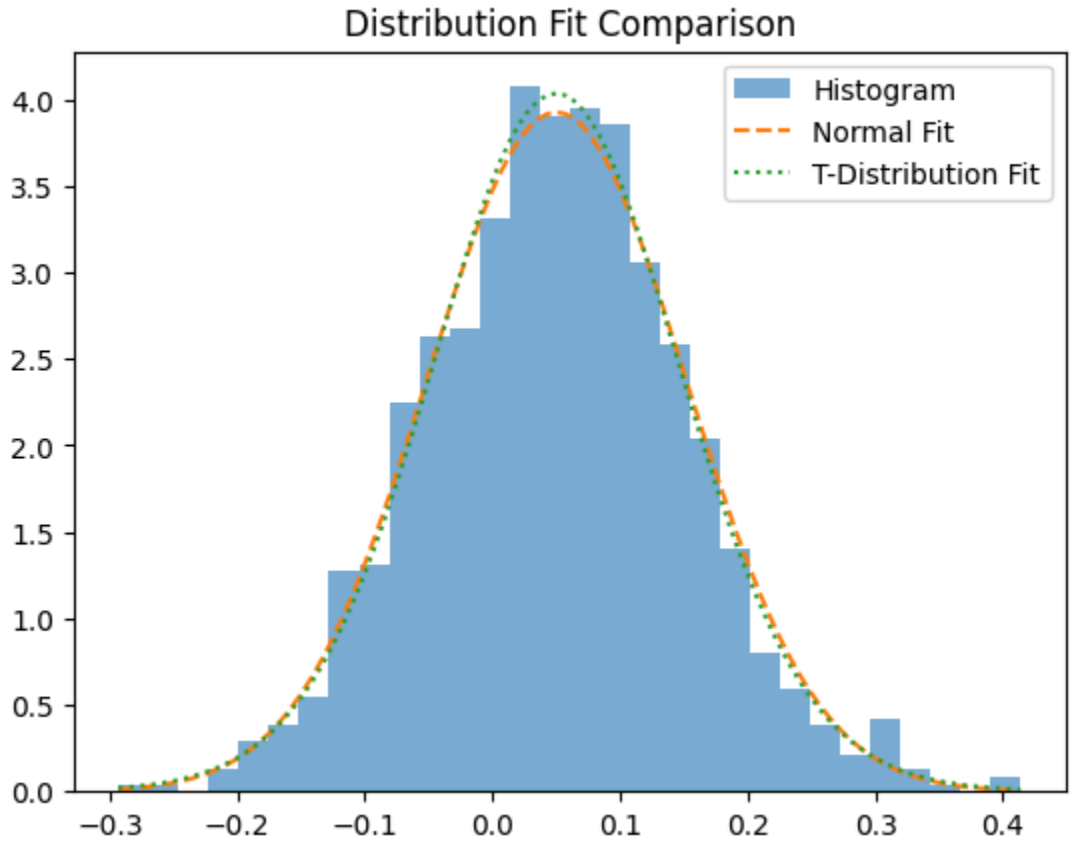
Problem1:

1. *PartA* Result:
    a. Mean: 0.05019795790476916
    b. Variance: 0.010332476407479588
    c. Skewness: 0.1204447119194402
    d. Kurtosis: 0.2229270674503816
2. *Part B:*



    a.
    b. Goodness-of-Fit Test Results:
        i. Normal Distribution: KS Statistic = 0.0128, p-value = 0.9962
        ii. T-Distribution: KS Statistic = 0.0127, p-value = 0.9965
    c. The results of the goodness-of-fit test indicate that both the Normal Distribution and T-Distribution provide an excellent fit for the dataset. The KS statistic for the Normal Distribution is 0.0128 with a p-value of 0.9962, while for the T-Distribution, the KS statistic is 0.0127 with a p-value of 0.9965. Since the p-values for both distributions are significantly higher than any conventional significance level, we fail to reject the null hypothesis that the data comes from either distribution. However, T-distribution is the better fit here because of smaller KS statistics and higher p-value.
3. *PartC:*
    a. Model Comparison using AIC and BIC:
        i. Normal Distribution: AIC = -1731.59, BIC = -1721.77

ii. T-Distribution: AIC = -1731.42, BIC = -1716.70
iii. Normal Distribution is the better fit based on AIC and BIC.

Problem 2

1. *PartA result:*
    a. Pairwise Covariance Matrix:
    b.
    c. x1 1.470484 1.454214 0.877269 1.903226 1.444361
    d. x2 1.454214 1.252078 0.539548 1.621918 1.237877
    e. x3 0.877269 0.539548 1.272425 1.171959 1.091912
    f. x4 1.903226 1.621918 1.171959 1.814469 1.589729
    g. x5 1.444361 1.237877 1.091912 1.589729 1.396186
2. *PartB*
    a. No, the matrix is not positive semi-definite because not all its eigenvalues are greater than or equal to zero.
3. *PartC:* Highham & Rebonato-Jackel
    a. Nearest PSD Matrix (Higham):
    b. [[1.61513295 1.44196041 0.89714421 1.78042572 1.43379434]
    c. [1.44196041 1.34696791 0.58508635 1.55455193 1.21140918]
    d. [0.89714421 0.58508635 1.29891578 1.11595578 1.07669234]
    e. [1.78042572 1.55455193 1.11595578 1.98316488 1.62137332]
    f. [1.43379434 1.21140918 1.07669234 1.62137332 1.40493616]]
    g.
    h. Nearest PSD Matrix (Rebonato-Jackel):
    i. [[1.47048437 1.32652813 0.84725458 1.62496988 1.36381777]
    j. [1.32652813 1.25207795 0.55831928 1.43363198 1.16431726]
    k. [0.84725458 0.55831928 1.272425 1.05649629 1.06233291]
    l. [1.62496988 1.43363198 1.05649629 1.81446921 1.54604392]
    m. [1.36381777 1.16431726 1.06233291 1.54604392 1.39618646]]
4. *PartD:*
    a. Covariance Matrix Using Overlapping Data:
    b.
    c. x1 0.418604 0.394054 0.424457 0.416382 0.434287
    d. x2 0.394054 0.396786 0.409343 0.398401 0.422631
    e. x3 0.424457 0.409343 0.441360 0.428441 0.448957
    f. x4 0.416382 0.398401 0.428441 0.437274 0.440167
    g. x5 0.434287 0.422631 0.448957 0.440167 0.466272
5. *PartE:*
    a. The covariance matrices from Part C and Part D exhibit notable differences due to their generation methods. The Part C matrices, derived using Higham's and Rebonato-Jackel methods, are mathematical adjustments of the input matrix to ensure positive semi-definiteness. Higham's method adjusts negative eigenvalues to zero, resulting in a matrix with dampened off-diagonal values, while Rebonato-Jackel preserves the original structure more effectively by

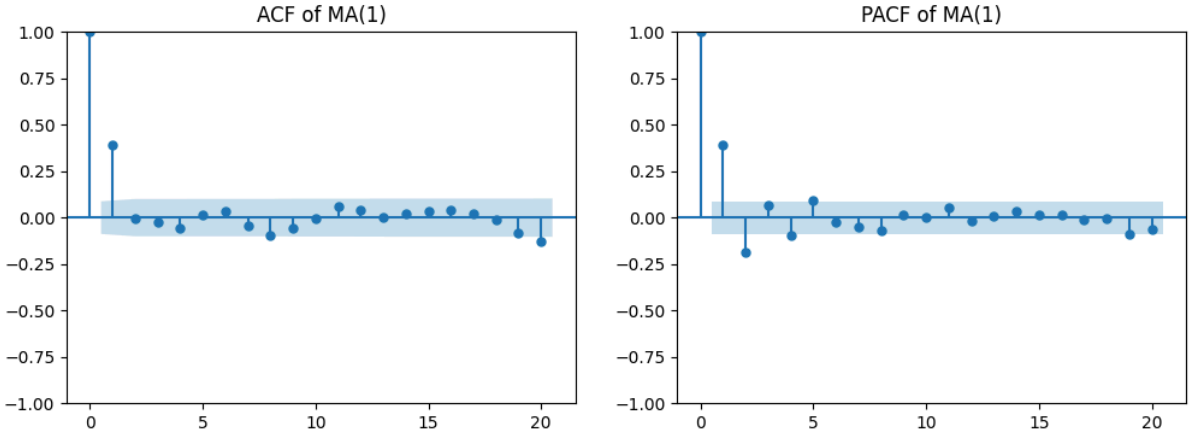
rescaling and maintaining higher off-diagonal values closer to the generating process (0.99). In contrast, the Part D matrix is empirically calculated using only overlapping data, reflecting actual pairwise relationships observed in the dataset. Due to missing or invalid data, the off-diagonal values in Part D are slightly weaker, as incomplete data reduces the strength of correlations. Despite these differences, all matrices retain diagonal elements near 1, reflecting preserved variances. The discrepancies highlight the trade-off between theoretical adjustments (Part C) and data-driven calculations (Part D).

Problem3:

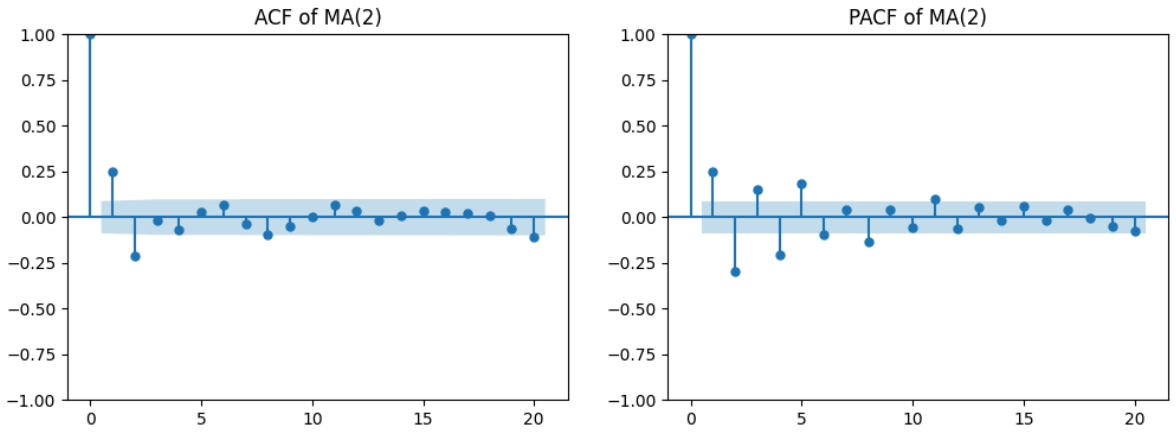
1. *PartA:*
    a. Mean vector: [0.04600157 0.09991502]
    b. Covariance matrix:
    c. [[0.0101622 0.00492354]
    d. [0.00492354 0.02028441]]
2. *PartB:*
    a. Conditional mean of X2 given X1=0.6: 0.3683249958609775
    b. Conditional variance of X2 given X1=0.6: 0.017898969645087522
    c. Conditional mean of X2 given X1=0.6 using OLS: 0.36832499586097767
    d. Estimated variance of X2 given X1 using OLS: 0.017898969645087526
    e. Slope (beta_1): 0.48449591027977523, Intercept: 0.07762744969311253
3. *PartC:*
    a. Simulation Results:
    b. Empirical Mean of X2 | X1=0.6: 0.2091168842544608
    c. Empirical Variance of X2 | X1=0.6: 0.5980409312602568

Problem4

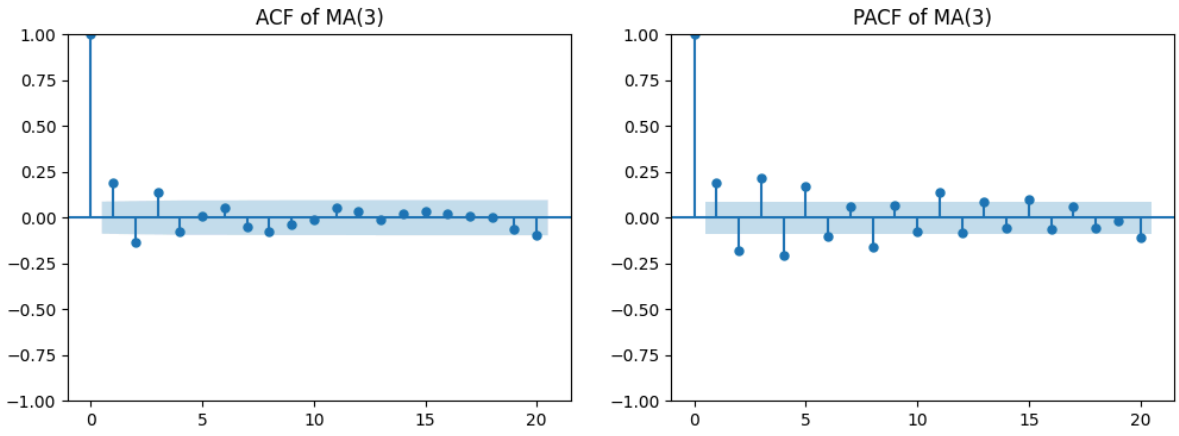
1. *PartA:*
    a.



b.



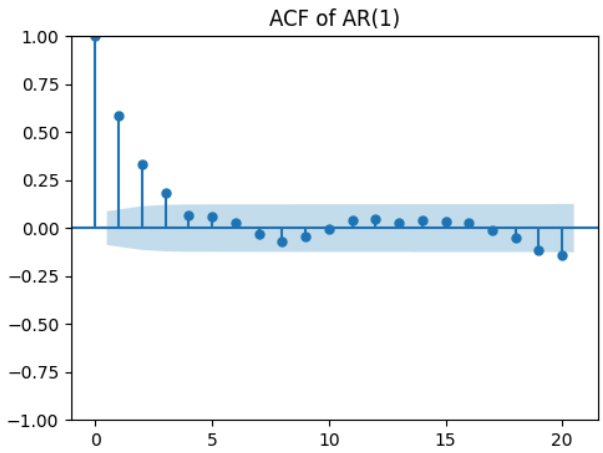
c.

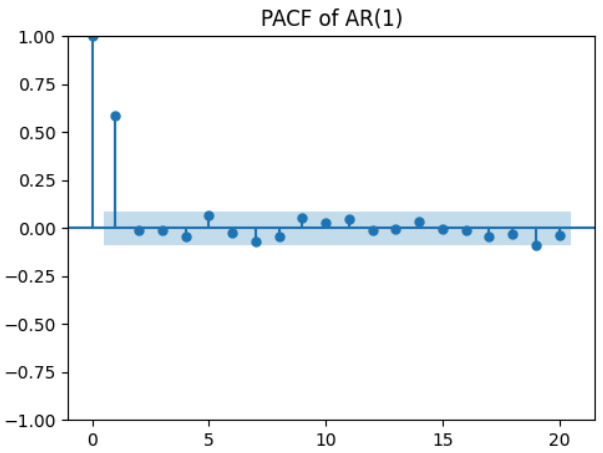


d. From the ACF and PACF graphs of the MA(2) process, we observe that the ACF shows significant spikes at lags 1 and 2, consistent with the moving average order, and quickly diminishes to near zero for higher lags. This indicates that the process is limited to dependencies on the last two time periods, as expected for an MA(2) model. The PACF, on the other hand, displays a strong initial spike at lag 1 but rapidly tapers off, which is characteristic of moving average processes. These observations confirm the finite lag structure of the MA(2) process and align with theoretical expectations for identifying MA processes through ACF and PACF patterns.

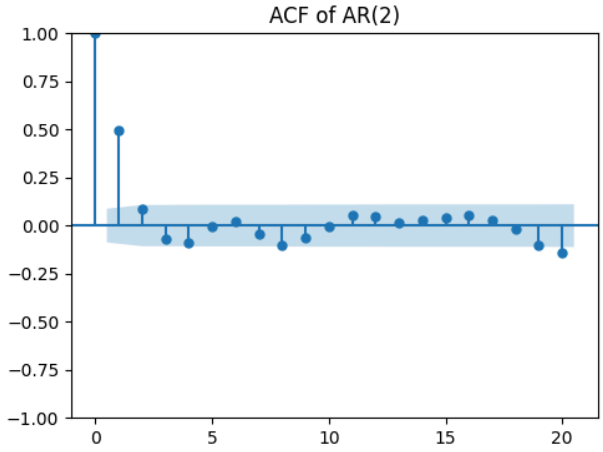
2. *PartB:*

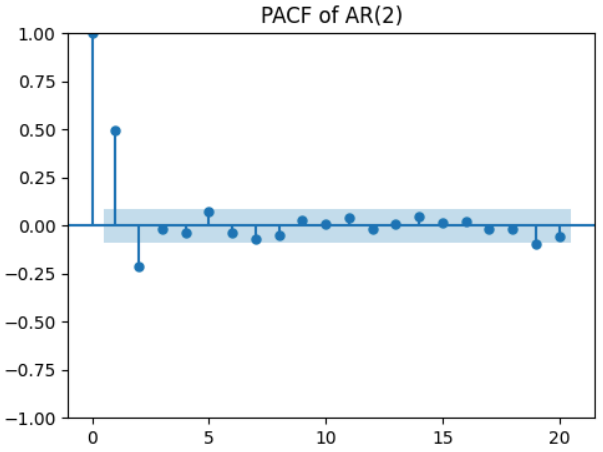
a.



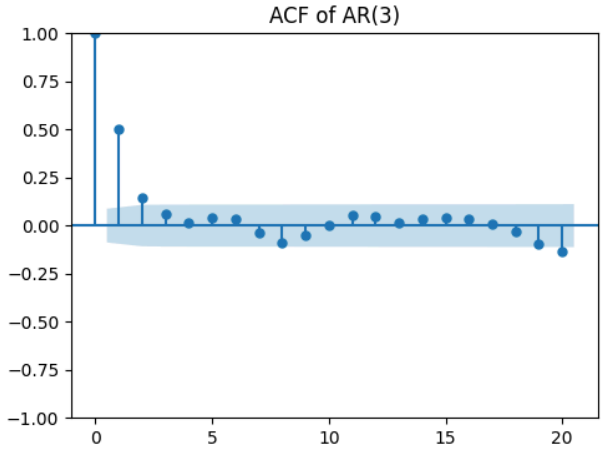


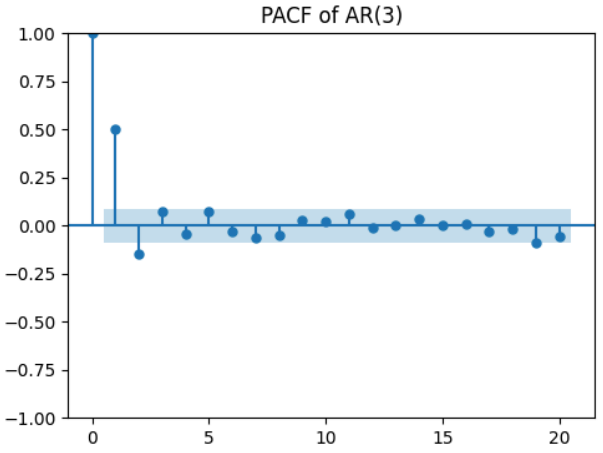
b.





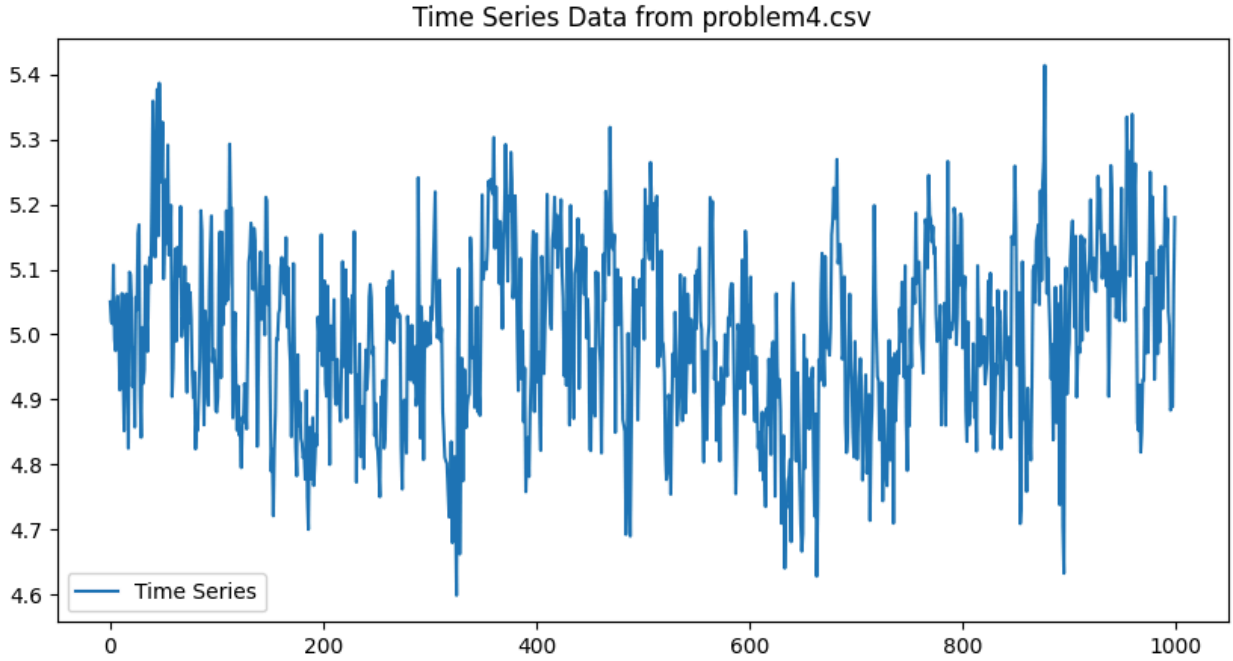
c.





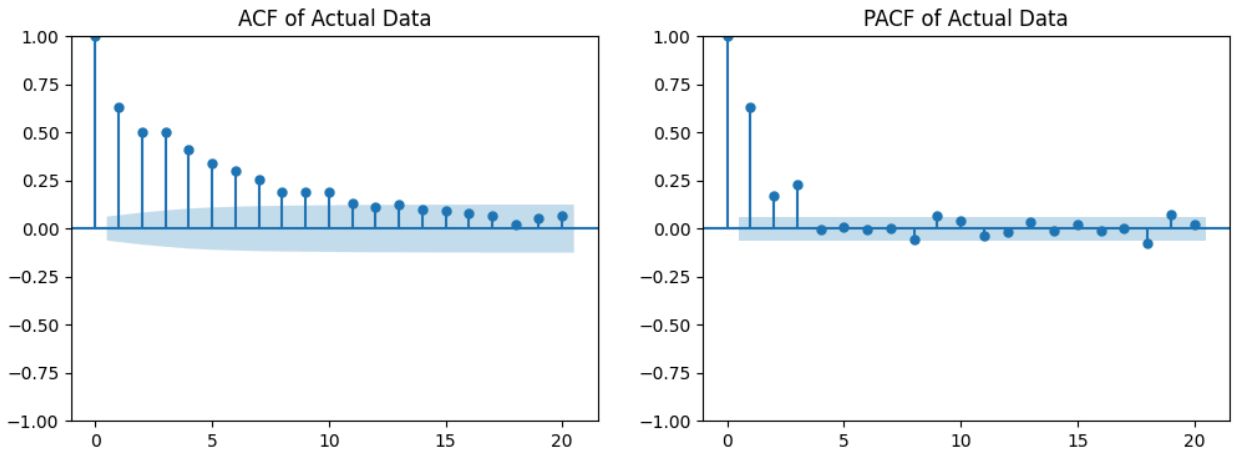
d. From the ACF and PACF graphs of the AR(1), AR(2), and AR(3) processes, we observe distinct patterns that align with the characteristics of autoregressive models. For AR(1), the ACF decays gradually, while the PACF shows a single significant spike at lag 1, indicating dependence on the immediate past value. In AR(2), the ACF decays more slowly, and the PACF has significant spikes at lags 1 and 2, reflecting dependencies on the two most recent observations. Similarly, for AR(3), the ACF decays even more gradually, and the PACF exhibits

significant spikes at lags 1, 2, and 3, consistent with a third-order AR process. These patterns demonstrate how the PACF effectively identifies the order of an AR process, while the ACF reveals the cumulative decay of correlations over time.



3. *PartC:*

a.



b.

c. The AR(3) model is chosen because the Partial Autocorrelation Function (PACF) plot shows significant spikes at lags 1, 2, and 3, while cutting off sharply after lag 3. This indicates that the current value of the time series is directly influenced by the three preceding values, and additional lags do not have significant independent contributions. The gradual decay observed in the Autocorrelation Function (ACF) plot further supports an Auto-Regressive process, as it is

characteristic of AR models. Together, these patterns strongly suggest that an AR(3) model is the most appropriate choice for capturing the underlying structure of the data.

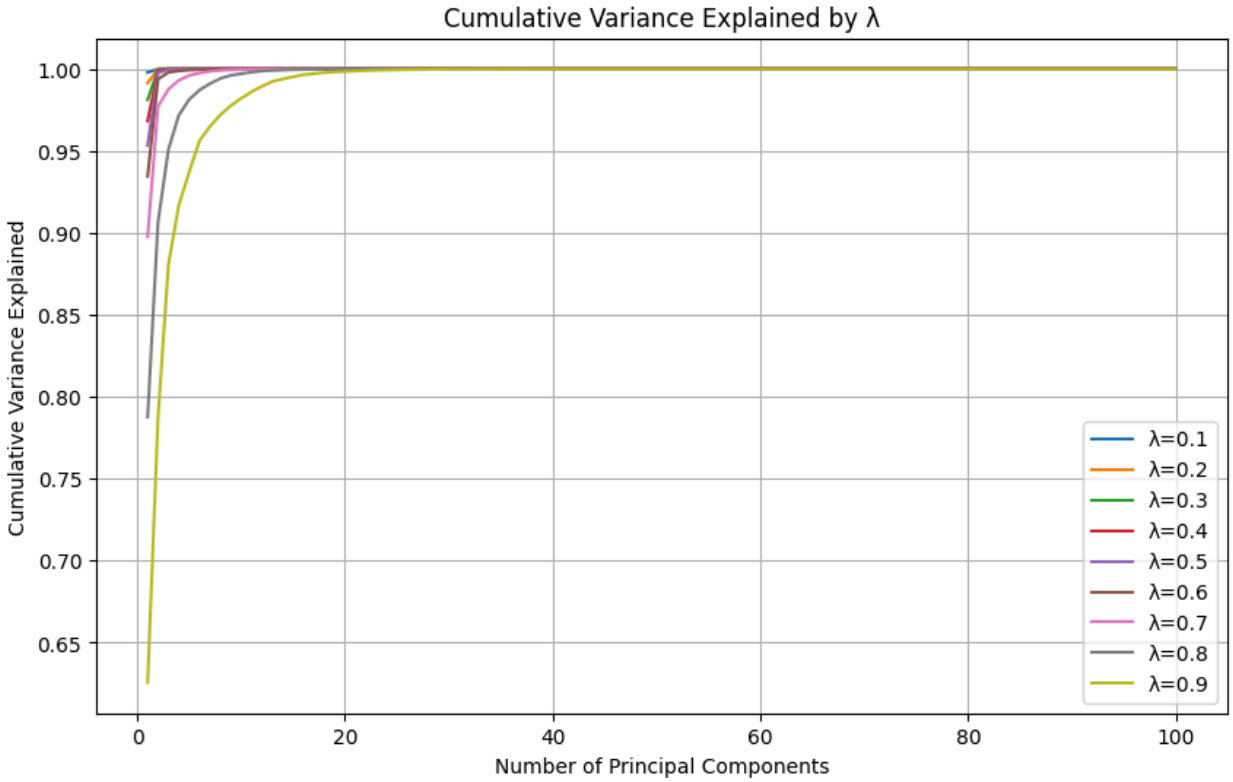
4. *PartD:*
    a. ARMA(0,1) - AICc: -1508.91
    b. ARMA(0,2) - AICc: -1559.23
    c. ARMA(0,3) - AICc: -1645.09
    d. ARMA(1,0) - AICc: -1669.08
    e. ARMA(1,1) - AICc: -1723.43
    f. ARMA(1,2) - AICc: -1728.85
    g. ARMA(1,3) - AICc: -1744.08
    h. ARMA(2,0) - AICc: -1696.07
    i. ARMA(2,1) - AICc: -1725.43
    j. ARMA(2,2) - AICc: -1739.99
    k. ARMA(2,3) - AICc: -1742.50
    l. ARMA(3,0) - AICc: -1746.24
    m. ARMA(3,1) - AICc: -1744.25
    n. ARMA(3,2) - AICc: -1742.29
    o. ARMA(3,3) - AICc: -1740.41
    p. Best model: ARMA(3, 0) with AICc: -1746.24
    q. Based on the comparison of AICc values, the ARMA(3,0) model provides the best fit for the time series data, with the lowest AICc value of -1746.24. This indicates that the model effectively balances goodness of fit and simplicity, capturing the dynamics of the data without unnecessary complexity. The residual analysis further supports this choice, as the residuals from the ARMA(3,0) model are randomly distributed around zero, resembling white noise. While other models such as ARMA(1,3) and ARMA(3,1) also have competitive AICc values, they are slightly higher, making ARMA(3,0) the most optimal. Lower-order models, like ARMA(1,1) and ARMA(2,2), show higher AICc values, indicating they fail to capture the full structure of the data. Thus, the ARMA(3,0) model is the best representation of the underlying time series behavior.

Problem5

1. *PartA:*
    a. Exponentially Weighted Covariance Matrix (λ=0.97):
    b. [[7.19328492e-05 5.39025561e-05 1.24211898e-04 ... 5.90676803e-05
    c. 1.26977172e-04 5.26657656e-05]
    d. [5.39025561e-05 1.39267406e-04 4.08463903e-05 ... 5.99870926e-05
    e. 8.39458050e-05 3.78230986e-05]
    f. [1.24211898e-04 4.08463903e-05 6.62806598e-04 ... 1.41563912e-05
    g. 3.21709386e-04 4.95822627e-05]
    h. ...
    i. [5.90676803e-05 5.99870926e-05 1.41563912e-05 ... 2.50136306e-04
    j. 8.43975411e-05 8.80937769e-05]

k. [1.26977172e-04 8.39458050e-05 3.21709386e-04 ... 8.43975411e-05
l. 7.38435342e-04 7.33886829e-05]
m. [5.26657656e-05 3.78230986e-05 4.95822627e-05 ... 8.80937769e-05
n. 7.33886829e-05 1.53172671e-04]]

2. *PartB:*



a.

3. *PartC*:
   a. The graph illustrates the impact of lambda on the cumulative variance explained by principal components in the exponentially weighted covariance matrix. Smaller lambda values lead to a faster accumulation of variance, with fewer principal components explaining a larger proportion of the total variance. This indicates that smaller lambda emphasizes recent data, making the covariance matrix more responsive to recent changes. In contrast, larger lambda values result in a slower accumulation of variance, spreading it more evenly across components, as these values give more uniform weight to historical data. This trade-off highlights that smaller lambda values are suited for capturing short-term dynamics, while larger lambda values are better for stability and long-term trends. The choice of lambda depends on the specific objectives of the analysis.

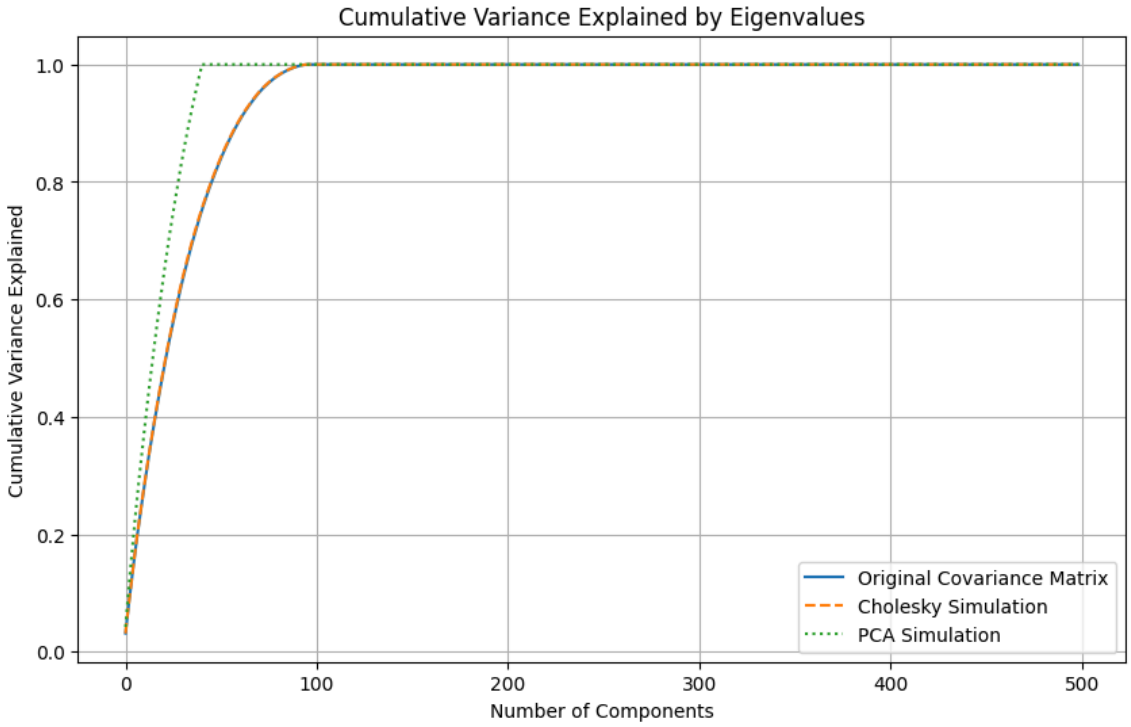
Problem6:

1. *PartA&B*: simulation
   a. Cholesky Simulation Shape: (10000, 499)
   b. Cholesky Simulation Time: 0.25327587127685547
   c. PCA Simulation Shape: (10000, 499)
   d. PCA Simulation Time: 0.02667689323425293
2. Part C:

a. Frobenius Norm (Cholesky): 0.0205586440
b. Frobenius Norm (PCA): 0.0806317687
c. The Frobenius norm results indicate that the Cholesky decomposition method produces a covariance matrix much closer to the original, with a norm of 0.0206, compared to 0.0806 for PCA. This suggests that the Cholesky method better preserves the covariance structure of the original data, as it directly factors the full covariance matrix. In contrast, PCA selects a subset of principal components, which inherently discards some information about the original correlations. As a result, PCA introduces more deviation from the true covariance matrix, leading to a higher Frobenius norm. This highlights a fundamental trade-off: Cholesky provides higher accuracy in capturing dependencies but is computationally expensive, while PCA sacrifices some precision for dimensionality reduction and computational efficiency.

3. Part d:

a.



b. The cumulative variance explained by the Cholesky simulation closely aligns with the original covariance matrix across all components, demonstrating that Cholesky accurately preserves the covariance structure by factoring the full covariance matrix without reducing dimensionality. In contrast, the PCA simulation reaches the 75% cumulative variance threshold much faster, requiring fewer components, as it explicitly reduces dimensionality by selecting the principal components that explain the most variance. While the Cholesky simulation maintains a near-identical eigenvalue distribution to the original matrix with no noticeable loss in explained variance, PCA sacrifices information about the covariance structure by truncating components beyond the 75% variance threshold. This results in a faster rise in cumulative variance but less precision in preserving the original matrix's characteristics. Overall, PCA is efficient for dimensionality reduction, trading off some accuracy, whereas Cholesky provides high fidelity in capturing the full covariance structure.

4. Part E:

    a. The Cholesky simulation took significantly longer (approximately 0.2533 seconds) compared to the PCA simulation (approximately 0.0267 seconds). This difference arises because Cholesky decomposition directly factors the full covariance matrix, requiring O(n3)O(n^3)O(n3) computational complexity for a matrix of size n×nn \times nn×n. In contrast, PCA leverages eigen decomposition but, in this case, involves fewer computations because it retains fewer principal components and efficiently reconstructs the reduced-dimensional data. While Cholesky is computationally expensive, it provides a more precise reconstruction of the covariance structure.
5. Part F:
    a. The key tradeoff between Cholesky and PCA lies in precision versus computational efficiency. Cholesky decomposition accurately preserves the original covariance structure, as reflected in the lower Frobenius norm (0.0206), but at the cost of higher computational time. PCA, on the other hand, sacrifices some accuracy (Frobenius norm of 0.0806) by truncating lower-variance components, making it faster and computationally more efficient. PCA is well-suited for dimensionality reduction tasks where speed and noise reduction are priorities, while Cholesky is ideal when preserving the full covariance structure is critical. The choice depends on whether the focus is on maintaining accuracy or optimizing for speed and resource efficiency.