Keese Phillips

FINTECH 545

8 February 2025

Project 1

Problem 1

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

- Mean 0.05019795790476916
- Variance 0.010332476407479588
- Skew 0.12044471191944014
- Kurtosis 0.2229270674503816

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

- I would use the normal distribution; with 1000 samples I rely on the central limit theorem that the normal distribution is a reasonable approximation of the true distribution.

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

- Although the data appears to be roughly normally distributed the t-distribution has ~28.7 degrees of freedom, because the degrees of freedom < 30, the t-distribution is the more conservative fit to model the data.

Problem 2

A. Calculate the pairwise covariance matrix of the data.

	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 matrix is not positive semi-definite because the matrix has two negative eigenvalues. The eigenvalues are: [6.78670573 0.83443367 -0.31024286 0.02797828 -0.13323183]

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

- Higham's method:

	x1	x2	x3	x4	x5
x1	1.470484	1.332524	0.886817	1.6287	1.400961
x2	1.332524	1.252078	0.622298	1.45423	1.217183
x3	0.886817	0.622298	1.272425	1.070369	1.057808
x4	1.6287	1.45423	1.070369	1.814469	1.577137
x5	1.400961	1.217183	1.057808	1.577137	1.396186

- Rebenato and Jackel:

	x1	x2	x3	x4	x5
x1	1.470484	1.327009	0.842583	1.624464	1.364833
x2	1.327009	1.252078	0.555421	1.433109	1.165906
x3	0.842583	0.555421	1.272425	1.052789	1.060424
x4	1.624464	1.433109	1.052789	1.814469	1.544993
x5	1.364833	1.165906	1.060424	1.544993	1.396186

D. Calculate the covariance matrix using only overlapping data.

	x1	x2	x3	x4	x5
x1	0.132972	0.144205	-0.021	0.087688	0.064467
x2	0.144205	0.170188	-0.03952	0.112221	0.072822
x3	-0.021	-0.03952	0.084718	-0.03176	0.003825
x4	0.087688	0.112221	-0.03176	0.079924	0.046841
x5	0.064467	0.072822	0.003825	0.046841	0.036917

E. Compare the results of the covariance matrices in C and D. Explain the differences. Note: the generating process is a covariance matrix with 1 on the diagonals and 0.99 elsewhere.

- Higham's and Rebenato and Jackel covariance matrices retain relatively high diagonals and values close to 0.99 elsewhere. This seems to align closely with the intended generating process with 1 on the diagonals and 0.99 elsewhere. Adjustments for positive semi-definiteness slightly reduce covariances but preserve the high-correlation structure. The covariance matrix using only overlapping data shows low diagonals and near-zero or negative values elsewhere. This covariance deviates from the generating process, indicating the overlapping data might lack sufficient observations and can cause distortions.

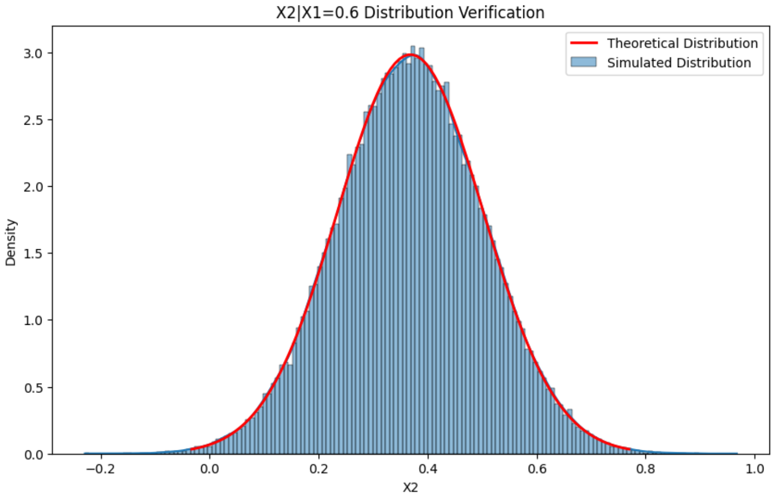
Problem 3

A. Fit a multivariate normal to the data.

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

- Conditional Distributions:
  - Mean: 0.3683
  - Variance: 0.0179
- OLS:
  - Slope: 0.4845
  - Intercept: 0.0776
  - Variance: 0.0179

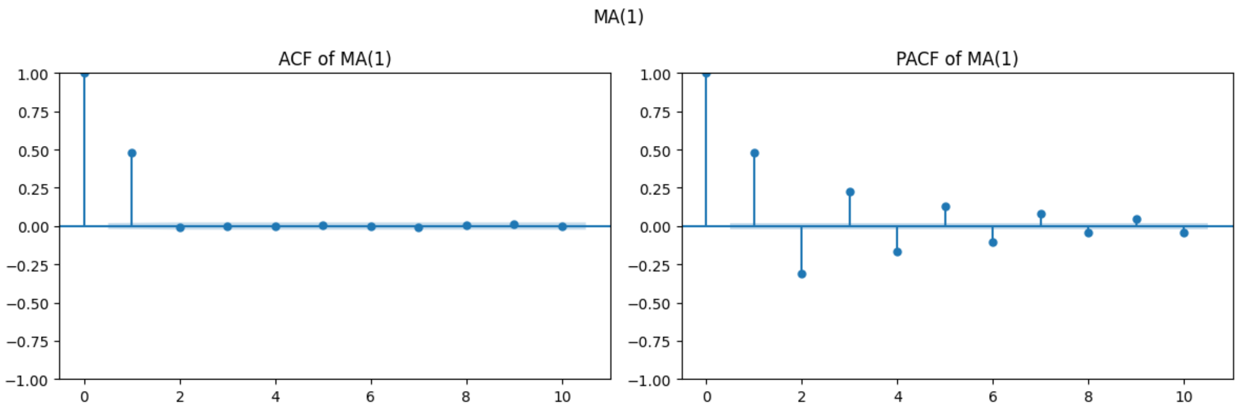
C. Given the properties of the Cholesky Root, create a simulation that proves your distribution of X2 | X1=0.6 is correct.

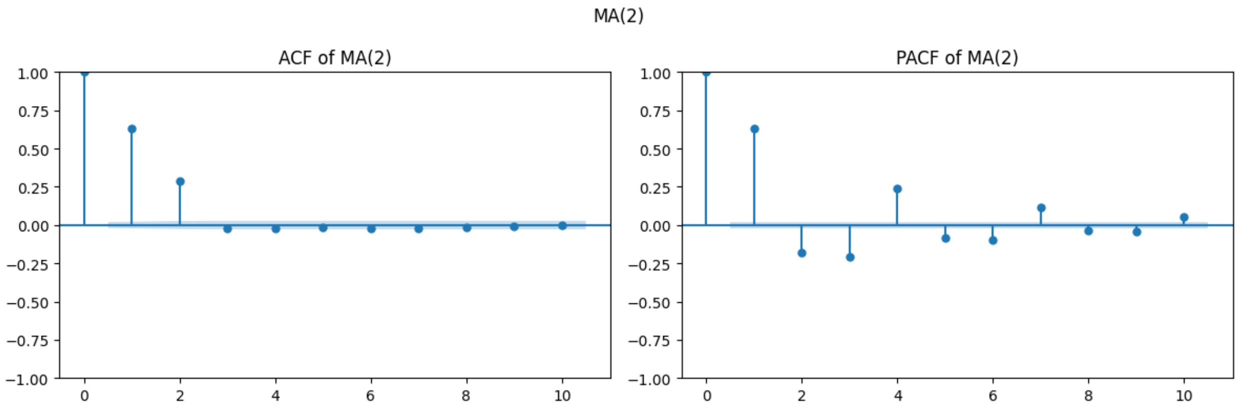


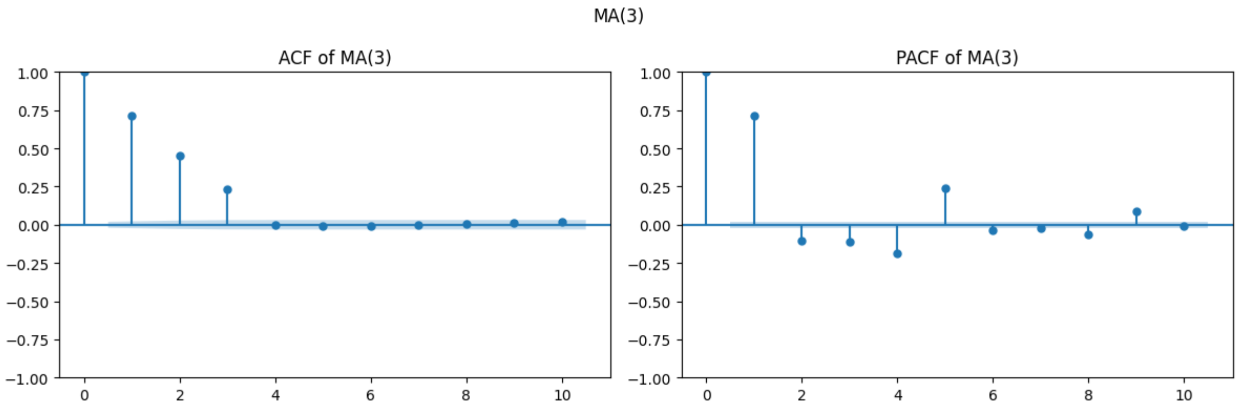
- Simulated Mean: 0.3685
- Simulated Variance: 0.0179

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?

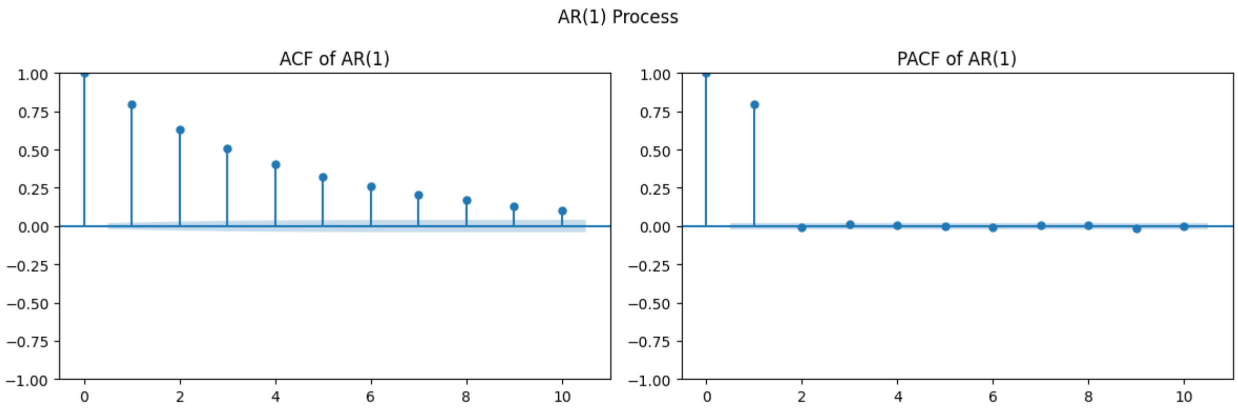


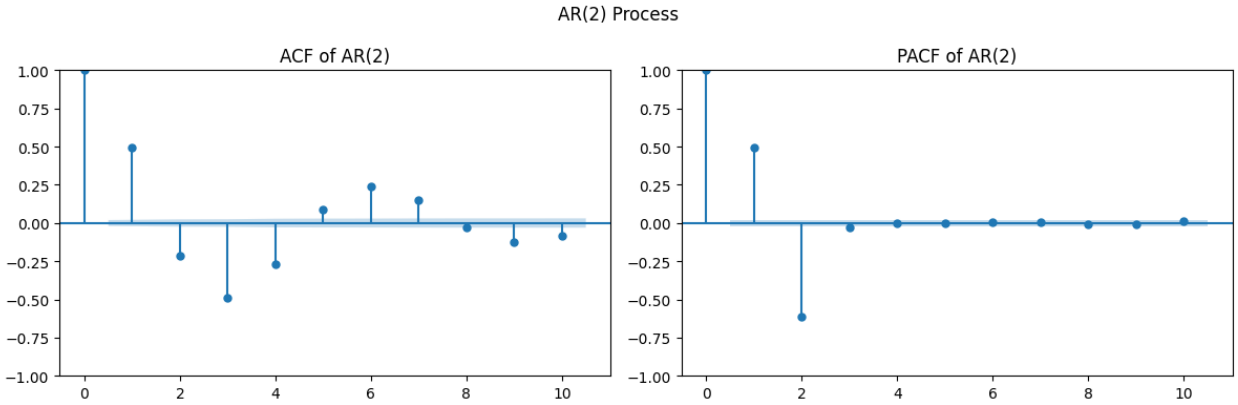


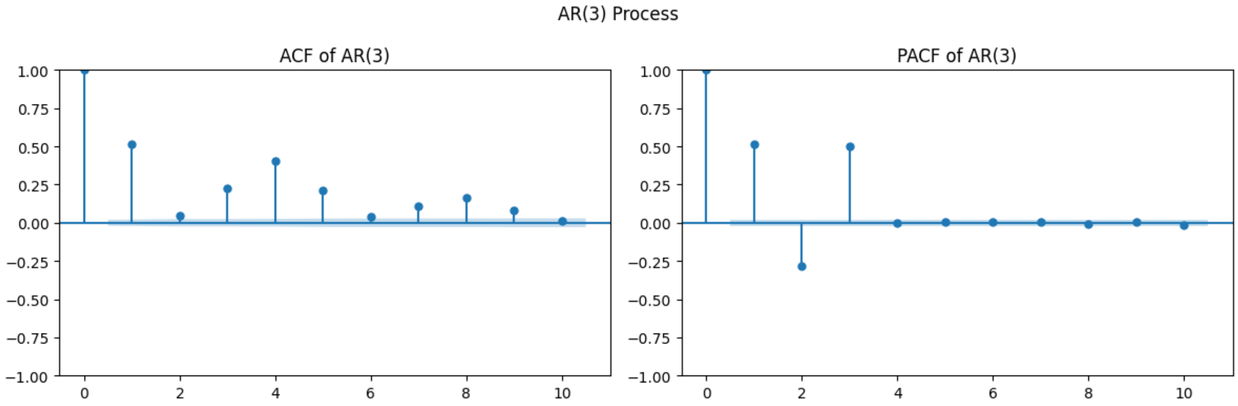


- The ACF graph shows a significant spike up to lag q. After lag q, the ACF drops to near-zero, followed by insignificant correlations. The PACF decays gradually, contrasted with the ACF's sharp cutoff. The ACF indicates the lag q for the MA model.

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

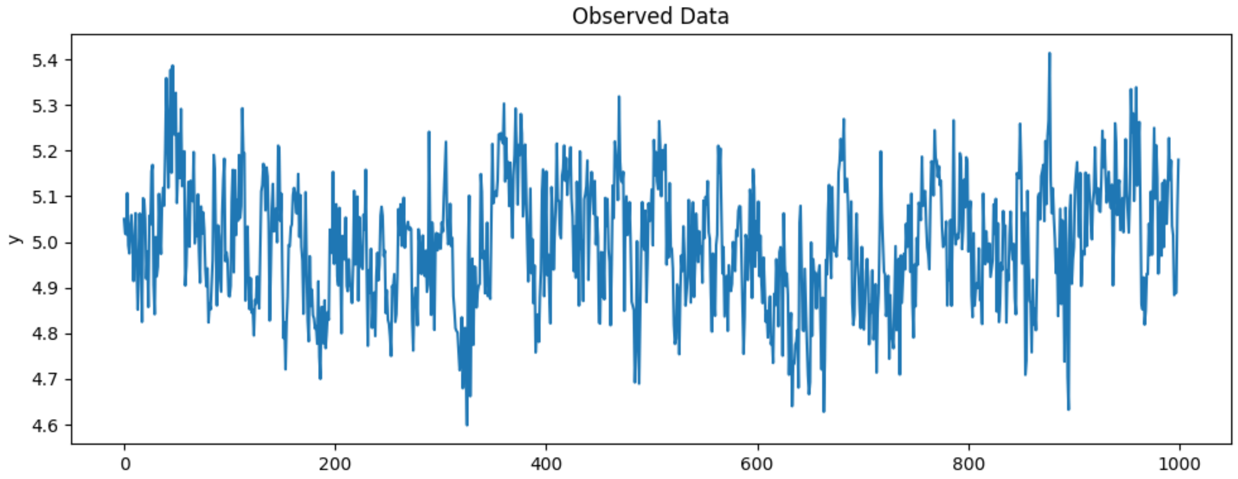


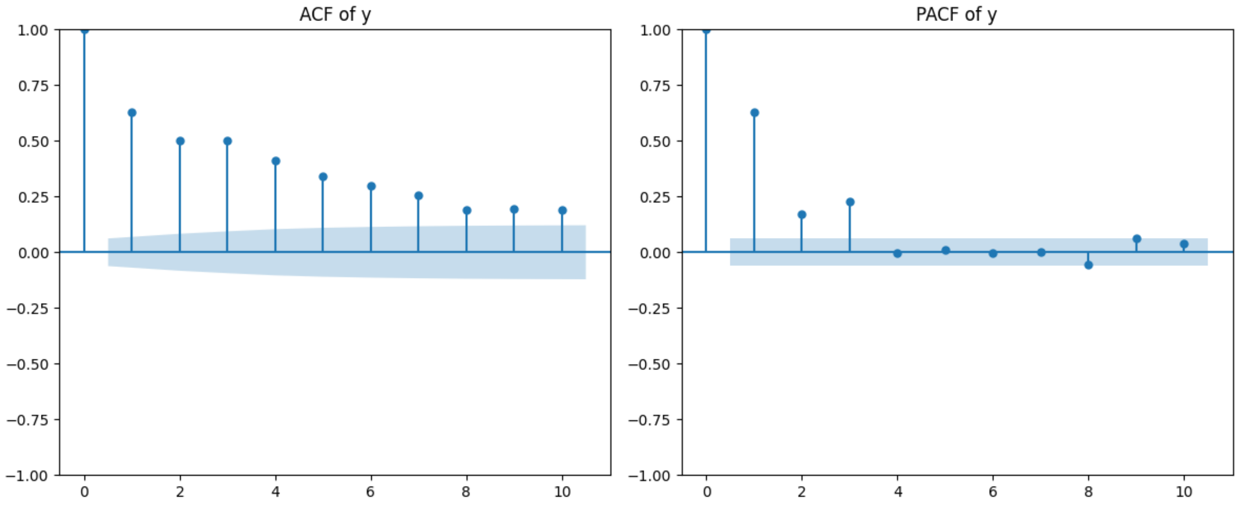




- The PACF shows a significant spike up to lag p. After lag p, the PACF drops to near-zero, followed by insignificant correlations. The ACF decays gradually, contrasted with the PACF's sharp cutoff. The PACF indicates the lag p for the AR model.

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





- The ACF graph seems to decay gradually compared with the PACF sharp cutoffs. Due to this behavior a AR model might be a good fit. Specifically an AR(3) model might be a good fit for the data because the PACF indicates a sharp cutoff after the 3rd lag..

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?

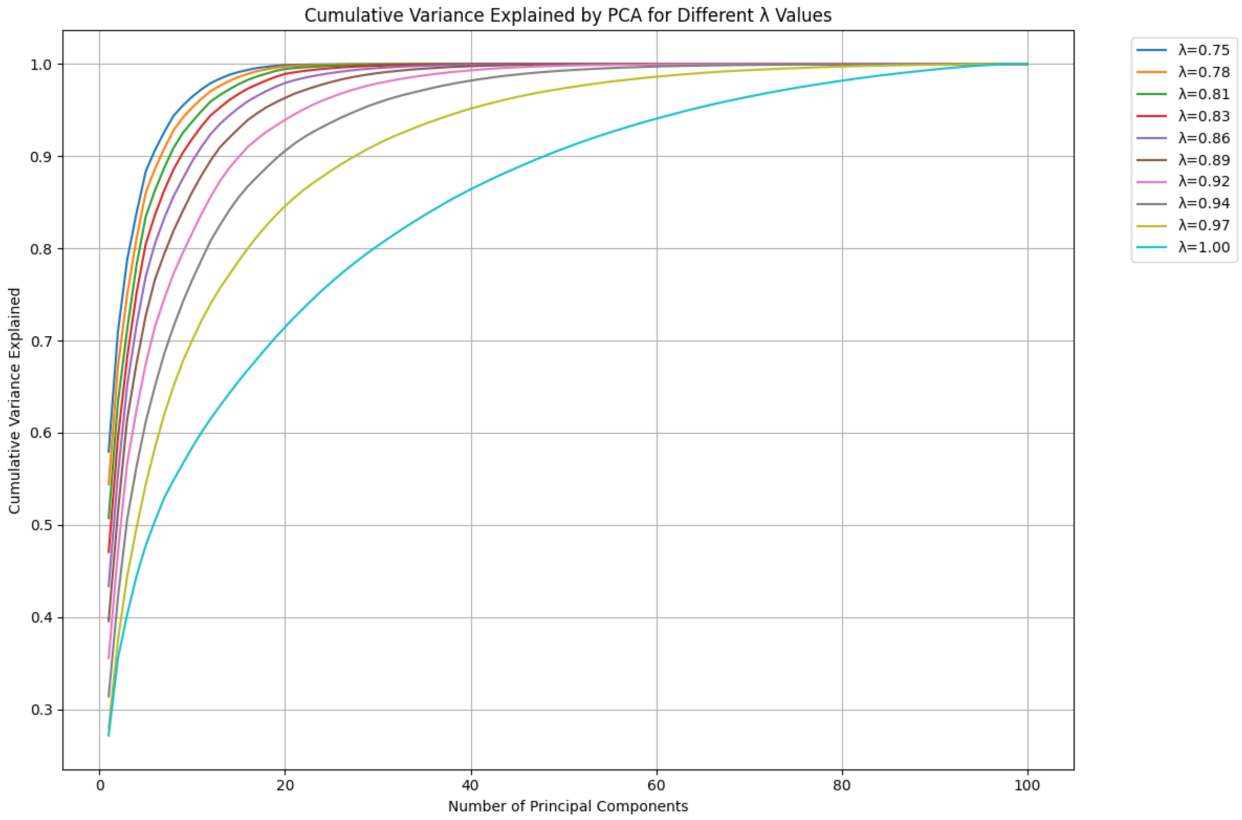
- The AR(3) model was the best fit, this is followed by several more higher order AR models before the MA models.

1. AR(3): -1746.26
2. AR(4): -1744.27
3. AR(2): -1696.08
4. MA(4): -1677.54
5. AR(1): -1669.09
6. MA(3): -1645.11
7. MA(2): -1559.24
8. MA(1): -1508.92

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.

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?

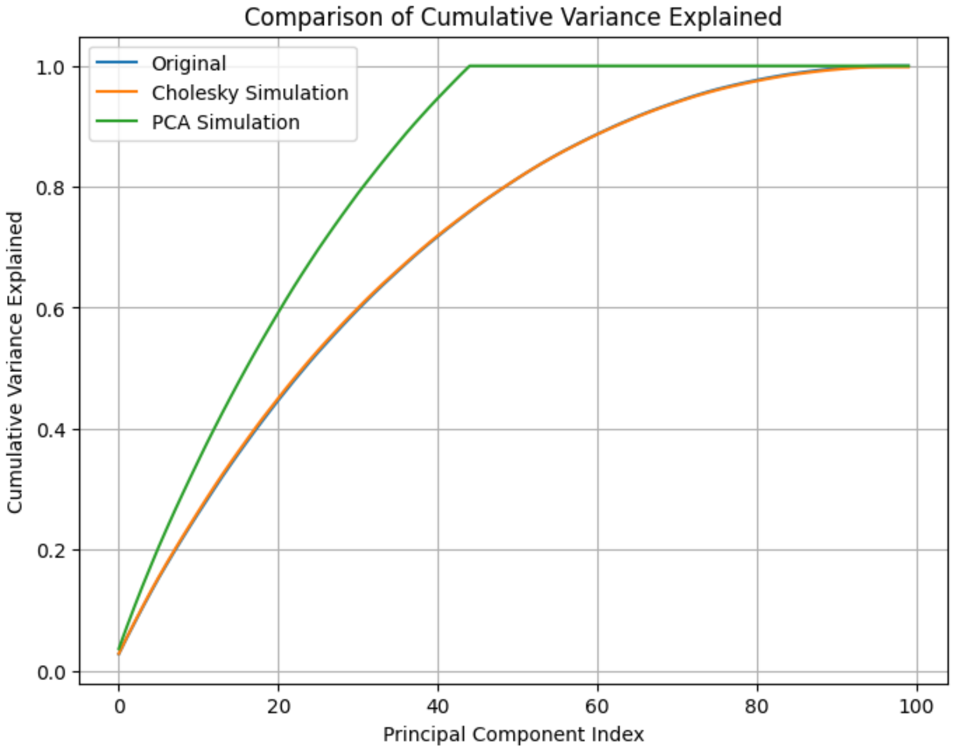
- The lower the lambda value the faster the cumulative weights approach 1. This means that the faster the cumulative weights approach 1, the less amount of time is actually included in the covariance matrix. The lower the lambda the lower the amount of time that will be included in the covariance matrix because the weight values past a certain point approach 0 and will not provide much towards computing the covariance matrix. Therefore, smaller lambda values concentrate on recent conditions while larger lambda values incorporate more history.

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.
B. Simulate 10,000 draws using PCA with 75% variance
C. Take the covariance of each simulation. Compare the Frobenius norm of these matrices to the original covariance matrix. What do you notice?
   - Cholesky Root: 0.0209
   - PCA: 0.0833
   - The Cholesky method produces a covariance matrix nearly identical to the original. The small norm confirms that the simulation accurately replicates the covariance structure, with minor discrepancies due to finite-sample noise. The PCA method introduces a larger deviation from the original covariance matrix. This is expected because PCA truncates components beyond the 75% variance threshold, sacrificing covariance fidelity for dimensionality reduction.
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 cumulative variance curve closely tracks that of the original covariance matrix. The Cholesky method preserves the eigenvalue structure of the original covariance matrix. This explains how the Cholesky decomposition accurately replicates the covariance structure. By applying a 75% variance threshold, PCA reduces the dimensionality of the data, sacrificing some of the original covariance matrix for a more compact representation.

E. Compare the time it took to run both simulations.
   - Cholesky: 0.2392 seconds
   - PCA: 0.1964 seconds

F. Discuss the tradeoffs between the two methods.
   - Cholesky is ideal for exact covariance reproduction but computationally heavy for large matrices. PCA trades off accuracy for efficiency by compressing data into fewer components, making it suitable for high-dimensional datasets where approximation is acceptable.