CSC3320 — Lab #5 — Problem Solving with Data Analysis

Outcomes

- Demonstrate the ability to apply knowledge of mathematics, science, and engineering to design and develop a data analytics application.
- Demonstrate the ability to design and develop a relational database application relational database.
- Understand the design tradeoffs in terms of development effort, scalability, and performance when allocating functionality between procedural programming and relational operations.

Assessment

- Given a portfolio of stocks and their daily closing prices for one year, develop a data analytics application using a high-level programming language and a relational database that optimizes the risk-adjusted return of the portfolio for the investor risk as measured by the Sharpe Ratio.
- The application is required to:
  - Integrate procedural programming using a high-level application programming language (e.g., Java, Python), with relational operations (SQL) using a relational database.
  - Utilize analytical processing queries in SQL.

Requirements

You are provided with the adjusted closing prices of a portfolio of 4 stocks and the S&P 500 ETF (SPY). SPY serves as your benchmark asset. The SPY represents a market-weighted index of the stock prices of the 500 largest publicly traded companies in the US. You will load this information into a database, then write a program that uses the Sharpe Ratio to compute the optimal mix of these four stocks.

The lab feature 3 main steps:

1. Create a portfolio database
2. Write a portfolio simulation function
3. Write a portfolio optimization function

Step 1: Create a portfolio database

1. Download the data analytics lab archive (zip file) from Canvas and unzip the archive into a convenient directory.
2. The archive contains `csv` data files for 4 stocks: [GOOG, CELG, NVDA, FB], and the SPY index we will use for our benchmark. Use the 2017 files.

   Examine the contents of the files.
3. Create load tables for each of the 5 files and load the stock data from the `csv` files into them. Note: You can use the `labstockanalytics_import_2017.sql` included in the lab archive for this purpose.

You will need the following schema for each file:

```
(Date, Open, High, Low, Close, Volume)
```

4. Load the data into the tables.
5. Create a portfolio table for your analysis with the following schema.

```
 (Date,
S1 Adjusted Close, S1 Cumulative Return, S1 Value,
S2 Adjusted Close, S2 Cumulative Return, S2 Value,
S3 Adjusted Close, S3 Cumulative Return, S3 Value,
S4 Adjusted Close, S4 Cumulative Return, S4 Value,
SPY Adjusted Close, SPY Cumulative Return, SPY Value,
Portfolio Cumulative Return, Portfolio Value)
```

Feel free to replace Sx with the actual stock ticker to improve the readability of your table.

Note that Close in the load file is the *adjusted* close.[1]

6. Insert the data from the load tables into the portfolio table. Be careful to maintain proper ordering of dates and adjusted closing prices, i.e., ascending date order. Leave the cumulative return and value fields for each stock and the portfolio `NULL` for now.
7. Update the cumulative return (normalized closing price) field for each stock and the SPY, where

$$\text{each day}'\text{s cumulative return} = \frac{\text{that day}'\text{s closing price}}{\text{first day}'\text{s closing price}}$$

.

Step 1 submission requirements

In your lab report include a snapshot of your portfolio table as defined above (prior to updating the table with simulation results).

Step 2: Write a portfolio *simulation* function

Once the data is loaded, you will use a programming language of your choice (I recommend Java or Python) to write a function that queries the database to compute the Sharpe Ratio for a particular portfolio weighting.

Here, a *portfolio weighting* is an assignment of weights to each of the four stocks indicating how much of your investment will be given to each stock. So a weighting of [0.1, 0.3, 0.2, 0.4] puts 10% of your money in S1, 30% in S2,20% in S3 and 40% in S4. For this lab, we will use 0.1 increment granularity for all portfolio weightings.

The `portfolio_evaluation_example.xlsx` spreadsheet provides an example of the calculations your function will perform, using a different, smaller portfolio of stocks. These calculations can get confusing; please consult the example as needed.

Requirements for your portfolio simulation function:

[1] See `http://finance.zacks.com/adjusted-closing-price-vs-closing-price-9991.html` for the difference.

Inputs A list of stock symbols, and a portfolio allocation of those stocks.

Outputs Standard deviation of daily returns, average of daily returns, Sharpe Ratio, and cumulative return for the portfolio (across the entire time frame)

Functionality

- You can assume the stock symbols provided are valid for your data set, but should check that the portfolio allocation sums to 1.0. We are running our simulation for the entire range of data/ dates.
- Based on the portfolio allocations (weights for each stock), calculate the cumulative portfolio return for each day. This will be the weighted average of the cumulative returns for each individual stock in the portfolio. In the example spreadsheet, this is the `Cum. Port. Ret.` column.

  Should this be done in SQL or code? You can try either.
- Based on the portfolio allocations (weights for each stock), calculate the value fields for each stock and the portfolio, and update your portfolio table.

  The daily value of a stock can be computed by:

  $$\text{daily value of stock} = \frac{\text{weight of stock} \times \text{portfolio value} \times \text{adjusted daily closing price of stock}}{\text{first day's closing price of stock}}$$

  or just

  $$\text{weight of stock} \times \text{portfolio value} \times \text{daily cumulative return for stock}$$

  Assume a total portfolio value of $1.

  The daily value of the portfolio can be calculated by summing the daily values of each stock.
- For the portfolio, calculate the Sharpe Ratio of the portfolio (normalized by n trading days):

  $$\text{Sharpe} = \frac{\sqrt{n} \times \text{avg}(R_a - R_b)}{\text{stddev}(R_a - R_b)}$$

  Where

  R_a = cumulative daily returns of the portfolio, and

  R_b = cumulative daily returns of the benchmark SPY.
- Finally, calculate the overall cumulative return for the portfolio

  $$\frac{\text{last daily value} - \text{first daily value}}{\text{first daily value}}$$

  This and the Sharpe ratio can be found in the bottom-right corner of the example spreadsheet.

Step 2 submission requirements

In your lab report include the output results of your simulation function and a snapshot of your portfolio table after running a simulation for [GOOG, CELG, NVDA, FB] with allocation `[0.3, 0.3, 0.2, 0.2]` for a start and end date that covers the whole dataset.

Also include a second set of results where you have changed the stock weights to improve the Sharpe ratio and list all of the weights you tried along the way to finding one that improved result.

Step 3: Write a portfolio *optimization* function

In the same program as your simulation function, write a function which finds the portfolio weighting with the highest Sharpe Ratio.

Inputs a list of stock symbols.

Outputs

- The portfolio weighting with the highest Sharpe ratio across the date that cover the entire dataset.
- Standard deviation of daily returns, average of daily returns, Sharpe ratio, and cumulative return for the portfolio at that portfolio weighting

Functionality Using your simulation function, iterate through 0.1 increment allocations for each of your 4 stocks.

Ensure the sum of the weightings is always equal to 1.0. For example `[1.0, 0.0, 0.0, 0.0]` and `[0.2, 0.2, 0.2, 0.4]` are valid allocation weights.

Return the portfolio allocation and required output values with the highest Sharpe ratio.

Step 3 submission requirements

In your lab report include the output results of your best performing (optimized) simulation values and a snapshot of your optimal portfolio table after running your optimization function. Again, correctness of results is important!

The Sharpe Ratio

The Sharpe ratio was named after William Sharpe (Nobel Prize in Economics, 1990). In finance, the Sharpe ratio is a way to examine the performance of an investment by adjusting for its risk. The Sharpe measure gives us a way of measuring the return relative to the risk of an investment when compared to a benchmark investment. The ratio measures the excess return (or risk premium) per unit of deviation in an investment relative to a given baseline investment.

The basic premises of the Sharpe Ratio are as follows:

- The greater the deviation in the price of an investment (volatility), the greater the risk.
- The greater the risk for an investment, the greater the return you should receive.

The Sharpe Ratio is expressed mathematically as

$$S_a = \frac{E\left[R_a - R_b\right]}{\sigma_a} = \frac{E\left[R_a - R_b\right]}{\sqrt{\mathrm{var}\left[R_a - R_b\right]}}$$

where

R_a is the asset return

R_b is the return on a benchmark asset, such as the risk-free rate of return or an index like the S&P 500.

$E[R_a - R_b]$ is the expected (mean) value of the excess return of the asset over the benchmark return

σ_a is the standard deviation of this excess return

The ratio on the left is the original Sharpe ratio; the ratio on the right is the ex-post Sharpe ratio that uses realized returns rather than just expected returns. We will use the ex-post Sharpe ratio (right equation).

References

- `http://en.wikipedia.org/wiki/Sharpe_ratio`
- Original 1994 paper by William Sharpe

Deliverables

Submit a report with the submission requirements outlined above for each step (1, 2, 3) in the assignment to Canvas as a single PDF file.

Your report should be written so that someone without access to this document (but with knowledge of stock analytics) could read and understand it.

Submit your project as a separate zip file archive, as part of the same Canvas submission.

Credits

This lab was developed by Dr. Jay Urbain and adapted from a Computational Finance assignment from Professor Tucker Balch of Georgia Tech.