Design and Testing of Algorithmic Trading Systems with Python

FINTECH 533

**Syllabus**

Professor:

Jake Vestal [jacob.vestal@duke.edu](mailto:jacob.vestal@duke.edu)

Description

This course serves as an introduction to the use of the Python Dash platform to apply the most important concepts used in quantitative finance to the development and evaluation of actual trading strategies. Students will learn to build practical financial models in Python and display results in interactive, visual web apps. The course has a practical focus - how to analyze trading strategies using the types of sophisticated analytical tools in wide use today. Students will design and implement their strategies using data from the Refiniv API and the Interactive Brokers paper trading system.

**COVID-19 Statement:**

Students in this course are expected to abide by the commitments they made in signing the Duke Compact to protect the health and safety of their fellow students, faculty, staff, families and neighbors. First time, minor violations of COVID-19 conduct expectations will be met with appropriate educational responses. However, anyone who fails to comply with the expectations of the Duke Compact more than once, or who flagrantly commits a serious violation that creates a health or safety risk to others in the Duke community, will be subject to more significant consequences, beginning with loss of the privilege to attend courses in-person and/or loss of access to campus, and moving up to suspension or expulsion.”

# Learning Objectives

The overarching objective of this course is to provide students the background knowledge needed to use Python to develop, implement, communicate, and evaluate the performance of algorithmic trading strategies that may include any combination of stocks, bonds, options, futures, and currencies.

By the end of the course, students will be able to:

1. Use Python to:
   * fetch market data from both streaming and stationary sources
   * algorithmically process data into executable trades
   * automatically place trade orders on a live trade execution system
2. Elegantly communicate financial information and algorithmic strategy results in a visual, interactive Web-enabled platform using Python’s Dash framework
3. Understand the mechanisms and market behavior of stocks, bonds, options, futures, and currencies and apply that understanding to practical trading situations
4. Apply advanced analytical techniques to the development of algorithmic trading strategies
5. Use GitHub to develop software in a team environment
6. Backtest, analyze, and evaluate the performance of a trading algorithm
7. Understand and apply the Golden Rules of algorithmic trading

# Labs / Office Hours:

There is no official lab for the course.

# Grading

Final grades will be determined as follows:

1. Homeworks are projects – each one takes the form of a small webapp. Your homework grades will be averaged into a **project average** grade. There are typically 3 to 5 projects in a semester, and some may be completed in teams depending on student interest and instructor approval.
2. There will be a quiz on course material before Spring Break. A study guide will be provided to with a week’s worth of notice and and a review session will be held. The quiz isn’t there to trip you up – it’s there to reinforce concepts and check your understanding.
3. Your final grade will be calculated as 85% project grade and 15% Quiz Grade.

# Course Schedule

## Building Blocks

* Refinitiv API
* IBKR API
* Python
* Basic UI with HTML and CSS
* Dash environment & reactive programming

## Working with One Strategy

* Dealing with series of returns
* Alpha & Beta
* Efficient Frontier & Sharpe Ratio

## Trading System Design

* Limit & market orders
* Trade Blotter
* Calendar & Trade ledgers

## Generalized Linear Model

* Link function
* Response variable behavior
* Use cases, distributions & support

## Working with Several Strategies

* Portfolio Optimization
* Parameter Estimation & Asset Allocation

## Dash

* Live plots
* Buttons that do things: trade, fetch data, run analysis
* Active table outputs
* 3d Plots
* HosHosting

## Going Live (time permitting)

* Create a paper trading account at Interactive Brokers
* Connect your app to the trader to make and analyze trades.

# Team Policies and Requirements

* *Designate roles within the team.* This could include a project manager, Git manager, frontend dev, data wrangler, etc.
* *Decide on a Git workflow that works best for your team.*This will be discussed more in class, but here are a few links to get you started:
  + https://www.atlassian.com/git/tutorials/comparing-workflows
  + https://resources.github.com/videos/github-best-practices/
  + https://about.gitlab.com/2015/01/27/7-git-personalities/
* *Agree on a common meeting time and what each member should have done before the meeting by way of preparation*.
* *Do the required individual preparation*. Each team member should attempt to produce their assigned deliverables and written work.
* *Submit deliverables*: All deliverables should be submitted by the deadline. Project advisors may specify additional means of submittal (e.g. hard copy, email, Dropbox, Google doc, etc.)
* *Review returned deliverables*. Make sure everyone understands why points were lost and how to correct errors.
* *Complete and submit on-line peer ratings for all team members when required*.
* *Be willing to address conflict if necessary*. Consult with the instructor if a conflict arises that can't be worked through by the team. Advisors and instructors do not like to be surprised. Bad news does not get better when withheld.

Omitting names from completed assignments, firing, and quitting  
If a team member refuses to cooperate on an assignment, his/her name should not be included on the completed work. If the non-cooperation continues, the team should meet with the course instructor and attempt to resolve the problem. If no resolution is achieved, the cooperating team members may notify the uncooperative member in writing that he/she is in danger of being fired, sending a copy of the memo to the course instructor. If there is no subsequent improvement, they should notify the individual in writing (cc instructor) that he/she is no longer part of the team.

Similarly, a student who is consistently doing most of the work on a team may issue a warning memo (copy to instructor) that he/she will quit unless more cooperation is forthcoming. The team should meet with the course instructor at this point so that the problem can be resolved, if possible. If the non-cooperation continues, the student doing the work may notify the others in writing (copy to instructor) that he/she is no longer part of the team.

*Students who are fired or who quit must find another design team unanimously willing to accept them as a member, otherwise they complete a project by themselves. This is not a desirable option and should only occur as a last resort.*

# Group Work Credit and Peer Reviews

Deliverables are expected to be the product of team efforts. There is opportunity, however, for team members to identify individual members who have made an outstanding contribution or who have contributed in a less-than-equal manner, including non-participation, to the phase. This will be accomplished in two ways:

1. The names on each team assignment should reflect those of group members who contributed to the report. If a group member fails to contribute or cooperate in the development of the report, his or her name should not be included. Group members who are not listed as contributing to a report or presentation will receive a grade of zero.
2. Group members will complete a very brief peer rating form on Sakai four times during the semester when the four team deliverables are turned in. This practice ensures that any mismatched opinions regarding individual contributions to the project are caught early, and credit is fairly assigned. Peer evaluations will be considered in assigning the final grade for each deliverable. Students who choose not to complete the peer evaluation will be penalized with a deduction of 0.1 from the peer weighting factor for that deliverable.
3. The instructor(s) will provide feedback via email to each student regarding team performance as appropriate.

# GitHub Permissions & Policies

Your project repositories are initially set to 'public' access by the instructor, meaning that anyone who wants to view & download the project will be able to. Your team may decide to restrict or grant viewing & download access to whatever audience it wants, so long as the instructor, the TA, and other class members are always able to view & clone the Master branch.

# This syllabus is a work in progress

While I will try my best to stick to this syllabus, the course is constantly evolving and everything in this document should be considered tentative.