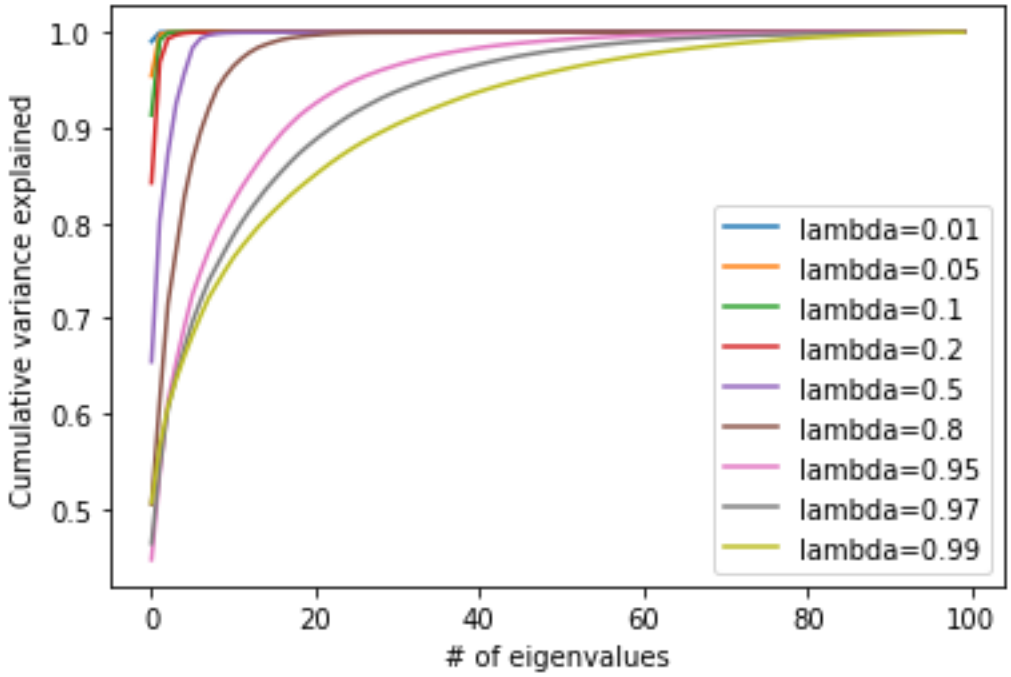
FINTECH 545 Week03 Project Report

Yilun Wu(yw528)

Problem 1:

The implementation of the routine for calculating an exponentially weighted covariance matrix is in the Problem 1 Section of code file (code.ipynb).

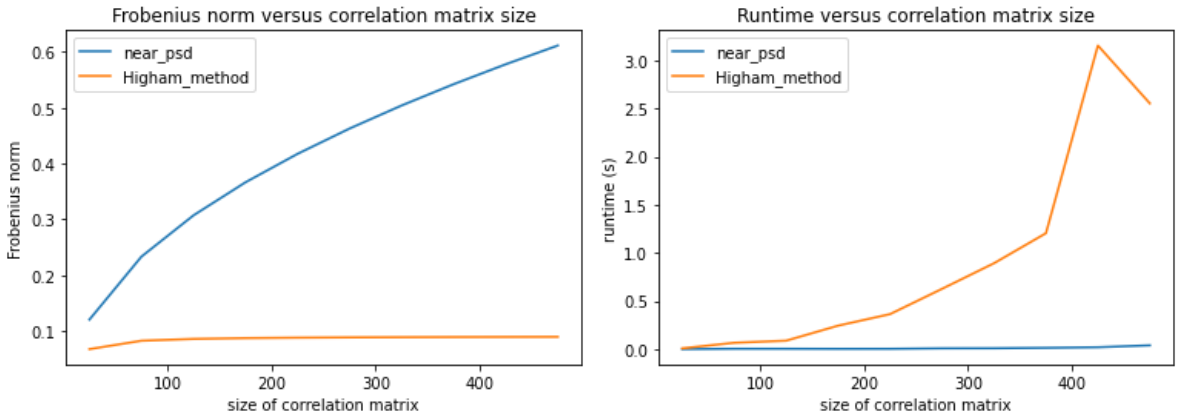
The plot of the cumulative variance explained by each eigenvalue for each λ chosen from [0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 0.95, 0.97, 0.99] are shown below:



According to the plot above, we can figure out that when the value of λ is higher, the cumulative variance explained increases slower as the number of principle components (eigenvalues) increases. Therefore, **the value of λ here is inversely proportional to the extent that we weight the most recently used information—a larger value of λ will result in more data (data further from recent) needed to explain the variance, while a smaller value of λ will result in less data further from recent needed to explain the variance.**

Problem 2:
The implementation of Higham's 2002 nearest psd correlation function and the confirmation that the matrix after fixing by near_psd() and Higham's method are both positive semi-definite are in the Problem 2 Section of code file (code.ipynb).

The comparison of the results of both function using the Frobenius Norm and runtime for various matrix sizes are shown below:

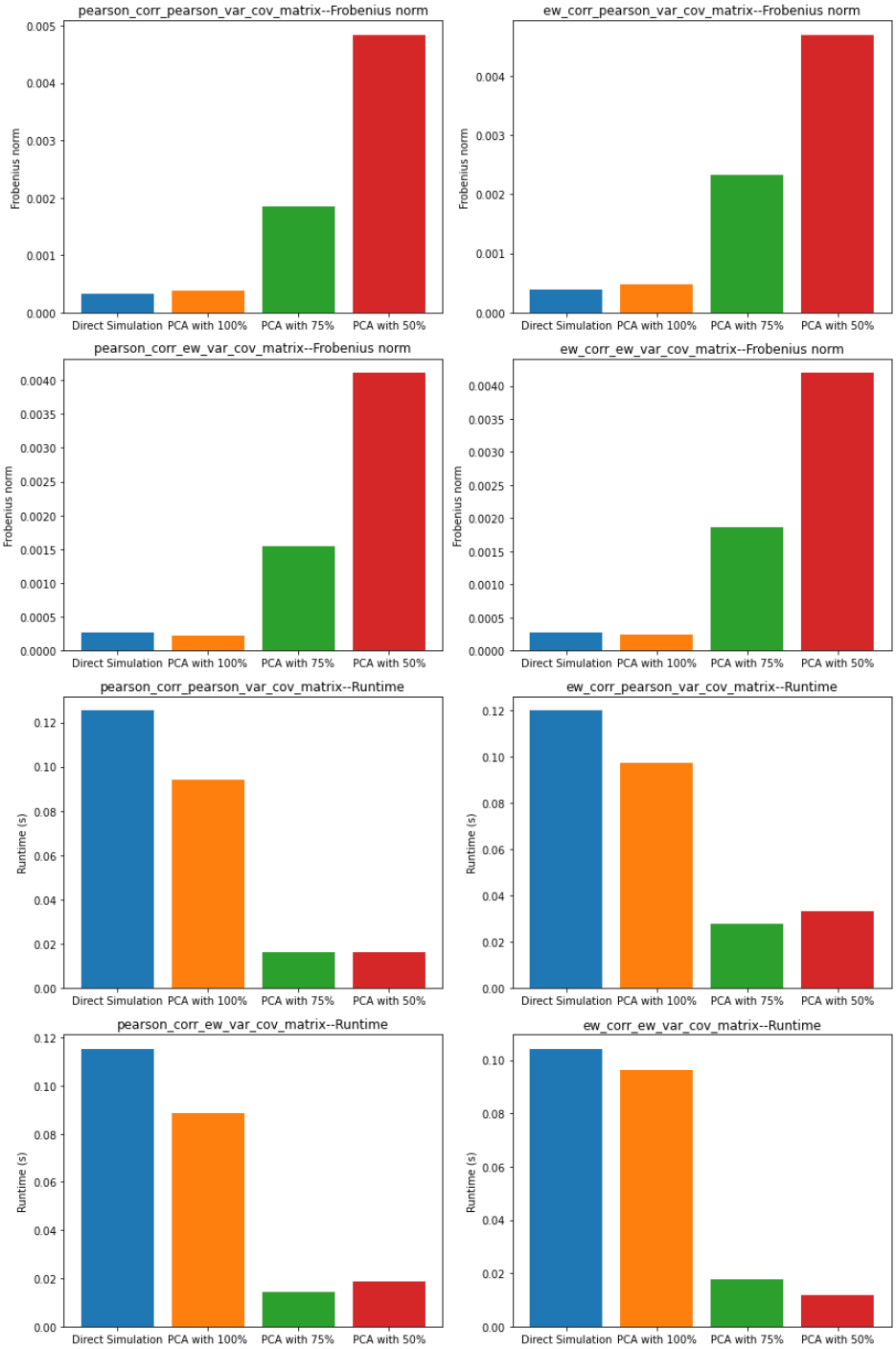


According to the two plots above, we can figure out that as the size of the correlation matrix increases, the Frobenius Norm (L2-Norm, also is the error term) of the matrix fixed by near_psd() function also increases, while the Frobenius Norm of the matrix fixed by Higham's method remains constant instead; on the other hand, the runtime of the near_psd() function remains constant as the size of the matrix increases, yet the runtime of Higham's method earns a heavily increasing trend in general.

Therefore, the pros for the near_psd() function is that **it is time-efficient**, and the cons for this function is that **it will have increasing error (measured by the Frobenius Norm as noted) as the size of the correlation matrix increases**; the pros for the Higham's method is that **it has constant error rate no matter what the correlation matrix size is**, yet the cons for this method is that **its runtime is proportional as the size of the correlation matrix increases, so it may result in a huge computational cost for large-size matrices**. Hence, from my perspective, I will choose to use the Higham's method if the correlation matrix size or the data size is in the scale of 100-100000 to ensure the accuracy, while I will choose to use the near_psd() function if the correlation matrix size is greater than 100000 to ensure the efficiency and reduce the computational cost instead.

Problem 3:
The implementation of a multivariate normal simulation that allows for simulation directly from a covariance matrix or using PCA with an optional parameter for % variance explained and the simulation, as well as the computation of the covariance of simulated values is in the Problem 3 Section of code file (code.ipynb).

The comparison of the simulated covariance to its input matrix using the Frobenius Norm and runtimes for each simulation are shown below:



According to the comparison plots above (on the previous page), we can figure out that the runtime for direct simulation is the highest, and then it's the PCA with 100% variance explained, and the runtime decreases as the percent of variance explained decreases for the PCA; and the Frobenius Norm (error) is about the same for the direct simulation and the PCA with 100% variance explained, and the Frobenius Norm increases as the percent of variance explained increases for the PCA. Therefore, we can note that there is a trade-off between the runtime and accuracy here—**as the percent of variance explained decreases for the PCA, the runtime decreases as well, yet the error increases at the same time, which indicates that the accuracy decreases**. The rationale for runtime is that the percent of variance explained is proportional to the size of the number of eigenvectors, which is proportional to the size of the covariance matrix, and thus smaller size of the covariance matrix requires lower computation cost and lower runtime. The rationale for the accuracy is that the percent of variance explained is proportional to the number of effective features, which is also proportional to the accuracy; if the number of effective features becomes lower as the percent of variance explained decreases, then the accuracy will also decrease.