Problem 1

Given the dataset in problem1.csv

A. Calculate the Mean, Variance, Skewness and Kurtosis of the data

Mean: 0.050366360906888966

Variance: 0.010314441602725719

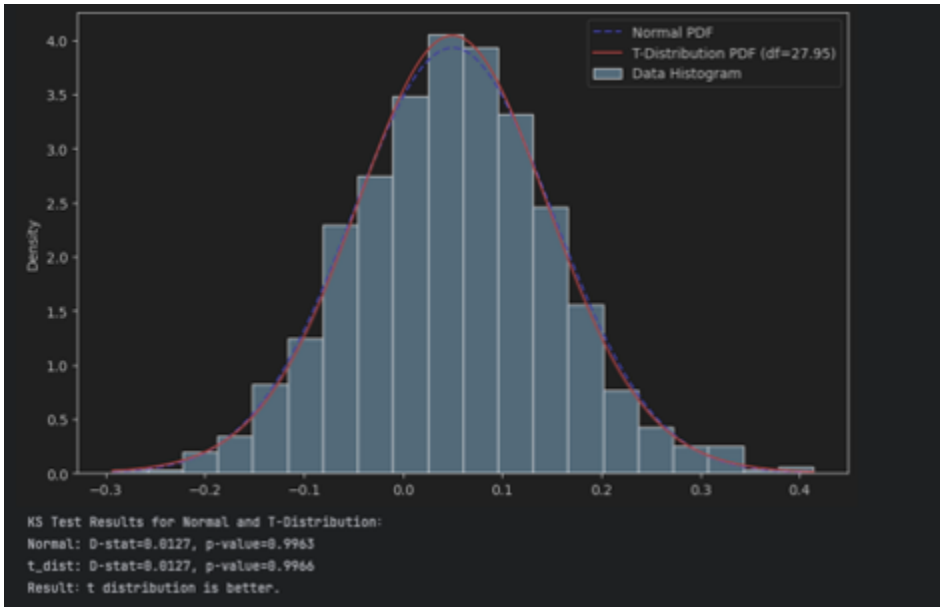
Skewness: 0.12046167291008684

Kurtosis (excess): 0.22908332509377516

B. Given a choice between a Normal Distribution and a T-Distribution, which one would you choose to model the data? Why?

A Normal Distribution seems like the better choice to model it. The mean is 0.050, and the variance is 0.010, which are within normal ranges. The skewness is 0.120, indicating that the data is almost perfectly symmetric, with only a very slight rightward lean. The excess kurtosis is 0.229, showing that the tails of the data are slightly heavier than those of a perfect Normal Distribution but still fairly close. Since these values align closely with the properties of a Normal Distribution, it seems like a more natural fit than a T-Distribution, which is used for data with much heavier tails.

However, using KS Test Results for Normal and T-Distribution:



Normal: D-stat=0.0127, p-value=0.9963

t_dist: D-stat=0.0127, p-value=0.9966

Result：t distribution is better.

C. Fit both distributions and prove or disprove your choice in B using methods presented in class.

AIC and BIC Comparison:

Normal Distribution: AIC = -1731.60, BIC = -1721.78

T-Distribution: AIC = -1731.53, BIC = -1716.81

Based on AIC, Normal Distribution fits better.

Based on BIC, Normal Distribution fits better.

Problem 2

A. Calculate the pairwise covariance matrix of the data.

Covariance Matrix:

	x1	x2	x3	x4	x5
x1	1.470484	1.454214	0.877269	1.903226	1.444361
x2	1.454214	1.252078	0.539548	1.621918	1.237877
x3	0.877269	0.539548	1.272425	1.171959	1.091912
x4	1.903226	1.621918	1.171959	1.814469	1.589729
x5	1.444361	1.237877	1.091912	1.589729	1.396186

B. Is the Matrix at least positive semi-definite? Why?

No, the covariance matrix is not positive semi-definite because not all eigenvalues are greater than or equal to zero.

C. If not, find the nearest positive semi-definite matrix using Higham’s method and the near-psd method of Rebenato and Jackel.

Nearest PSD Matrix (Higham):

[[1.61513295 1.44196041 0.89714421 1.78042572 1.43379434]

[1.44196041 1.34696791 0.58508636 1.55455192 1.21140918]

[0.89714421 0.58508636 1.29891579 1.11595578 1.07669233]

[1.78042572 1.55455192 1.11595578 1.98316489 1.62137332]

[1.43379434 1.21140918 1.07669233 1.62137332 1.40493616]]

Nearest PSD Matrix (rebonato_jackel):

[[1.47048437 1.32652813 0.84725458 1.62496988 1.36381777]

[1.32652813 1.25207795 0.55831928 1.43363198 1.16431726]

[0.84725458 0.55831928 1.272425 1.05649629 1.06233291]

[1.62496988 1.43363198 1.05649629 1.81446921 1.54604392]

[1.36381777 1.16431726 1.06233291 1.54604392 1.39618646]]

D. Calculate the covariance matrix using only overlapping data.

Covariance Matrix Using Overlapping Data:

	x1	x2	x3	x4	x5
x1	0.418604	0.394054	0.424457	0.416382	0.434287
x2	0.394054	0.396786	0.409343	0.398401	0.422631
x3	0.424457	0.409343	0.441360	0.428441	0.448957
x4	0.416382	0.398401	0.428441	0.437274	0.440167
x5	0.434287	0.422631	0.448957	0.440167	0.466272

E. Compare the results of the covariance matrices in C and D. Explain the differences.

The nearest PSD matrix Higham and rebonato_jackel are closer to the original theoretical covariance structure because they adjusts the matrix mathematically to fix

any issues. Higham's method sets negative eigenvalues to zero, resulting in a matrix with reduced off-diagonal values. And, the Rebonato-Jackel method better preserves the original structure by rescaling and keeping the off-diagonal values closer to the generating process (0.99).On the other hand, the covariance matrix from overlapping data shows the limits of the dataset since there isn't enough data to make an accurate estimate.

Problem 3

A. Fit a multivariate normal to the data.

Mean vector: [0.04600157 0.09991502]

Covariance matrix:

[[0.0101622 0.00492354]

[0.00492354 0.02028441]]

B. Given that fit, what is the distribution of X2 given X1=0.6. Use the 2 methods described in class.

Method 1: Conditional distribution formula

Conditional mean of X2 given X1=0.6: 0.3683249958609775

Conditional variance of X2 given X1=0.6: 0.017898969645087522

Method 2: Using the OLS Method

Conditional mean of X2 given X1=0.6 using OLS: 0.3683249958609775

Estimated variance of X2 given X1 using OLS: 0.017881070675442437

Slope (beta_1): 0.48449591027977496, Intercept: 0.07762744969311251

C. Given the properties of the Cholesky Root, create a simulation that proves your

distribution of X2 | X1=0.6 is correct.

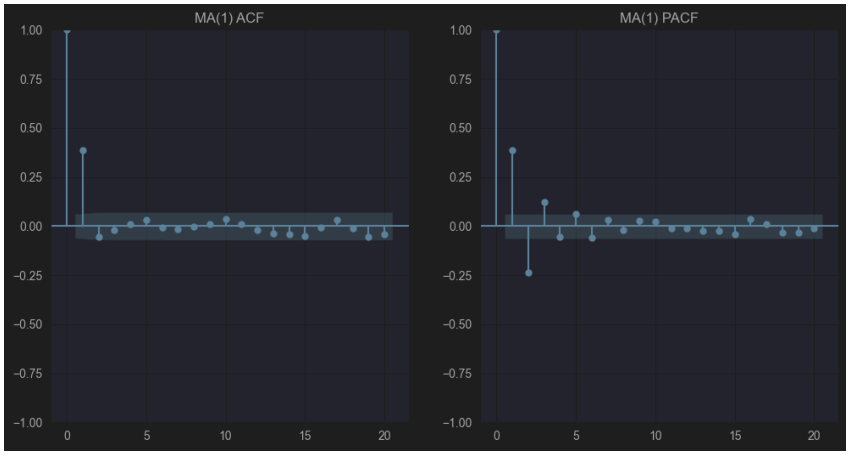
Simulated mean of X2 | X1=0.6: 0.34517989612039235

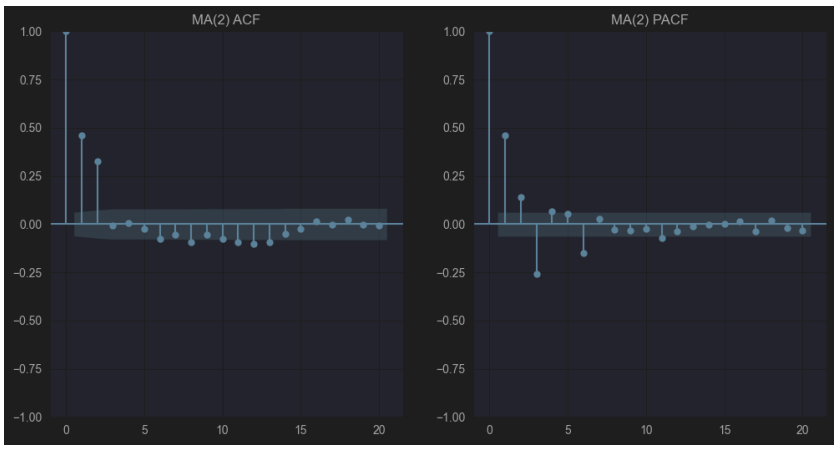
Simulated variance of X2 | X1=0.6: 0.015088302683434123

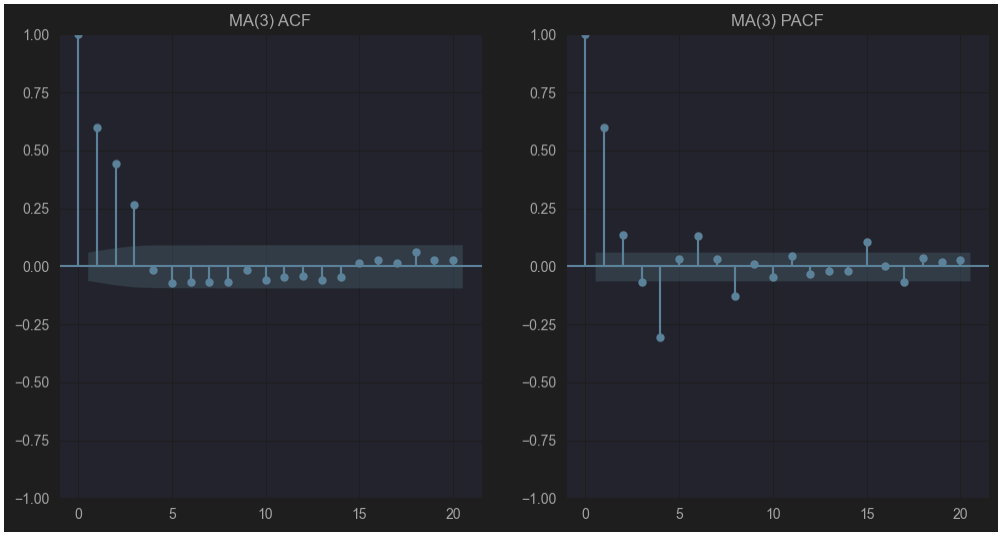
The simulated mean (0.3452) is close to the analytical mean (0.3683) but slightly lower. The simulated variance (0.0151) is also close to the analytical variance (0.0179) but slightly underestimated. The simulated mean and variance are slightly underestimated, likely due to the smaller sample size near X_1 = 0.6 and the threshold-based filtering (0.05).

Problem 4

A. Simulate an MA(1), MA(2), and MA(3) process and graph the ACF and PACF of each. What do you notice?

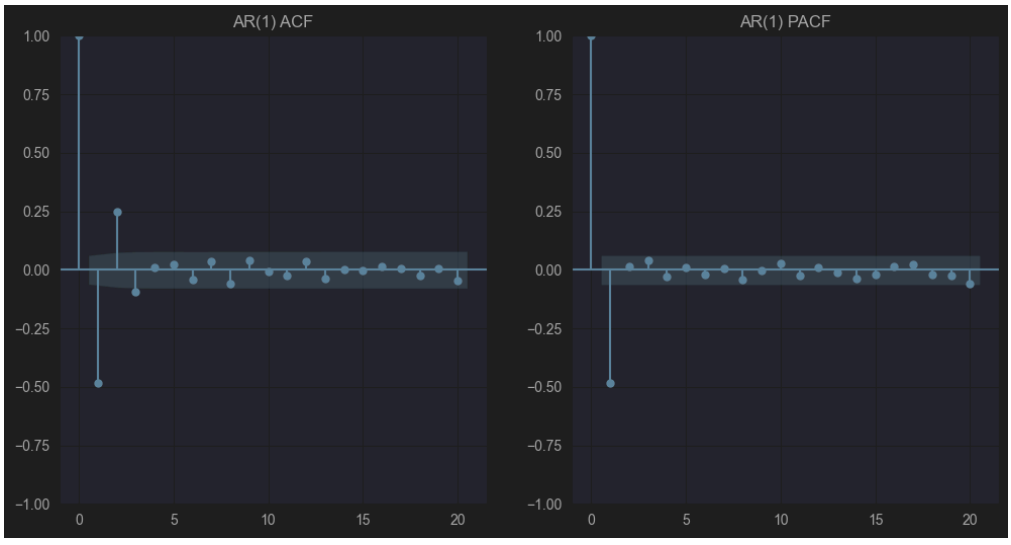


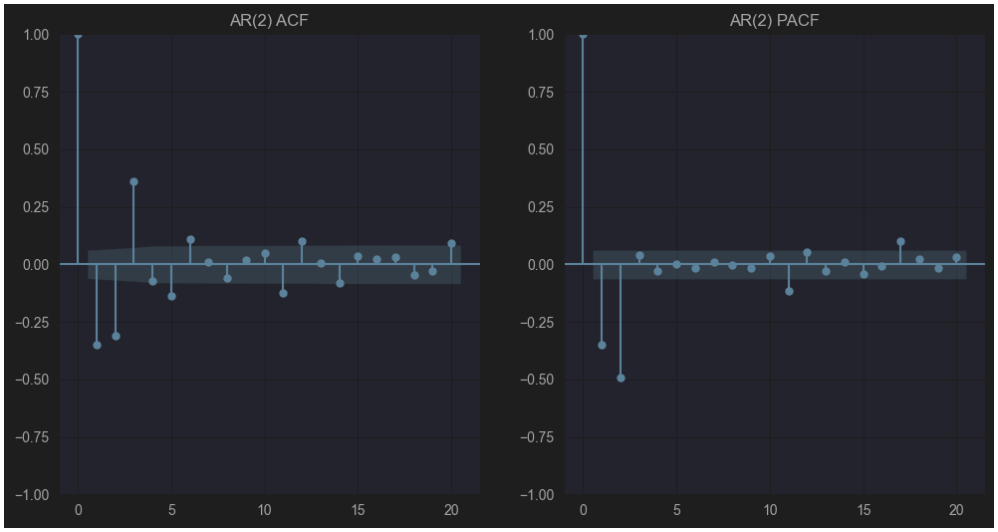


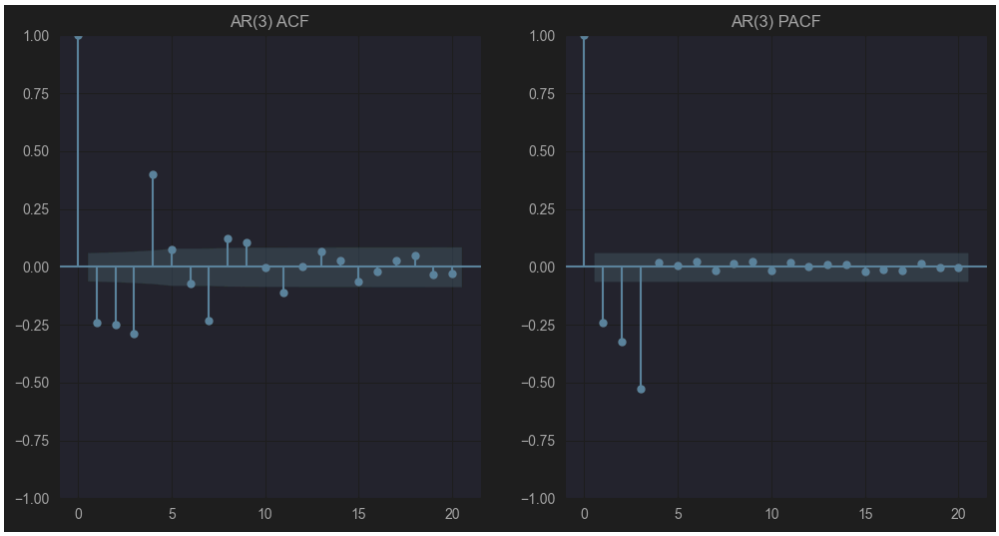


From the graphs, we could notice that the ACF truncates after order q and the PACF gradually decays, which satisfy the ACF and PACF properties of the MA process. MA(1): The ACF shows a sharp decay after lag 1, with a significant spike at lag 1, while the PACF decays quickly. MA(2): The ACF shows a more complex structure with significant spikes up to lag 2, while the PACF decays after lag 1. MA(3): Similar to MA(2), the ACF shows significant lags up to 3, and the PACF shows a quick decay after lag 1.

B. Simulate an AR(1), AR(2), and AR(3) process and graph the ACF and PACF of each. What do you notice?







The PACF helps identify the order of the autoregressive model by showing significant lags, while the ACF reveals the autocorrelation behavior.For AR(1), the ACF gradually decays, while the PACF shows a significant spike at lag 1, indicating its dependence on past direct values. AR(2): The ACF shows a more complex structure with significant correlations up to lag 2. The PACF shows a spike at lag 2. AR(3): Similar to the AR(2) case, the ACF reflects more complexity with significant lags, and the PACF has a spike at lag 3, indicating AR(3) could fit.

C. Examine the data in problem4.csv. What AR/MA process would you use to model the data? Why?

The PACF truncates after lag 3, while the ACF gradually decays. So the AR(3) model should be chosen.

D. Fit the model of your choice in C along with other AR/MA models. Compare the AICc of each. What is the best fit?

	AR Order	MA Order	AIC
0	0	0	-1162.397518
1	0	1	-1508.927033
2	0	2	-1559.250932
3	0	3	-1645.132969
4	1	0	-1669.089267
5	1	1	-1723.454539
6	1	2	-1728.894414
7	1	3	-1744.140141
8	2	0	-1696.091685
9	2	1	-1725.465935

10 2 2 -1740.046363

11 2 3 -1742.580430

12 3 0 -1746.281721

13 3 1 -1744.309922

14 3 2 -1742.372427

15 3 3 -1740.525912

Best model: AR(3.0), MA(0.0) with AIC = -1746.2817209033037

Based on the comparison of AICc values, the ARMA(3,0) model provides the best fit for data, with the lowest AICc value of -1746.28.

Problem 5

A. Create a routine for calculating an exponentially weighted covariance matrix. If you have a package that calculates it for you, verify it produces the expected results from the testdata folder.

Exponentially Weighted Covariance Matrix with 0.97 lambda:

[[7.22898672e-05 5.42719158e-05 1.25794884e-04 ... 6.02118473e-05

1.28030564e-04 5.23812959e-05]

[5.42719158e-05 1.39649533e-04 4.24840975e-05 ... 6.11708118e-05

8.50356109e-05 3.75287951e-05]

[1.25794884e-04 4.24840975e-05 6.69825413e-04 ... 1.92295240e-05

3.26380029e-04 4.83209496e-05]

...

[6.02118473e-05 6.11708118e-05 1.92295240e-05 ... 2.53803118e-04

8.77734377e-05 8.71821118e-05]

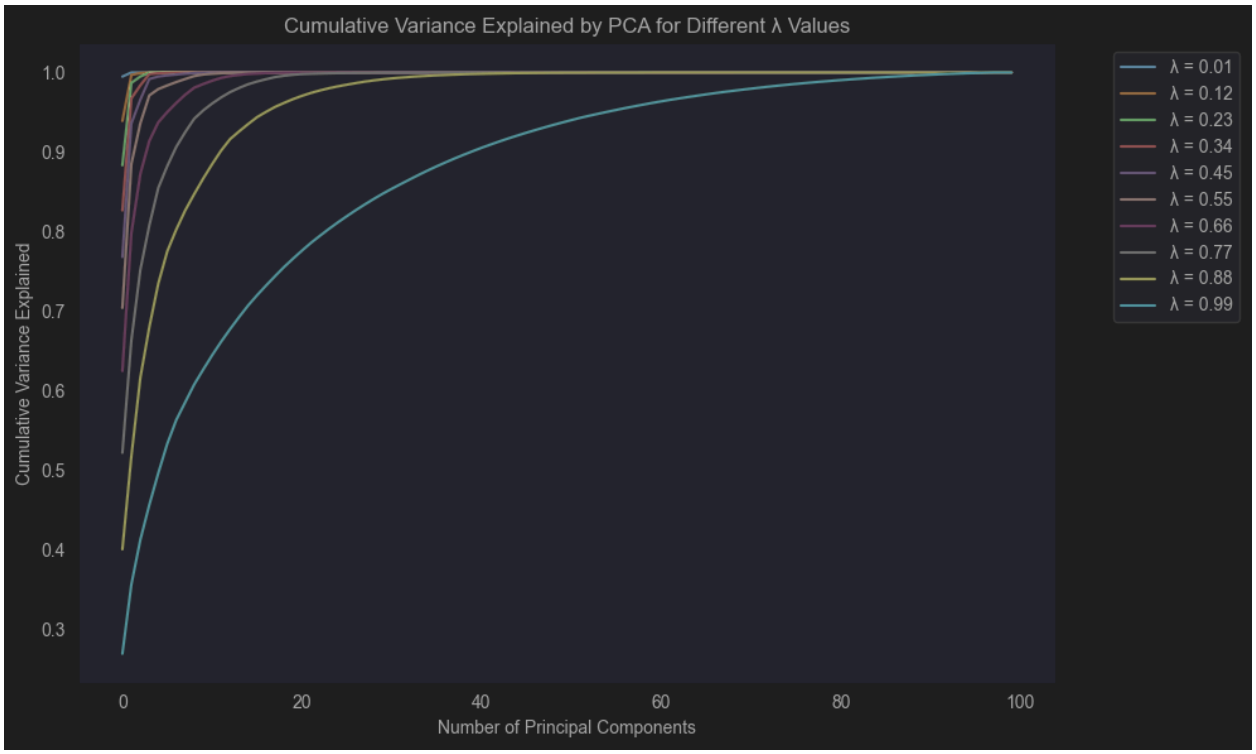
[1.28030564e-04 8.50356109e-05 3.26380029e-04 ... 8.77734377e-05

7.41543403e-04 7.25493470e-05]

[5.23812959e-05 3.75287951e-05 4.83209496e-05 ... 8.71821118e-05

7.25493470e-05 1.53399334e-04]]

B. Vary λ. Use PCA and plot the cumulative variance explained of λ in (0,1) by each eigenvalue for each λ chosen.



C. What does this tell us about the values of λ and the effect it has on the covariance matrix?

λ controls the weight given to recent versus historical data.

A higher λ (e.g., 0.99) gives more weight to recent data, resulting in a smoother covariance matrix that is less sensitive to recent fluctuations.

A lower λ (e.g., 0.01) gives more weight to historical data, making the covariance matrix more sensitive to recent changes in the data.

Each λ value affects the computation of the covariance matrix and thus the proportion of variance explained by the principal components. Larger values of λ result in a more even distribution of variance across principal components, requiring more components to explain the same amount of variance. In contrast, smaller values of λ concentrate variance in fewer principal components, allowing the first few components to explain a larger proportion of the total variance.

Problem 6

Implement a multivariate normal simulation using the Cholesky root of a covariance matrix.

Implement a multivariate normal simulation using PCA with percent explained as an input.

A. Simulate 10,000 draws using the Cholesky Root method.

Cholesky Simulation Shape: (10000, 500)

B. Simulate 10,000 draws using PCA with 75% variance

PCA Simulation Shape: (10000, 500)

C. Take the covariance of each simulation. Compare the Frobenius norm of these matrices to the original covariance matrix. What do you notice?

Frobenius Norm (Cholesky): 0.02179562160305022

Frobenius Norm (PCA): 0.08316923525944199

The Frobenius norm of the Cholesky method is small (0.0218):

This indicates that the covariance matrix simulated by the Cholesky method is very close to the original covariance matrix.

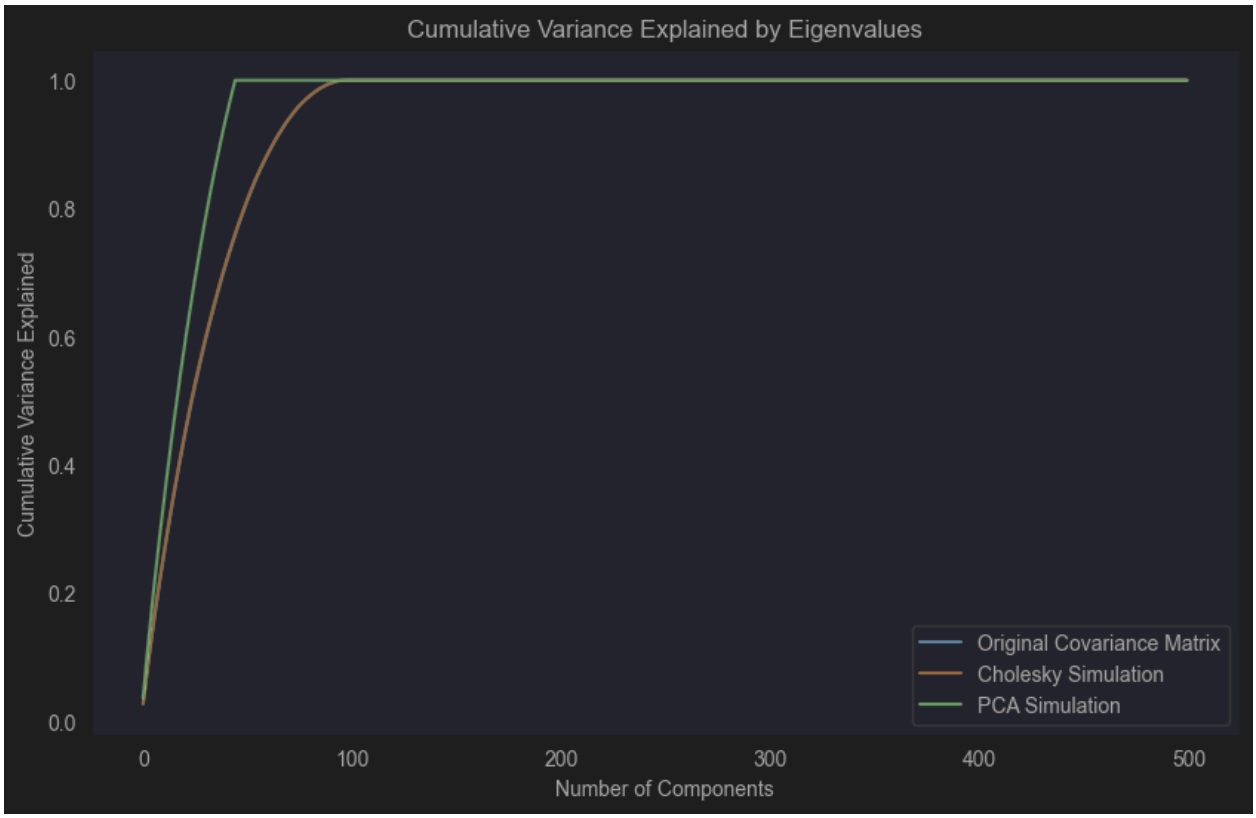
The Cholesky method retains the complete structure of the original covariance matrix and therefore the simulation results are more accurate.

The Frobenius norm of the PCA method is larger (0.0832):

This indicates that the covariance matrix simulated by the PCA method is more different from the original covariance matrix.

The PCA method loses some information by dimensionality reduction (only 75% of the variance is retained), so the simulation results are more different from the original matrix.

D. Compare the cumulative variance explained by each eigenvalue of the 2 simulated covariance matrices along with the input matrix. What do you notice?



The Cholesky Simulation (orange line) is very close to the original (blue line, which is covered by orange line), indicating that the Cholesky method does a great job of preserving the structure of the original covariance matrix, maintaining a very similar variance explanation curve.

The PCA Simulation (green line) explains only 75% of the variance by design, so the curve flattens out earlier, showing a more pronounced drop in variance explained after the first few components. PCA's dimensionality reduction leads to some loss of information, which is evident in the variance explained by each component.

E. Compare the time it took to run both simulations.

Cholesky Simulation Time: 0.11326003074645996

PCA Simulation Time: 0.015511035919189453

Cholesky Method took significantly more time than the PCA Method. This difference in time is expected because the Cholesky method requires performing matrix decompositions (Cholesky decomposition) and generating correlated random samples, which can be computationally expensive, especially as the number of components or the matrix size increases. The PCA method typically involves dimensionality reduction, which is less computationally demanding.

F . Discuss the tradeoffs between the two methods.

The Cholesky simulation is more accurate in preserving the structure of the original covariance matrix, but it requires more computational effort.

The PCA simulation is faster but loses some of the covariance structure due to dimensionality reduction, leading to less accurate results in terms of matching the original matrix.