NYU TANDON SCHOOL OF ENGINEERING

Finance & Risk Engineering

FRE-GY.6191
Advanced Topics in Financial Technology: Visualization Lab

Francisco Rubio, PhD
francisco.rubio@nyu.edu
Office hours: before and after class – Tuesdays 5-6pm.
Class meeting: Tuesdays, 6pm – 8:40pm

Course Overview and Goals

This course provides financial engineering students the opportunity to code a large financial visualization project in Python. The class focuses on web implementation for visualizing financial applications for common financial engineering problems like generating, displaying, and updating Monte Carlo simulations, generating, and visualizing financial time series, and creating dynamic graphs based on user inputs. The class will provide a quick introduction to HTML5 and CSS as well as intruding graphic libraries needed to provide visual object that are easily displayed in web form. In addition, students will learn to provide effective communication mechanisms between calculation and visualization layers.

Upon completion of this course, students will be able to:

- effectively visualize financial data and learn about the differences between static and dynamic data visualization techniques.
- separate financial calculations from financial data visualization layers in a way that the calculation layer can be efficiently used to quickly update charts and figures.
- fully develop an application capable of taking user requests, download financial time-series data, running financial calculations, and efficiently displaying and updating charts and figures, all coded in Python.

Course Requirements

Since the course has an important financial calculation component, it is important that students have as co-requisites (students may take FRE 6191 while taking):

- FRE 6083: Quantitative Methods in Finance
- FRE-6103: Valuation for Financial Engineering

Assignments

Students will develop and present a financial web application capable of downloading data, analyzing it, running calculation and Monte Carlo simulations, and presenting the results. Each app will solve a different financial problem to be assigned in the first week of class; possible app topics can be computing option prices, analyzing a pair-trading strategy, calculating portfolio VaR, among others. These applications will be presented during the last class. Students will be required to turn in weekly advances on this web application as homework. This way, the project will be continuously built throughout the semester.

Grading

The grade for this course will be determined according to the following formula:

Assignments/Activities	% of Final Grade
Class participation	10%
Weekly project reviews/Homework	40%
Final Project Presentation	50%

Letter Grades

Letter grades for the entire course will be assigned as follows:

Letter Grade	Points	Percent
A	4.00	Example: 92.5% and higher
A-	3.67	Example: 90.0 – 92.49%
B+	3.33	Example: 87.5% - 89.99%
B	3.00	Example: 82.5% - 87.49%
B-	2.67	Example: 80% - 82.49%
C+	2.33	Example: 77.5% - 79.99%
C	2.00	Example: 70.0% - 77.49%
F	.00	Example: 69.99% and lower

Course Schedule

Week 1: Basic principles of HTML and Web Server Definitions.

- Understanding the basics of HTML code creation.
- A brief introduction to styles.
- Tutorial on initializing a web server.
- Project Part 0: introduction to the class project.

Week 2: Why Dash? A review of financial data visualization.

- An introduction of web development with python.
- Creating dynamic HTML components.
- Creating static graphs and figures with Plotly.
- Project Part I: downloading, formatting, and presenting time-series data in a web app.

Week 3: Using callbacks to update visual outputs Part I.

- Beyond matplolib: an introduction to dynamic data visualization with Plotly.
- An introduction to server and client callbacks and subroutines.
- Project Part II: adding end users' ability to request data and update the results from Part I. The time-series would include the ability to change the time-period, i.e.: 1 day, 1 month, 1 year, etc.

Week 4: Using callbacks to update visual outputs Part II.

- An introduction to the layout components and Bootstrap for Plotly Dash
- Creating a responsive form that allows users to take different actions depending on inputs.
- Adding "memoization" to an expensive calculation.
- Project Part III: adding a calculation and Monte-Carlo simulation layer to Part II. The results from the simulation would be presented so that convergence can be graphically displayed.

Week 5: Handling Dash drawbacks.

- Implementing callbacks with multiple inputs and outputs.
- Learning to handle requests from multiple inputs that affect the same output.
- Learning to handle situations that require the same input and output.
- Learning to handle component that have not been created.
- Project Part IV: at this point the project would include the ability for end-users to enter the number of iterations in the MC simulation, and if the request is repeated, memoization should prevent it from calculating it again.

Week 6: Presenting data in a clean way.

- An introduction to the Dash DataTable.
- Learning how to style a DataTable and apply conditional formatting.
- Project Part V: adding a DataTable to the project show the MC results.

Week 7: Presentations

- Project presentation.

Course Materials

Required Textbooks & Materials

- HTML & CSS: The Good Parts: Better Ways to Build Websites That Work, ISBN-13: 978-0596157609
- Interactive Data Visualization with Python, ISBN-13: 978-1800200944
- Python Data Science Handbook, ISBN-13: 978-1491912058
- Dash's documentation, found here

Additional textbooks

- Flask Web Development: Developing Web Applications with Python 2nd Edition, ISBN-13: 978-1491991732

Resources

- **Access your course materials**: NYU Classes (nyu.edu/its/classes)
- **Databases, journal articles, and more**: Bern Dibner Library (library.nyu.edu)
  NYU Virtual Business Library (guides.nyu.edu/vbl)
- **Obtain 24/7 technology assistance:** Tandon IT Help Desk (soehelpdesk@nyu.edu, 646.997.3123)
  NYU IT Service Desk (AskIT@nyu.edu, 212-998-3333)

Policies

Academic Misconduct

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.

B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:

1. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
2. Fabrication: including but not limited to, falsifying experimental data and/or citations.
3. Plagiarism: Intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
4. Unauthorized collaboration: working together on work that was meant to be done individually.
5. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission have been received from the course instructor(s) or research adviser involved.
6. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.

Disability Disclosure Statement

Academic accommodations are available for students with disabilities. Please contact the **Moses Center for Students with Disabilities** (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

Inclusion Statement

The NYU Tandon School values an inclusive and equitable environment for all our students. I hope to foster a sense of community in this class and consider it a place where individuals of all backgrounds, beliefs, ethnicities, national origins, gender identities, sexual orientations, religious and political affiliations, and abilities will be treated with respect. It is my intent that all students' learning needs be addressed both in and out of class, and that the diversity that students bring to this class be viewed as a resource, strength and benefit. If this standard is not being upheld, please feel free to speak with me.