New York University Tandon School of Engineering
Department of Finance and Risk Engineering
Course syllabus FRE 7241 **Algorithmic Portfolio Management**
Spring 2019
Professor Jerzy Pawlowski
Tuesdays at 6PM; Rogers Hall, Room #227

Contact information: jp3900@nyu.edu
Phone: 201-936-9026
Office hours: TBA

Course Description:

The course will apply the R programming language to algorithmic portfolio management. The course will also apply machine learning techniques, such as backtesting (cross-validation), dimensionality reduction, and parameter regularization (shrinkage).

Course Objectives:

Students will learn to fit ARIMA and GARCH time series models, and to apply them to forecasting. They will learn to optimize portfolios under different constraints and risk-return objectives. Students will learn to build trading strategies using autoregressive and momentum models, and to simulate their out-of-sample performance using backtesting (cross-validation). They will learn to improve their performance by applying dimensionality reduction and shrinkage techniques. Students will implement real-time trading strategies via the API of Interactive Brokers. They will learn to present their results using interactive plots. They will also learn to download data, to input and output data from R, and to scrub and format the data.

Course Structure

The course will consist of lectures, homework assignments, and in-class tests. There will be no final exam or project. The assignments will consist of extensive coding exercises, designed for practical applications.

Readings

The required readings will be the course slides and other texts uploaded to NYU Classes. There will be no required textbook, but a recommended textbook is:
David Ruppert, Statistics and Data Analysis for Financial Engineering (Springer Texts in Statistics) ISBN-10: 1461427495 & ISBN-13: 978-1461427490
link: Ruppert_Statistics_and_Data_Analysis
Also recommended are:
Grant Farnsworth, Econometrics in R
Norman Matloff, The Art of R Programming

Other Required Course Materials

Students will be required to install on their laptop computers the R interpreter and the RStudio integrated development environment (IDE), and to become proficient with the R

Studio IDE. Students will be required to bring their laptop computers and run R during all the lectures.
To download the R Interpreter: http://cran.us.r-project.org
To download the RStudio Development Environment: http://www.rstudio.com/ide

Course Requirements

Students will be required to study the course slides and other texts uploaded to NYU Classes. Students will also be required to run and analyze all the R code contained in the course slides.

Course Pre-requisites

FRE6123 Financial Risk Management and Asset Pricing, and graduate standing. The R language is considered to be challenging, so this course requires some programming experience with other languages such as C++ or Python. Students should also have knowledge of basic statistics (random variables, statistical estimators, hypothesis testing, linear regression, etc.)

Grading

Grading will be based on homework assignments and in-class tests, in which students will be required to write extensive R code. There will be no final exam or project. Each homework and test will be graded and assigned a numerical score, based on its difficulty and on the correctness of the solution. The final course letter grade will be derived from the cumulative numerical scores obtained for all the homeworks and tests.

Lecture topics

Lecture #1:

- Simulating geometric Brownian motion.
- Time evolution of stock prices.
- The log-normal probability distribution of stock prices.
- Time evolution of random sub-portfolios.
- Crossover strategies using moving average technical indicators.
- Trend-following and mean-reverting (contrarian) strategies.
- Optimal parameters of crossover strategies.
- Ensembles of crossover strategies.
- Creating interactive plots using packages dygraphs and shiny.
- Performing rolling aggregations over time series.
- Backtesting (cross-validation) of out-of-sample strategy performance.
- Momentum strategies for ETF and stock portfolios.
- Backtesting momentum strategies and momentum crashes.

Lecture #2:

- Eigenvectors and eigenvalues of matrices.
- Singular value decomposition (SVD).
- Regularized inverse of matrices.
- Formula objects and regression analysis.
- Regression goodness of fit: t-values and p-values, R-squared, and F-statistic.

- Regression diagnostics: Q-Q plots and the Durbin-Watson test.
- Predictions from linear regression and their confidence intervals.
- The autocorrelation function and the Ljung-Box test.
- ARIMA and Vector Autoregressive (VAR) time series models.
- Stationary processes and their characteristic equations.
- Integrated and unit-root processes.
- The Augmented Dickey-Fuller (ADF) test for unit roots.
- Partial autocorrelations.
- Calibrating ARIMA models and order selection using the Akaike and Bayesian information criteria.
- The Yule-Walker equations.
- Time series forecasting using ARIMA and VAR models.
- Backtesting ARIMA and VAR forecasting models, and their mean squared errors (MSE).
- Overfitting and parameter regularization (shrinkage).
- Meta-parameter optimization and the bias-variance tradeoff.

Lecture #3:

- Compiling C++ programs into R functions using package Rcpp.
- Fast matrix algebra in C++ using package RcppArmadillo.
- Strategy backtesting in C++ using packages Rcpp and RcppArmadillo.
- Estimating and modeling volatility.
- Range volatility estimators of OHLC time series.
- Simulating the Ornstein-Uhlenbeck process.
- GARCH volatility models.
- Multi-dimensional optimization.
- Calibrating GARCH models using the maximum-likelihood method.
- Volatility forecasting.
- The efficient frontier and the Capital Market Line.
- Capital Asset Pricing Model (CAPM): the market portfolio, the Security Market Line.
- Performing rolling regressions over time series using package Rcpp.
- Calculating rolling stock betas using the Kalman filter.
- Risk-adjusted performance measures: Sharpe, Calmar, and Sortino ratios.
- Beta-adjusted performance measures: Treynor ratio, Jensen's alpha, information ratio.
- Tail risk measures: Value at Risk (VaR) and Conditional Value at Risk (CVaR).

Lecture #4:

- Global portfolio optimization using package DEoptim.
- Portfolio optimization with weight constraints.
- Maximum return portfolio using linear programming.
- Minimum variance and maximum Sharpe ratio portfolios.
- Mean-variance portfolio optimization using the package quadprog for quadratic programming.
- Backtesting out-of-sample performance of optimized portfolios.
- Constrained portfolio optimization using coefficient shrinkage.

- Correlation matrix estimation and Cholesky decomposition.
- Principal Component Analysis (PCA) and factor models.
- Dimensionality reduction using PCA.
- The Hurst exponent and the variance ratio test.

Lecture #5:

- Principal Component Analysis (PCA) of stock, bond, and currency portfolios.
- The Engle-Granger two-step cointegration procedure.
- Granger causality.
- Pairs trading and statistical arbitrage.
- Financial and commodity futures contracts.
- Chaining together futures prices.
- VIX futures contracts.
- Contango and backwardation of VIX futures curve.
- VIX futures investing.
- High frequency and intraday time series data.
- Trade and Quote (TAQ) data.
- Communicating with Interactive Brokers via its API using package IBrokers.
- Downloading data from Interactive Brokers.
- Live event processing using event wrapper and callback functions.
- Placing market trade orders via the API of Interactive Brokers.
- Placing and cancelling limit orders in a programmatic callback loop.
- Executing real-time trading strategies via the API of Interactive Brokers.

Lecture #6:

- Static asset allocation strategies: stocks and bonds, all weather portfolios.
- Rebalancing strategies between stocks and bonds: constant dollar allocations, risk parity.
- Equal-weighted and cap-weighted stock indices, cap-weighted indices as momentum strategies.
- Portfolio objectives: maximum Sharpe, minimum correlation, minimum variance (or CVaR), low beta.
- Active asset allocation strategies.
- Measuring market timing skill: Merton-Henriksson and Treynor-Mazuy tests.
- Measuring portfolio selection skill using random portfolios.
- Factor investing and smart beta portfolios.

Lecture #7:

- Date and time objects: the POSIX date format and time zones.
- Time series objects using package xts: downloading, reading, scrubbing, plotting, saving.
- Package quantmod for quantitative financial modeling.
- Downloading financial data from the internet: Wharton WRDS, Yahoo Finance, Quandl, FRED Federal Reserve.
- Creating an R package on GitHub, containing C++ code with Rcpp.
- Optimizing R code for speed and memory usage.

Moses Center Statement of Disability

If you are student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities (CSD) at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.

NYU School of Engineering Policies and Procedures on 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 has 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.