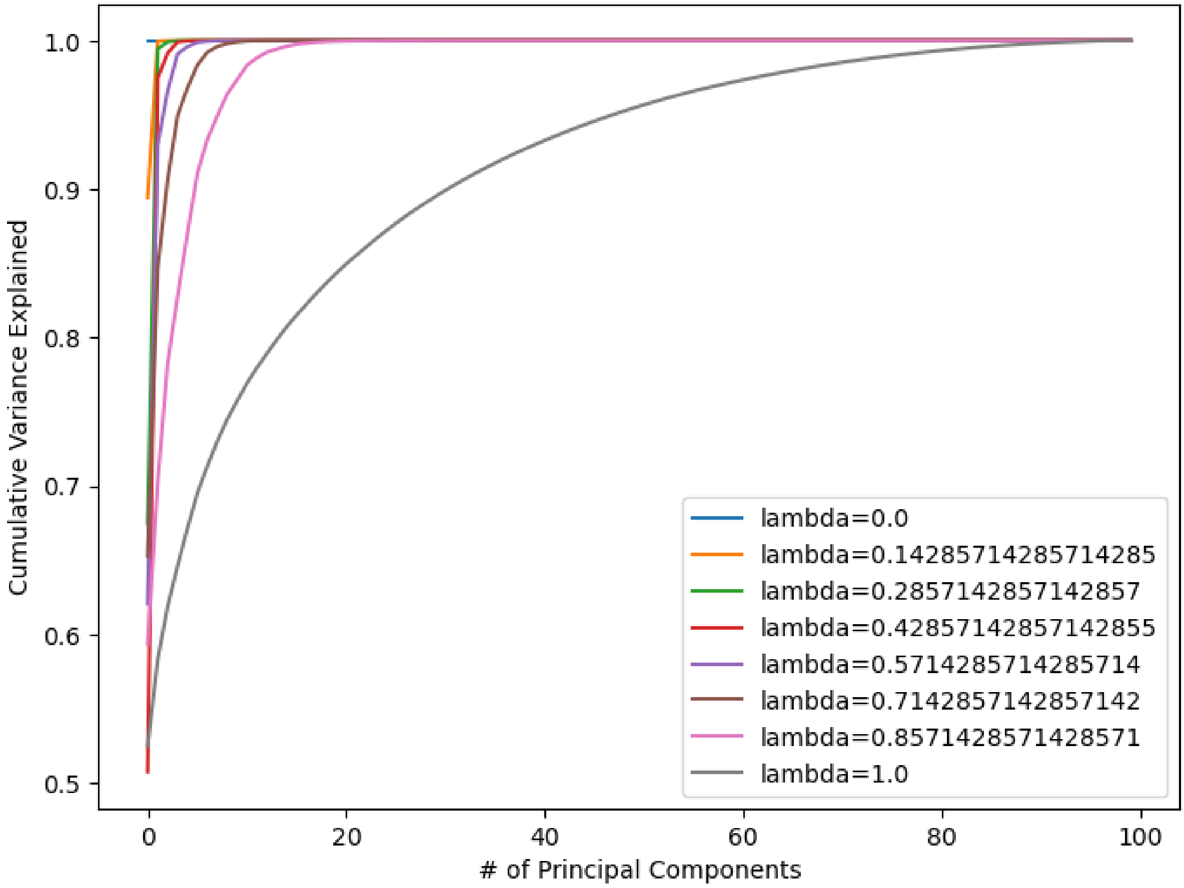
Zhiqiang Tian Fintech 545 09/23/2023

Week03 Project

Problem 1:

The implementation of the exponentially weighted covariance matrix is shown in the code.

The plot of the cumulative variance is shows below:



The interpretation of the above chart is: financial data often involves the concept of the time value of data. The physical interpretation of λ (lambda) is the weight assigned to recent data points. When λ is larger, more weight is assigned to the most recent data points, and the weights for distant data points are relatively smaller. This means that the overall trend in the data is not as heavily considered, resulting in a need for more principal components to capture the data's variations. Conversely, when more weight is assigned to recent data points and the weights for distant data points decrease, earlier changes are given more consideration, leading to a requirement for fewer principal components.

In summary, the magnitude of λ determines the emphasis placed on recent data compared to historical data. A larger λ places greater importance on recent changes, requiring more principal components to reflect these changes. A smaller λ places more emphasis on the overall trend in the data, resulting in a need for fewer principal components. Choosing the appropriate value of λ is crucial in financial data analysis, as it influences how we interpret and utilize information in time series data.

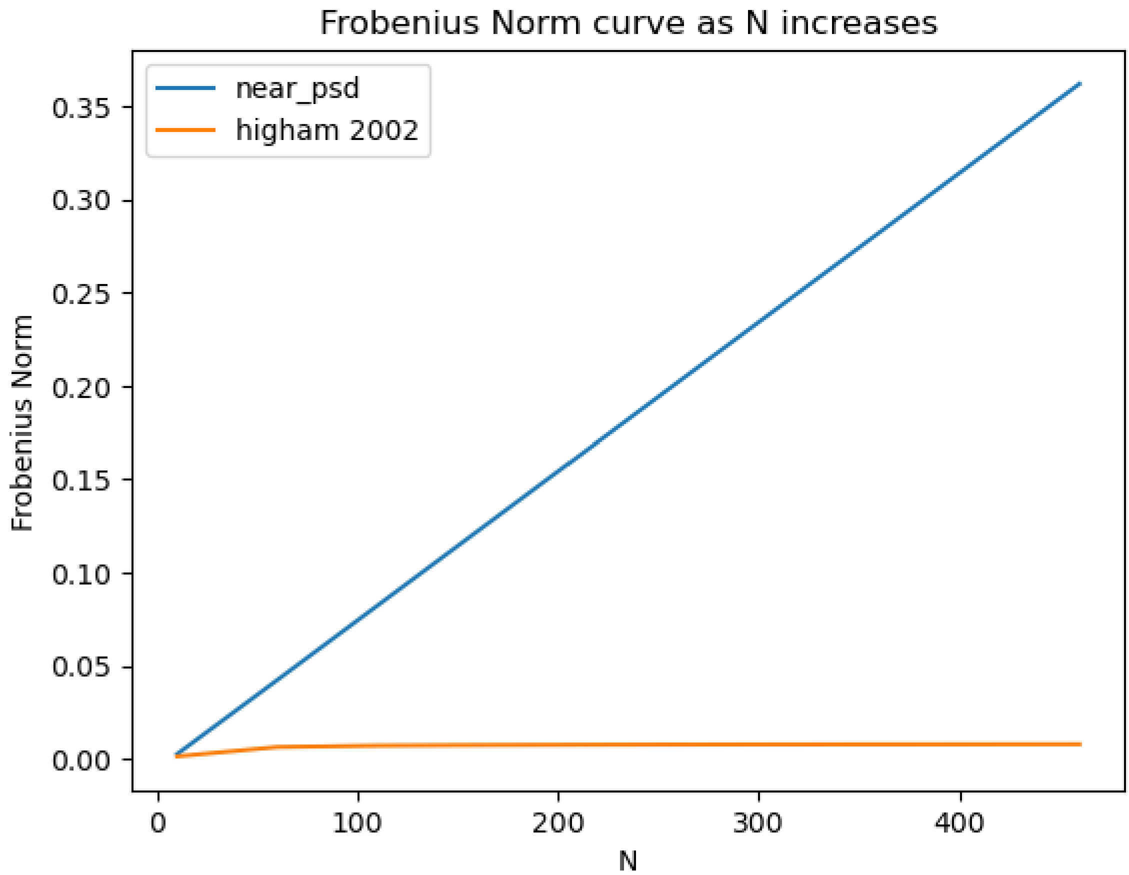
Problem 2:

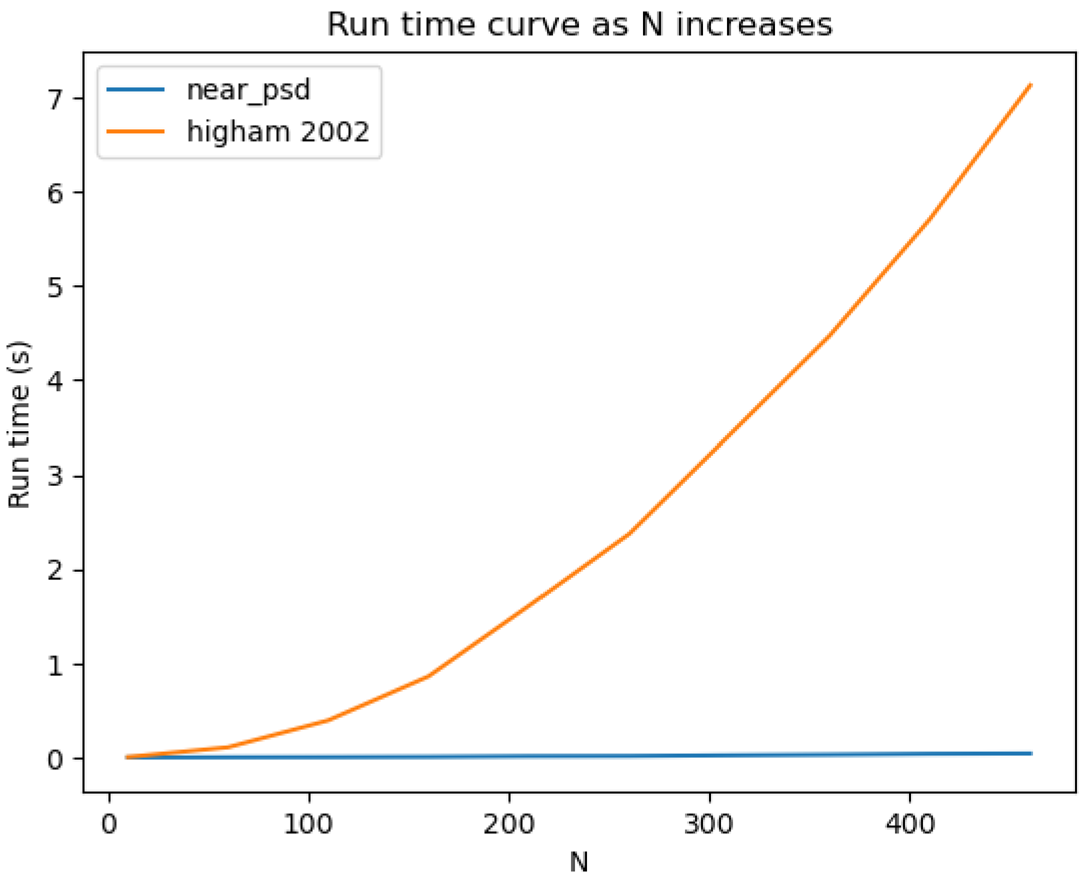
The process of implementation of chol_psd(), near_psd() and Higham's 2002 is in the code.

I have confirmed that the matrix after using near_psd() and Higham's method is PSD:

```
Original matrix:  False
Matrix fixed with near_psd():  True
Matrix fixed with Higham_psd():  True
```

The result of near_psd() and Higham's method is as follows:





As N increases, Higham's Frobenius Norm remains almost constant, while the Frobenius Norm of near_psd significantly increases. This indicates that Higham's method produces more accurate results. With increasing N, the runtime of Higham's method also increases significantly, whereas the runtime of near_psd increases only slightly.

Therefore, we can conclude as follows: The advantages of Higham's method lie in its robust implementation and effectiveness with large matrices, despite the longer runtime for large matrices. On the other hand, `near_psd` excels in handling small matrices quickly, with short runtimes even for large matrices, but it exhibits a sharp increase in error as the matrices grow larger.

I would choose to use `near_psd` when I need to swiftly correct a small non-PSD matrix, and I would opt for Higham's method when runtime is not a critical concern, as it provides a more accurate outcome, especially for larger matrices.

Problem 3

The result is as follows:

```
1
Direct simulation time is  0.15433216094970703
When explained variance =  1 simulation time is  0.1537330150604248
When explained variance =  0.75 simulation time is  0.10342788696289062
When explained variance =  0.5 simulation time is  0.08644628524780273


2
Direct simulation time is  0.10660195350646973
When explained variance =  1 simulation time is  0.11661100387573242
When explained variance =  0.75 simulation time is  0.08849215507507324
When explained variance =  0.5 simulation time is  0.06874370574951172


3
Direct simulation time is  0.09734606742858887
When explained variance =  1 simulation time is  0.10938882827758789
When explained variance =  0.75 simulation time is  0.08109402656555176
When explained variance =  0.5 simulation time is  0.05746912956237793


4
Direct simulation time is  0.11453008651733398
When explained variance =  1 simulation time is  0.10532307624816895
When explained variance =  0.75 simulation time is  0.08717703819274902
When explained variance =  0.5 simulation time is  0.05606508255004883
```

Reducing the explained variance comes at the cost of saving some time but results in decreased accuracy. With direct simulation, we can obtain the most accurate results by using the covariance matrix directly to generate simulated values. However, this approach may take a long time to run, especially when dealing with a large number of stocks and a long time period.

On the other hand, using PCA for value simulation reduces computational time since we only utilize the principal components, which capture most of the data's variability. As the explained variance decreases, the time required for the simulation also decreases, but the difference between the simulated covariance and the input covariance matrix increases.

When the explained variance is 100%, the simulated covariance closely matches the input covariance matrix, indicating accurate capture of relationships between stocks. As the explained variance decreases to 75% and 50%, the disparity between the simulated covariance and the input covariance matrix increases, indicating less precise capture of stock relationships.

In conclusion, while PCA simulations with 75% and 50% explained variance may be less accurate, they still capture some of the stock relationships and may be useful for specific applications.