New York University Tandon School of Engineering
Department of Finance and Risk Engineering
Course syllabus FRE 6871 **R in Finance**
Spring 2019
Professor Jerzy Pawlowski
Mondays at 6PM; Rogers Hall, Room #227

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

Course Description:

The course will introduce the R programming language and its applications to finance. The course will emphasize the applications of machine learning techniques to forecasting.

Course Objectives:

Students will learn to build and calibrate various models for regression, classification, and clustering analysis. They will learn to apply these models to out-of-sample forecasting. Students will learn to apply Monte Carlo simulation to evaluate risk and return characteristics of financial assets. They will learn to present their results using interactive plots. They will also learn essential finance related tasks such as data downloading, input and output, scrubbing, and pivoting

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: Norman Matloff, The Art of R Programming (link)

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

There are no formal course pre-requisites. But 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:

- The R workspace and environment variables.
- R data structures: character strings, vectors and matrices, lists and data frames.
- R object classes, attributes, and object coercion.
- Subsetting, filtering, binding, and sorting operations on vectors, matrices, lists, and data frames.
- Logical operators, control structures, and data validation.
- Iteration and loops.
- Vectorized functions for vector and matrix computations.
- Calculating and plotting probability distributions: Normal, Chi-squared, and Student's t-distribution.
- Hypothesis testing and statistical tests: Shapiro-Wilk, Jarque-Bera, Ljung-Box.
- Creating interactive plots using Shiny apps.

Lecture #2:

- Defining new functions, binding function arguments, and the dots function argument.
- Higher order functions, functionals, and anonymous functions.
- Performing loops using the apply functionals.
- Generating pseudo-random numbers.
- Calculating the standard errors of statistical estimators using bootstrap simulation.
- Monte Carlo simulation using parallel computing with package parallel.
- Simulating the first passage time of Brownian Motion.
- Monte Carlo variance reduction techniques, including antithetic sampling and importance sampling.
- Bootstrapping from empirical distributions.

Lecture #3:

- Credit portfolio defaults under the Vasicek model.
- Calculating credit portfolio Value at Risk (VAR) and Conditional Value at Risk (CVAR).
- Calculating the standard errors of VAR and CVAR using bootstrap simulation.
- Simulating correlated defaults using Cholesky decomposition.

- Collateralized debt obligations (CDOs).
- Simulating CDO tranche losses.

Lecture #4:

- Eigenvectors and eigenvalues of matrices.
- Singular value decomposition (SVD).
- Regularized inverse of matrices.
- Principal component analysis (PCA) of the yield curve.
- Dimensionality reduction using PCA.

Lecture #5:

- 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.
- Principal component regression (PCR).
- Out-of-sample forecasting using regression models with shrinkage.
- Logistic regression for classification and credit scoring models.
- Type I and Type II errors in hypothesis tests and the ROC curve.

Lecture #6:

- The split-apply-combine procedure for data aggregation.
- Debugging functions and validating arguments.
- Handling warnings, errors, and exceptions.
- Exception handling in loops.

Lecture #7:

- Data input and output: reading and writing to files
- Scrubbing bad and missing data.
- Package quantmod for quantitative financial modeling.
- Downloading financial data from the internet: Wharton WRDS, Yahoo Finance, Quandl, FRED Federal Reserve website.
- Combining R with Excel.
- Running R code from Excel spreadsheets.
- The S3 object-oriented programming system, generic functions and their methods.
- Loading and exploring R packages, the R search path and package namespaces.

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.