Columbia University
MATH GR5260 Spring 2024
Programming for Quantitative & Computational Finance

Ka Yi NG
kyn@math.columbia.edu

Course Overview

This course covers programming applications to finance with a focus of tackling real world challenges frequently seen in Capital Markets. Topics may include creating tools for yield curves calibration; valuation of financial instruments; portfolio risk measurements and their management; Limit systems for effective risk controls and compliance; and selected topics that drive latest fintech innovations. Students will learn the underlying theory, practitioner concerns, and coding techniques, software designs with hands-on experience in creating and implementation of financial technology systems and tools using a programming language.

For Spring 2024 students will be using Python for lab activities.

Prerequisite: Math GR 5010 Intro to the Math of Finance. Strong background in linear algebra, Calculus, elementary analysis (e.g. limits), probability theory and statistics, stochastic calculus (e.g. Ito's lemma) are required.

Textbooks:
There is no standard textbook for the course, recommended reference books are:
Python for Finance: Mastering Data-Driven Finance by Yves Hilpisch (ISBN: 978-1492024330)
Machine Learning in Finance: From Theory to Practice by M. Dixon, I. Halperin, P.Bilokon (ISBN: 978-3030410674)
Options, Futures and Other Derivatives, 10th Edition, by John Hull (ISBN: 978-0134472089)
Monte Carlo Methods in Financial Engineering by Paul Glasserman (ISBN: 978-0387004518)

Required Work and Grading Policy:
There will be about 4-5 homework assignments, in-class lab exercises and two written examinations.
Grading is based on the following weighting schemes:
40% homework assignments, 20% in-class exercises, 20% Exam 1, 20% Exam 2

Teaching Assistant:
Zhixin Li (Email: zl3219@columbia.edu)
Xiaofan Zheng (Email: xz3135@columbia.edu)

Office Hours:
TBA

Statement of academic integrity:
Students are expected to truthfully represent their course participation and submit their own work. Students are allowed to collaborate on assignments however, they are not allowed to copy one another's code. The code should be your own code. If students are caught copying, all participants would receive zero for that assignment. Use of online sources are allowed with proper citation.
Students should adhere to the GSAS policy on Academic integrity and Responsible Conduct of Research.

Outline (subject to change)

Each class consists of a theory session and a hands-on programming lab session. Students are required to bring their own laptops for lab activities and in-class exercises.

Week	Topics to cover
1-2	Python basics: IDE, programming concepts and features
3-4	Monte Carlo simulations, derivatives pricing
5-6	Pandas basics: time series, data handling, historical volatility
7	Value-at-Risk: Parametric, MC, historical
8	Exam 1
9	Spring break
10	Object-oriented programming and design patterns
11-15	Machine learning overview, Scikit-learn, SVM, random forests, neural networks, tensorflow (if time permits)

Topics include:

- Python basics: Python IDE, Basic programming features, Python syntax, Modules and packages, Data structures and types, List, Arrays, Functions, Error and exception handling
- Numpy basics: Numpy arrays and vectorization, Random number generators
- Data visualizations: 2D and 3D Plotting
- Pandas basics: Series, Dataframe, Input/Output operations, reading / writing / analyzing data in various formats, basic statistics, data aggregation, fetching data from web pages
- Scikit-Learn basics: data preprocessing and machine learning model training
- Object-oriented programming, Classes and objects, Members and methods, Inheritance/Polymorphism, Interface and implementation

Use case studies include:

- Valuation of financial instruments
- Monte Carlo simulations
- Portfolio risk measurements, scenario analysis
- Historical volatilities
- Value at risk
- Time series forecasting