FinTech 545 – Quantitative Risk Management

Synopsis

Quantitative Risk Management offers a hands-on introduction to the science and implementation of risk analytics. Topics include probability theory, regression and time series analysis, risk metrics such as Value at Risk and Expected Shortfall, derivative valuation methods, stress testing and scenario analysis, factor models, and portfolio construction and optimization.

Each week students will be expected to implement topics learned in either the C/C++, Python, or Julia programming language. Homework projects will test students' knowledge of the theory and their ability to solve real world risk management problems. In the final 3 weeks of the semester, students will work to consolidate their code into a risk management software platform. They will use that platform for the final exam.

Success in the risk management industry rests not just on good analysis, but also the ability to present results. Besides grading for correct answers, a component of the weekly grade will rest on the student's presentation of results.

Class Schedule

The class schedule is tentative and subject to change.

Date	Topics	Project
1/8/2025	Introduction, Course Syllabus and Expectations, Review Univariate Statistics	No
1/13/2025	Multivariate Statistics, Regression Models	No
1/20/2025	Martin Luther King Day - No Class	No
1/27/2025	Financial Data -- Common issues, Monte Carlo Methods	Yes - Due 2/1
2/3/2025	Catch up and Q&A	No
2/10/2025	Value at Risk	No
2/17/2025	Advanced VaR and Expected Shortfall	No
2/24/2025	Options	Yes - Due 3/1
3/3/2025	Options II and Portfolio Construction	No
3/10/2025	Spring Break - No Class	No
3/17/2025	Attribution and Risk Parity	No
3/24/2025	Advanced Topics	Yes - Due 3/29
3/31/2025	Speaker	No
4/7/2025	Speaker	No
4/14/2025	Catch up and Q&A	No
5/1/2025	Final - Hudson 125	No

Grading

Final grade will be calculated as

- 60% weekly projects
- 40% final exam

Homework Projects

Students will be assigned a project testing and expanding on their knowledge from the topics presented up to that week. Projects will include data and questions – students are expected to analyze the data and present their findings.

Projects will be due electronically by 8am Saturday on the dates outlined above. Students are expected to turn in the following

1. A paper in PDF format answering the questions presented.
2. Code and other files used to create results.

After each project is due 1 or more students will be called upon to present their project. Verbal presentation and defense of results will not harm the student's grade, but an excellent presentation can add bonus points. If a student feels strongly that they would like to present, please let me know prior to class (preferably on Saturday when you turn in your work).

Projects will be graded on a 20 point scale. 18 points will be awarded based on the paper demonstrating a mastery of the subject matter. 2 points will be awarded based on clear and concise code.

The paper must contain all findings and opinions and present them in a logical manner. Graphs, and tables are useful tools for presenting ideas. In the professional world, you will often be one of the smartest people in the room, and you will likely be the most knowledgeable on topics you present. Communication is key. Remember that not everyone has your knowledge, so present the logic behind a stated fact or opinion. Minimal code should be present in the paper – leave the code in the code files.

You must write good, well documented, code that someone else can read and follow. In practice, code you write will often be passed to others. Each week, I will attempt to run a select group of students' code. You must provide instructions on running your code and replicating your results.

In the professional world, your code will often outlive your time on a project. Your ability to write programs that can be picked up by others is a highly valued skill.

Code must be written in the C, C++, Python, R, or Julia programming languages. If a student wishes to use another language, it must be approved by the instructor first. Students can also use applications such as Excel for the creation of tables and charts for presentation. However, all analysis must be done in code.

Students are encouraged to work with their classmates on these projects. Peer feedback is crucial to finding and filling the logical gaps in your arguments and to better understanding the topics. However, all work products must be the student's own. Sharing of code libraries (outside of libraries publicly available) is not permitted.

Final Exam

Prior to the final exam, students will have compiled a library of risk management analysis routines. These routines will be the culmination of the weekly projects.

The final exam will include data and questions. Students will use their libraries to analyze the data and answer the questions. Some questions will require discussion. The more complete and easy to use the analytic library, the more time the student will have to answer discussion questions.

The final exam will be in person. The exam is open notes, but the internet may only be used to access sites for programming syntax help (StackOverFlow, etc.). Proctors will enforce this during the test. All work MUST be the student's own. Plagiarism of any kind will not be tolerated.

Office Hours

I will not have set office hours. I will be available to meet with students on an as needed basis. Please email me with questions. I will attempt to answer via email. If needed, we will set up a time to meet in person to discuss.

Our TAs may, from time to time, host in person or virtual office hours. These will be communicated through Canvas.

Previous Exams and Code Unit Tests

The class GitHub contains old finals and their solutions as well as a library of unit tests for your code libraries. These provide a good foundation for the types of questions you will be expected to solve on the final.