SSMIF Quant Coding Challenge
Spring 2025

Contents

0 Introduction

If you are receiving this document, it means that you have successfully qualified for the second round of applications for the Quant side of the Stevens Student Managed Investment Fund (SSMIF). Below, you will find information regarding the questions that will help you understand the format of the challenge, as well as transparent guidelines to ensure your submissions meet our expectations. We wish you all the best and hope to review your submissions soon!

0.1 Guidelines

The structure of the coding challenge will involve 1 mandatory question, which all applicants are required to answer. In addition to this question, applicants will be required to answer 1 other question of their choice. However, if you choose to do the development challenge (Quant Dev & Optimization Coding Challenge), you will *not* need to submit the mandatory question. The deadline for this challenge is **Friday, February 21st, 2025 at 11:59 PM** and must be submitted through the following google form: `https://forms.gle/RHJzrUPQutphFUZS9`.

There are 4 team-specific challenges to choose from. While the optional challenges are of slightly varying difficulty, the challenge you choose to complete will increase your likelihood of being considered for that specific team, among other applicants that have submitted that specific challenge. You may also complete and submit multiple of the optional challenges, which will reflect accordingly when considering your application for the appropriate teams. Any available materials for these challenges (data, submission templates, etc.) will be provided in the zip file attached to the email.

Where feasible, ChatGPT may be used in your code - however, any lines of code written by ChatGPT must be cited through a 1-line comment above it. The same applies to all other sources, online or not, including forums like Stack Overflow, and sites similar to it (the only exception is documentation relevant to a function/module used in your code). Collaborating with other applicants or non-applicants is forbidden and will only decrease the chances of your application moving to the next round.

Please read each section carefully to ensure that you understand all instructions and available tools pertaining to the challenge. All challenges have notes clarifying possible discrepancies. Any generally vague statements are intentionally worded to encourage creativity. However, you may forward any additional questions regarding nuances in the challenge to Head of Quant, Shyam Parikh, at `sparikh10@stevens.edu`. All submissions must strictly follow the requirements listed on this google form for your submission to be considered. Bonus points go to typing your code.

1 Question 1 - General Question

1.1 Background

One of the key qualities of a strong quantitative analyst is the ability to quickly devise and implement trading strategies. This assignment is designed to help you develop skills in working with financial data and identifying profitable opportunities through analysis. This problem will be a great example for you to learn how to analyze data and find specific trends within your own analysis.

You will be provided with historical price data for four individual securities and an ETF that reflects the performance of these securities. Your objective is to identify potential arbitrage opportunities between the ETF and its underlying assets.

1.2 Part 1: Data Cleaning

In real-world finance, large datasets often contain missing, inconsistent, or corrupted values. Your first task is to clean the dataset effectively, ensuring it is accurate and usable for analysis. Be prepared to explain your approach and justify the methods you used to clean the data.

1.3 Part 2: Strategy Development

Once the dataset is prepared, you will analyze the relationship between the ETF and its underlying securities to identify profitable trading opportunities. Develop and implement a trading strategy that looks to maximize return while minimizing risk.

You will also visualize your trades by plotting:

- **Asset prices** to visually show the price you are entering and leaving at.
- **Blue arrows** to indicate entry points (when you open a position).
- **Red arrows** to indicate exit points (when you close a position).

1.4 Part 3: Summary & Explanation

The ability to articulate your thought process and methodology is a critical skill for quantitative analysts. In this section, you will provide a clear explanation of:

- The trading strategy you implemented.
- The logic and mathematics behind your approach.
- How you generate profit on your strategy.

Now that you understand the scope of the assignment, you will find further details in the provided *first_last_trading_strat.ipynb* file. Download, edit, and complete the notebook, then submit your final work via the designated Google Form.

2 Question 2 - Development Challenge

The SSMIF Quant side aims to be a systematic quant division that utilizes modern framework and technology to run our models and generate trade signals. The Quant Development and Optimization team ("Dev" team) is a critical part of SSMIF Quant that glues all the other teams together via our end-to-end financial data forecast pipeline, as well as our web interface.

The coding challenge to join SSMIF Quant Dev will be to build a full-stack application, utilizing a time-series database to provide insight in a model portfolio to analysts.

You will be given a draft portfolio of a quant firm that trades quarterly for the past 10 years with over 50 tickers. Your application should calculate historical portfolio value based on data pulled from Yahoo Finance, performance compared to our benchmark (the S&P 500), and the portfolio sector breakdown over time. For the sake of calculating performance, you can assume purchases are done on the date the statement came out, so on the first day, we have bought every stock on that day.

If you are selected to move forward in the application process, you will be expected to explain the challenges that you have encountered while developing this application. Your code should be well documented and maintainable, following the common conventions of your programming language and framework of choice. Lastly, your work will be submitted in a git repo compressed in a zip archive.

For any questions, please do not hesitate to reach out to `tzhu22@stevens.edu`.

2.1 Requirements

- A full-stack application
- A table showing the current holdings (ticker, quantity, day and total change, market value, unit cost, total cost
- A table for all trades
- A line graph of portfolio sector breakdown over time
- A graph of portfolio holding value over time
- A graph of portfolio performance over time against the S&P 500
- A time-series database to support the above graphs (ClickHouse, InfluxDB, etc...)
- Yahoo Finance ingestion component in the backend to take in ne wstock ticker data every day when the market closes (running in the backend)
- Documentation on how to deploy the code in the README file of the repo

2.2 Optional Requirements

It is deeply encouraged to select the most options possible.

- Graph of portfolio Sharpe ratio over time (you are highly encouraged to do this)
- Use `https://github.com/robertmartin8/PyPortfolioOpt` to optimize the current portfolio with the max_sharpe strategy and generate trade signals to get us to the optimized weights (the trade signals can just be displayed in the frontend) (exclude currencies)
- Docker-compose, Dockerfile

- AuthN & AuthZ
- Packaged with nix or k8s deployment
- ci containing linting, integration tests, etc..
- Hosted version accessible
- Handling of non-equity assets listed in the holdings, pay special care of multi-sector ETFs (breaking down a multi-sector ETF by their holdings to figure out the exact sector distribution of that ETF)

3 Question 3 - Multi-factor Model

3.1 Background

The Factor Model team aims to predict market behavior, specifically across 11 market sectors, to optimize funding allocations. This challenge will demonstrate your ability to research, code, and explain quantitative models.

Multiple teams on SSMIF Quant work on researching, developing, and implementing multi-factor models to forecast the returns of multiple asset classes such as equities, sector indexes, and fixed-income ETFs.

A multi-factor model is defined as:

$$\rho^{\text{fwd}} = \alpha + \beta_1 F_1 + \cdots + \beta_n F_n$$

where:

- ρ^{fwd} is the predicted forward-looking return for the asset.
- $\{F_1, \ldots, F_n\}$ is the list of factors.
- $\{\beta_i, \ldots, \beta_n\}$ is the list of the asset's sensitivity to the factors.
- α is the excess return not explained by factors.

This generates the forward-looking expected returns for the asset, which we can then use to optimize the weightings for each asset in our portfolio, with the goal of maximizing return and minimizing volatility.

In the CSV file, you have access to the past 20 years of the following daily data for 4 stocks (AAPL, NVDA, JPM, GS):

- Stock Price
- Historical Market Cap
- Book Value per Share
- Total Equity
- Total Assets
- P/E Ratio
- Operating Income
- Market (S&P 500) Price
- 3-Month Treasury Bill

Complete the the *first_last_fm.ipynb* template attached. More detailed instructions are in the the template.

Direct any questions about this challenge to Jason Bhalla at `jbhalla@stevens.edu`.

3.2 Part 1: Capital Asset Pricing Model

The Capital Asset Pricing Model is a standard one-factor model that generates expected forward-looking returns for an asset.

In the template file, implement the Capital Asset Pricing Model to generate a list of each stock's expected forward-looking return.

3.2.1 Part 1A: Data Preparation

Import the data from the *multifactor_security_data* file.

3.2.2 Part 1B: CAPM Model

Construct a multi-factor model, using at least 3 factors, to predict the expected forward-looking returns of each stock.

3.2.3 Part 1C: Backtesting

Test your expected returns against the actual returns.

3.3 Part 2: Multi-factor Model

Your second task is to build an equity-specific multi-factor model with the data.

You can implement any multi-factor model that you want, as long as it *utilizes at least 3 factors*. Some of the data isn't perfect - that's intentional! You can either research a multi-factor model and implement it, or you can create some combination of factors to create your own new model. As illustrated in the multi-factor model equation, your final goal is to have a list of the expected forward-looking returns for each of the 4 stocks.

3.3.1 Part 1A: Data Preparation

Import the data from the *multifactor_security_data* file.

3.3.2 Part 1B: Multi-factor Model

Construct a multi-factor model, using at least 3 factors, to predict the expected forward-looking returns of each stock.

3.3.3 Part 1C: Backtesting

Test your expected returns against the actual returns.

3.4 Part 3: Comparison

Compare the Capital Asset Pricing Model with your multi-factor model. Which performs better? Why?

3.5 Submission

Submit the provided Jupyter notebook, *first_last_mfm.ipynb*. Ensure your code is well-documented and easy to understand.

4 Question 4 - Fixed Income Factor Model Challenge

4.1 Background

The Multi-Factor Fixed Income Strategies Team is dedicated to optimize its fixed income allocations by accurately predicting the returns and volatility of US Treasuries using interest rate models. The primary goal of fixed income is to provide a hedge for our equity holdings through utilizing mathematical and statistical modeling techniques to value bonds, manage risk, and work with interest rates. This challenge will assess your ability to work with bond data and thoughtfully implement different techniques and models to forecast future measures.

4.2 Part 1: Bootstrapping

In the first part, your goal is to utilize bootstrapping to obtain the spot rate for each bond at each point in time. You will be using the data 'yield_data.csv' which contains the coupon rates and prices for each bond. The spot rate (also known as the zero rate) is the rate of return that the investor earns without collecting coupon payments. Bootstrapping is a method used for obtaining spot rates for fixed-income instruments that pay coupons. Using these spot rates, you are to generate a spot rate curve for each year (so 11 total spot rate curves). Note down any insights you observe.

Please write the following functions:

- **bootstrapped_rates**: For every bond, calculate the zero rate using the bootstrapping method. For example, you will have to get the spot rate for the 1 year from 2015, the 2 year from 2015, etc until and including the bonds from the last year (2025). You can choose how you want to return these rates.
- **plot_spot**: For each year, plot out the rates you obtained from bootstrapped_rates. You will have to do this for each year (all bonds from 2015, 2016, etc), so you should have 11 graphs total.

4.3 Part 2: Stochastic Interest Rate Model

In the second part of the challenge, you will use a stochastic interest rate model of your choice to forecast short-term rate evolution as well as price impacts on a hypothetical universe of bonds. You will also explore bond duration and convexity as risk measures and alternate price forecasting tools.

You will use the current SOFR rate as your starting point. You can use any interest rate model you like. If you're unsure of where to start, we recommend either of the Ho-Lee or CIR models. Resources for these models are provided below. For parameters, choose whatever you think is reasonable. Make sure you understand the model you're implementing!

Please write the following functions:

- **plot_rate_paths**: Use your interest rate model to forecast and plot 100 sample short-term (overnight) rate paths over the next 4 years.
- **plot_yield_curve**: A key characteristic of fixed income is the extent to which rates affect each other. To explore this, use your interest rate model to generate ONE overnight rate path. Use this overnight rate path to forecast the evolution of longer-term rates, specifically the 3-month, 6-month, 1 year and 2 year rates. How will changes in the overnight rate impact these rates? Include comments or a text box explaining and justifying your approach.
- **plot_bond_prices**: Using the yield curve generated above, forecast the price evolution of the following bonds over a 2-year period:

  - 3-month zero-coupon bond
  - 6-month zero-coupon bond
  - 1-year zero-coupon bond
  - BONUS: 2-year coupon bond paying 1% semiannually

- **calculate_duration**: A function that calculates the duration of a bond given its time to maturity, yield and semiannual coupon rate
- **calculate_convexity**: A function that calculates the convexity of a bond given its time to maturity, yield and semiannual coupon rate

BONUS: Calculate the duration and convexity of a 2-year bond paying 1% coupons semiannually. Calculate the bond's NPV using the first entry in your 2-year rate path. Using the rest of the rate path, estimate the price of the bond over time using its duration and convexity to relate price changes to rate changes. Do these results match with your original price forecast? If so, why might they differ? Comment on your results.

Resources:

- CIR: `https://www.finance-tutoring.fr/the-cox-ingersoll-ross-(cir)-model-simply-explained/`
- Ho-Lee: `https://www.bensblog.tech/fixed_income/HoLee_Model/`

4.4 Submission

Submit a copy of the template provided in the format *first_last_fi.ipynb*.

4.5 Additional Comments

Remember to read all of the instructions carefully and to comment your code. Cite all code that was not written by you, including if you got it from external sources like ChatGPT.

Good luck!

5 Question 5 - Risk Challenge

5.1 Background

Risk management analysts are skilled in understanding, using, and applying various statistical metrics to assess portfolio performance and strength. Risk metrics are crucial in managing a portfolio as they provide a clear view of the potential downside of investments and enable more informed decision-making. Examples of widely utilized metrics across the industry include Beta, Maximum Drawdown, 95% Value at Risk (VaR), Volatility, Sharpe Ratio, and many others.

5.2 Challenge Breakdown

In this challenge, you will monitor the risk metrics of a portfolio and use one or more of these metrics to establish a risk tolerance for the portfolio. The code provided gives the infrastructure to build a portfolio containing multiple stocks. The returns of the portfolio are not the primary focus of this challenge; instead, we are interested in your ability to mitigate the risk of the portfolio at levels you deem significant while still allowing for potential gains.

5.3 Part 1: Risk Metrics

In this part, you will be implementing the calculate_metrics() method which will calculate the 95% VaR, Volatility, Max Drawdown (choose your horizon), and Sharpe Ratio for each trading day. In addition, you will create your own unique statistical measurement that you believe captures risk efficiently. This risk metric can be based on preexisting metrics, or it can be entirely brand new. Comment on your code/functions explaining why this metric is an effective and usable risk metric. You may edit any prewritten functions as needed for this. These metrics (including your newly designed one) should be updated in each data frame in the dictionary (ticker_holdings) and updated in the portfolio holdings with every iterating trading day. This structure will enable you to use the risk metrics to mitigate risk in Part 2.

5.4 Part 2: Risk Tolerance

In Part 2, you will use the risk metrics calculated in Part 1 to mitigate the risk of your portfolio. The objective is to determine weights for each stock so that the portfolio remains within your specified level of risk (e.g., maintaining a Sharpe Ratio above 1.5 and a 30-day Volatility below 3% at any given time). You may use one or multiple risk metrics to optimize the portfolio. You will implement your strategy under the rebalance_portfolio() method, using the designated buy() and sell() methods within the given Portfolio class. Since we are holding only long positions and not trading options, risk management will exclusively come from adjusting the portfolio weights. Note: use an initial balance of $100,000.

5.5 Part 3: Data Visualization

For the final part of this challenge, implement the last method to display data relevant to the portfolio. Plot the value of your risk-metric of choice over time along with your portfolio returns, as well as the values of your portfolio's risk metrics over the investment duration. Along with the portfolio's risk metrics, plot the corresponding metrics for a portfolio composed of a market index, such as the S&P 500.

5.6 Submission

Please submit your solution in a python file and name your file as *first_last_risk.py* (ex. *john_doe_risk.py)*

6 BONUS: Question 6 - Mental Math Challenge

One of the most fun and challenging parts of pursuing a career in quantitative finance is the mental math that comes with it. As a bonus problem, we ask that you complete the test on standard settings and submit your score in the form of a screenshot. Quants tend to consistently get around 50 and above. *This is based on the honor system. You should be prepared to do this live and demonstrate the same ability.*

The link to the game can be found here: `https://arithmetic.zetamac.com/`