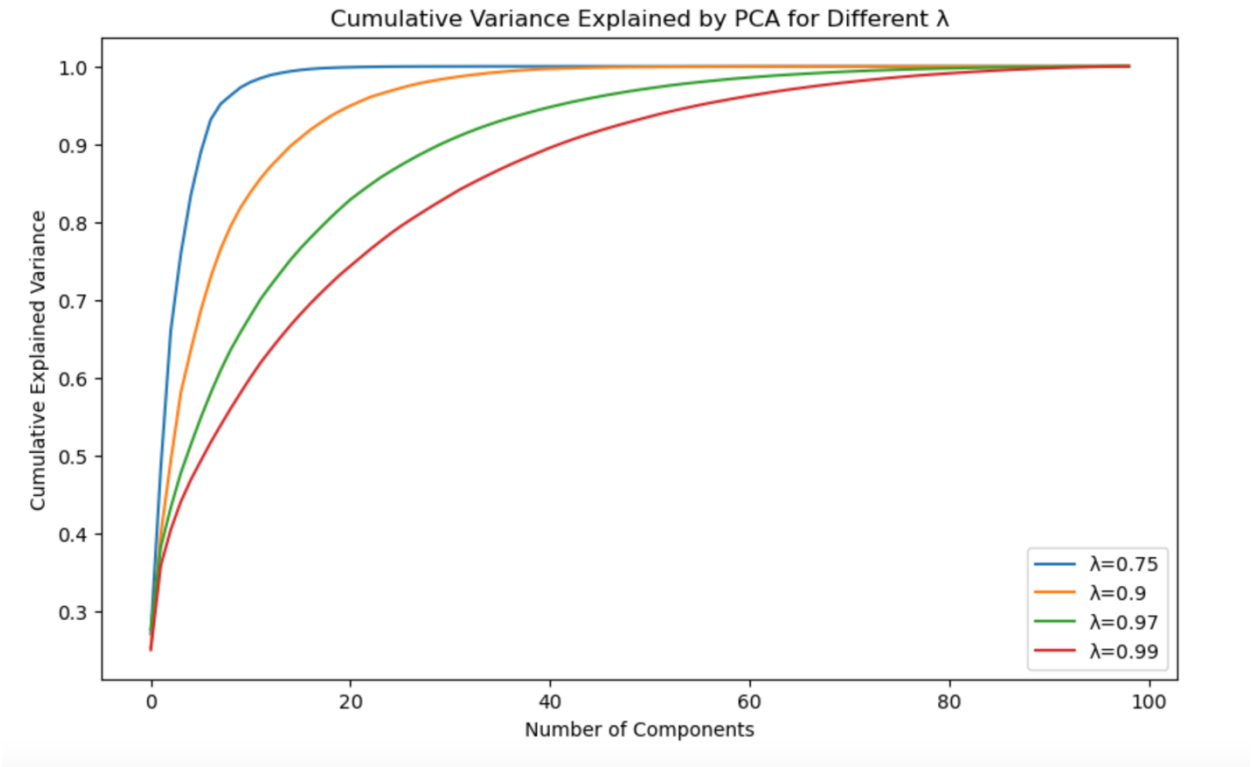
Problem 1



When λ is lower, more recent data dominates, causing the first few eigenvalues to explain most of the variance, and the covariance matrix becomes lower rank.
When λ is higher, both recent and older data contribute more evenly, resulting in a full-rank covariance matrix where the variance is spread across more principal components.
Summary: As λ decreases, the covariance matrix increasingly emphasizes more recent data points due to the higher weights assigned to the most recent observations.

Problem 2

Comparing methods for N=500 with Frobenius norm:

- near_psd Frobenius norm: 0.063644, Time: 0.027349s
- Higham's method Frobenius norm: 0.063643, Time: 0.037242s

Comparing methods for N=1000 with Frobenius norm:

- near_psd Frobenius norm: 0.063972, Time: 0.126580s
- Higham's method Frobenius norm: 0.063971, Time: 0.220852s

Comparing methods for N=1500 with Frobenius norm:

- near_psd Frobenius norm: 0.064082, Time: 0.408260s
- Higham's method Frobenius norm: 0.064081, Time: 0.831507s

In terms of accuracy (based on the Frobenius norm), both methods perform nearly identically regardless of the matrix size. This suggests that both near_psd() and Higham's method are equally effective at producing a positive semi-definite (PSD) matrix that closely matches the original matrix.

The near_psd method is consistently faster than Higham's method across all matrix sizes. However, as N increases, both methods require more runtime for larger matrices.

near_psd:

- **Pros**:
  - Faster execution, especially for larger matrices.
  - Produces a very close approximation to the original matrix (as seen by the small Frobenius norm).
- **Cons**:
  - May be less robust for certain edge cases or numerical stability.

Higham's Method:

- **Pros**:
  - A well-established and mathematically rigorous approach to finding the nearest PSD matrix.
  - Slightly more flexible and can handle a broader range of cases.
- **Cons**:
  - Slower than near_psd, especially as the matrix size increases.
  - For larger matrices, the increased runtime may become a bottleneck.

Use near_psd when you need fast results, especially when working with larger matrices.

Use Higham's Method when you think accuracy and robustness are more important than speed. And when you are working with small-to-medium-sized matrices, where the increased runtime is not a significant concern.

Problem 3

Simulating for Pearson Corr + Pearson Var

- **Direct Simulation: Frobenius Norm = 0.000000, Time = 0.089982s**

- **Covariance matrix of simulated values (Direct):**
  [[4.00581531e-08 1.50546940e-08 2.41822170e-08 ... 6.26648490e-09
  9.15190051e-09 5.23934202e-09]
  [1.50546940e-08 2.47455163e-08 3.15508860e-08 ... 6.55106847e-09
  5.87640841e-09 5.46593507e-09]
  [2.41822170e-08 3.15508860e-08 1.06012871e-07 ... 1.24078226e-08
  1.54050115e-08 1.11758569e-08]
  ...
  [6.26648490e-09 6.55106847e-09 1.24078226e-08 ... 1.36216452e-07
  6.22329944e-08 9.30011461e-09]
  [9.15190051e-09 5.87640841e-09 1.54050115e-08 ... 6.22329944e-08
  1.06553414e-07 7.06682525e-09]
  [5.23934202e-09 5.46593507e-09 1.11758569e-08 ... 9.30011461e-09
  7.06682525e-09 1.34462480e-08]]

- **PCA (100%): Frobenius Norm = 0.000000, Time = 0.081863s**
- **Covariance matrix of simulated values (PCA 100%):**
  [[4.02045128e-08 1.52422784e-08 2.42832541e-08 ... 6.19841603e-09
  9.94669005e-09 5.18652139e-09]
  [1.52422784e-08 2.44924924e-08 3.13658287e-08 ... 5.98278992e-09
  6.62916491e-09 5.51540224e-09]
  [2.42832541e-08 3.13658287e-08 1.05802614e-07 ... 1.25892934e-08
  1.79900797e-08 1.15180714e-08]
  ...
  [6.19841603e-09 5.98278992e-09 1.25892934e-08 ... 1.36227975e-07
  6.24520961e-08 9.94131423e-09]
  [9.94669005e-09 6.62916491e-09 1.79900797e-08 ... 6.24520961e-08
  1.07789728e-07 7.69576873e-09]
  [5.18652139e-09 5.51540224e-09 1.15180714e-08 ... 9.94131423e-09
  7.69576873e-09 1.37768422e-08]]

- **PCA (75%): Frobenius Norm = 0.000001, Time = 0.015762s**
- **PCA (50%): Frobenius Norm = 0.000001, Time = 0.006105s**

Simulating for Pearson Corr + EWCM Var

- **Direct Simulation: Frobenius Norm = 0.000000, Time = 0.083780s**
- **Covariance matrix of simulated values (Direct):**
  [[3.10413076e-08 1.49315490e-08 2.83408998e-08 ... 5.60367510e-09

1.21815288e-08 2.93909200e-09]
[1.49315490e-08 3.12533805e-08 4.65831123e-08 ... 6.42142280e-09
1.10194788e-08 4.02997777e-09]
[2.83408998e-08 4.65831123e-08 1.80370513e-07 ... 1.54791030e-08
3.27895142e-08 9.59581383e-09]
...
[5.60367510e-09 6.42142280e-09 1.54791030e-08 ... 1.44062768e-07
9.00586025e-08 6.28342857e-09]
[1.21815288e-08 1.10194788e-08 3.27895142e-08 ... 9.00586025e-08
2.13028463e-07 6.96797621e-09]
[2.93909200e-09 4.02997777e-09 9.59581383e-09 ... 6.28342857e-09
6.96797621e-09 5.97314416e-09]]

- **PCA (100%): Frobenius Norm = 0.000000, Time = 0.085950s**
- **Covariance matrix of simulated values (PCA 100%):**
[[3.15023534e-08 1.50983706e-08 2.82616987e-08 ... 4.51182112e-09
1.06857385e-08 3.13056147e-09]
[1.50983706e-08 3.14221884e-08 4.61578468e-08 ... 6.12120100e-09
9.35465085e-09 4.07682354e-09]
[2.82616987e-08 4.61578468e-08 1.79002329e-07 ... 1.47970797e-08
2.96738746e-08 9.97807139e-09]
...
[4.51182112e-09 6.12120100e-09 1.47970797e-08 ... 1.39138342e-07
8.80827965e-08 6.38096494e-09]
[1.06857385e-08 9.35465085e-09 2.96738746e-08 ... 8.80827965e-08
2.11629623e-07 7.06184557e-09]
[3.13056147e-09 4.07682354e-09 9.97807139e-09 ... 6.38096494e-09
7.06184557e-09 6.08569563e-09]]

- **PCA (75%): Frobenius Norm = 0.000000, Time = 0.014837s**
- **PCA (50%): Frobenius Norm = 0.000001, Time = 0.006649s**

Simulating for EWCM Corr + EWCM Var

- **Direct Simulation: Frobenius Norm = 0.000155, Time = 0.079652s**
- **Covariance matrix of simulated values (Direct):**
[[1.76201937e-04 9.66734024e-05 1.17336619e-04 ... 2.54894047e-05

9.28931616e-05 1.86498171e-05]
[9.66734024e-05 1.78529860e-04 1.55638718e-04 ... 3.00189572e-05
4.46658980e-05 3.95997327e-05]
[1.17336619e-04 1.55638718e-04 4.19044133e-04 ... 2.07145285e-05
6.72479052e-05 7.96669180e-05]
...
[2.54894047e-05 3.00189572e-05 2.07145285e-05 ... 3.74807159e-04
2.08830564e-04 4.54925841e-05]
[9.28931616e-05 4.46658980e-05 6.72479052e-05 ... 2.08830564e-04
4.60638813e-04 4.61052137e-05]
[1.86498171e-05 3.95997327e-05 7.96669180e-05 ... 4.54925841e-05
4.61052137e-05 7.84648874e-05]]

- **PCA (100%): Frobenius Norm = 0.000212, Time = 0.082374s**
- **Covariance matrix of simulated values (PCA 100%):**
[[1.81214279e-04 9.73381135e-05 1.20567285e-04 ... 2.29043724e-05
9.31322096e-05 1.91464048e-05]
[9.73381135e-05 1.76261222e-04 1.54811685e-04 ... 3.11805230e-05
4.58072180e-05 3.92105551e-05]
[1.20567285e-04 1.54811685e-04 4.25585998e-04 ... 2.18077783e-05
6.90694610e-05 7.88828971e-05]
...
[2.29043724e-05 3.11805230e-05 2.18077783e-05 ... 3.82203101e-04
2.10528286e-04 4.90799688e-05]
[9.31322096e-05 4.58072180e-05 6.90694610e-05 ... 2.10528286e-04
4.65260056e-04 4.70766768e-05]
[1.91464048e-05 3.92105551e-05 7.88828971e-05 ... 4.90799688e-05
4.70766768e-05 7.90515735e-05]]

- **PCA (75%): Frobenius Norm = 0.001088, Time = 0.013378s**
- **PCA (50%): Frobenius Norm = 0.002216, Time = 0.006019s**

Simulating for EWCM Corr + Pearson Var

- **Direct Simulation: Frobenius Norm = 0.000331, Time = 0.074833s**
- **Covariance matrix of simulated values (Direct):**
[[2.25610439e-04 9.66144454e-05 1.01273896e-04 ... 2.33968809e-05

7.06494414e-05 2.99846708e-05]
[9.66144454e-05 1.39204817e-04 1.05698012e-04 ... 2.54455541e-05
2.78910658e-05 4.97751506e-05]
[1.01273896e-04 1.05698012e-04 2.50745635e-04 ... 1.26038789e-05
3.42703781e-05 8.87791925e-05]
...
[2.33968809e-05 2.54455541e-05 1.26038789e-05 ... 3.67265666e-04
1.46935582e-04 6.85750273e-05]
[7.06494414e-05 2.78910658e-05 3.42703781e-05 ... 1.46935582e-04
2.32555496e-04 4.98111295e-05]
[2.99846708e-05 4.97751506e-05 8.87791925e-05 ... 6.85750273e-05
4.98111295e-05 1.72075030e-04]]

- **PCA (100%): Frobenius Norm = 0.000323, Time = 0.080953s**
- **Covariance matrix of simulated values (PCA 100%):**
[[2.27430134e-04 9.65678374e-05 1.03435364e-04 ... 2.86819075e-05
7.50290479e-05 3.31266674e-05]
[9.65678374e-05 1.38926621e-04 1.07490539e-04 ... 2.78281541e-05
3.03334599e-05 5.20201940e-05]
[1.03435364e-04 1.07490539e-04 2.53907083e-04 ... 1.94147154e-05
3.82560462e-05 9.26158258e-05]
...
[2.86819075e-05 2.78281541e-05 1.94147154e-05 ... 3.63192394e-04
1.45444197e-04 7.01466470e-05]
[7.50290479e-05 3.03334599e-05 3.82560462e-05 ... 1.45444197e-04
2.31298011e-04 5.13205769e-05]
[3.31266674e-05 5.20201940e-05 9.26158258e-05 ... 7.01466470e-05
5.13205769e-05 1.77301545e-04]]

- **PCA (75%): Frobenius Norm = 0.001729, Time = 0.013487s**
- **PCA (50%): Frobenius Norm = 0.003729, Time = 0.005994s**

Trade-offs Between Time and Accuracy:

- **Direct Simulation** and **PCA (100%)**:
  - **Accuracy**: These methods provide the best accuracy with Frobenius norms close to zero, meaning they closely match the input covariance matrix.
  - **Speed**: They are relatively slower compared to **PCA (75%)** and **PCA (50%)**, but the difference is usually within a few milliseconds.
- **PCA (75%)** and **PCA (50%)**:
  - **Accuracy**: These methods lose some accuracy due to the reduction in the number of principal components retained, especially for complex covariance structures.
  - **Speed**: They are much faster, especially **PCA (50%)**.