Week04 - Project Report

Ruchen Shi

Problem1

To simulate price P_t at time t with three types of price returns:

1. Classical Brownian Motion

$$P_t = P_{t-1} + r_t$$

2. Arithmetic Return System

$$P_t = P_{t-1}\left(1 + r_t\right)$$

3. Log Return or Geometric Brownian Motion

$$P_t = P_{t-1}e^{r_t}$$

I set T=1000, sigma=0.01 which is the standard deviation more close to the realistic situation, and make the initial price P_0=100. Initialize random normal sets r_t with mean of 0 and standard deviation of sigma. And then do 10000 times simulation of P_t. Then find the average and standard deviation of the P_t list.

Classical Brownian Motion:
The expected price after 1000 times motion is 100 because mean of r_t is zero.
The expected standard deviation is calculated by sigma*sqrt(T).

Arithmetic Return System:
The expected price after 1000 times motion is 100 because mean of r_t is zero.
The expected standard deviation is calculated by the mean of all standard deviation of P_(t-1).

Geometric Brownian Motion:
The expected price is calculated by P_0*exp(sigma^2/2*T), which is 105.13
The expected standard deviation is calculated by P_0 sqrt((exp(sigma^2*T)-1)* exp(2*mu*T)),where mu =(sigma^2/2)

According to the result below, we can see that the mean and standard deviation are almost the same, which match our expectations.

```
1. Classical:
Mean of P: 99.9995009149625
Expected Mean of P: 100
Standard deviation of P: 0.31222576415457065
Expected Standard deviation of P: 0.31622776601683794
2. Arithmatic:
Mean of P: 99.9903678600618
Standard deviation of P: 32.16371793562093
Expected Mean of P: 100
Expected Standard deviation of P: 32.14905628028684
3. Log Return:
Mean of P: 105.03364562608112
Standard deviation of P: 34.179668433907466
Expected Mean of P: 105.12710963760242
Expected Standard deviation of P: 34.09279103923912
```

Problem2

My return_calculate() function logic is that:

1. The function starts by checking if the specified date_column exists in the DataFrame. If not, it raises a ValueError.
2. Then it removes the date_column from the DataFrame, leaving only the price data for the calculations.
3. Return Calculation:
    - If the method is "DISCRETE", it calculates the percentage change in price, which is the simple return from one period to the next.
    - If the method is "LOG", it calculates the logarithmic returns, which is the natural logarithm of the ratio of successive prices.
4. The first row after the return calculation will always be NaN because the percentage change from an undefined previous point can't be calculated, so the function drops the first NaN row.
5. The function then adds the date information back to the returns DataFrame, now aligned with the returns data (since the first date, which corresponds to the NaN return, is excluded).
6. Finally, the function returns a DataFrame that contains the calculated returns along with the dates.

The results are below:

```
Value at Risk (VaR) for META at the 5% level using different methods:
VaR_Normal: -0.0543
VaR_EWMA: -0.0301
VaR_T: -0.0431
VaR_AR1: -0.0538
VaR_Historical: -0.0395
```

1. VaR_Normal (5.43%): This value is calculated under the assumption that returns are normally distributed. This is the simplest VaR calculation, which might not account for the actual distribution of stock returns that often exhibit fat tails (i.e., higher kurtosis than the normal distribution).
2. VaR_EWMA (3.01%): This method uses an Exponentially Weighted Moving Average to give more weight to recent observations when calculating variance. It assumes a normal distribution but can react more quickly to changes in volatility. The value is less negative than the simple normal VaR, suggesting less risk, which could indicate that recent volatility has been lower than the long-term average.
3. VaR_T (4.31%): The T-distribution is more appropriate for financial returns data that may not be normally distributed. The T-distribution has heavier tails than the normal distribution, which can account for the potential for extreme losses. The T-distribution-based VaR is between the normal and historical VaR values.
4. VaR_AR1 (5.38%): This method uses an Autoregressive model of order 1 to forecast the next period's return and then calculate VaR. The AR(1) model captures the serial correlation in returns, if any. The result is quite close to the normal VaR, suggesting that the simple past value may be a strong predictor of the next value, or that the returns are close to being normally distributed.
5. VaR_Historical (3.95%): This method does not assume a specific distribution and uses actual historical return data to calculate VaR. This method can capture the actual observed range of outcomes, including any skewness or kurtosis in the returns distribution. The historical VaR is more conservative than the other methods.

Summary:
The Historical Simulation VaR is the most conservative. The normal and AR(1) VaR values are similar, implying that returns could be close to normally distributed or that yesterday's return is a good predictor of today's return. The EWMA VaR reflects recent market conditions more and can be more sensitive to recent volatility. The T-distribution VaR provides a middle ground, reflecting the possibility of extreme outcomes better than the normal distribution.

Problem3

Normal Distribution with EWMA:

1. I developed a script that ingests portfolio and daily stock price information from CSV files to compute the Value at Risk (VaR) for individual portfolios and cumulatively. This script is constructed using the pandas and NumPy libraries for efficient data handling and calculations.
2. Initially, I extract the stock listings and their corresponding holdings amounts from the portfolio data utilizing pandas. Following that, I employ NumPy to ascertain the daily returns for each stock portfolio based on the provided daily price data.

3. Subsequently, I determine the exponentially weighted covariance matrix for each individual portfolio, setting lambda at 0.94. This method is extended to compute a comprehensive covariance matrix encompassing all portfolios, with recent data receiving higher emphasis, reflecting its greater significance in the context of financial markets.
4. Utilizing the latest stock prices and the existing holdings, I establish the current valuation for each individual portfolio and for the aggregate of all portfolios. To conclude, I apply a 0.95 confidence level and the inverse cumulative distribution function of a standard normal distribution to calculate the VaR for each portfolio and in total. VaR serves as a statistical metric that gauges the potential depreciation in an investment's value over a specified timeframe.

The output is below:

```
Portfolio A VaR: $15242.09
Portfolio B VaR: $7775.09
Portfolio C VaR: $17836.00
Total VaR: $38039.28
```

We can see that the risk of C is higher than A and B. The risk of A is higher than B. The total VaR is smaller than the sum of risk A, B, and C.
The above analysis shows that VaR doesn't have subadditivity, so it is not a coherent risk measure.

Historical Simulation:
Since Historic Simulation has no assumption of the distribution of returns, it is more reliable and logical.
The ouput is below:

```
Portfolio A VaR: $14264.14
Portfolio B VaR: $10437.30
Portfolio C VaR: $19194.54
Total VaR: $40359.78
```

We could see that the historical VaR in general is larger than the above estimator. It indicates that the normal assumption underestimates the probability of some extreme events.
I prefer historical simulation, because it relies on actual historical market data to estimate the potential for future losses. This method is grounded in real-world outcomes rather than theoretical distributions, which can make it more intuitive and easier to explain to others who may not have deep statistical knowledge. It captures the actual behavior and volatility of the market, including the occurrences of extreme events, without assuming that returns are normally distributed. This approach also has the benefit of simplicity and transparency; it doesn't require complex modeling assumptions or parameters beyond the choice of the historical window and confidence level. However, if we don't have enough data, it would have more bias.