Asset Allocation Quantitative Researcher

Problem #1: What is the half-life of the covariance estimate?

Many of our portfolios are built to target a very specific risk budget and annualized standard deviation. The attached Excel file (worksheet "Problem #1"") provides you with 2 years of daily data. You will use all this data to estimate monthly covariance using the methodology described in the attached PDF by Litterman and Winkelmann, allowing for 2 days of serial correlation in the data and decaying the data at 1% per day. For the avoidance of doubt, let's assume 21 business days in a month. Skim the PDF, focusing on covariance calculation.

Using MATLAB or Python, build a portfolio with an expected volatility of 10% per year that has exactly the risk allocation shown row 1 of the Excel file. For our purposes, an asset's risk contribution is measured by its covariance contribution, which is the asset's weight in the portfolio times the row of the covariance matrix corresponding with the asset times the column vector of all the asset weights in the portfolio. You will see that if you sum risk contributions calculated in this way, you will arrive at total portfolio variance. To reiterate, asset i's covariance contribution is equal to:

$$\Sigma_i = w_i \cdot \Omega_i \cdot w$$

where:

- w_i is the weight of asset i,
- Ω_i is the i_{th} row of the covariance of returns, and
- w is the column vector of portfolio weights.

There is no requirement that weights sum to 1.

Deliverable:

- an Excel file
    - your estimate of monthly covariance based on the daily data, which is the data driving the construction of your portfolio. Tell us in your email: what is the half-life of the covariance estimate?
    - the weights on each asset that get you the targeted risk you seek
    - the expected risk weight achieved using your numerical methodology
- a script file
    - Clearly annotated code in which you describe what you are doing in each step and your rationale for doing so.
    - You will need to use numerical methods to solve for the portfolio weights that achieve your risk target; For Matlab users, I recommend the solver lsqnonlin.

Problem #2: Use SQL to generate portfolio statistics

We prefer to calculate statistics on the fly, as needed, rather than storing them separately in a database. We also prefer to use the same program to access the data and to calculate such statistics. As a result, there are numerous occasions where we use SQL to calculate metrics that may be more simply calculated in MATLAB or Python.

Create an SQL function to calculate compound (not cumulative sum) returns over multiple frequencies

Download the free version of Microsoft SQL 2017 Express (https://www.microsoft.com/en-us/sql-server/sql-server-downloads)

- Create a schema and store the attached excel file (from problem #1) into a table
- Create a function that compounds the daily returns provided into month-to-date (mtd), quarter-to-date (qtd) and year-to-date (ytd) frequencies.

The function should take the following:

- @startDate
- @endDate
- @frequency

The function should return mtd, qtd, and ytd returns for each day

Deliverable: an Excel file with your inputs and results and the SQL script (text file) used to generate them

Add a flavor where one has multiple timestamps per each business day

1) Retrieve the latest timestamped value per each business day using a stored procedure, and then apply the compounding operations
2) A variation on #1 where you are asked to get the latest point in time value (that is the update timestamp <= the business date), and then apply the compounding operations

Deliverable: the SQL script (text file) used to generate them

Problem #3: Examine a multi-object class and explain its contents

Members of the Quantitative Research team often need to dissect code that they did not write.

- See the attached MATLAB function (UpdateAUM.m)
- Identify the main objects of this class
- Explain the functionality of each object
- State how to call this class to realize each of the aspect

Deliverable: an annotated version of the MATLAB file provided

Problem #4: Implement methodology described in an academic paper

Read and apply the technique described in the following paper using Python (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2660984)

The Excel file contains the hypothetical daily returns for the test strategy in the worksheet labeled "Problem #4". The expected Sharpe ratio of the strategy is assumed to be 0.5

Provide feedback on how to possibly improve the technique to be able to assess if/when a strategy should be retired.

Deliverable:

- A python code (Jupyter Notebook preferably)
- A document containing critical analysis of the findings in the paper (applicability, limitations, theoretical assumptions, …) and suggesting possible enhancements