Numerical Linear Algebra: 10th December 2023

Project 2: SVD applications

Dafni Tziakouri

1. Least Squares problem

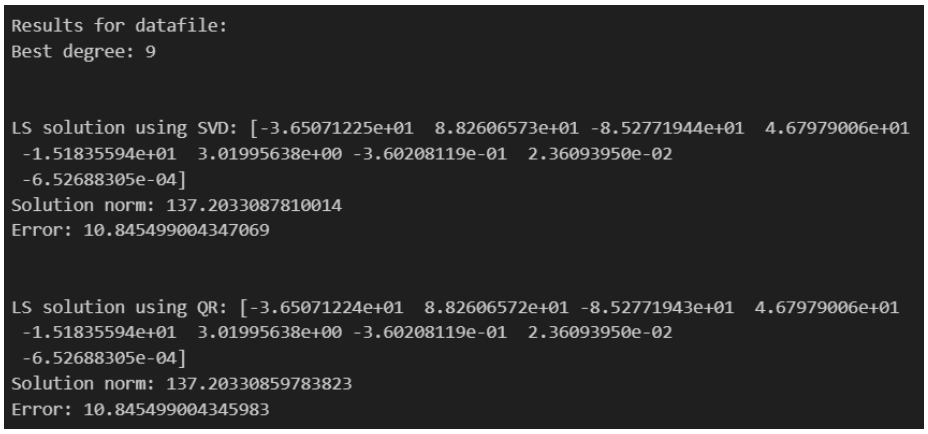
Write a program to solve the LS problem using SVD. Compute the LS solution for the datasets datafile and datafile2.csv that were used in pr4: QR factorization and least square problems. Compare the results using SVD with those obtained from the QR solution of the LS problems.

Initially, we will establish two functions dedicated to solving the Least Squares problem. One function will tackle the problem through Singular Value Decomposition (SVD), while the other will employ QR factorization.

In addressing the problem through QR factorization, our approach hinges on the matrix rank, denoted as rank(A). This leads to two scenarios: one where the matrix is full-rank and another where it is rank-deficient. In the case of a full-rank matrix, we proceed with QR factorization followed by solving the system through back substitution. On the other hand, if the matrix is rank-deficient, we choose to employ QR factorization with pivoting. This not only addresses the rank deficiency but also enhances the numerical stability of the solution.

The optimal degree identified as the best performer in our analysis is 9. This suggests that a polynomial of degree 9 provides a good trade-off between complexity and accuracy in fitting the given data. It's essential to highlight that the considered degrees range from 3 to 10, and the selection criterion prioritizes the degree with the least error in the solution, a criterion consistently met at approximately 10.8455 for both the QR and SVD solving methods. (“datafile”)

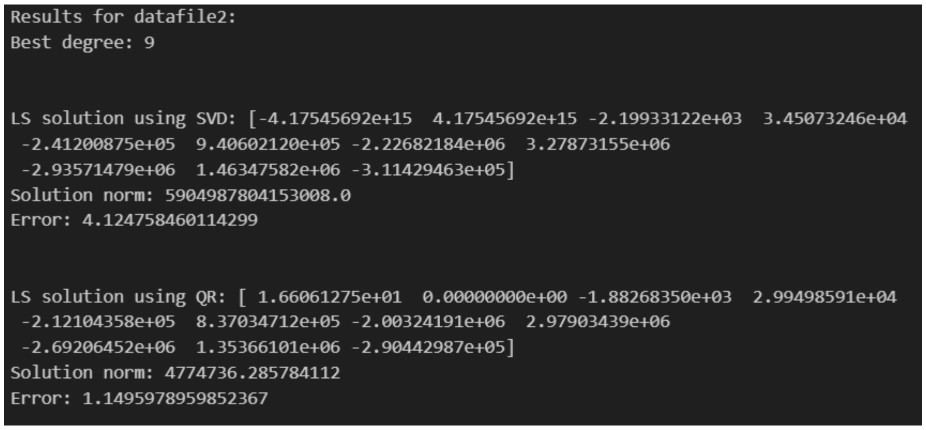
Now, let's analyze the outcomes for both datasets. In the instance of the first dataset ("datafile"), both the SVD and QR methods yield an error of 10.8455, accompanied by a solution with a norm of 137.2033. Turning to the second dataset ("datafile2"), we observe an error of 1.1496, coupled with a solution norm of approximately 4774736.



```
Results for datafile:
Best degree: 9


LS solution using SVD: [-3.65071225e+01  8.82606573e+01 -8.52771944e+01  4.67979006e+01
 -1.51835594e+01  3.01995638e+00 -3.60208119e-01  2.36093950e-02
 -6.52688305e-04]
Solution norm: 137.2033087810014
Error: 10.845499004347069


LS solution using QR: [-3.65071224e+01  8.82606572e+01 -8.52771943e+01  4.67979006e+01
 -1.51835594e+01  3.01995638e+00 -3.60208119e-01  2.36093950e-02
 -6.52688305e-04]
Solution norm: 137.20330859783823
Error: 10.845499004345983
```



```
Results for datafile2:
Best degree: 9


LS solution using SVD: [-4.17545692e+15  4.17545692e+15 -2.19933122e+03  3.45073246e+04
 -2.41200875e+05  9.40602120e+05 -2.22682184e+06  3.27873155e+06
 -2.93571479e+06  1.46347582e+06 -3.11429463e+05]
Solution norm: 5904987804153008.0
Error: 4.124758460114299


LS solution using QR: [ 1.66061275e+01  0.00000000e+00 -1.88268350e+03  2.99498591e+04
 -2.12104358e+05  8.37034712e+05 -2.00324191e+06  2.97903439e+06
 -2.69206452e+06  1.35366101e+06 -2.90442987e+05]
Solution norm: 4774736.285784112
Error: 1.1495978959852367
```

The SVD method produces similar solutions for both datasets. However, for “datafile2”, the solution's norm is extremely high, indicating potential numerical instability. This might be attributed to ill-conditioning of the matrix, leading to a significant amplification of errors.

The QR method yields stable solutions for both datasets. However, for “datafile2”, the solution's norm is significantly smaller than that obtained using the SVD method, suggesting improved numerical stability.

Singular value decomposition appears to operate without explicit rank considerations, making it more straightforward to handle. However, its precision in delivering results may be compromised. In contrast, when we have knowledge of the matrix dimensions, allowing us to determine the simplicity or complexity of the solution, QR factorization becomes a more favorable option. This is because the procedure adapts based on whether the matrix is full rank or rank deficient.

2. Graphics compression

2.1: The SVD factorization has the property of giving the best low rank approximation matrix with respect to the Frobenius and/or the 2-norm to a given matrix. State properly the previous statement and write down the corresponding proofs for the Frobeni.

The Singular Value Decomposition (SVD) factorization states that for any real or complex matrix A of size mxn, there exist matrices U (orthogonal matrix of size mxn), Σ (diagonal matrix of size mxn), and V^T(conjugate transpose of V , where V is an orthogonal matrix of size mxn) such that: $A = U\Sigma V^T$

The singular values in Σ are ordered in decreasing order:$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $\mathrm{p} = \min(\mathrm{m}, \mathrm{n})$.

The best low-rank approximation of A of rank k is given by: $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, where u_i and v_i are the i-th columns of U and V, respectively.

To show that the matrix A_k is the best rank k approximation to A in both the Frobenius and the 2-norm, we may follow the steps of the proof found in [1] reference.

First we show that the matrix $A_k$is the best rank k approximation to A in the Frobenius norm. For that we will need the following 2 lemmas:

Lemma 4.6: *The rows of A_k are the projections of the rows of A onto the subspace V_k spanned by the first k singular vectors of A.*

Lemma 4.8: $\|\mathrm{A} - \mathrm{A_k}\|_2^2 = \sigma_{\mathrm{k}+1}^2$.

Proof for the above lemmas can be found in reference [1].

Theorem 4.7: For any matrix B of rank at most k, $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F} \leq \|\mathrm{A} - \mathrm{B}\|_\mathrm{F}$

Proof: Let B minimize $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F}^2$ among all rank k or less matrices. Let V be the space spanned by the rows of B. The dimension of V is at most k. Since B minimizes $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F}^2$, it must be that each row of B is the projection of the corresponding row of A onto V , otherwise replacing the row of B with the projection of the corresponding row of A onto V does not change V and hence the rank of B but would reduce $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F}^2$. Since each row of B is the projection of the corresponding row of A, it follows that $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F}^2$ is the sum of squared distances of rows of A to V. Since A_k minimizes the sum of squared distance of rows of A to any k -dimensional subspace, it follows that $\|\mathrm{A} - \mathrm{A_k}\|_\mathrm{F} \leq \|\mathrm{A} - \mathrm{B}\|_\mathrm{F}$. ■

Theorem 4.9: Let A be an $\mathrm{n} \times \mathrm{d}$ matrix. For any matrix B of rank at most k,

$$\|\mathrm{A}-\mathrm{A}_{\mathrm{k}}\|_2 \leq \|\mathrm{A}-\mathrm{B}\|_2$$

Proof: If A is of rank k or less, the theorem is obviously true since$\|\mathrm{A}-\mathrm{A}_{\mathrm{k}}\|_2 = 0$. Thus assume that A is of rank greater than k. By Lemma 4.8, $\|\mathrm{A}-\mathrm{A}_{\mathrm{k}}\|_2^2 = \sigma_{\mathrm{k}+1}^2$. Now suppose there is some matrix B of rank at most k such that B is a better 2-norm approximation to A than Ak. That is, $\|\mathrm{A}-\mathrm{B}\|_2 < \sigma_{k+1}$. The null space of B, Null (B), (the set of vectors v such that Bv = 0) has dimension at least d − k. Let v1, v2,..., vk+1 be the first k + 1 singular vectors of A. By a dimension argument, it follows that there exists a $\mathrm{z} \neq \mathbf{0}$ in

$$\mathrm{Null\ (B)} \cap \mathrm{Span}\ \{\mathrm{v1}, \mathrm{v2}, \ldots, \mathrm{vk}+1\}\ .$$

Scale z so that |z| = 1. We now show that for this vector z, which lies in the space of the first k + 1 singular vectors of A, that$(\mathrm{A}-\mathrm{B})\,\mathrm{z} \geq \sigma_{k+1}$. Hence the 2-norm of A − B is at least σ_{k+1} contradicting the assumption that $\|\mathrm{A}-\mathrm{B}\|_2 < \sigma_{k+1}$. First

$$\|\mathrm{A}-\mathrm{B}\|_2^2 \geq |(\mathrm{A}-\mathrm{B})\mathrm{z}|^2.$$

Since$\mathrm{Bz} = 0$,

$$\|\mathrm{A}-\mathrm{B}\|_2^2 \geq |\mathrm{Az}|^2\ .$$

Since z is in the Span {v1, v2,..., vk+1}

$$|\mathrm{Az}|^2 = \left|\sum_{i=1}^{n} \sigma_i u_i v_i^T z\right|^2 = \sum_{i=1}^{n} \sigma_i{}^2\,(v_i^T z)^2 = \sum_{i=1}^{k+1} \sigma_i{}^2\,(v_i^T z)^2 \geq \sigma_{k+1}{}^2 \sum_{i=1}^{k+1} (v_i^T z)^2 = \sigma_{k+1}{}^2.$$

It follows that

$$\|\mathrm{A}-\mathrm{B}\|_2^2 \geq \sigma_{k+1}{}^2,$$

contradicting the assumption that $\|\mathrm{A}-\mathrm{B}\|_2 < \sigma_{k+1}$. This proves the theorem. ■

2.2: Use the previous results to obtain a lossy compressed graphic image from a .jpeg graphic file. A .jpeg graphic file can be read as a matrix using the function scipy.ndimage.imread(). Use SVD decomposition to create approximations of lower rank to the image. Compare different approximations. The function scipy.misc.imsave() can be useful to save the approximated graphic files as .jpeg. The code must generate different compressed files for a given graphic file. Hence, to organize the output files, the name of the compressed file must reflect the percentage of the Frobenius norm captured in each compressed file. Use different .jpeg images (of different sizes and having letters or pictures) and compare results.

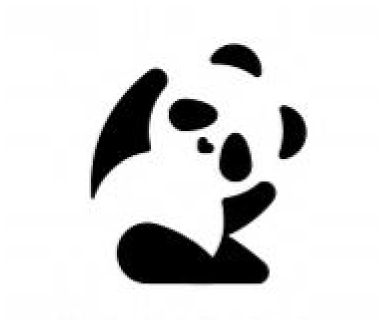
For this segment of the exercise, we opted to employ two distinct black and white images: one featuring a panda drawing and the other incorporating text in the form of the alphabet. We deemed these images to be particularly intriguing for comprehending the approximations involved.

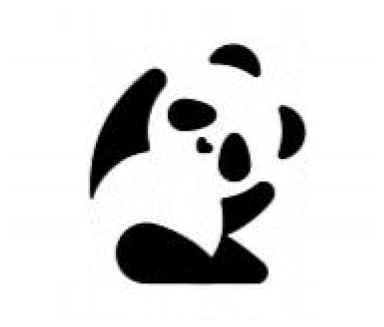
To generate diverse approximations, we employed five distinct ranks: 1, 5, 25, 50, and 100. Experimenting with these various values enabled us to observe and compare distinct outcomes. The function implemented in this phase yields the Frobenius norm percentage (%) captured by the approximation, the relative error of the norm, and the image.

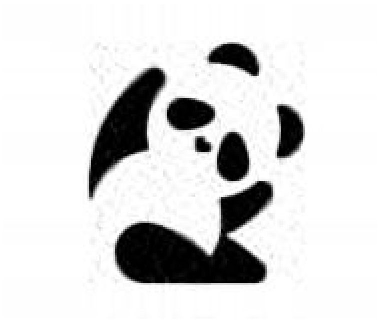
Now, we will showcase the images utilized along with the varied results obtained for the various percentages of Frobenius norm captured.

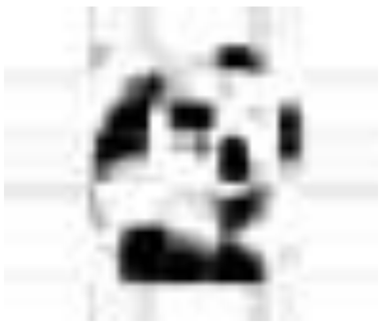
<u>Image1-Panda:</u>

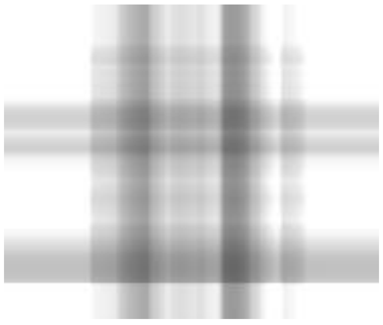
(a) Original image


(b) Rank-100


(c) Rank-50

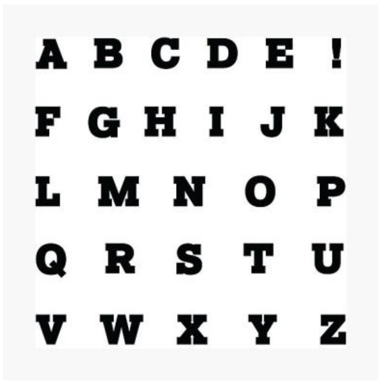

(d) Rank-25


(e) Rank-5

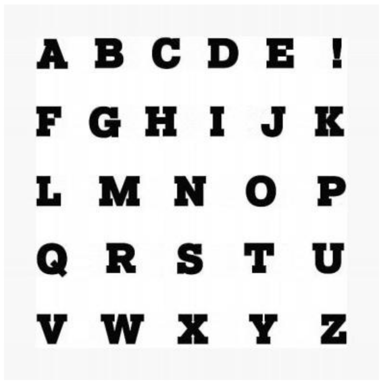

(f) Rank-1

At the lowest rank, the discernibility of the panda's shape is notably challenging. However, with each incremental increase in rank approaching 100, the visibility of the panda's form improves, reflected in the percentage of Frobenius norm approaching 100%. Simultaneously, the relative error exhibits a decreasing trend. In addition, we can notice that the biggest change happens between ranks 5 and 1.

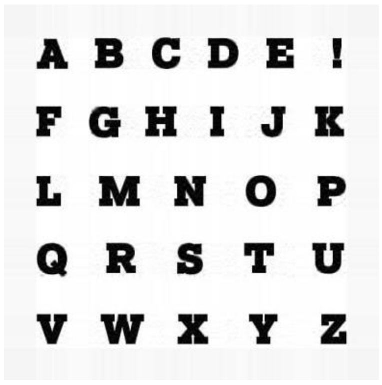
<u>Image2- Alphabet:</u>



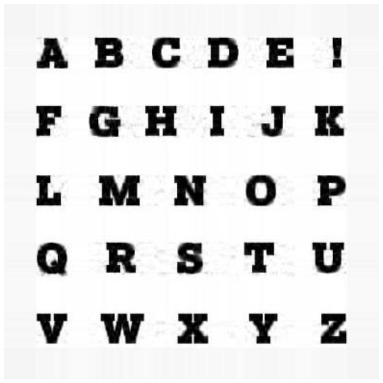
(a) Original image



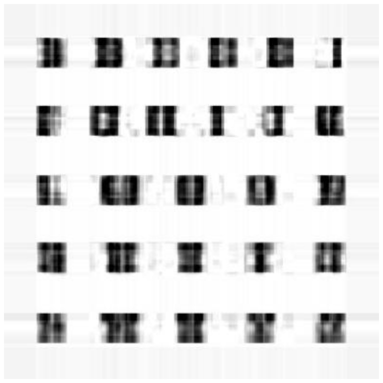
(b) Rank-100



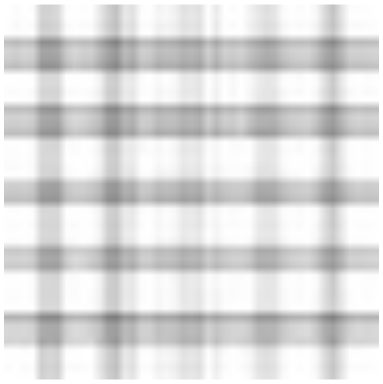
(c) Rank-50



(d) Rank-25



(e) Rank-5



(f) Rank-1

As anticipated, the structure of Image 2 (alphabet) closely mirrors that of Image 1 (panda). It becomes evident that as we increase the rank, comprehension of the letters becomes significantly more accessible. Larger rank values, such as 50 and 100, notably enhance the clarity of the letters compared to the challenges faced with lower rank values like 5 and 1.

These experiments underscore the importance of possessing a thorough understanding of the image being processed when undertaking image compression. The outcomes reveal that a more nuanced knowledge of the image significantly influences the effectiveness of compression techniques.

3. Principal component analysis (PCA)

We will present a comprehensive explanation of the results and processes for both components of this section in a unified manner.

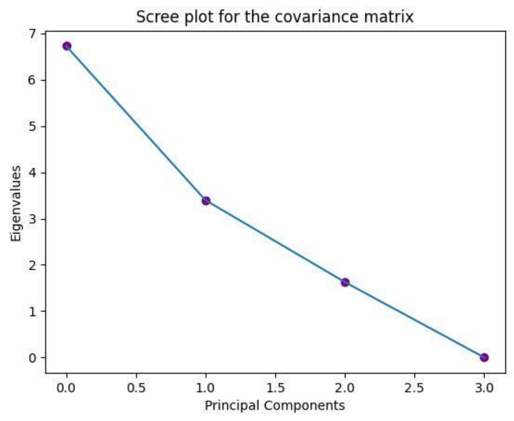
1. The file example.dat contains a dataset of 16 observations of 4 variables. Perform PCA analysis using both the covariance matrix and the correlation matrix. The code must write down the portion of the total variance accumulated in each of the principal components, the standard deviation of each of the principal components and the expression of the original dataset in the new PCA coordinates.

2. The file RCsGoff.csv contains data from the experiment reported in [3]. Each observation consists in measuring the amount of a total number of 58581 genes. There are a total of 20 observations grouped by day of observation. The code must perform a PCA analysis on the covariance matrix. The output file must contain rows with the following format Sample,PC1,PC2,. . . ,PC20,Variance where Sample stands for day0 rep1,...,day18 rep3 (i.e. the different observations) and PCi stands for the coordinate of the principal component of the observation. Finally variance is the portion of the total variance accumulated in each of the principal components.

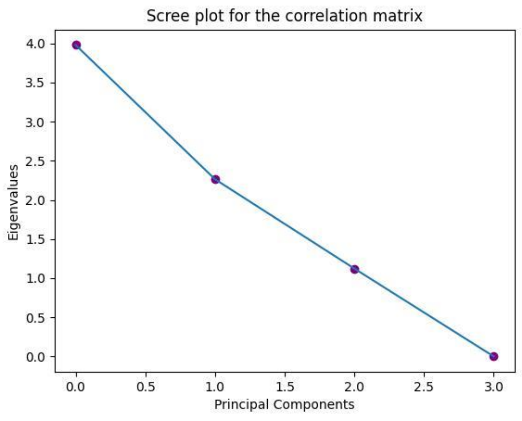
We will try to explain the datasets using both the covariance matrix and the correlation matrix. First of all we will briefly explain the methods used to for obtaining the Principal Components:

- *Scree plots:* offer a visual representation of matrix eigenvalues, ordered by size. Those to the left of the curve's "elbow" are considered significant, while those on the right are less crucial.
- *The Kaiser Rule:* involves retaining only principal components with eigenvalues greater than 1.
- *The 3/4 rule:* involves determining the minimum number of Principal Components necessary to encompass 75% of the cumulative variance.

Let’s start with the “example.dat” dataset, we will plot the scree plots pointing out the “elbows”:



(a) Covariance Matrix



(b) Correlation Matrix

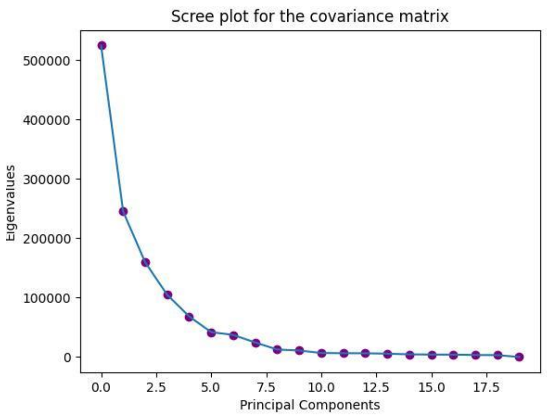
The dataset has a small number of eigenvectors and eigenvalues. However, by analyzing the covariance matrix and correlation matrix, it seems feasible to discard two of them for both matrices. Consequently, only two principal components would be retained.

Following the Kaiser rule, each matrix would be defined by three principal components. It's worth noting that the Kaiser Rule typically recommends selecting more components compared to the "elbow" method in the Scree plot.

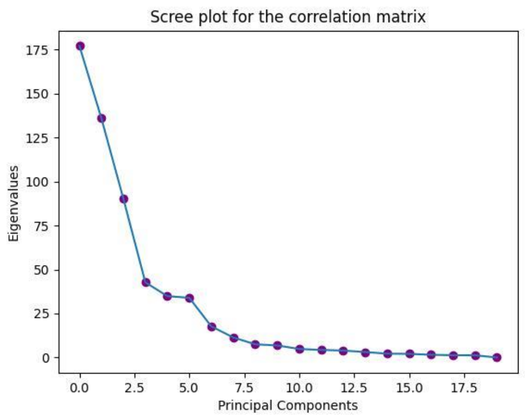
In the given dataset, the covariance matrix exhibits a principal component that captures over 75% of the total variance independently. As a result, the 3/4 Rule applied to this matrix suggests retaining only 1 Principal Component for this dataset. Conversely, for the correlation matrix, it is advisable to retain 2 Principal Components according to the 3/4 Rule. This assessment has been validated through code.

The preceding analysis was conducted using code, and you can examine the details in the provided Python file.

Repeating the same analysis for the RCsGoff.csv dataset:



(a) Covariance Matrix



(b) Correlation Matrix

The Scree Plot analysis recommends retaining 5 Principal Components for the Covariance Matrix. However, the Correlation Matrix presents some irregularities, featuring two small "elbows." Therefore, considering the second "elbow," it is advisable to retain 6 or 7 Principal Components for the correlation matrix.

The application of the Kaiser Rule to the covariance matrix and correlation matrix identifies 19 Principal Components associated with eigenvalues exceeding 1.

Finally, following the 3/4 rule, we are directed to retain 2 Principal Components for both matrices. This decision stems from the fact that the top variance Principal Components accumulate up to 72% and 15% variance in the case of the covariance matrix, and 50% and 30% in the case of the correlation matrix.

References:

[1] https://www.cs.princeton.edu/courses/archive/spring12/cos598C/svdchapter.pdf