Week 3 Project

Question 1

First, implement the function to calculate the exponential weighted covariance matrix based on the algorithm in the note.

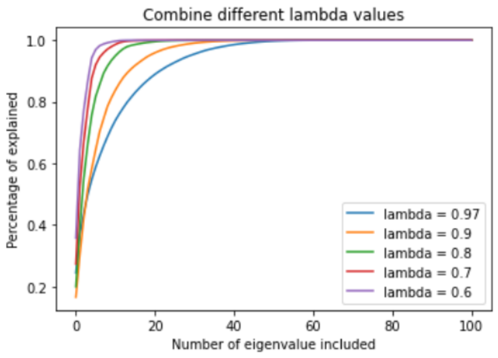
$$\widehat{cov(x, y)} = \sum_{i=1}^{n} w_{t-i}\left(x_{t-i} - \overline{x}\right)\left(y_{t-i} - \overline{y}\right)$$

In this algorithm, we first need to get the predict weight for each row of data, which means we need to normalize the weight by dividing it with the sum of weight.

$$w_{t-i} = (1 - \lambda)\,\lambda^{i-1}$$

$$\widehat{w_{t-i}} = \frac{w_{t-i}}{\sum_{j=1}^{n} w_{t-j}}$$

Then use PCA to calculate the cumulative covariance explained by each eigenvalue for each λ chosen.



From the graph above, we can see that the higher λ is, the more weight we put on the most recent data, the smaller number of eigenvalues we need to explain the whole data.

Question 2

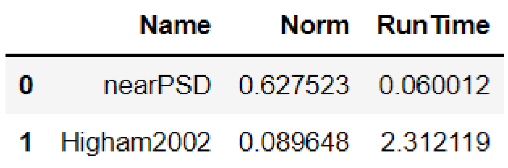
In this problem, we need to test and compare two methods for finding the nearest positive semi-definite matrix for any non-PSD matrix.

The first method was invented by Rebonato and Jackel in 1999. They first get the eigenvalues and eigenvectors of the matrix, and replace all the negative eigenvalues to be 0, then they construct a diagonal scaling matrix T to find the matrix B such that B times the transpose of B is the nearest PSD of the input matrix.

$$B = \sqrt{T}\, S \sqrt{\Lambda'}$$

The second method was invented by Higham in 2002. The goal of this method is to find the PSD matrix A such that the $||A - C||$ is minimized. This method uses two projection of the input matrix and weighted with an identical matrix W to make corrections until we reach the desired tolerance.

We build a non-PSD matrix, and test with these two methods, and compare their precision and runtime, and the result is:

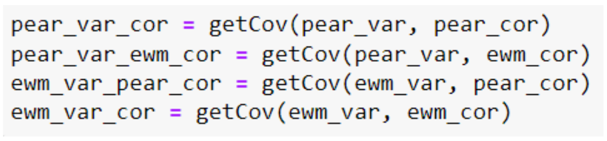


	Name	Norm	RunTime
0	nearPSD	0.627523	0.060012
1	Higham2002	0.089648	2.312119

This shows that Higham’s method is more precise, but it also takes significantly more time to run. This is because Higham’s method takes loops until it gets the desired tolerance, and Rebonato and Jackel’s method only change the eigenvalues and get the result. However, I think Higham’s method is better because it can be flexible by changing the tolerance.

Question 3

In this problem, we first need to get the variance and correlation matrix of the data using standard Pearson and exponentially weighted method, then combine them to get 4 different types of the covariance matrix.



```
pear_var_cor = getCov(pear_var, pear_cor)
pear_var_ewm_cor = getCov(pear_var, ewm_cor)
ewm_var_pear_cor = getCov(ewm_var, pear_cor)
ewm_var_cor = getCov(ewm_var, ewm_cor)
```

From the multivariate normal distribution, we get that:

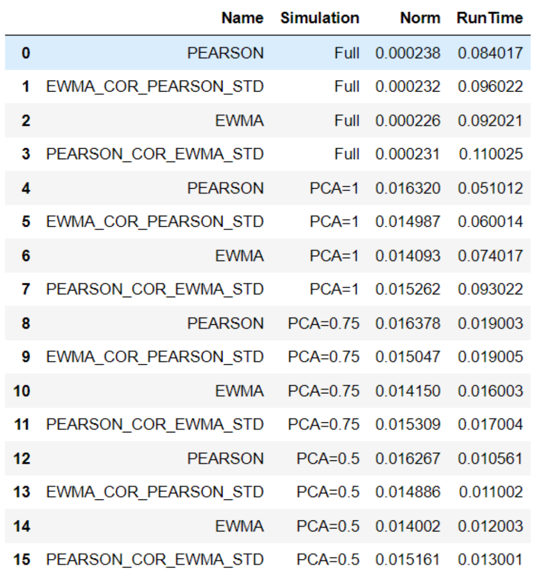
if $X \sim N(\mu, \Sigma)$,

then there exists $\mu \in R^n, L \in R^{n \times m}$ such that $X = LZ + \mu$

where $Z \sim N(0,1), i \in [1, m]$ and $\Sigma = \mathrm{LL}'$

The result is similar to the Cholesky Factorization and PCA where we can find a matrix time its transpose approximates the covariance of the original data, so we will simulate them to see the result.

For the simulation, we used full rank matrix for the Cholesky Factorization, and PCA with 50%, 75% and 100% explained, and compared the precision using Frobenius norm.



	Name	Simulation	Norm	RunTime
0	PEARSON	Full	0.000238	0.084017
1	EWMA_COR_PEARSON_STD	Full	0.000232	0.096022
2	EWMA	Full	0.000226	0.092021
3	PEARSON_COR_EWMA_STD	Full	0.000231	0.110025
4	PEARSON	PCA=1	0.016320	0.051012
5	EWMA_COR_PEARSON_STD	PCA=1	0.014987	0.060014
6	EWMA	PCA=1	0.014093	0.074017
7	PEARSON_COR_EWMA_STD	PCA=1	0.015262	0.093022
8	PEARSON	PCA=0.75	0.016378	0.019003
9	EWMA_COR_PEARSON_STD	PCA=0.75	0.015047	0.019005
10	EWMA	PCA=0.75	0.014150	0.016003
11	PEARSON_COR_EWMA_STD	PCA=0.75	0.015309	0.017004
12	PEARSON	PCA=0.5	0.016267	0.010561
13	EWMA_COR_PEARSON_STD	PCA=0.5	0.014886	0.011002
14	EWMA	PCA=0.5	0.014002	0.012003
15	PEARSON_COR_EWMA_STD	PCA=0.5	0.015161	0.013001

From the result, we can see that the precision is negatively related to the runtime. The full rank simulation has the best precision, while the precision of PCA decreases as the required amount of explanation decreases.