

English Language Learners

Emory University welcomes students from around the country and the world, and the unique perspectives international and multilingual students bring enrich the campus community. To empower multilingual learners, an array of support is available including language and culture workshops and individual appointments. For more information about English Language Learning support at Emory, please contact the ELLP Specialists at `https://writingcenter.emory.edu`. No student will be penalised for their command of the English language.

Assignments and Grading Policy

Problem Sets (50%). There will be ten problem sets throughout the course. These assignments are designed to reinforce concepts covered in lectures and readings, and to provide hands-on practice with statistical programming. Problem sets will include a mix of theoretical questions and practical applications. They will be assigned regularly and must be completed individually. You may discuss your work with other colleagues as long as you do not copy entire sentences, just changing a few words. If you worked with other students, please write down their names on your assignment. Please also acknowledge any sources you used in your work, including textbooks, articles, and AI resources. *Any assignment submitted after the due date/time will be penalised by 10% per day.* Please submit your assignments as Jupyter Notebooks (`.ipynb`) or `.pdf` files via Canvas or email until midnight on the due date.

Class Quizzes (30%). Students will also take five in-class quizzes throughout the semester. These quizzes will be based on the lectures from the previous weeks. They will be designed to test your understanding of the material and your ability to apply the concepts to new problems. Quizzes will be open-book and open-notes, and students have the entire class period to complete them. They are individual assessments, and students are not allowed to discuss the questions with their colleagues in class.

Final Project (20%): For the final project, you will work in groups of three to four to create a short report. This report will require you to apply the tools and methods we have covered in class to a real-world dataset. You should host your report in a GitHub repository, use Quarto for the document, and employ SQL for data manipulation. Make sure to include visualisations and statistical analyses as well. The project is due on the last day of class. You can find the instructions in the course GitHub repository.

Grading Scale

Each student's final grade will be based on the following after rounding up to the nearest point:

Grade	A	A-	B+	B	B-	C	D	F
Range	91%–100%	86%–90%	81%–85%	76%–80%	71%–75%	66%–70%	60%–65%	<60%

Course Outline and Suggested Readings

The lecture notes cover all the necessary material for the course, and the weekly suggested readings are recommended for those who want to deepen their understanding of the course topics. As mentioned above, the course outline is subject to change, and I will update the syllabus if needed. Please remember to check the course GitHub repository regularly. Lecture notes, assignments, and other materials will be posted there as the course progresses.

Module 01: Introduction, Computational Literacy, and Command Line Interface (CLI)

Friday, May 16:

- Syllabus and course repository: `https://github.com/danilofreire/qtm350-summer`.
- Lecture 01: Welcome to QTM 350 - Introduction.
- Lecture 02: Computational Literacy.
- Course Tutorials: How to Install Anaconda, Jupyter, PostgreSQL, VSCode, and Open a Free Educational Account on GitHub.

Suggested references:

- Cleveland, W. S. (2001). Data science: An action plan for expanding the technical areas of the field of statistics. International Statistical Review, 69(1), 21-26.
- Donoho, D. (2017). 50 Years of Data Science. Journal of Computational and Graphical Statistics, 26(4), 745-766.
- Breiman, L. (2001). Statistical Modeling: The Two Cultures (with Comments and a Rejoinder by the Author). Statistical Science, 16(3), 199-231.
- Brady, H. E. (2019). The Challenge of Big Data and Data Science. Annual Review of Political Science, 22(1), 297-323.
- Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., & Hoffman, M. M. (2019). Machine Learning for Integrating Data in Biology and Medicine: Principles, Practice, and Opportunities. Information Fusion, 50, 71-91.
- Campbell-Kelly, M., Aspray, W. F., Yost, J. R., Tinn, H., & Díaz, G. C. (2023). Computer: A History of the Information Machine. Routledge.
- Shalf, J. (2020). The Future of Computing beyond Moore's Law. Philosophical Transactions of the Royal Society A, 378(2166), 20190061.
- Al-Hashimi, H. M. (2023). Turing, von Neumann, and The Computational Architecture of Biological Machines. Proceedings of the National Academy of Sciences, 120(25), e2220022120.
- Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35.

- Videos: David J. Malan - Abstraction, Khan Academy - Hexadecimal Number System, Matthias Wandel - Marble Adding Machine, Crash Course - Early Computing and Electronic Computing (the last two are quite entertaining!).

Monday, May 19:

- Lecture 03: Encoding Information & Introduction to Programming.
- Lecture 04: Command Line Interface.
- **Assignment 01:** Problem Set 01.

Suggested references:

- Janssens, J. (2021). Data Science at the Command Line: Obtain, Scrub, Explore, and Model Data with Unix Power Tools (2nd ed.). O'Reilly Media.
- Levy, J. (2024). The Art of Command Line. GitHub.
- Shotts, W. (2019). The Linux Command Line: A Complete Introduction. No Starch Press.
- Healy, K. (2019). The Plain Person's Guide to Plain Text Social Science. Chapters 1-5.
- Kerr, D. (2024). Effective Shell.
- Irianto, I. (2021). Learn Vim (the Smart Way).
- Neil, D. (2015). Practical Vim: Edit Text at the Speed of Thought. Pragmatic Bookshelf.
- Dennis, J. Your problem with Vim is that you don't grok vi. (Stack Overflow).
- Vim Adventures. (Instructor's note: this is a fun, albeit cringy, way to learn Vim).
- Videos: freeCodeCamp - Command line crash course, Percy Grunwald - Absolute beginner guide to the macOS terminal, NetworkChuck - 50 macOS tips and tricks using terminal

Module 02: Version Control with Git and GitHub

Wednesday, May 21:

- Lecture 05: Command Line Interface Continued.
- Lecture 06: Version control with Git and GitHub.
- **Assignment 01 due (5%).**

Suggested references:

- Chacon, S. and Straub, B. (2014). Pro Git. Apress. (Instructor's note: this is *the book* on Git).
- GitHub tutorials: GitHub skills (recommended), Git guides, GitHub learning lab, Best practices for repositories.

Friday, May 23:

- Lecture 07: More Git and GitHub: pull requests, issues, pages, and collaboration features.
- Lecture 08: Practice.
- Kahoot Quiz.
- **Assignment 02:** Problem Set 02.

Suggested references:

- Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R. M., Blin, K., & Vizcaíno, J. A. (2016). Ten Simple Rules for Taking Advantage of Git and GitHub. PLOS Computational Biology, 12(7), e1004947.
- Beckman, M. D., Çetinkaya-Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J., & Tackett, M. (2021). Implementing version control with git and GitHub as a learning objective in statistics and data science courses. Journal of Statistics and Data Science Education, 29(sup1), S132-S144.
- Escamilla, E., Klein, M., Cooper, T., Rampin, V., Weigle, M. C., & Nelson, M. L. (2022). The Rise of GitHub in Scholarly Publications. arXiv preprint arXiv:2208.04895.

(**More lectures to be added soon**)