Problem 1

A.

Mean: 0.0502

Variance: 0.0103

Skewness: 0.1204

Kurtosis: 0.2229

B.

I choose Normal Distribution for following reasons:

(1) symmetrical: The histogram of data shows the distribution of data is close to symmetrical.Besides, according to 1A, the skewness of data is 0.1204, close to 0, which means that the data is close to symmetrical， without significant left or right bias.

(2) no significant heavy tails: According to 1A, the kurtosis of data is 0.2229, close to 0, which means no significant heavy tails.

C.

normal distribution: $\mu = 0.0502$, $\sigma = 0.1016$

AICc of normal distribution: -1731.5747

t distribution: $v = 28.7102$, $\mu = 0.0499$, $\sigma = 0.0980$

AICc of t distribution: -1731.3943

AICc values determine model performance, the smaller the AICc, the better the model. According to AICc result, the normal distribution has a slightly lower AICc than the t-distribution (difference = 0.1804), indicating that it provides a better fit. The reason why the difference is so small is that the degrees of freedom of t-distribution is high(df = 28), and t-distribution with high degree of freedom is very close to normal

distribution. Therefore, the judgement that using normal distribution to fit given data is proved.

Problem 2

A.

Pairwise 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.

The Pairwise Covariance Matrix is not positive semi-definite, because the smallest eigenvalue is -0.3102 < 0.

C.

PSD calculated by Higham's method':

	x1	x2	x3	x4	x5
x1	1.470484	1.332361	0.884378	1.627602	1.399556
x2	1.332361	1.252078	0.619028	1.450604	1.214450
x3	0.884378	0.619028	1.272425	1.076846	1.059658
x4	1.627602	1.450604	1.076846	1.814469	1.577928
x5	1.399556	1.214450	1.059658	1.577928	1.396186

PSD calculated by Rebonato & Jackel's method:

	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.

the covariance matrix using only 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.

The results obtained from Method C (PSD correction using Higham's and Rebonato & Jackel's methods) and Method D (covariance matrix using only overlapping data) show significant differences in their values. The key observation is that the covariance values in Method C are considerably higher than those in Method D, which indicates that the approach used to handle missing data has a major impact on the final results.

In Method C, all available data were used, even if some observations contained missing values. The computed covariance matrix was then adjusted to ensure that it was at least positive semi-definite (PSD). This correction helped preserve much of the original structure of the theoretical covariance matrix, which was generated with ones on the diagonal and 0.99 elsewhere. As a result, the covariance values remained relatively high, with off-diagonal elements ranging from 0.9 to 1.7, which aligns more closely with the expected theoretical structure.

In contrast, Method D only considered complete cases, meaning that any row containing missing values was entirely excluded from the computation. As a consequence, the effective sample size for each variable pair was significantly reduced. This led to a systematic underestimation of covariance values, with off-diagonal elements dropping to approximately 0.39 to 0.47, which is much lower than the expected 0.99. The reduction in covariance values suggests that the missing data had a substantial impact on the accuracy of the estimation, causing a deviation from the theoretical covariance matrix.

The key takeaway is that Method C is generally more reliable in preserving the covariance structure when missing data are present. By correcting negative eigenvalues and enforcing a PSD structure, it maintains a reasonable approximation to the true covariance matrix, even if some data points are missing. On the other hand, Method D introduces bias by discarding a significant portion of the data, leading to an artificially lower covariance estimate.

In conclusion, when dealing with missing data, applying a PSD correction (Method C) is a preferable approach as it utilizes all available information and ensures a valid covariance matrix. In contrast, restricting calculations to complete cases (Method D) may lead to significant underestimation, particularly when the dataset has a substantial amount of missing values. Therefore, PSD correction methods provide a more accurate representation of the underlying covariance structure compared to simply removing incomplete data.

Problem 3

A.

According to given data, the mean and covariance matrix are calculated as follows:

$$\mu = \begin{pmatrix} 0.04600157 \\ 0.09991502 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 0.0101622 & 0.00492354 \\ 0.00492354 & 0.02028441 \end{pmatrix}$$

Therefore,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0.04600157 \\ 0.09991502 \end{pmatrix} \begin{pmatrix} 0.0101622 & 0.00492354 \\ 0.00492354 & 0.02028441 \end{pmatrix}\right)$$

B.

① Method 1:

according to A.

$\mu = \begin{pmatrix} 0.04600157 \\ 0.09991502 \end{pmatrix}$, $\mu_1 = 0.04600157$, $\mu_2 = 0.09991502$

$\Sigma = \begin{pmatrix} 0.0101622 & 0.00492354 \\ 0.00492354 & 0.02028441 \end{pmatrix}$ $\Sigma_{11} = 0.0101622$ $\Sigma_{12} = 0.00492354$ $\Sigma_{21} = 0.00492354$ $\Sigma_{22} = 0.02028441$

$X_2 \mid X_1 = 0.6 \sim N(\bar{\mu}, \bar{\Sigma})$

where,

$\bar{\mu} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(0.6 - \mu_1)$

$= 0.09991502 + \frac{0.00492354}{0.0101622} \times (0.6 - 0.04600157)$

≈ 0.3683

$\bar{\Sigma} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$

$= 0.02028441 - 0.00492354 \times \frac{1}{0.0101622} \times 0.00492354$

≈ 0.017

$\therefore X_2 \mid X_1 = 0.6 \sim N(0.3683, 0.017)$.

② Method 2.

In OLS,

$$E(X_2|X_1) = \beta_0 + \beta_1 X_1$$

where,

$$\beta_1 = \frac{Cov(X_1, X_2)}{Var(X_1)} = \frac{\sigma_{12}}{\sigma_1^2} = \frac{0.00487354}{0.0101622} \approx 0.47945$$

$$\beta_0 = \mu_2 - \beta_1 \mu_1 = 0.09991502 - \frac{0.00487354}{0.0101622} \times 0.04600157 \approx 0.0778$$

$$\therefore E(X_2|X_1) = 0.0778 + 0.47945 X_1$$

$$\therefore E(X_2|X_1 = 0.6) = 0.0778 + 0.47945 \times 0.6 \approx 0.3683$$

$$Var(X_2|X_1) = \sigma_2^2 - \frac{\sigma_{12}^2}{\sigma_1^2} = 0.02023491 - \frac{(0.00487354)^2}{0.0101622} \approx 0.0179$$

$$\therefore X_2|X_1=0.6 \sim N(0.3683, 0.0179)$$

C.

$$X = LZ + \mu$$

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

Therefore,

$$X_1 = L_{11}Z_1 + \mu_1, \quad Z_1 = \frac{X_1 - \mu_1}{L_{11}}$$

$$X_2 = L_{21}Z_1 + L_{22}Z_2 + \mu_2 = L_{21}\frac{X_1 - \mu_1}{L_{11}} + L_{22}Z_2 + \mu_2, \text{ where } Z_2 \sim N(0,1)$$

Calculate L in python, the result is $L = \begin{bmatrix} 0.10080772 & 0 \\ 0.04884093 & 0.13378703 \end{bmatrix}$

Given $X_1 = 0.6$ *and result calculated in A, we have*:

$$\mu_1 = 0.04600157$$

$$\mu_2 = 0.09991502$$

$$L_{11} = 0.10080772$$

$$L_{21} = 0.04884093$$

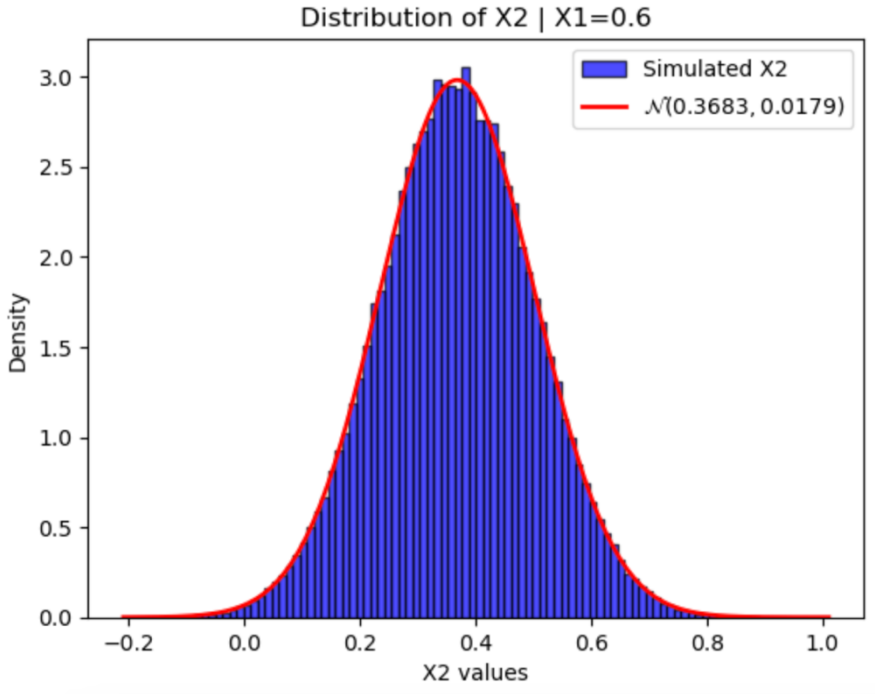
$$L_{22} = 0.13378703$$

Generate 10000 standard normal distribution sample, and substitute the values into the formula $X_2 = L_{21}\frac{X_1 - \mu_1}{L_{11}} + L_{22}Z_2 + \mu_2$, then calculate the mean and variation of samples. The result is:

Conditional mean of $X_2|X_1$ =0.6 is 0.36801553596789105

Conditional variance of $X_2|X_1$ =0.6 is 0.01779291551001577

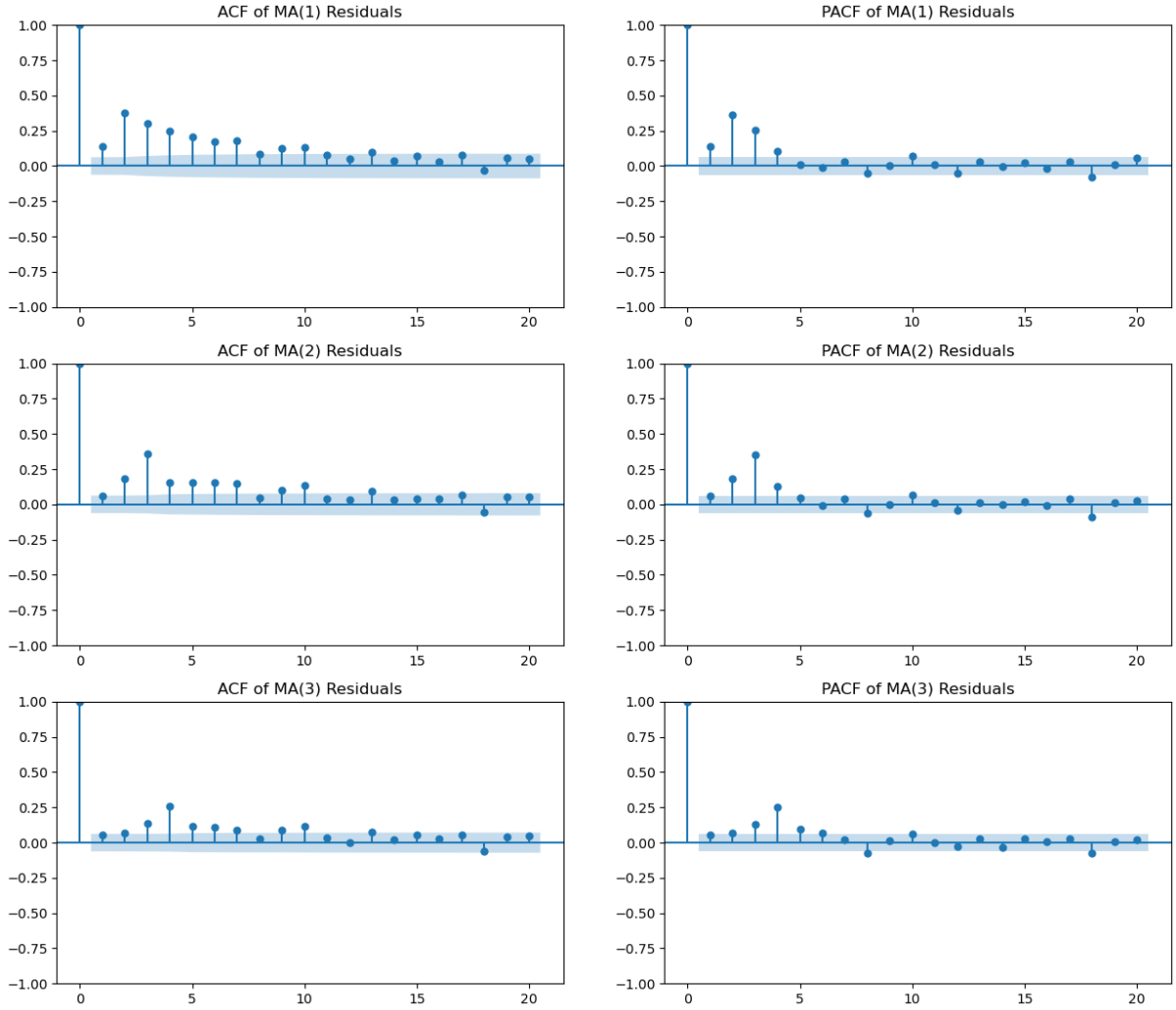
Then plot simulation normal distribution of $X_2|X_1 = 0.6 \sim N(0.3683, 0.0179)$ (calculated in B) and results in the same graph:



According to calculated result and graph, distribution of $X_2|X_1 = 0.6 \sim N(0.3683, 0.0179)$ calculated in B is proved.

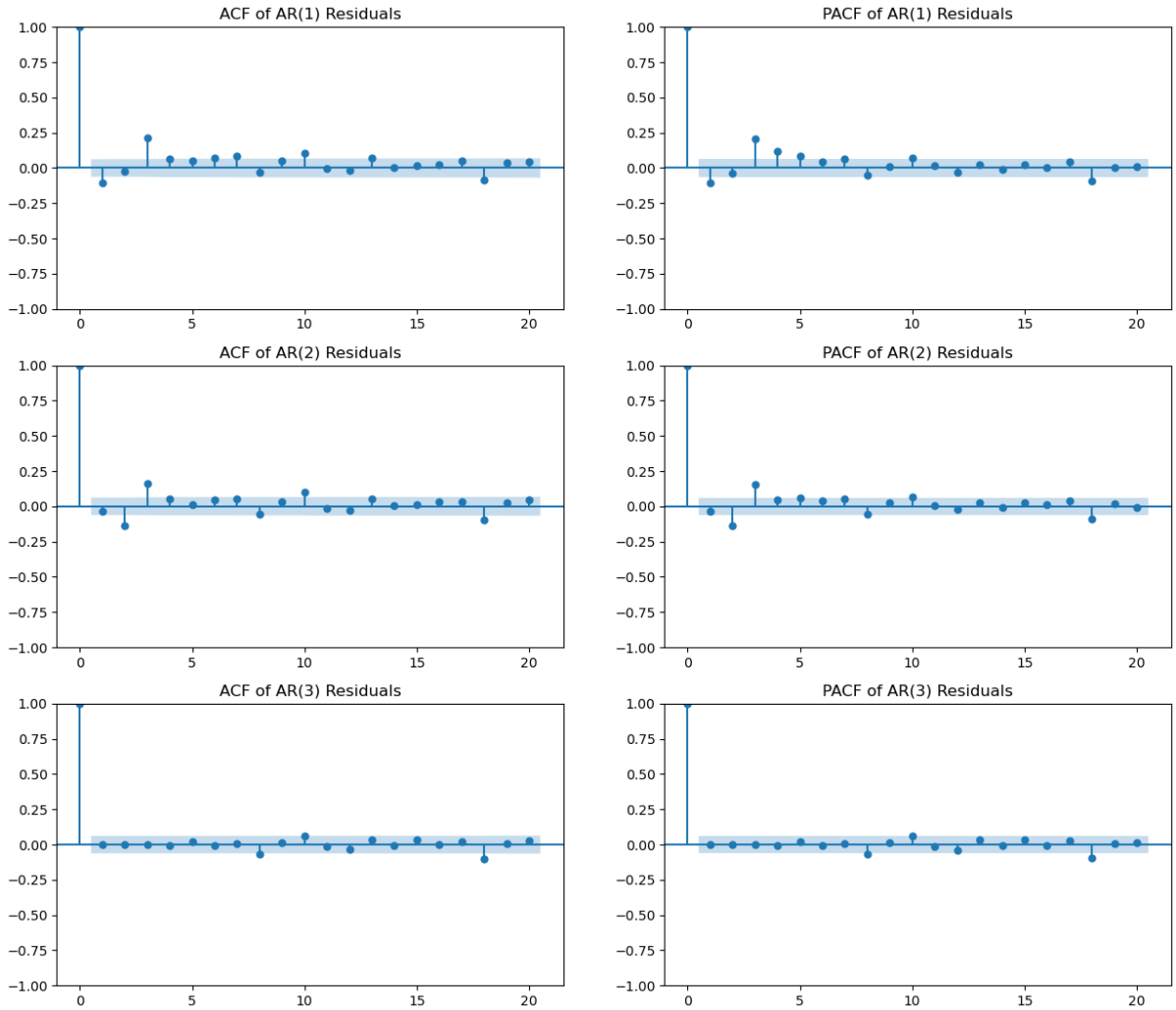
Problem 4

A.



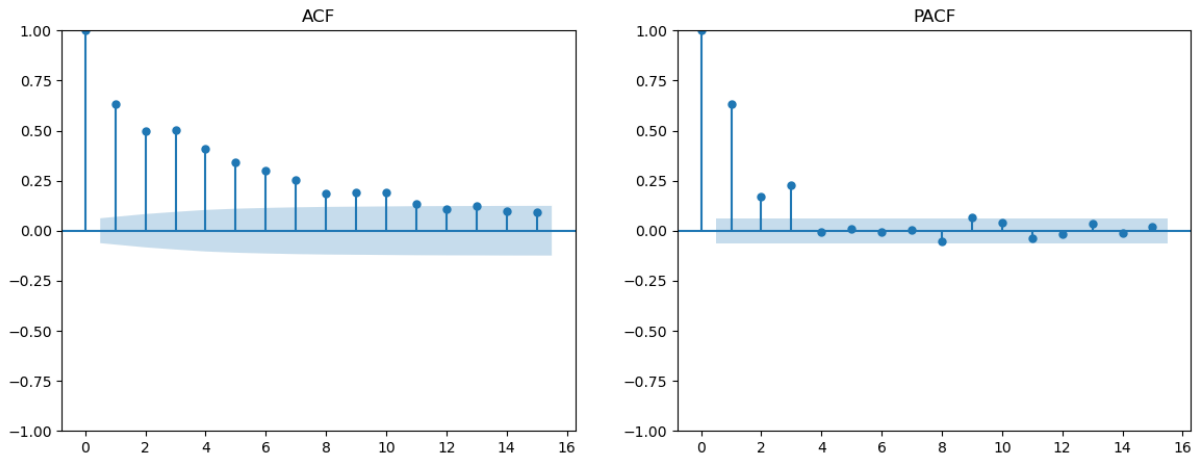
After fitting the data with MA(1), MA(2), and MA(3) models, the ACF and PACF plots of the residuals are examined. Ideally, well-fitted MA models should produce residuals with white noise properties, showing no autocorrelation. However, the ACF and PACF plots indicate significant autocorrelation in the residuals, suggesting that the MA models do not fit the data well.

B.



After fitting the data with AR(1), AR(2), and AR(3) models, the ACF and PACF plots of the residuals are analyzed. For well-fitted AR models, the residuals should exhibit white noise properties without autocorrelation. The ACF and PACF plots for AR(1) and AR(2) show truncation at lag 3, indicating that the models are underfitting. In contrast, the residuals from AR(3) demonstrate white noise characteristics, indicating that AR(3) provides a better fit.

C.



I would like to model the data using AR process. Because the ACF plot exhibits a tailing pattern, while the PACF plot shows a sharp cutoff. Based on these characteristics, it is appropriate to select an AR (Autoregressive) model to fit the original data.

D.

After calculating AICc of AR(1) ~ AR(10), AR(3) has the smallest AICc. Therefore, AR(3) is the best fit.

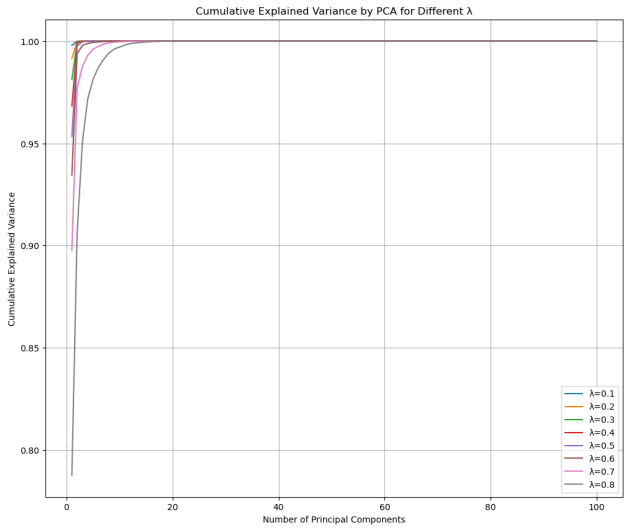
Problem 5

A.

Code as bellow:

```python
def populate_weights(n: int, lambda_factor: float):
    """calculate exponential weights

    Args:
        n (int): length of data
        lambda_factor (float): decay factor λ between 0 an 1

    Returns:
        _type_: weight array, cumulative weight array
    """
    w = np.zeros(n)
    cumulative_w = np.zeros(n)
    total_w = 0.0

    for i in range(n):
        w[i] = (1 - lambda_factor) * lambda_factor ** (i + 1)
        total_w += w[i]
        cumulative_w[i] = total_w

    w /= total_w
    cumulative_w /= total_w

    return w, cumulative_w


def exponentially_weighted_covariance_matrix(df: pd.DataFrame, lambda_factor: float):
    """calculate exponentially weighted covariance matrix

    Args:
        df (pd.DataFrame):
        lambda_factor (float): decay factor λ between 0 an 1

    Returns:
        _type_: exponentially weighted covariance matrix
    """
    n = len(df)
    weights, _ = populate_weights(n, lambda_factor)
    weights = weights[::-1]
    df = df.loc[:, df.columns != "Date"]
    means = (df.T * weights).sum(axis=1)

    mean_centered = df - means

    variables = df.columns
    cov_matrix = pd.DataFrame(index = variables, columns = variables, dtype = float)

    for i in variables:
        for j in variables:
            cov_matrix.loc[i, j] = np.sum(weights * mean_centered[i] * mean_centered[j])

    return cov_matrix
```

B.



C.

The value of λ in the exponentially weighted covariance matrix determines how past data is weighted relative to recent data. Smaller λ values place greater emphasis on recent data, leading to a covariance matrix that captures short-term fluctuations and is more responsive to recent changes. Conversely, larger λ values assign more weight to historical data, resulting in a smoother covariance matrix that reflects long-term trends. The PCA results and the cumulative variance explained curves show that smaller λ values produce more dynamic eigenvalue distributions, emphasizing the dominant principal components influenced by recent data. Larger λ values lead to more evenly distributed eigenvalues, highlighting the importance of long-term consistency in the data. This illustrates the trade-off between capturing short-term volatility and maintaining a broader historical perspective in covariance matrix estimation.

Problem 6

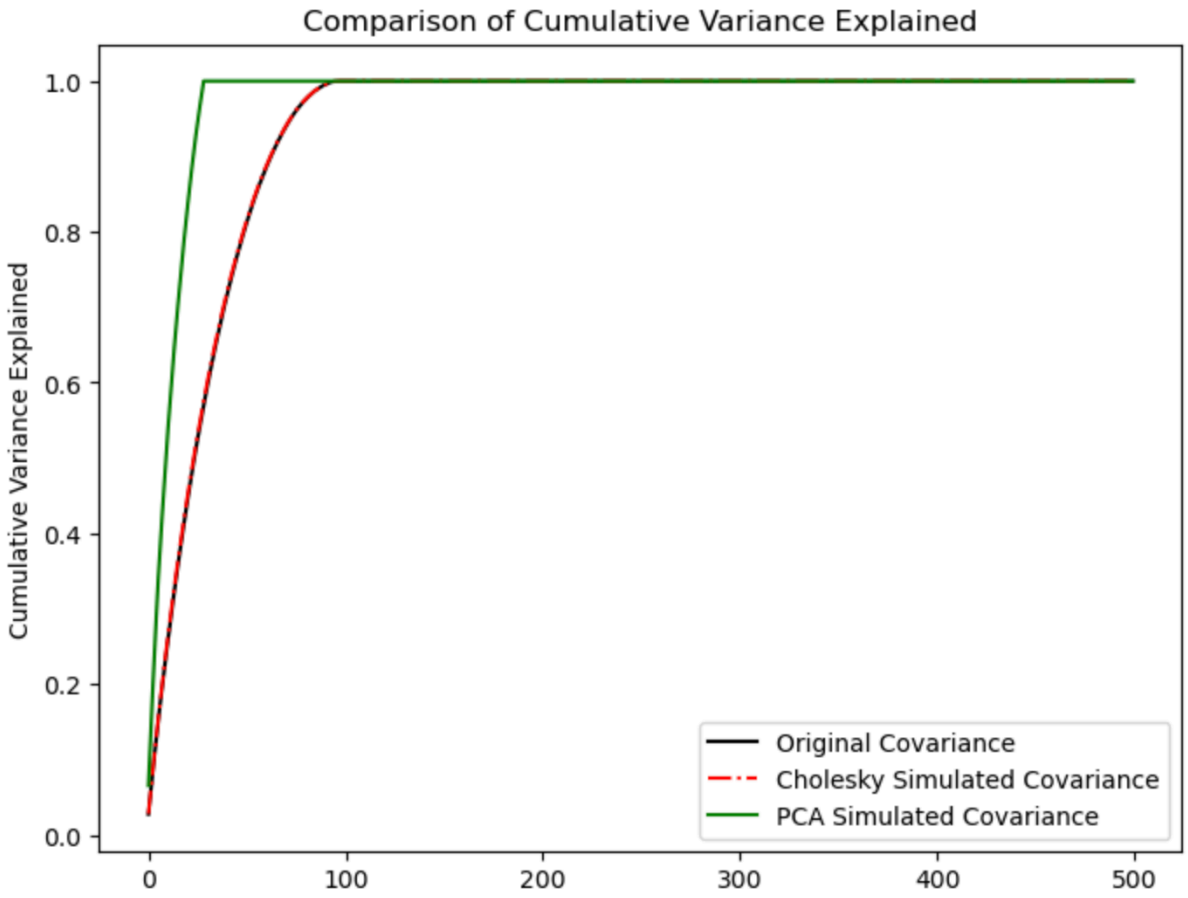
C.

frobenius_cholesky: 0.021250368442464126

frobenius_pca: 0.26028844361926623

The Cholesky method shows a significantly smaller Frobenius norm compared to the PCA method. This indicates that the covariance matrix simulated using the Cholesky method is much closer to the original covariance matrix, as the Cholesky decomposition fully preserves the structure of the original covariance.

D.



The Cholesky method closely matches the original covariance matrix, as it retains the entire variance structure without any truncation.

The PCA-simulated covariance matrix, while explaining the majority of the variance, shows cumulative variance that exceeds the original and Cholesky-simulated covariance matrices across all principal components. This deviation indicates that the PCA simulation inflates the variance contributions in certain components to meet the 75% variance threshold, resulting in overestimation compared to the original.

This overestimation is a consequence of PCA emphasizing the retained principal components, which can distort the cumulative variance when compared to the original covariance structure.

E.

time cost by Cholesky Root method: 0.2745 s

time cost by PCA method: 0.1165 s

The PCA method is significantly faster because it reduces the dimensionality of the data by selecting a limited number of principal components (those that explain 75% of the variance). The Cholesky method, on the other hand, performs a full decomposition of the covariance matrix, which is computationally more expensive.

F

(1) Cholesky Method:

a. Advantages:

- Fully preserves the covariance structure, leading to a simulated matrix that is nearly identical to the original (as seen in the Frobenius norm comparison).
- Retains all variance and eigenvalue information, making it ideal for high-accuracy simulations.

b. Disadvantages:

- Computationally slower compared to PCA.
- Does not provide dimensionality reduction, which might be unnecessary in some scenarios.

(2) PCA Method:

a. Advantages:

- Faster simulation due to reduced dimensionality, making it suitable for large-scale datasets or applications where speed is critical.
- Offers dimensionality reduction, which is useful for data compression or exploratory analysis.

b. Disadvantages:

- Cumulative variance consistently overestimates the true variance structure, as seen in the graph.
- The resulting covariance matrix deviates more significantly from the original, as indicated by the larger Frobenius norm.