Fintech 545 - Project1

Problem 1

Given the dataset in **problem1.csv**, answer the following:

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

Answer:

- **Mean** 0.05019795790476916
- **Variance** 0.010332476407479581
- **Skewness** 0.1204447119194402
- **Kurtosis** 0.22908332509377516

B. Choose Between Normal and T-Distribution

Given the statistical characteristics of the dataset:

- Would you model the data using a **Normal Distribution** or a **T-Distribution**?
- Justify your choice based on the data properties.

Answer:

Normal Distribution:

1. Parameters: mu = 0.050366360906888966, sigma = 0.1015091958216428
2. KS Test Statistic: 0.01274823663658442
3. KS Test p-value: 0.9962811660027789

T Distribution:

1. Parameters: df = 27.95470932568757, loc = 0.05002790293792376, scale = 0.09781715958134082
2. KS Test Statistic: 0.012673598952560733
3. KS Test p-value: 0.9965693525994647

Based on the analysis, I would choose the **Normal Distribution** to model the data. Here are the key reasons for this decision:

1. The skewness is very low (0.12), indicating that the data are nearly symmetric. Additionally, the excess kurtosis is very near 0 (0.23), suggesting that the tails of the

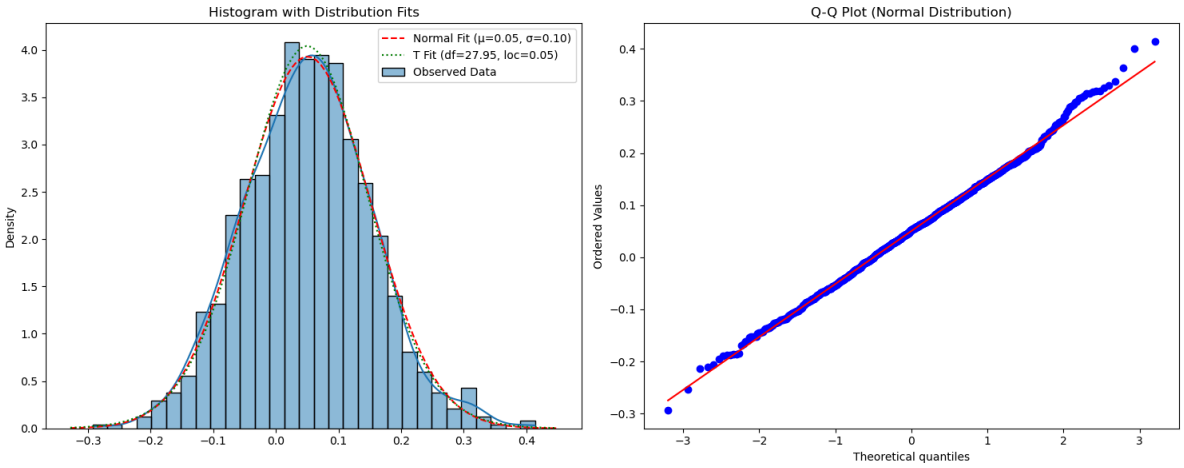
data are not significantly heavier than those of a normal distribution. These values support the idea that the data are approximately normally distributed.

2. Both the Normal and T distributions yield very high KS test p-values, which indicates that neither model can be statistically rejected based on the KS test alone. The KS test suggests that the observed data are consistent with both distributions.However, because the differences in the KS test statistics are extremely small and the descriptive statistics (with nearly zero skewness and only a slight excess kurtosis) point toward a nearly symmetric, light-tailed distribution, there is little evidence to support the need for the extra flexibility of the T distribution. Additionally, the estimated degrees of freedom for the T distribution are quite high (around 28), meaning that it essentially approximates a Normal distribution.

C. Fit Both Distributions and Evaluate the Choice

1. Fit both a **Normal Distribution** and a **T-Distribution** to the dataset.
2. Use statistical methods presented in class to verify which model fits better.
3. Compare results using appropriate goodness-of-fit tests.

Answer:



C. Distribution Selection: Selection Criteria:

1. Normal Distribution p-value: 0.9963
2. T Distribution p-value: 0.9966

Normal Distribution Fit:

- Parameters: mu = 0.0504, std = 0.1015
- Log-Likelihood: 867.7987
- AIC: -1731.5974
- BIC: -1721.7839

T-Distribution Fit:

- Parameters: df = 27.9547, loc = 0.0500, scale = 0.0978
- Log-Likelihood: 868.7629
- AIC: -1731.5259
- BIC: -1716.8056

From the **histogram with fitted distributions**, we can observe that both the normal and T-distributions follow the data closely. The **Q-Q plot** shows that the empirical quantiles align well with the theoretical quantiles of a normal distribution, though slight deviations exist in the tails.

Based on the AIC/BIC comparisons, both models fit the data very well. However, the Normal distribution shows a slightly lower (better) AIC and BIC than the T-distribution.

Problem 2

A. Calculate the pairwise covariance matrix of the data.

Pairwise Covariance Matrix:

	x1	x2	x3	x4	x5
x1	1.4705	1.4542	0.8773	1.9032	1.4444
x2	1.4542	1.2521	0.5395	1.6219	1.2379
x3	0.8773	0.5395	1.2724	1.1720	1.0919
x4	1.9032	1.6219	1.1720	1.8145	1.5897
x5	1.4444	1.2379	1.0919	1.5897	1.3962

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

Answer: ***No***. Here are the eigenvalues: [6.78670573 0.83443367 -0.31024286 0.02797828 -0.13323183]. The matrix is not positive semi-definite because it has negative eigenvalues (-0.310 and -0.133), despite pairwise covariances being positive.

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

Answer:

Nearest PSD Matrix (Higham) | | Col1 | Col2 | Col3 | Col4 | Col5 | |--------|---------|---------|---------|---------|---------| | **Row1** | 1.6151 | 1.4420 | 0.8971 | 1.7804 | 1.4338 | | **Row2** | 1.4420 | 1.3470 | 0.5851 | 1.5546 | 1.2114 | | **Row3** | 0.8971 | 0.5851 | 1.2989 | 1.1160 | 1.0767 | | **Row4** | 1.7804 | 1.5546 | 1.1160 | 1.9832 | 1.6214 | | **Row5** | 1.4338 | 1.2114 | 1.0767 | 1.6214 | 1.4049 |

Nearest PSD Matrix (Rebonato-Jackel) | | Col1 | Col2 | Col3 | Col4 | Col5 | |--------|--------|---------|---------|---------|---------| | **Row1** | 1.4705 | 1.3265 | 0.8473 | 1.6250 | 1.3638 | | **Row2** | 1.3265 | 1.2521 | 0.5583 | 1.4336 | 1.1643 | | **Row3** | 0.8473 | 0.5583 | 1.2724 | 1.0565 | 1.0623 | | **Row4** | 1.6250 | 1.4336 | 1.0565 | 1.8145 | 1.5460 | | **Row5** | 1.3638 | 1.1643 | 1.0623 | 1.5460 | 1.3962 |

D. Calculate the covariance matrix using only overlapping data

Answer:

Overlapping Covariance Matrix | | x1 | x2 | x3 | x4 | x5 | |--------|--------|--------|--------|--------|--------| | **x1** | 0.4186 | 0.3941 | 0.4245 | 0.4164 | 0.4343 | | **x2** | 0.3941 | 0.3968 | 0.4093 | 0.3984 | 0.4226 | | **x3** | 0.4245 | 0.4093 | 0.4414 | 0.4284 | 0.4490 | | **x4** | 0.4164 | 0.3984 | 0.4284 | 0.4373 | 0.4402 | | **x5** | 0.4343 | 0.4226 | 0.4490 | 0.4402 | 0.4663 |

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.

The Higham (and Rebonato–Jackel) methods adjust the original pairwise covariance matrix to enforce positive semidefiniteness while largely preserving the high covariance structure implied by the generating process (with 1’s on the diagonal and 0.99 off-diagonals). As a result, the adjusted matrix retains relatively high values, reflecting the near-perfect correlations intended by the model, even though some modifications were necessary to correct the negative eigenvalues present in the raw estimates.

In contrast, the covariance matrix computed using only overlapping (complete-case) data shows substantially lower values. This occurs because restricting the analysis to only those observations where all variables are present dramatically reduces the effective sample size, leading to an underestimation of both variances and covariances. Thus, while the Higham adjustment preserves the overall structure of the original estimates, the overlapping data method suffers from sample reduction and bias, resulting in a matrix that deviates significantly from the intended generating process.

Problem 3

Given the data in problem3.csv

A. Fit a multivariate normal to the data.

Answer:

Mean vector (mu): [0.04600157 0.09991502]

Covariance matrix (Sigma) | | Col1 | Col2 | |--------|---------|---------| | **Row1** | 0.0102 | 0.0049 | | **Row2** | 0.0049 | 0.0203 |

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

Answer:

Method 1: Conditional Distribution Formula

- **Conditional Mean** = 0.3683
- **Conditional Variance** = 0.0179

Method 2: OLS Method

- **Beta (slope)** = 0.4845
- **Alpha (intercept)** = 0.0776
- **Predicted Mean** = 0.3683
- **Estimated 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

Answer: Our simulation using Cholesky decomposition shows that our theoretical calculations for the conditional distribution of X2|X1=0.6 are correct. The simulated results give a conditional mean of about 0.362 and a conditional variance of 0.0165. These values are close to the theoretical mean of 0.368 and variance of 0.018 from Part B.

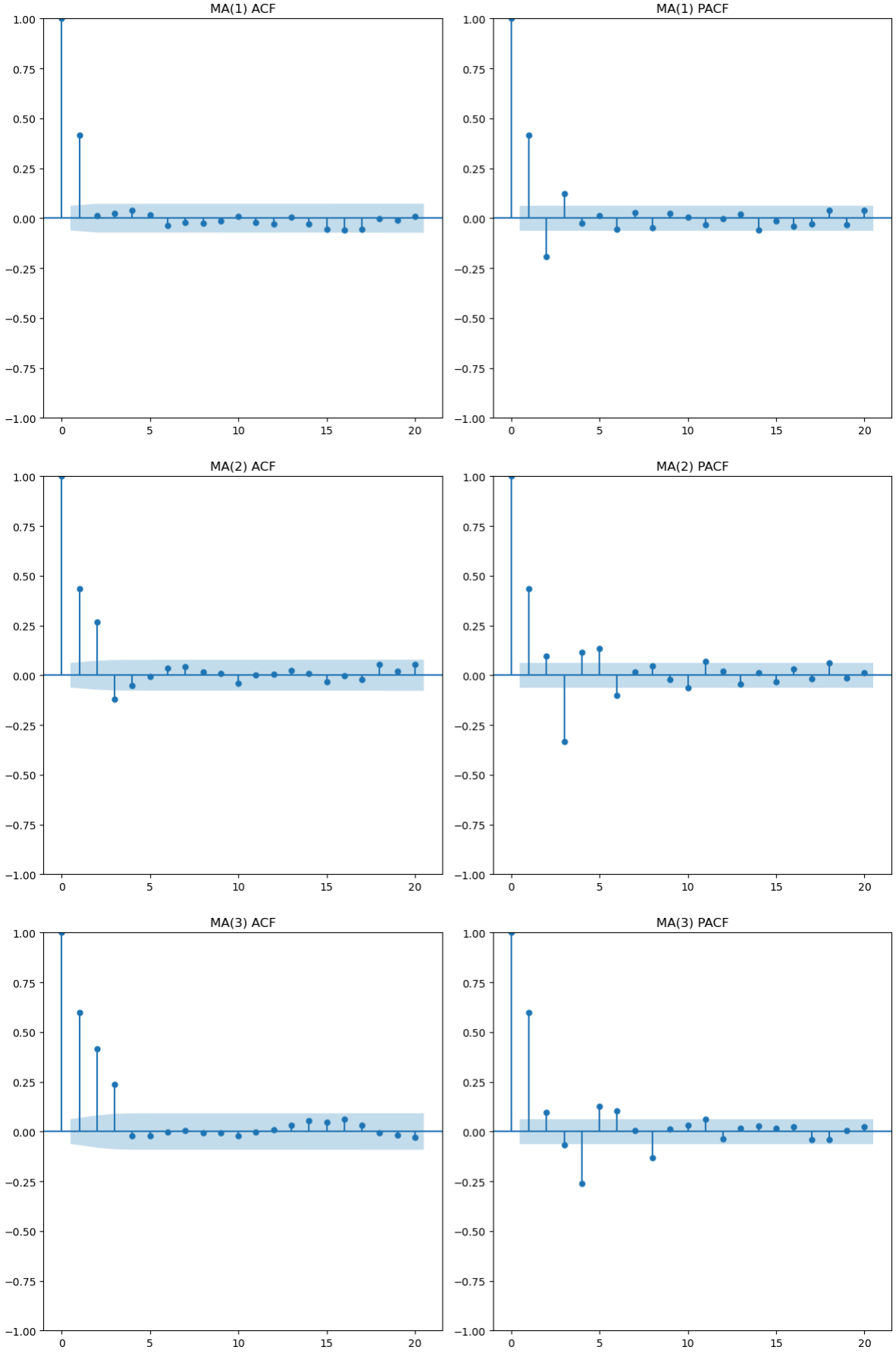
The small differences of about 1-2% come from the randomness in simulation and the use of a threshold approach to approximate conditional values. The histogram of X1 shows that the Cholesky decomposition generated correlated normal random variables. The red dashed line at X1=0.6 marks where we condition the distribution.

Problem 4

Given the data in problem4.csv

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

Answer:



1. MA(1) Process:

- ACF: The ACF shows a significant spike at lag 1 and then quickly drops to near zero for higher lags. This is characteristic of an MA(1) process, where the correlation only persists for one lag.
- PACF: The PACF has a single significant spike at lag 1, and the rest are within the confidence bounds, confirming the order of the moving average process.

1. MA(2) Process:

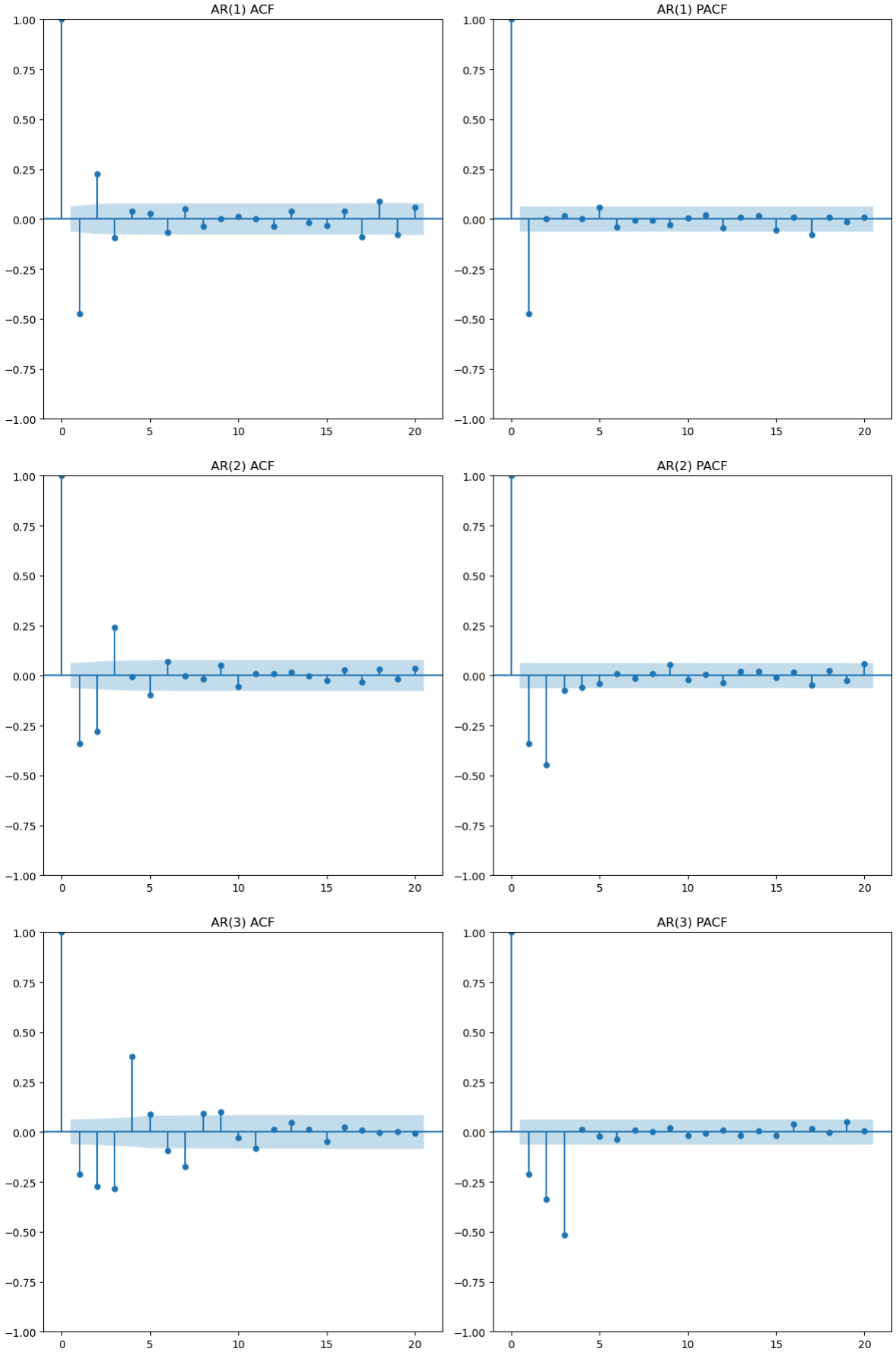
- ACF: The ACF exhibits significant spikes at lags 1 and 2, after which it rapidly drops to near zero. This aligns with the expected pattern for an MA(2) process.
- PACF: The PACF shows a more dispersed pattern, with some values outside the confidence bounds at lower lags, but generally, it does not cut off sharply.

1. MA(3) Process:

- ACF: The ACF shows significant spikes at lags 1, 2, and 3, then decays rapidly to zero, which matches the theoretical expectation for an MA(3) process.
- PACF: The PACF has multiple significant spikes at lower lags (especially around lag 3), with no clear cutoff.

Conclusion: ACF for an MA(q) process cuts off after q lags, meaning that correlations exist up to lag q and are zero afterward. PACF for an MA(q) process does not cut off but instead exhibits a trailing decay. This is because each lag is indirectly correlated through earlier terms.

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



1. AR(1) Process:

- ACF: The ACF shows an exponential decay, meaning the correlations decrease gradually over time. This is characteristic of an AR(1) process.
- PACF: The PACF has a sharp cutoff after lag 1, with only the first lag being significant. This aligns with the theoretical expectation that an AR(1) process only has direct dependence on the first lag.

1. AR(2) Process:

- ACF: The ACF exhibits the values oscillate slightly while gradually decreasing. This behavior is typical of an AR(2) process.
- PACF: The PACF shows significant spikes at lags 1 and 2, then cuts off at higher lags. This confirms the presence of two autoregressive terms.

1. AR(3) Process:

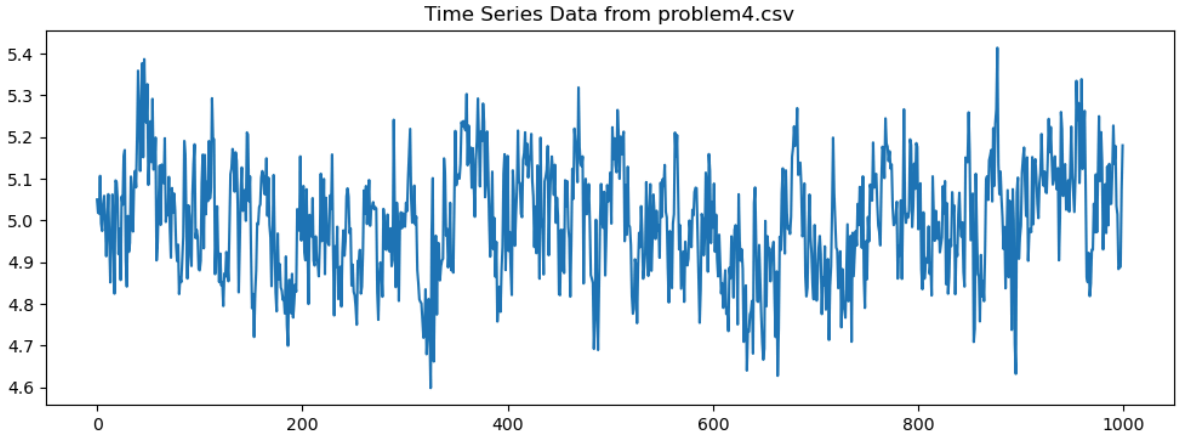
- ACF: The ACF continues to exhibit a damped sinusoidal pattern, but the oscillations persist for more lags before vanishing.
- PACF: The PACF shows significant spikes at lags 1, 2, and 3, before cutting off. This confirms an AR(3) structure.

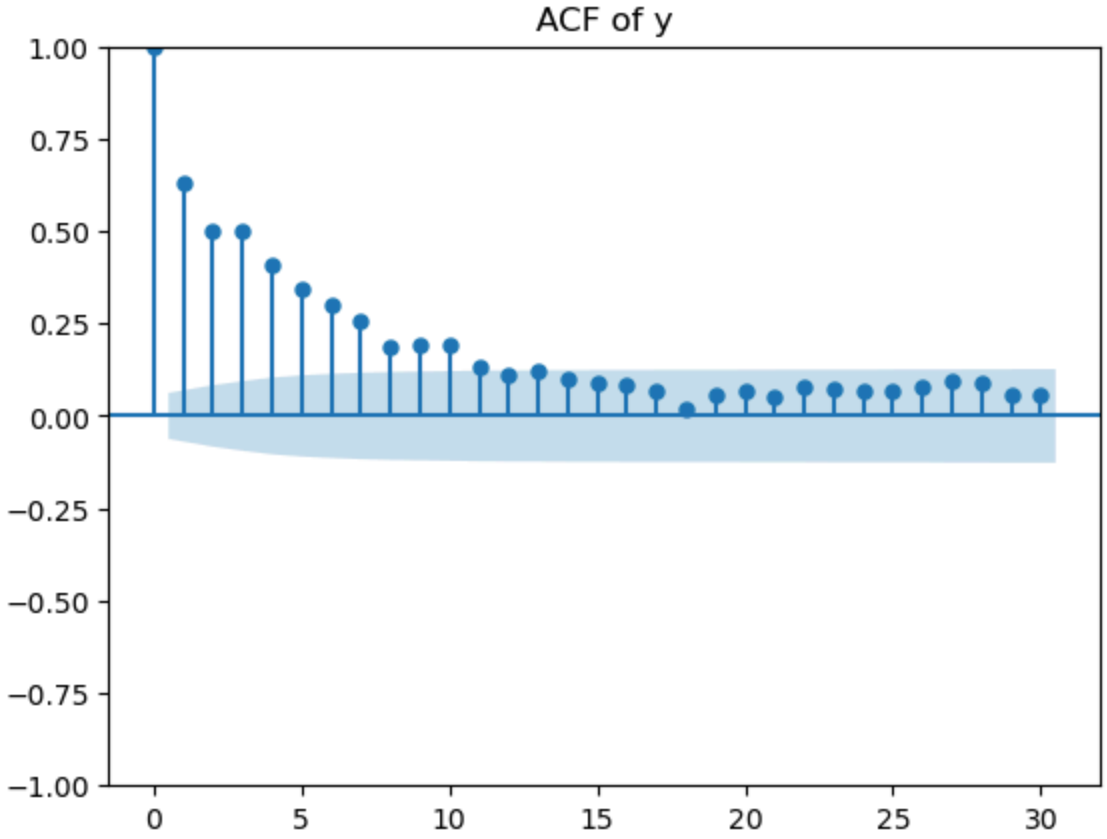
Conclusion：

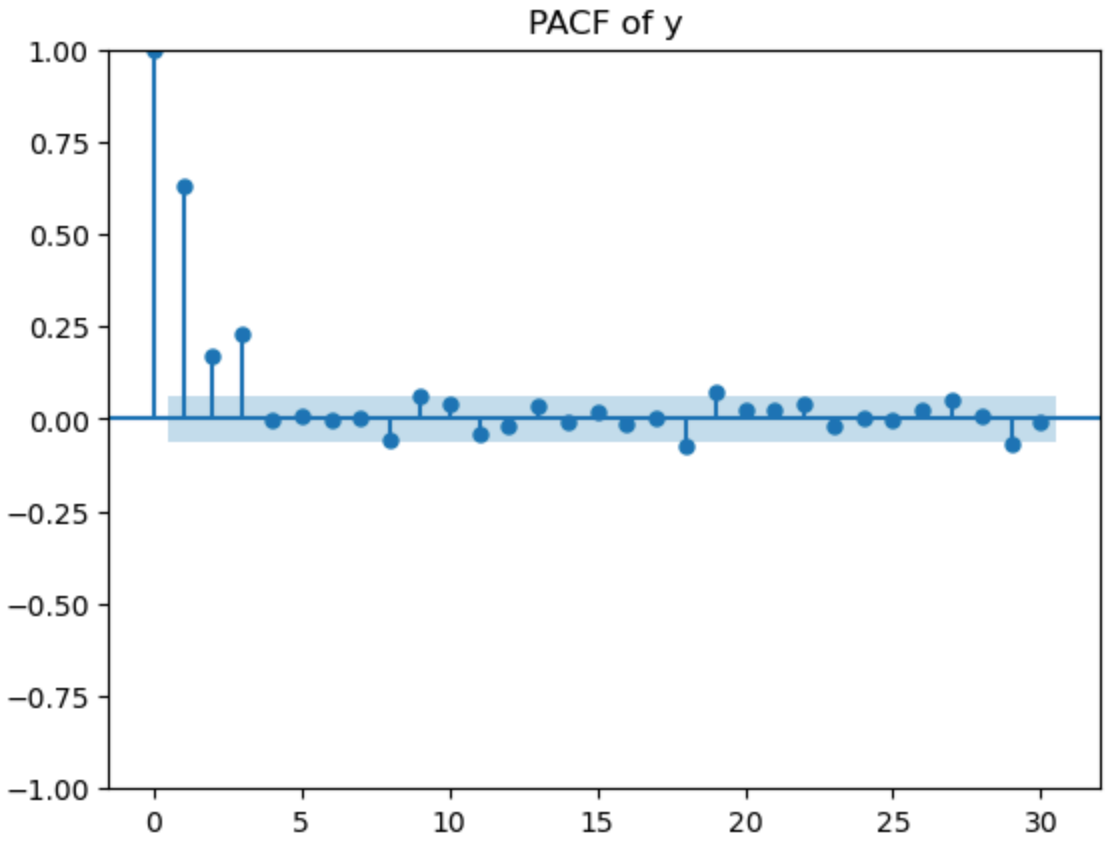
AR processes have ACFs that decay gradually, unlike MA processes where the ACF cuts off after a certain lag. The PACF of an AR(p) process cuts off sharply at lag p, which helps identify the order of the model. The damped sinusoidal pattern in ACF suggests the presence of complex roots in the characteristic equation, which occurs in AR(2) and AR(3) models.

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

Answer:


Time Series Data from problem4.csv
5.4
5.3
5.2
5.1
5.0
4.9
4.8
4.7
4.6
0
200
400
600
800
1000


ACF of y
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30



1. Time Series Plot: The series appears to be stationary with fluctuations around a mean level, without obvious trends or seasonality.
2. ACF: The ACF shows a gradual decay, indicating a potential moving average (MA) component.
3. PACF: The PACF cuts off sharply after lag 2, suggesting an autoregressive (AR) process with an order around 2.

Conclusion： The PACF behavior (significant lags at 1 and 2, followed by a cutoff) suggests an AR(2) process. Since both AR and MA components are present, an ARMA(2,q) model may be appropriate.

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

Answer:

Order	AICc
(1, 0, 0)	-1466.66
(2, 0, 0)	-1571.40
(3, 0, 0)	-1670.94
(0, 0, 1)	4693.45
(0, 0, 2)	3531.43

Order	AICc
(0, 0, 3)	2634.25
(2, 0, 2)	**-1685.41**
(3, 0, 3)	-1678.98

Best Model:

Based on AICc, the best model is **(2, 0, 2) → ARMA(2,2)**.

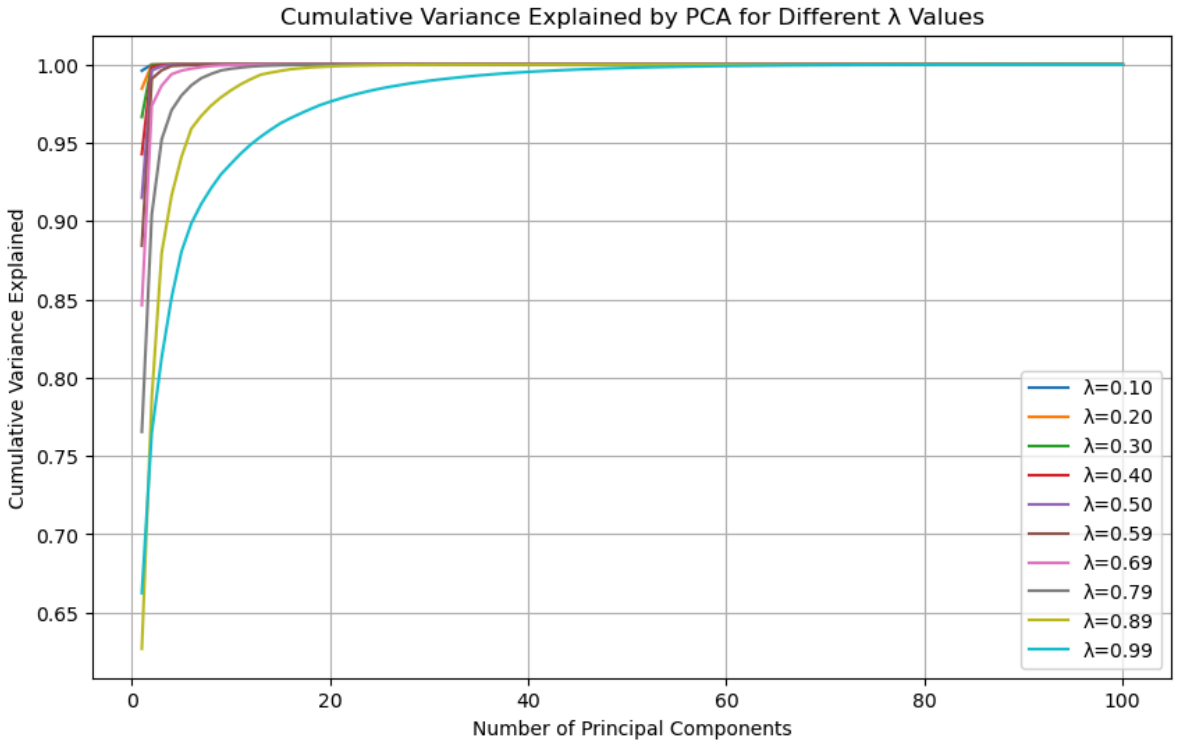
Problem 5

Given the stock return data in DailyReturns.csv.

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 (λ=0.97):
 [[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 covariancematrix?

Answer:

A higher λ (e.g., 0.97) places more weight on recent data while still incorporating past observations, leading to a smoother and more stable covariance matrix. This results in lower variability in covariance estimates, making the model less sensitive to short-term fluctuations. In contrast, a lower λ would emphasize recent changes more strongly, causing the covariance structure to shift more dynamically over time.

From the PCA results, we see that for high λ, only a few principal components explain most of the variance, indicating a well-structured covariance matrix with strong correlations among assets. Lower λ values would distribute variance across more components, making the model more reactive to recent market shifts. Choosing λ depends on the balance between stability and responsiveness—higher values are useful for long-term trends, while lower values are better for short-term adaptations.

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.
Using the covariance matrix found in problem6.csv
```

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

Answer:

=== Part A: Cholesky Simulation ===

Adjusted Matrix Eigenvalues (first 10): [0.05796942+0.j 0.05575631+0.j 0.05340276+0.j 0.05286599+0.j 0.0511631 +0.j 0.05039024+0.j 0.04708343+0.j 0.04626092+0.j 0.04513198+0.j 0.04424075+0.j]

- Is Adjusted Matrix SPD? True
- Cholesky Simulation Shape: (10000, 500)

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

Answer:

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?

Answer:

=== Part C: Frobenius Norm Comparison ===

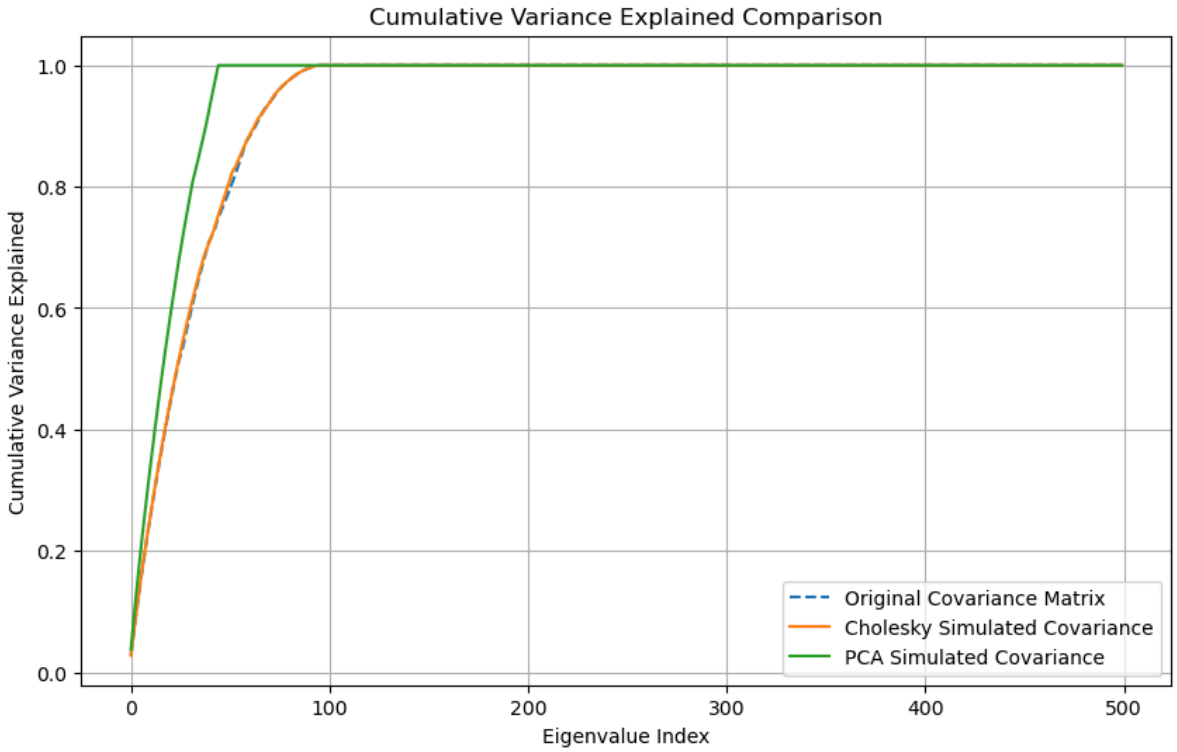
Frobenius Norm (Cholesky): 0.0206

Frobenius Norm (PCA): 0.2603

- Cholesky Simulation Frobenius Norm: 0.0206 (Lower, meaning closer to the original covariance). Cholesky is much more precise in preserving the original covariance structure.
- PCA Simulation Frobenius Norm: 0.2603 (Higher, meaning more deviation from the original covariance). PCA loses some covariance information because it reduces dimensionality by retaining only 75% of variance.

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

Answer



- Cholesky closely matches the original covariance, preserving most of the structure.
- PCA deviates after a certain number of components, reflecting its loss of variance due to dimensionality reduction.

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

Answer:

- Cholesky Simulation Time: 0.161154 seconds
- PCA Simulation Time: 0.023059 seconds

- **PCA is significantly faster than Cholesky**, taking only **~14% of the time** required for Cholesky.

- This difference is expected because **PCA reduces the number of dimensions**, while Cholesky operates on the **full covariance matrix**.
- **Cholesky requires full matrix factorization**, which is computationally more expensive than the **eigenvalue-based transformation in PCA**.

F. Discuss the tradeoffs between the two methods.

Both **Cholesky decomposition** and **PCA-based simulation** are useful methods for generating multivariate normal samples, but they have different strengths and weaknesses.

1️⃣ Cholesky Simulation: High Accuracy, Higher Cost

The **Cholesky method** maintains the **exact covariance structure** of the original data. Since it factorizes the covariance matrix directly, it produces simulations that **perfectly preserve correlations** between variables. However, it **requires more computational effort**, especially when dealing with very large datasets, and it **only works if the covariance matrix is positive semi-definite (PSD)**.

✅ Pros

- **Exact covariance structure is preserved**
- **More reliable for financial modeling and risk analysis**
- **No variance loss, all information is retained**

❌ Cons

- **Computationally expensive for large matrices**
- **Requires a PSD matrix, otherwise adjustments are needed**
- **Not ideal if dimensionality reduction is needed**

2️⃣ PCA Simulation: Faster, but Less Accurate

The **PCA method** reduces the number of dimensions while keeping the **most important variance**. This makes it **faster and more memory-efficient**, but it also **loses some covariance structure** in the process. PCA is particularly useful when dealing with **high-dimensional data** where keeping all variables is unnecessary. However, the **simulated covariance matrix is only an approximation** of the original.

✅ Pros

- **Faster computation, especially for high-dimensional datasets**
- **Does not require a PSD matrix (handles ill-conditioned data well)**
- **Can reduce noise by removing less important components**

❌ Cons

- **Loses some covariance structure due to dimensionality reduction**
- **Not ideal if exact correlations must be maintained**
- **Requires choosing a variance threshold (e.g., 75%), which affects accuracy**

3️⃣ Side-by-Side Comparison

Feature	Cholesky Simulation	PCA-Based Simulation
Speed	❌ Slower for large matrices	✅ Faster due to dimensionality reduction
Accuracy	✅ More accurate, exact covariance	❌ Less accurate, variance loss occurs
Handles Non-PSD Matrix?	❌ No, requires adjustment	✅ Yes, works naturally
Dimensionality Reduction	❌ No, keeps all variables	✅ Yes, reduces dataset size

Feature	Cholesky Simulation	PCA-Based Simulation
Best for	When accuracy matters most (finance, risk)	When speed and efficiency matter (ML, high-dim data)

4️⃣ Final Thoughts

- **Use Cholesky** when we need **precise covariance replication**, and computational cost is not a major issue.
- **Use PCA** when **reducing dimensionality** is important, even if it means sacrificing some accuracy.
- If **storage and speed are the priority**, PCA is a better choice.
- If **financial modeling or risk assessment** is the goal, Cholesky is preferred.
