Fintech545 Project2

Jieyang Ran

September 22, 2024

1 Problem 1 Use the stock returns in DailyReturn.csv for this problem. DailyReturn.csv contains returns for 100 large US stocks and as well as the ETF, SPY which tracks the S&P500.

1.1 Create a routine for calculating an exponentially weighted covariance matrix. Vary $\lambda \in (0, 1)$. Use PCA and plot the cumulative variance explained by each eigenvalue for each λ chosen. What does this tell us about values of λ and the effect it has on the covariance matrix?

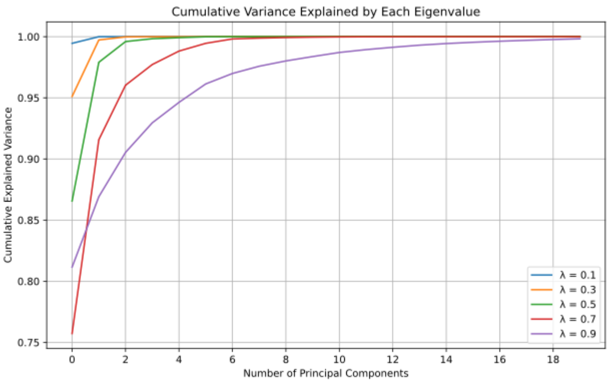

Figure 1: 1.1

In the above graph, we used $\lambda \in \{$ 0.1, 0.3, 0.5, 0.7, 0.9 $\}$ to do PCA and plot the cumulative variance. From this graph, we can conclude that the lower λ is, the higher the explanation power of the first few components are. It indicates that stock returns are more closely related to recent data.

2 Problem 2

Copy the chol_psd(), and near_psd() functions from the course repository - implement in your programming language of choice. These are core functions you will need throughout the remainder of the class.

Implement Higham's 2002 nearest psd correlation function. Use near_psd() and Higham's method to fix the matrix. Confirm the matrix is now PSD.

Compare the results of both using the Frobenius Norm. Compare the run time between the two. How does the run time of each function compare as N increases?

Based on the above, discuss the pros and cons of each method and when you would use each.

Method	$N = 100$	$N = 300$	$N = 500$	$N = 700$	$N = 900$
near_psd()	0.0	0.0155	0.0262	0.0473	0.0837
Higham_psd()	0.0321	0.4706	1.3037	3.3721	6.6424

Table 1: Running time as N increases.

Method	$N = 100$	$N = 300$	$N = 500$	$N = 700$	$N = 900$
near_psd()	0.0744	0.2341	0.3938	0.5535	0.7131
Higham_psd()	0.0071	0.0079	0.0080	0.0081	0.0081

Table 2: Difference between the original matrix and the transformed PSD matrix using the method of Frobenius norm.

From these two graphs, we can tell that the running time of near_psd() is shorter than Higham_psd(). But the result of Higham_psd() is more accurate than near_psd(). As N increases, the running time of Higham_psd() increases far more rapidly than that of near_psd().

Pros and Cons: The method of near_psd() has short running time but relatively low accuracy. The method of Higham_psd() has long running time, but relative high accuracy. When N is small, I prefer using Higham_psd(), as the running time won't be too long and accuracy is high. When N is large, I'd like to sacrifice accuracy for a shorter runtime.

3 Problem 3

Implement a multivariate normal simulation that allows for simulation directly from a covariance matrix or using PCA with an optional parameter for % variance explained. If you have a library that can do these, you still need to implement it yourself for this homework and prove that it functions as expected.

Generate a correlation matrix and variance vector 2 ways:
1. Standard Pearson correlation/variance (you do not need to reimplement the cor() and var() functions).
2. Exponentially weighted $\lambda = 0.\ 97$

Combine these to form 4 different covariance matrices. (Pearson correlation + var()), Pearson correlation + EW variance, etc.)

Simulate 25,000 draws from each covariance matrix using:
1. Direct Simulation
2. PCA with 100% explained.
3. PCA with 75% explained.
4. PCA with 50% explained.

Calculate the covariance of the simulated values. Compare the simulated covariance to it's input matrix using the Frobenius Norm (L2 norm, sum of the square of the difference between the matrices). Compare the run times for each simulation.

What can we say about the trade offs between time to run and accuracy.

The running time for direct simulation is longer because it involves performing the Cholesky decomposition on the entire covariance matrix. In contrast, simulation using PCA is quicker, especially when using fewer principal components.

Direct simulation provides the highest accuracy since it uses the complete covariance matrix. The accuracy of PCA-based simulation decreases as the explained variance percentage decreases. When using 100% explained variance, the results are very close to those of direct simulation. However, using 75% and 50% explained variance leads to a reduction in accuracy while significantly decreasing computation time.

Method	*Direct*	$PCA(100\%explained)$	$PCA(75\%explained)$	$PCA(50\%explained)$
difference	0.00000	0.0000700523	0.0000700523	0.0000700527
running time(seconds)	76.60534	0.14103	0.04730	0.04688

Table 3: Pearson correlation + var().

Method	*Direct*	$PCA(100\%explained)$	$PCA(75\%explained)$	$PCA(50\%explained)$
difference	0.00000	0.0001268310	0.0001268314	0.0001268318
running time(seconds)	83.39563	0.13783	0.03094	0.03429

Table 4: Pearson correlation + ew_var().

Method	*Direct*	$PCA(100\%explained)$	$PCA(75\%explained)$	$PCA(50\%explained)$
difference	0.00000	0.0001022292	0.0001022295	0.0001022296
running time(seconds)	83.53966	0.15306	0.03799	0.04552

Table 5: EW correlation + var().

Method	*Direct*	$PCA(100\%explained)$	$PCA(75\%explained)$	$PCA(50\%explained)$
difference	0.00000	0.0001846160	0.0001846167	0.0001846172
running time(seconds)	84.95572	0.14629	0.03447	0.03752

Table 6: EW correlation + ew_var().