

Spring 2025

FRE-GY 6883 Financial Computing

Course Final Project

General Requirement

You are required to complete class projects in teams of approximately five members (Tuesday Evening section: 5-6 people per team, Saturday Afternoon section: 4-5 people per team). We will have a total of 5 teams in our class. You should elect one member of your team to be the team leader. Teams, once formed, cannot be changed midway through the project. The team leader is responsible for facilitating the project planning, and the entire team will plan the project under the team leader's guidance. Planning involves identifying what needs to be done (tasks), who should do it (resources), when tasks should be completed (time frames), and how tasks are best sequenced (dependencies).

The team leader is required to provide a brief weekly update of your team's progress.

Each team will submit PowerPoint slides and all project files, including source code and executables, in a tar/zip format to our course website three days before the presentation day (May 7, 2024, 11:55 pm for the Saturday session, and May 10, 2025, 11:55 pm for the Tuesday session). The PowerPoint presentation should include your research on Russell 3000 stocks, focusing on their earnings, a diagram of the project design (UML is preferred), class declarations and data structures, and program outputs. All the teams are requested to present and demonstrate their projects. Team members will be quizzed with written questions based on the team's project source code. Each team can resubmit their presentation once on the designated day if necessary. Your project will be evaluated based on its program efficiency, complexity, and the success of your demonstration and presentation. The team leader will make all the submissions.

Project Description

Evaluate the impact of quarterly earnings reports on stock price movement

Programming Requirements:

- Use **liburl** to retrieve historical price data from eodhistoricaldata.com: A function retrieves the adjusted close prices for selected **Russell 3000 stocks** and **IWV** (Russell 3000 ETF used as market benchmark) into memory.
- Create a set of classes, such as a class for stock to handle EPS (earnings per share) estimate and price information.

- Use member functions or independent functions for all calculations. Overload a few arithmetic operators for vector/matrix.
- The stocks and their corresponding price information for each group should be stored in an STL map, with stock symbols as the keys.
- The expected AAR, AAR STD, and expected CAAR and CAAR STD for 3 groups are presented in a matrix. The row of the matrix is the group#, and matrix columns are for AAR, AAR-STD, CAAR, CAAR-STD
- Use gnuplot to show the CAAR from all three groups in one graph.
- Your program should be able to:
 - Retrieve historical price data for all selected stocks. Parse the retrieved data for dates and adjusted closing prices.
 - Calculate AAR, AAR-STD, CAAR, CAAR-STD for each group
 - Populate the stock maps and AAR/CAAR matrix.
 - Show the gnuplot graph with CAAR for all three groups.
- Your program should have a menu of 5 options:
 - Enter N to retrieve $2N+1$ days of historical price data for all stocks (you need to validate user input to make sure $N \geq 30$ and ≤ 60).
 - Pull information for one stock from one group:
 - Daily Prices
 - Cumulative Daily Returns
 - The group the stock belongs to
 - Earning Announcement Date, Period Ending, Estimated, Reported Earnings, Surprise and Surprise %.
 - Show AAR, AAR-STD, CAAR and CAAR-STD for one group.
 - Show the gnuplot graph with CAAR for all three groups.
 - Exit your program.

Calculation Details:

1. Based on the 4th quarter of 2024 earnings announcement for Russell 3000 stocks (See the Earnings Announcements sheet), sort all the surprise% in ascending order, and split all the stocks into three groups with **relatively equivalent numbers** of stocks:
 - i. Highest surprise group: Beat Estimate Group
 - ii. Lowest surprise group: Miss Estimate Group
 - iii. The rest of the stocks in between the Meet Estimate Group
2. Define day “zero” for a stock as the day the earning is announced.
3. Implement Bootstrapping:
 - a. Randomly selecting 30 stocks from each group, a total of 90 stocks.
 - b. Use libcurl lib to retrieve $2N+1$ days of historical prices for Russell 3000 stocks and ETF ticker IWV (used as market benchmark) around the date of

earning release (You could enhance our class example for this purpose). N is an integer that must be greater or equal to 30 and will be entered by users. Users will be warned if there are not enough historical prices for $2N+1$.

- c. For each stock, calculate the daily returns R_{it} for $N-1$ days before the day “zero” and N days after, such as $t = -60, -59, \dots, -1, 0, 1, \dots, 59, 60$:

$$R_{it} = \log(\text{Price}_t / \text{Price}_{t-1})$$

Using **adjusted daily closing price** for your calculation

- d. Calculate the corresponding daily return R_{mt} for IWV for the same days.
- e. Define abnormal returns as the difference $AR_{it} = R_{it} - R_{mt}$.
- f. Calculate average daily abnormal returns for each group of stocks (with M stocks, $M = 30$ in our case) for all $2N$ reference days:

$$AAR_t = \frac{1}{M} \sum_{i=1}^M AR_{it}$$

- g. Cumulate the returns on the first T days to CAAR:

$$CAAR = \sum_{t=-N+1}^T AAR_t, T \text{ could be } -N+1, -N+2, \dots, N-2, N-1, N$$

- h. Repeat steps a to g 40 times to create 40 samplings and then calculate:
- Average AAR and CAAR for 40 samplings for each group
 - Standard Deviation of AAR and CAAR for each group
4. Generate a gnuplot chart showing the average CAAR of all three groups and discuss the impact the earning releases have on their stock prices. Is there any conclusion you could draw from your project?

GitHub Classroom Final Project Group Assignment Invitation. Please join only ONE Group according to your team number on the team project Excel sheet:

Tuesday Session:

Saturday Session:

Project Tasks:

Task 1: Earnings research: sort stocks from Russell 3000 into 3 groups based on their earnings and EPS Estimate based on Zacks.

Task 2: Project Design:

- a) Create classes and data structures such as vectors, matrices, and maps.
- b) Figure out how to handle historical price retrieval from eodhistoricaldata.com for all IWW 3000 stocks and the benchmark IWW, and parse the retrieved data.
- c) Figure out how to implement your Bootstrap algorithm.
- d) Write member functions or independent functions (with operator overloading) for all the calculations.
- e) Design and implement a menu.

Task 3: Figure out how to implement menu options and graph your results with gnuplot.

Task 4: Divide the project into modules and assign team members to work on each module.

Task 5: Module Integration and Testing

Task 6: Presentation Preparation.