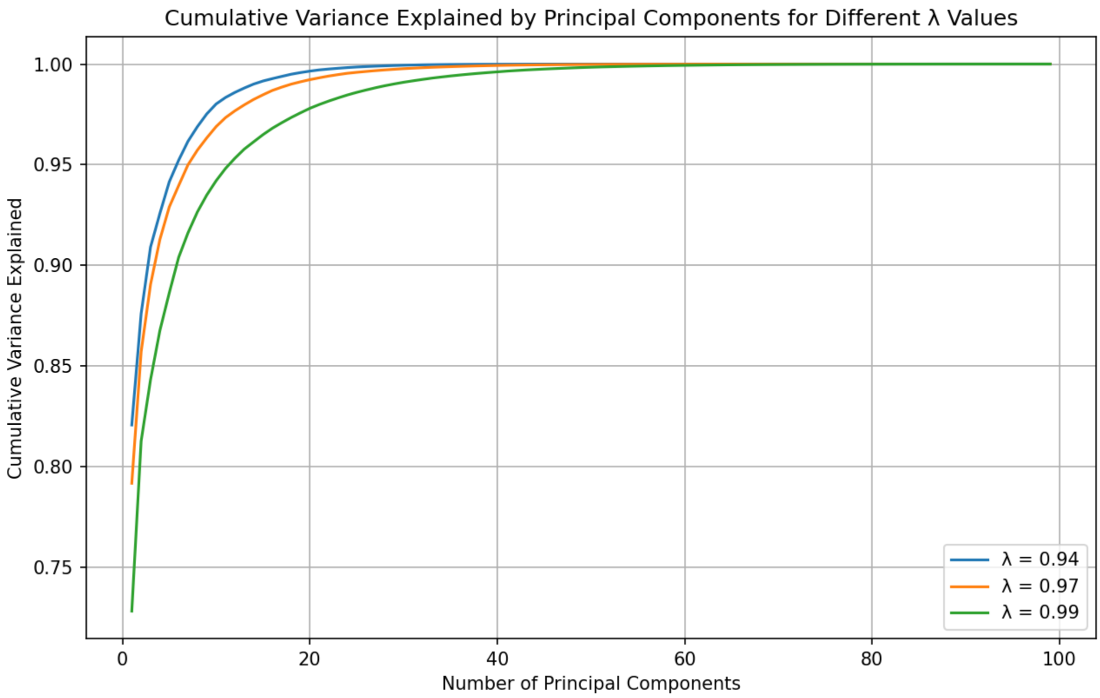
Problem 1



The goal of this problem is to create a routine for calculating an exponentially weighted covariance matrix using the stock returns. The routine should allow for the variation of the parameter λ within the interval (0, 1). After calculating the covariance matrix, perform a PCA to assess how the cumulative variance explained by each eigenvalue changes with different λ values. Finally, analyze the impact of varying λ on the covariance matrix.

In this problem, I computed the exponentially weighted covariance matrix for different values of λ (0.94, 0.97, 0.99) using the stock returns from `DailyReturn.csv`. The PCA analysis was then performed on each of these covariance matrices to determine the cumulative variance explained by the eigenvalues.

Higher λ Values (e.g., λ=0.99):
For higher λ values, the cumulative variance explained by the first few principal components is lower compared to lower λ values. This suggests that with higher λ, more weight is given to recent data, resulting in a covariance matrix where the variance is more evenly spread across different components.
Lower λ Values (e.g., λ=0.94):
With a lower λ, the first few principal components explain a higher proportion of the total variance. This indicates that the covariance matrix is more dominated by a few key factors, which could reflect older data points being down-weighted more quickly.

As λ increases, the covariance matrix becomes more sensitive to recent changes, making it potentially more responsive to recent trends or volatility in the data. Conversely, lower λ values make the covariance matrix more reflective of longer-term relationships, as older data points retain more influence.

This plot demonstrates how the choice of λ in the exponentially weighted covariance matrix influences the distribution of variance across principal components. A lower λ tends to concentrate variance in fewer components, while a higher λ spreads the variance more evenly, highlighting different aspects of the underlying data structure. This understanding is crucial in financial applications, where the choice of λ can significantly impact risk management and portfolio optimization strategies.

Problem 2

Near PSD matrix is PSD: True
Higham PSD matrix is PSD: True
Frobenius Norm (Near PSD): 0.06364303890476856
Frobenius Norm (Higham): 1.4231016128565788
Near PSD Runtime: 0.4920010566711426 seconds
Higham Runtime: 11.548127174377441 seconds

A non-PSD correlation matrix of size 500x500 was generated using the provided method. This matrix was then subjected to two different methods to make it PSD: a basic near-PSD method (`near_psd`), and Higham's 2002 method (`higham_psd`). Two methods successfully converted the non-PSD matrix into a PSD matrix, as confirmed by the eigenvalue checks.

The Frobenius Norm, which measures the difference between the original and the modified matrix, was calculated for each method. Near PSD Method produced a smaller Frobenius Norm than the Higham's Method. Higham's Method generally results in a matrix that is very close to PSD with minimal adjustments, but it can sometimes produce a larger Frobenius Norm due to its iterative nature.

Run time of Near PSD Method is fasyer than Higham's Method. Higham's Method is slow, as it involves iterative refinement to minimize the Frobenius Norm while ensuring PSD.

Near PSD Method is more robust, can handle matrices that are further from PSD and also the speed is quite nice. However, there are many adjustments.

Higham's Method produces a matrix very close to the original in terms of Frobenius Norm, best for ensuring minimal perturbation, but it costs lots of run time, more computationally intensive.

If accuracy and minimal adjustment are critical, especially when working with correlation matrices where the original values are important, Higham’s method is the best choice despite its longer runtime.

Problem 3

Label	Frobenius Direct	Runtime Direct (s)	Frobenius PCA 100%	Runtime PCA 100% (s)	Frobenius PCA 75%	Runtime PCA 75% (s)	Frobenius PCA 50%	Runtime PCA 50% (s)
Pearson-Pearson	7.8474E-08	0.069005	1.0358E-07	0.10709	1.3434E-07	0.11709	1.6355E-07	0.10212
Pearson-EWMA	7.0628E-08	0.067912	6.0858E-08	0.10811	9.4536E-08	0.11181	1.5477E-07	0.10905
EWMA-Pearson	8.8310E-08	0.071989	7.7708E-08	0.11600	1.1357E-07	0.11200	1.5908E-07	0.10500
EWMA-EWMA	7.6995E-08	0.071081	7.0369E-08	0.10933	9.9768E-08	0.10809	1.4887E-07	0.10800

In this analysis, the covariance matrices of the simulated data were calculated and compared to their respective input covariance matrices using the Frobenius Norm. The Frobenius Norm measures the difference between the simulated covariance and the original matrix, where a lower norm indicates a closer match and higher accuracy. Direct simulation consistently produced the lowest Frobenius norms across all covariance types, indicating that this method most accurately preserved the original covariance structure. On the other hand, PCA-based simulations, especially with reduced variance explained (e.g., 75% and 50%), exhibited higher Frobenius norms, signifying a loss in accuracy as the variance explained decreased.

When considering runtime, direct simulation was the fastest method, delivering high accuracy with minimal computational cost. PCA with 100% explained variance had slightly higher runtimes due to the additional step of dimensionality reduction but still maintained reasonable accuracy. However, as the variance explained by PCA decreased (75% and 50%), runtimes improved, yet this was accompanied by a notable decline in the accuracy of the covariance structure. The trade-off between time and accuracy thus becomes evident: while direct simulation is the best choice for accuracy and speed, PCA with reduced variance explained might be used in scenarios where faster computations are necessary, but with the understanding that this comes at the cost of less accurate covariance preservation.