QTM 350 - Data Science Computing

Danilo Freire

Fall 2024

E-mail: danilo.freire@emory.edu
Office Hours: Mon-Fri Afternoon
Office: 36 Eagle Row, room 480

Web: github.com/danilofreire/qtm151
Class Hours: Mon/Wed, 16:00-16:50
Classroom: XXX

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 command line interfaces and vim, 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 confidently 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 14:30 to 15:45 in the Maths and Science Centre - E208. 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. 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, versatile, 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 PostgreSQL for database management. You can download PostgreSQL from the official website. Please also install pgAdmin and the VS Code extension for PostgreSQL to interact with the database.

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 effectively.

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, PostgreSQL, VS Code, GitHub Copilot, and open a free educational account on GitHub. Please follow these tutorials as soon as possible to ensure that you have all the necessary tools for the course.

Office Hours

I am very flexible with office hours, but it is easier to contact me via email. Feel free to send me a message any time 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. I am in the Department of Quantitative Theory and Methods almost every weekday. 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 Accommodation 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 automatically be graded for half points.* To accommodate unexpected circumstances, your lowest assignment grade will be automatically dropped at the end of the semester. *The same applies to in-class quizzes.* Please submit all 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 50 minutes to complete them. They are individual assessments, and students are not allowed to discuss the questions with their colleagues in class.

Final Project (20%). The final project will consist of a short report, created using Jupyter and using one of the datasets shared on the course GitHub repository. Further instructions will be provided in class. The final project will be due on the last day of class.

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 to Python, Jupyter, and GitHub

Wednesday, August 28:

- Syllabus and course repository: https://github.com/danilofreire/qtm350.
- Lecture 01: Welcome to QTM 350 - Introduction.
- 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.

Monday, September 02: Labour Day (no class)

Wednesday, September 04:

- Lecture 02: Computational Literacy.
- **Assignment 01:** Problem Set 01

Suggested references:

- 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!).

Module 02: Introduction to the Command Line Interface and Version Control

Monday September 09:

- Lecture 03: The Command Line Interface (CLI), Shell Basics, and File Management.

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.
- Shotts, W. (2019). The Linux Command Line: A Complete Introduction. No Starch Press.
- Levy, J. (2024). The Art of Command Line. GitHub.
- Healy, K. (2019). The Plain Person's Guide to Plain Text Social Science. Chapters 1-5.

Wednesday, September 11:

- Lecture 04: Command line tools, text files, scripting, and basics of Vim.
- **Assignment 01 due (5%).**
- **Assignment 02:** Problem Set 02.

Suggested references:

- 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.
- Videos: freeCodeCamp - Command line crash course, Percy Grunwald - Absolute beginner guide to the macOS terminal, NetworkChuck - 50 macOS tips and tricks using terminal

Monday, September 16:

- Lecture 05: Version control with git and GitHub.

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.

Wednesday, September 18:

- Lecture 06: More GitHub: pull requests, issues, pages, and collaboration features.
- **Assignment 02 due (5%).**
- **Assignment 03:** Problem Set 03.

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.

Monday, September 23:

- Lecture 07: Quiz 01: git and Github (6%).

Module 03: Literate Programming with Markdown, Quarto, and Jupyter

Wednesday, September 25:

- Lecture 08: Using Markdown, Jupyter, and Quarto for Reproducible Reports.
- **Assignment 03 due (5%).**
- **Assignment 04:** Problem Set 04.

Suggested references:

- Quarto official website.
- Awesome Quarto: https://github.com/mcanouil/awesome-quarto. Note: this repository contains dozens of tutorials, examples, and resources.
- Çetinkaya-Rundel, M. & Lowndes, J. S. (2022) Keynote talk: Hello Quarto: Share • Collaborate • Teach • Reimagine. Slides and source code. This is one of the nicest Quarto presentations I have seen.
- Getting Started with Quarto (YouTube). Note: Posit (formerly RStudio) has a series of tutorials on Quarto on their YouTube channel. You can find their playlist here.
- Markdown Guide.
- Jupyter Notebooks Documentation.
- Codecademy - How to use Jupyter Notebooks
- Course tutorial: Jupyter and Markdown

Monday, September 30:

- Lecture 09: Presentations with Quarto and GitHub Pages.

Suggested references:

- Quarto Documentation - Presentations and Websites.
- GitHub Pages Documentation.
- French, J. (2023). Creating Websites with Quarto and GitHub Pages (YouTube Playlist).
- Taylor, I. (2022). Publishing a Quarto Site to GitHub Pages

Wednesday, October 02:

- Lecture 10: **Quiz 02: Literate Programming (6%)**.
- **Assignment 05:** Problem Set 05.
- **Assignment 04 due (5%).**

Module 04: AI-Assisted Programming

Monday, October 07:

- Lecture 11: Introduction to AI-Assisted Programming and Chatbots.

Suggested references:

- Cihon, P. & Demirer, M. (2023). How AI-powered software development may affect labor markets. Brookings Institution
- Poldrack, R. A., Lu, T., & Beguš, G. (2023). AI-assisted Coding: Experiments with GPT-4. arXiv preprint arXiv:2304.13187.
- Lau, S & Guo, P. (2023). From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In Proceedings of the 2023 ACM Conference on International Computing Education Research V.1 (ICER '23 V1), August 07–11, 2023, Chicago, IL, USA. ACM, New York, NY, USA 16 Pages.
- Linus Torvalds Discusses the Impact of AI on Programming (YouTube).

Wednesday, October 09:

- Lecture 12: AI-Assisted Programming with GitHub Copilot.
- **Assignment 05 due (5%).**
- **Assignment 06:** Problem Set 06.

Suggested references:

- GitHub Copilot Documentation.
- Using GitHub Copilot in your IDE: Tips, Tricks, and Best Practices
- Using GitHub Copilot in the Command Line
- Coding with an AI Pair Programmer: Getting Started with GitHub Copilot (YouTube)
- GitHub Copilot YouTube Playlist
- Labadze, L., Grigolia, M., & Machaidze, L. (2023). Role of AI Chatbots in Education: Systematic Literature Review. International Journal of Educational Technology in Higher Education, 20(1), 56. ## Module 05: Data Manipulation with Python

Monday, October 14: Fall Break (no class)

Wednesday, October 16:

- Lecture 13: Python Data Types, Boolean Logic, and Control Structures.
- **Assignment 06 due (5%).**
- **Assignment 07:** Problem Set 07.

Suggested references:

- Python Documentation: An Informal Introduction to Python.
- Python Documentation: More Control Flow Tools.
- Python Documentation: Compound Statements.
- NumPy Documentation: Quickstart Tutorial.
- Programiz: Math Operations in Python.
- Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd ed.). No Starch Press. Chapter 02.
- Severance, C. (2016). Python for Everybody: Exploring Data in Python 3. CreateSpace Independent Publishing Platform. Chapters 3-11 (Note: Read only the chapters which interest you).

Monday, October 21:

- Lecture 14: Pandas for Data Analysis: Loading, Cleaning, and Exploring Data.

Wednesday, October 23:

- Lecture 15: Pandas for Data Analysis: Data Wrangling and Aggregating.
- **Assignment 07 due (5%).**
- **Assignment 08:** Problem Set 08.

Suggested references:

- McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (3rd ed.). O'Reilly Media. Chapter 05: Getting Started with Pandas.
- VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media. Chapter 3: Data Manipulation with Pandas.
- McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (3rd ed.). O'Reilly Media. Chapter 07: Data Cleaning and Preparation.
- DataCamp: Pandas Tutorial: DataFrames in Python.
- Real Python: Pandas Tutorial: DataFrames in Python.

Monday, October 28:

- Lecture 16: **Quiz 03: Python for Data Analysis (6%)**.

Module 06: Introduction to SQL Databases

Wednesday, October 30:

- Lecture 17: Introduction to PostgreSQL: Data Types, Tables, and Queries.
- **Assignment 08 due (5%).**
- **Assignment 09:** Problem Set 09.
- **Instructions for the Final Project**.

Suggested references:

- Mode Analytics: SQL Tutorial.
- Real Python: SQL Databases and SQLite.
- Khan Academy: SQL Basics. (Note: Khan Academy is a great resource for learning SQL and other programming languages).
- Coursera: PostgreSQL for Everybody.
- PostgreSQL Tutorial.
- PostgreSQL Documentation: SQL Commands. (Note: For reference only).

Monday, November 04:

- Lecture 18: Importing SQL Data into Python.

Wednesday, November 06:

- Lecture 19: Merging Tables in SQL.
- **Assignment 09 due (5%).**
- **Assignment 10:** Problem Set 10.

Suggested references:

- Pandas Documentation: SQL Databases.
- Real Python: Working with SQLite Databases Using Python and Pandas.
- Mode Analytics: SQL Joins.
- PostgreSQL Documentation: Joins Between Tables.

Monday, November 11:

- Lecture 20: **Quiz 04: SQL Databases (6%)**.

Module 07: Parallel Computing

Wednesday, November 13:

- Lecture 21: Parallel Computing with Dask.
- **Assignment 10 due (5%).**

Suggested references:

- Dask Documentation
- Dask Tutorial
- Coiled - Intro to Dask Tutorial (YouTube).
- Rocklin, M. (2017). Dask: Flexible Library for Parallel Computing in Python. In Proceedings of the 16th Python in Science Conference (Vol. 126, p. 130).

Monday, November 18:

- Lecture 22: Application: Parallelising Data Analysis with Dask and AutoML.

Suggested references:

- Dask Documentation: Machine Learning.
- He, X., Zhao, K., & Chu, X. (2021). AutoML: A Survey of the State-of-The-Art. Knowledge-based systems, 212, 106622.
- TPOT Documentation.

Module 08: Containers and Reproducibility

Wednesday, November 20:

- Lecture 23: Dependency Management, Virtual Environments, and Containers.

Suggested references:

- Docker Documentation

Monday, November 25:

- Lecture 24: Docker for Data Science.

Wednesday, November 27: Thanksgiving Break (no class)

Monday, December 02:

- Lecture 25: **Quiz 05: Dask, Docker and Containers (6%)**.

Wednesday, December 04:

- Lecture 26: Review and Final Project Discussion.

Monday, December 09:

- **Final Project due (20%).**