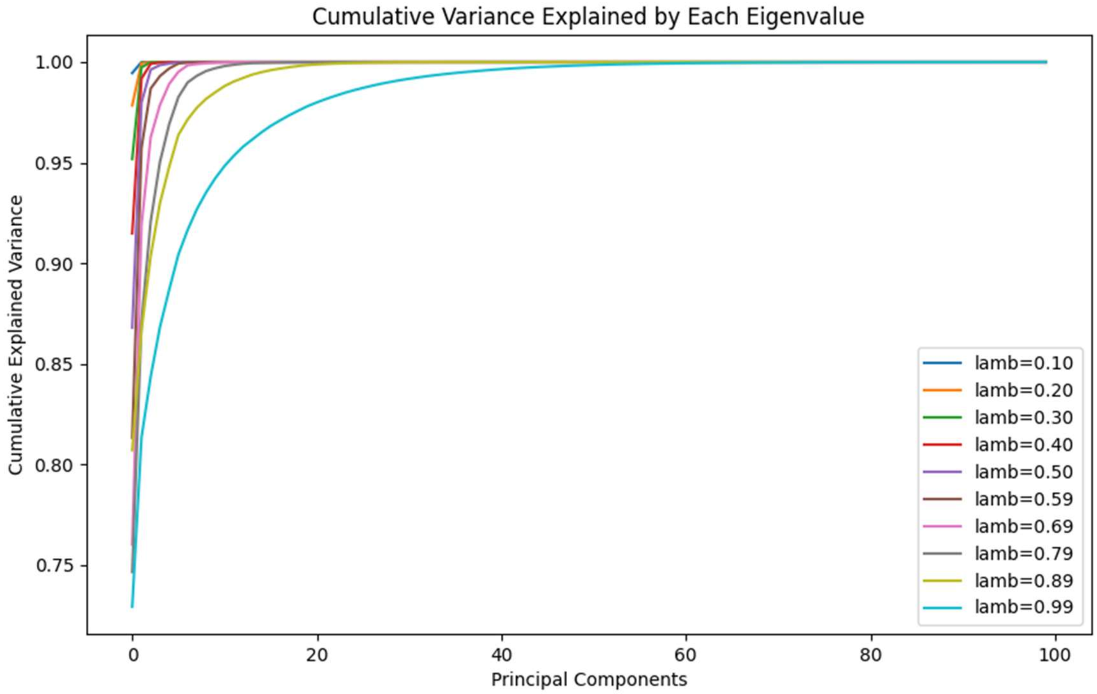
Prob1




By using PCA formula, I plot the cumulative variance explained by eigenvalue with lambda from 0.1 to 0.99. based on the graph, we can find that when the lambda increases, the change of cumulative variance decreases. Thus, they have an inverse relationship. Smaller lambda shows recent fluctuations which reflects short-term variations, and larger lambda shows past fluctuations which reflects long-term variations.

Prob2

```
Matrix size: 100x100
near_psd runtime: 0.05094 seconds, Frobenius norm: 17742.37732
Higham runtime: 3.20505 seconds, Frobenius norm: 44.50326
Matrix size: 200x200
near_psd runtime: 0.10542 seconds, Frobenius norm: 78661.45387
Higham runtime: 9.03733 seconds, Frobenius norm: 92.12451
Matrix size: 500x500
near_psd runtime: 1.18636 seconds, Frobenius norm: 682162.08438
Higham runtime: 111.90684 seconds, Frobenius norm: 236.96745
```

After checking the results of both functions using the Frobenius Norm and runtime for different matrix size, based on different matrix size, near-psd always has short runtime than Higham, while near-psd has larger Frobenius norm than Higham, On the other hand, the changes of runtime for near-psd is less obvious than Higham, and the changes of Frobenius norm for Higham is less obvious than near-psd. To sum up, near-psd is faster than Higham, but Higham is more accurate than near-psd.

Prob3

COV1: Pearson correlation + Pearson variance

COV2: Pearson correlation + Exponentially weighted variance

COV3: Exponentially weighted covariance matrix

COV4: Exponentially weighted covariance with Pearson variance

Covariance Matrix 1

Method: direct, Frobenius Norm: 0.00018, Runtime: 0.20067 seconds

Method: pca_100, Frobenius Norm: 5.19367, Runtime: 0.19195 seconds

Method: pca_75, Frobenius Norm: 3.32334, Runtime: 0.14209 seconds

Method: pca_50, Frobenius Norm: 3.30750, Runtime: 0.14362 seconds

Covariance Matrix 2

Method: direct, Frobenius Norm: 0.00000, Runtime: 0.21080 seconds

Method: pca_100, Frobenius Norm: 4.58535, Runtime: 0.16270 seconds

Method: pca_75, Frobenius Norm: 2.63465, Runtime: 0.16299 seconds

Method: pca_50, Frobenius Norm: 2.64797, Runtime: 0.16857 seconds

Covariance Matrix 3

Method: direct, Frobenius Norm: 0.00000, Runtime: 0.17869 seconds

Method: pca_100, Frobenius Norm: 3.32440, Runtime: 0.12302 seconds

Method: pca_75, Frobenius Norm: 2.00636, Runtime: 0.17930 seconds

Method: pca_50, Frobenius Norm: 2.00281, Runtime: 0.17794 seconds

Covariance Matrix 4

Method: direct, Frobenius Norm: 0.00000, Runtime: 0.25390 seconds

Method: pca_100, Frobenius Norm: 2.64216, Runtime: 0.18262 seconds

Method: pca_75, Frobenius Norm: 1.41301, Runtime: 0.16165 seconds

Method: pca_50, Frobenius Norm: 1.41004, Runtime: 0.17487 seconds

The output is shown above. Based on the simulation, the result shows that the direct method had lowest Frobenius Norm which provides more accurate covariance estimates.

However, for the running time, direct method always takes more time than PCA method. Moreover, when look inside the PCA method, 100% PCA tends to be less accurate and take less time. Therefore, there is a trade-off inside these methods. When we seek for accuracy, we should sacrifice time, if we want to spend less time, we have to sacrifice our accuracy. There is similar result in problem 2. We cannot pursue both time and accuracy at the same time, there's always a trade-off inside.