Problem 1

Using the code for generating Exponentially weighted covariance matrix and for PCA simulation with different lambda values, we could get the following graph. As lambda gets larger, the more eigen value are required for the explanation of variance. In addition, we could also find the for small lambda value, the weights are more heavily weighted for the most recent daily return.

A graph of different colored lines

Description automatically generated

Problem 2.

This code will:

A computer screen shot of text

Description automatically generated

Generate a non-positive definite correlation matrix of size 500x500.

Fix it using both Higham's method and near\_psd().

Compare the runtimes and Frobenius norms for both methods.

A screen shot of a computer program

Description automatically generated

I run the code in a Julia environment to observe the results for the specified matrix size (n=500). And I explore how runtime changes with matrix size (N), through adjusting the value of n and run the corresponding functions accordingly. According to the result, we could find that, in general, the Forbenius Norm of the Higham is always quite close to 0 as the sample size increase, while the other Forbenius norm is increase as sample size increase. The run time for the higham method is slower, especially as sample size is larger than 500, while for the near psd, the runtime is stable as sample size get larger, and it is faster than the higham as sample size gets larger.

Problem 3

For this problem, we generate a random covariance matrix. Calculate the correlation matrix and variance vector using standard Pearson correlation/variance and calculate the correlation matrix and variance vector using exponentially weighted correlation/variance with λ = 0.97.

For the accuracy comparison:

The direct method and PC with 100% explain how the lowest and highest accuracy and as the percentage explained decreases PC become less accurate. As percent explains decreases, less features are incorporated into the PC model and thus less accurate. According to the result, we could find that PCA with 50% explanation is the most inaccurate, as the explained percentage gets larger, the accuracy is increased, and the direct simulation is the most accurate one with lowest error.

For the run time:

The direct calculation cost the most time for simulation, while the that PCA with 50% explanation is the most fast one. As the percentage of the explained becomes larger, the operation process time gets smaller. So there is a trafe off between the accuracy and run time of the code.