Master of Science in Quantitative Finance

Course Code: QF 627

Course Title: Programming and Computational Finance

Instructor : Dr. Zhao Yibao

Title : Senior Lecturer of Quantitative Finance

Date Prepared : 2018

PRE-REQUISITE/CO-REQUISITE/MUTUALLY EXCLUSIVE COURSE(S)

None

COURSE AREA

(1) MSc in Quantitative Finance Core

GRADING BASIS

Graded

COURSE UNIT

1.0 CU

FIRST OFFERING TERM

Academic Year: AY2018/2019

Academic Term: Term 2

COURSE DESCRIPTION

The Python programming language has unique strengths and charm, and is increasingly considered as the *de facto* go-to language in finance industry, as it is suitable not just for quick proto typing but also library building for production environment. It is therefore important for students to familiarize themselves with modern programming techniques in Python.

The course is aimed at students with programming background and who have studied some financial models, but do not know how to put the two together. The objective is to teach students not just how to implement models in Python, but more importantly how to think in a structured way.

Writing efficient code is an integral part of a quantitative analyst's job function. It is not sufficient to merely be good in mathematical modelling and derivation – a quant must be proficient in coding to implement models as well. This is a useful skillset, as it allows quants to leverage on computation power to build toy models, test it, develop further, and eventually developing it into a production-ready implementation. In addition, computational thinking also cultivate a logical and systematic thinking process, providing a rigorous framework to tackle and solve real world problems.

This course also provides a comparison between Python and MATLAB.

LEARNING OBJECTIVES

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

- Gain aptitude in writing Python programs, and familiarity with useful Python packages like numpy, scipy, pandas etc.
- Use Python to solve financial problems, including regressions, statistical analysis, and other computational tasks

ASSESSMENT METHODS

The various key assessment components are as follows:

Attendance & Class Participation	20%
In-Class Exercise	15%
Homework Assignment	15%
Final Examination	50%

INSTRUCTIONAL METHODS AND EXPECTATIONS

Seminars
Weekly seminars providing an exposition on programming and computational finance, covering Python and MATLAB.

Attendance & Class Participation
Class Participation is a central part of the learning process for you and your classmates. When you contribute, you help others to learn. Peer evaluation is used to assess your contribution.

In-Class Exercises
Five (5) in-class exercises will help students to understand in depth those examples given in the notes. Late submission is accepted, but penalty will apply.

Homework Assignment
Five (5) homework assignments are designed to help students to have a better grasp of the course materials. Late submission is accepted, but penalty will apply.

Final Examination
There will be a 2-hour closed-book final exam.

CLASS TIMINGS

The course is taught in eight 3.5-hour sessions over seven weeks. There will be a 15-minute break after 1.5-hour in each class.

RECOMMENDED TEXT AND READINGS

A course pack will be offered for this course. The following reference books are recommended for additional reading:

1. Wes McKinney (2012) "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython"
2. Jake VanderPlas (2016) "Python Data Science Handbook". Available online at: https://github.com/jakevdp/PythonDataScienceHandbook
3. Brett Slatkin (2015) "Effective Python: 59 Specific Ways to Write Better Python"

SESSIONS PLAN

Session	Topics	Assignments/Activities
1-2	**MATLAB Basics in Comparison to Python** • Syntax (Comments, Indentation, Line Joining, etc.) • Data Types • Variables, Operators and Expressions • Assignment Statements • Flow Controls (if Statements, for Statements, etc.) • M-Files Scripts and User-Defined Functions	✓ In-Class Exercise S0102 ✓ Homework S0102
3-5	**Data Manipulation and Visualization in Python and MATLAB** • Data Importing/Exporting • Creation and Selection • Applying Functions • Handling Missing Data, Grouping, Concatenation, Merging, Reshaping, Sorting, etc. • Plotting	✓ In-Class Exercise S0305 ✓ Homework S0305
6-8	**Scientific Tools in Python and MATLAB** • Equation Solving • Optimization • Differentiation, Integration and ODE Solving • Interpolation • Linear Algebra (Matrix Operations, Eigenvalues and Eigenvectors, Singular Value Decompositions, etc.) • Regression • Statistical Tests • Random Variables and Distributions	✓ In-Class Exercises S06, S07 and S08 ✓ Homework S06, S07 and S08