Numerical Linear Algebra - Project 2

Sunday, December 10th, 2023 at 11:55PM

Madison E. Chester

1. **Least Squares Problem**

   The least squares problem involves finding the vector, x, that minimizes the residual vector Ax-b in a linear system where A is an mxn matrix and b is a given vector. In this exercise, two methods are employed to solve the least squares problem: singular value decomposition (SVD) and QR factorization.

   SVD -

   SVD decomposes the matrix A into three matrices, U, **Σ**, and V^T. The least squares solution can be obtained using these three matrices. The least squares solution vector, x_svd, is calculated as follows: x_svd = V^T * **Σ**^(-1) * U^T * b.

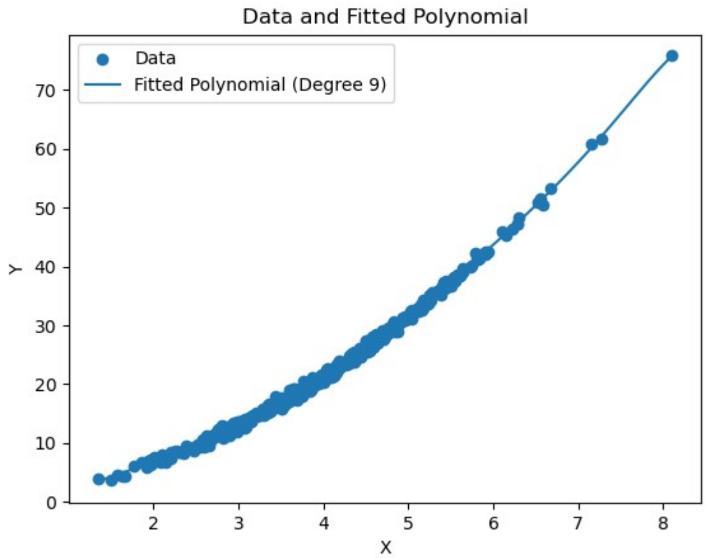
   QR Factorization -

   QR factorization decomposes the matrix A into the product of an orthogonal matrix Q and an upper triangular matrix R. The least squares solution can be obtained by solving the triangular system. In the case of a rank-deficient matrix, pivoting is used. This addresses the rank deficiency and enhances the numerical stability of the solution simultaneously.

   Data -

   Datasets include dades.txt and dades_regressio.csv. The least squares problem is solved for each dataset with degrees ranging from 3 to 10, and the selection criteria prioritizes the degree with the smallest error. Finding the best degree for fitting involves a tradeoff between complexity and accuracy. It should be noted that for dades_regressio.csv the degree to fit was already defined, so performing the search was not explicitly necessary.

   Results -




```
Best degree using SVD for datafile: 9
LS solution using SVD for datafile (Degree 9):
[-3.65071223e+01  8.82606571e+01 -8.52771943e+01  4.67979006e+01
 -1.51835594e+01  3.01995638e+00 -3.60208119e-01  2.36093950e-02
 -6.52688305e-04]
Error using SVD for datafile: 10.845499004347069
Norm of the solution using SVD for datafile: 137.2033085032067
Polynomial Coefficients:
[-6.52688305e-04  2.36093950e-02 -3.60208119e-01  3.01995638e+00
 -1.51835594e+01  4.67979006e+01 -8.52771943e+01  8.82606571e+01
 -3.65071223e+01]

Best degree using QR for datafile: 9
LS solution using QR for datafile (Degree 9):
[-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]
Error using QR for datafile: 10.845499004346472
Norm of the solution using QR for datafile: 137.20330855659404
Polynomial Coefficients:
[-6.52688305e-04  2.36093950e-02 -3.60208119e-01  3.01995638e+00
 -1.51835594e+01  4.67979006e+01 -8.52771943e+01  8.82606572e+01
 -3.65071224e+01]

Best degree using SVD for datafile2: 3
LS solution using SVD for datafile2 (Degree 3):
[ 4.63553184e+15 -4.63553184e+15 -9.52805090e+02  1.44785817e+04
 -9.71644693e+04  3.60187959e+05 -8.04451497e+05  1.11153883e+06
 -9.30293116e+05  4.31972012e+05 -8.53157274e+04]
Error using SVD for datafile2: 16.734508051030023
Norm of the solution using SVD for datafile2: 6555631995485197.0
Polynomial Coefficients:
[ -85315.72736023  431972.01202581 -930293.1157997  1111538.82898275]

Best degree using QR for datafile2: 3
LS solution using QR for datafile2 (Degree 3):
[ 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]
Error using QR for datafile2: 1.1495978960699962
Norm of the solution using QR for datafile2: 4774736.28643376
Polynomial Coefficients:
[ -290442.98693795  1353661.00712666 -2692064.52358153  2979034.39154896]
```

The SVD and QR solution for the dades.txt file yielded nearly identical results. However, when it comes to the dades_regression.csv file, the results differ considerably. The extremely high norm value when using SVD points to the potential numerical instability or ill-conditioning of the problem, meaning the matrix may be near singular. It can be seen that the error and solution norm are substantially smaller when using QR factorization which indicates that as a method, it is more suited to handle ill-conditioned matrices.

Conclusion -

Because SVD operates without rank considerations, it is more straightforward to handle, yet solutions are not as robust. That being said, when we are aware of the matrix dimensionality, QR factorization is a more desirable option as it adapts depending on the matrix's rank.

2. **Graphics Compression**

Proof:

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 $mxn$), $\Sigma$ (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 $\Sigma$ are ordered in decreasing order:$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where p=min(m,n).

The best low-rank approximation of $A$ of rank $k$ is given by: $A_k = \Sigma \sigma_i u_i v_i^T$ , 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, 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: 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: $\|A-A_k\|_2^2=\sigma_{(k+1)}^2$ .

Theorem: For any matrix $B$ of rank at most $k$, $\|A-A_k\|_F \leq \|A-B\|_F$

Proof: Let $B$ minimize $\|A-A_k\|_F^2$ 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 $\|A-A_k\|_F^2$, 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 $\|A-A_k\|_F^2$. Since each row of $B$ is the projection of the corresponding row of $A$, it follows that $\|A-A_k\|_F^2$ 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 $\|A-A_k\|_F \leq \|A-B\|_F$.

Theorem: Let $A$ be an $n \times d$ matrix. For any matrix $B$ of rank at most $k$, $\|A-A_k\|_2 \leq \|A-B\|_2$

Proof: If A is of rank k or less, the theorem is obviously true since $\|A-A_k\|_2=0$. Thus, assume that A is of rank greater than k. By Lemma, $\|A-A_k\|_2^2=\sigma_{(k+1)}^2$. Now suppose there is some matrix B of rank at most k such that B is a better 2-norm approximation to A than $A_k$. That is, $\|A-B\|_2 < \sigma_{(k+1)}$. The null space of B, Null (B), (the set of vectors v such that $Bv = 0$) has dimension at least $d - k$. Let $v_1, v_2,..., v_{k+1}$ be the first $k + 1$ singular vectors of A. By a dimension argument, it follows that there exists a $z\neq0$ in Null (B) $\cap$ Span $\{v_1,v_2,...,v_{k+1}\}$. 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$(A - B) z \geq \sigma_{k+1}$. Hence the 2-norm of $A-B$ is at least $\sigma_{k+1}$ contradicting the assumption that $\|A-B\|_2 <\sigma_{k+1}$. First $\|A-B\|_2^2\geq|(A-B)z|^2$. Since $Bz = 0$, $\|A-B\|_2^2\geq|Az|^2$ .Since z is in the Span $\{v_1, v_2,..., v_{k+1}\}$ $|Az|^2=|\Sigma\sigma_i u_i v_i^T z|^2=\Sigma\sigma_i^2(v_i^T z)^2=\Sigma\sigma_i^2(v_i^T z)^2\geq\sigma_{(k+1)}^2\Sigma(v_i^T z)^2=\sigma_{(k+1)}^2$. It follows that $\|A-B\|_2^2\geq \sigma_{(k+1)}^2$, contradicting the assumption that $\|A-B\|^2<\sigma_{k+1}$. This proves the theorem.

Reference:

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

The goal of this section was to apply singular value decomposition (SVD) to JPEG images. SVD is known for providing a low-rank approximation matrix concerning the Frobenius and 2-norm.

SVD Approximation -

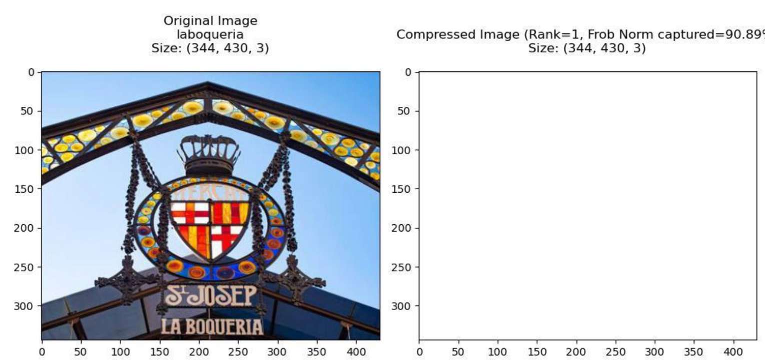
This function takes an image matrix and a rank as input and performs SVD to generate an approximation matrix of the specified rank. The function is designed to work with both grayscale and color images, treating each color channel independently.

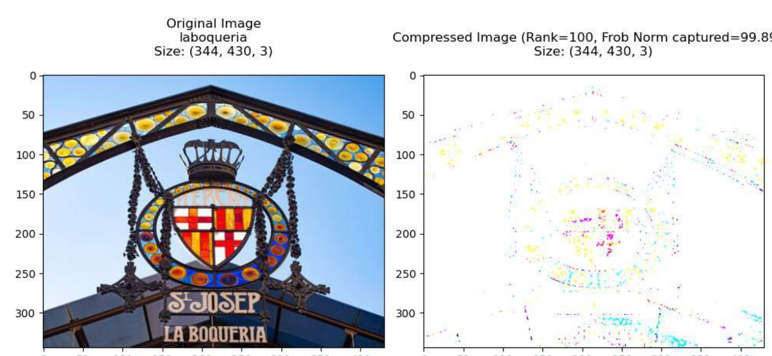
Image Compression -

This function reads a JPEG image file and converts it to a matrix. It then performs SVD compression on each color channel separately and combines them to form the compressed image. The compression is carried out for various rank values, and the resulting images are saved with filenames reflecting the compression ratio in terms of the Frobenius norm.

Visualization -

To assess the quality of compression, I created a plot_images function for visualizing the original and compressed images side by side. This aids in observing the impact of different rank approximations on image quality. For example, below I will the laboqueria.jpeg side by side with rank 1 and rank 100:



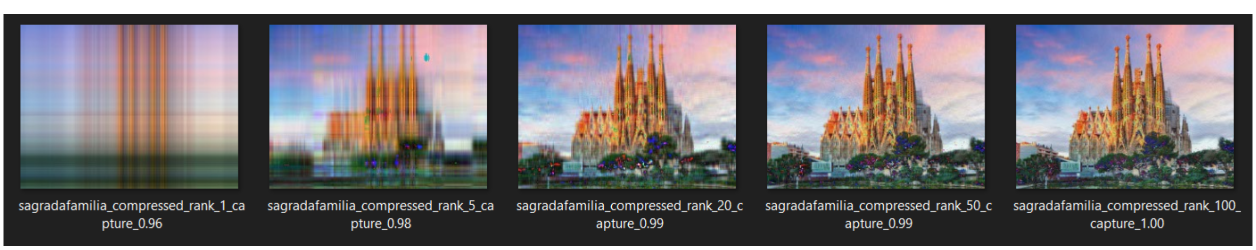
Data -

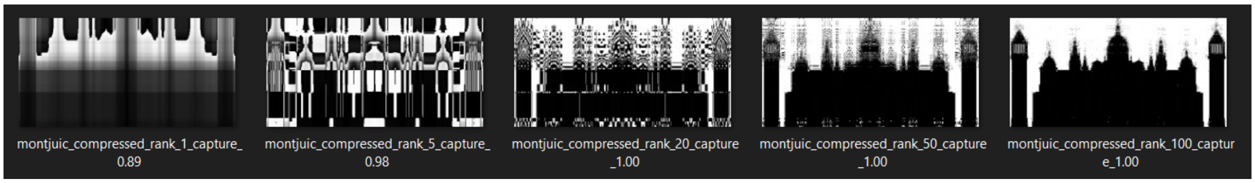
I applied the compression technique to three JPEG images of different sizes and content: sagradafamilia.jpg, laboqueria.jpg, and montjuic.jpg. The sagradafamilia.jpg file is the largest size, the laboqueria.jpg file contains some letters, and the montjuic.file is in black and white. The compression was performed for rank values of [1, 5, 20, 50, 100].

Results -

```
Original size of sagradafamilia: (1536, 2048, 3)
Saved compressed image: compressed_images\sagradafamilia_compressed_rank_1_capture_0.96.jpeg - Compression Ratio: 0.9561
Saved compressed image: compressed_images\sagradafamilia_compressed_rank_5_capture_0.98.jpeg - Compression Ratio: 0.9804
Saved compressed image: compressed_images\sagradafamilia_compressed_rank_20_capture_0.99.jpeg - Compression Ratio: 0.9900
Saved compressed image: compressed_images\sagradafamilia_compressed_rank_50_capture_0.99.jpeg - Compression Ratio: 0.9943
Saved compressed image: compressed_images\sagradafamilia_compressed_rank_100_capture_1.00.jpeg - Compression Ratio: 0.9968

Original size of laboqueria: (344, 430, 3)
Saved compressed image: compressed_images\laboqueria_compressed_rank_1_capture_0.91.jpeg - Compression Ratio: 0.9089
Saved compressed image: compressed_images\laboqueria_compressed_rank_5_capture_0.96.jpeg - Compression Ratio: 0.9601
Saved compressed image: compressed_images\laboqueria_compressed_rank_20_capture_0.99.jpeg - Compression Ratio: 0.9858
Saved compressed image: compressed_images\laboqueria_compressed_rank_50_capture_1.00.jpeg - Compression Ratio: 0.9954
Saved compressed image: compressed_images\laboqueria_compressed_rank_100_capture_1.00.jpeg - Compression Ratio: 0.9989

Original size of montjuic: (304, 612)
Saved compressed image: compressed_images\montjuic_compressed_rank_1_capture_0.89.jpeg - Compression Ratio: 0.8915
Saved compressed image: compressed_images\montjuic_compressed_rank_5_capture_0.98.jpeg - Compression Ratio: 0.9841
Saved compressed image: compressed_images\montjuic_compressed_rank_20_capture_1.00.jpeg - Compression Ratio: 0.9982
Saved compressed image: compressed_images\montjuic_compressed_rank_50_capture_1.00.jpeg - Compression Ratio: 0.9998
Saved compressed image: compressed_images\montjuic_compressed_rank_100_capture_1.00.jpeg - Compression Ratio: 1.0000
```

The output indicates the compression ratio achieved for each rank. For instance, a compression ratio of 0.98 implies that 98% of the Frobenius norm is captured by the compressed image. The plotted images can be used to provide a visual representation of the compression results. As expected, higher-rank approximations tend to preserve more details, resulting in images appearing closer to the original. The compression becomes more noticeable with lower rank values. Compressed images with their corresponding rank and compression ratio can be found in the compressed_images folder, while the original images can be found in the original_images folder, but here is a clipping below:




Conclusion -

Higher rank values lead to better image quality preservation, but at the cost of larger file sizes. The choice or rank therefore depends on the desired tradeoff between image quality and compression ratio. The effectiveness of the compression also varies with the characteristics of the images. Images with more intricate details will require higher-rank approximation to maintain visual integrity. Lastly, the percentage of Frobenius norm captured serves as a useful metric for interpreting the compression results. It provides a qualitative measure of how much information is retained in the compressed image compared to the original.

## 3. Principal Component Analysis (PCA)

PCA is a technique for dimensionality reduction and identifying the main components of a dataset. The goal is to transform the original variables into a new set of uncorrelated variables (principle components) which capture most of the variance in the data.

Methods -

Firstly, to define the methods for choosing the number of 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

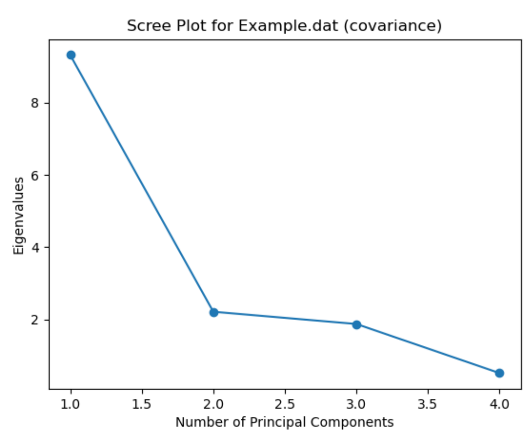
Kaiser Rule: retaining only principal components with eigenvalues > 1

3/4 rule: determining the minimum number of principal components necessary encompass 75% of the cumulative variance

example.dat -

Taking a look at the covariance matrix analysis, it can be seen that principal component 1 dominates as it captures 66.98% of the total variance. Subsequent components add less to the overall variance. In total though, 100% of variance is explained. Based on the kaiser rule, three principal components are needed, and based on the 3/4 rule, only two principal components are needed.

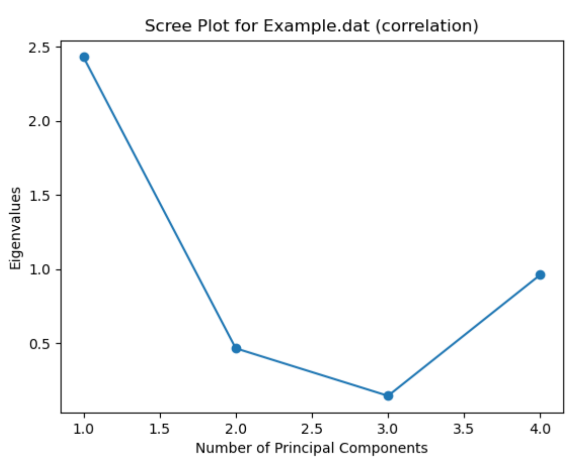
```
Covariance Matrix Analysis:
[[ 4.5625      0.5875     -2.7        -2.20416667]
 [ 0.5875      2.19583333 -0.43333333 -0.3125    ]
 [-2.7        -0.43333333  4.          3.03333333]
 [-2.20416667 -0.3125      3.03333333  3.1625    ]]

Accumulated total variance in each principal component:
[0.66979347 0.82867082 0.96319398 1.        ]

Standard deviation of each principal component:
[3.05353618 1.48718027 1.36845699 0.71580055]

PCA coordinates of original dataset:
[[ 2.33616589  0.02763159 -0.6113487  -0.42371803]
 [ 4.35336868 -2.12675047 -1.42283085  0.37065439]
 [ 1.10570266 -0.24063832 -1.79814637 -0.49789879]
 [ 3.68471106  0.48403484  2.14002302 -1.05860044]
 [ 1.42180965  2.90827971 -1.20204912  0.29523406]
 [ 3.34953464 -1.37262419 -0.50488667 -0.39157212]
 [ 4.11264074  0.15458217  2.47946554  1.08455516]
 [ 1.73086512  0.29514266 -0.92928776  0.25522494]
 [-2.81688024  0.58975511 -0.43183679 -0.73662089]
 [-3.79757714 -2.1654601   0.24020439  1.26222463]
 [-3.30409906  1.04539952  0.81479819 -0.76674108]
 [-1.49693922  2.98454685 -0.75368372  0.81873847]
 [-2.39927754 -1.1891191   0.3810924  -0.75561259]
 [-1.7836498  -0.00720863  0.22554479  0.72759173]
 [-2.2613339  -0.19769371  2.49658599 -0.03260905]
 [-4.23504154 -1.18987794 -1.12364434 -0.15085038]]

Scree Plot:
PC1: 9.3241
PC2: 2.2117
PC3: 1.8727
PC4: 0.5124

Kaiser rule: 3

3/4 rule: 2
```



Looking at the scree plot, it seems as though the appropriate number of principal components is two.

```
Correlation Matrix Analysis:
[[ 1.          0.1856123  -0.63202219 -0.5802668 ]
 [ 0.1856123   1.         -0.14621516 -0.11858645]
 [-0.63202219 -0.14621516  1.          0.85285436]
 [-0.5802668  -0.11858645  0.85285436  1.        ]]

Accumulated total variance in each principal component:
[0.66979347 0.82867082 0.96319398 1.        ]

Standard deviation of each principal component:
[3.05353618 1.48718027 1.36845699 0.71580055]

PCA coordinates of original dataset:
[[ 2.33616589  0.02763159 -0.6113487  -0.42371803]
 [ 4.35336868 -2.12675047 -1.42283085  0.37065439]
 [ 1.10570266 -0.24063832 -1.79814637 -0.49789879]
 [ 3.68471106  0.48403484  2.14002302 -1.05860044]
 [ 1.42180965  2.90827971 -1.20204912  0.29523406]
 [ 3.34953464 -1.37262419 -0.50488667 -0.39157212]
 [ 4.11264074  0.15458217  2.47946554  1.08455516]
 [ 1.73086512  0.29514266 -0.92928776  0.25522494]
 [-2.81688024  0.58975511 -0.43183679 -0.73662089]
 [-3.79757714 -2.1654601   0.24020439  1.26222463]
 [-3.30409906  1.04539952  0.81479819 -0.76674108]
 [-1.49693922  2.98454685 -0.75368372  0.81873847]
 [-2.39927754 -1.1891191   0.3810924  -0.75561259]
 [-1.7836498  -0.00720863  0.22554479  0.72759173]
 [-2.2613339  -0.19769371  2.49658599 -0.03260905]
 [-4.23504154 -1.18987794 -1.12364434 -0.15085038]]

Scree Plot:
PC1: 2.4303
PC2: 0.4647
PC3: 0.1438
PC4: 0.9612

Kaiser rule: 1

3/4 rule: 2
```

For the correlation matrix, many elements remain the same. Of course the matrix itself changes but the accumulated total variances and standard deviations do not. Even the PCA coordinates remain the same as well.



Finally, we see that according to the scree plot for the correlation matrix, the “elbow” is at two principal components. According to the kaiser rule, one principal component is needed and lastly, according to the 3/4 rule, two principal components are again needed.

RCsGoff -

For this analysis, the covariance matrix was used.

```
Covariance Matrix

              day0_rep1     day0_rep2     day0_rep3     day1_rep1  ...    day11_rep3    day18_rep1    day18_rep2    day18_rep3
day0_rep1    1.108231e+07  1.286051e+07  1.516597e+07  1.758920e+07  ...  2.455060e+07  2.358572e+07  2.133558e+07  2.009956e+07
day0_rep2    1.286051e+07  1.525812e+07  1.760607e+07  2.079202e+07  ...  2.887978e+07  2.751102e+07  2.483155e+07  2.344860e+07
day0_rep3    1.516597e+07  1.760607e+07  2.077202e+07  2.407518e+07  ...  3.361436e+07  3.226632e+07  2.917755e+07  2.749037e+07
day1_rep1    1.758920e+07  2.079202e+07  2.407518e+07  3.619859e+07  ...  5.123880e+07  4.245728e+07  3.829737e+07  3.612157e+07
day1_rep2    1.708377e+07  2.011116e+07  2.337321e+07  3.503706e+07  ...  4.963191e+07  4.119196e+07  3.718591e+07  3.504968e+07
day1_rep3    1.613578e+07  1.907018e+07  2.207316e+07  3.326314e+07  ...  4.707818e+07  3.899262e+07  3.518736e+07  3.318038e+07
day2_rep1    1.006552e+07  1.178888e+07  1.374774e+07  1.885798e+07  ...  2.627871e+07  2.478523e+07  2.243532e+07  2.113029e+07
day2_rep2    1.004665e+07  1.174285e+07  1.372090e+07  1.893662e+07  ...  2.638191e+07  2.477252e+07  2.242906e+07  2.111082e+07
day2_rep3    1.026988e+07  1.195719e+07  1.402024e+07  1.924714e+07  ...  2.684857e+07  2.533707e+07  2.296092e+07  2.160100e+07
day4_rep1    2.116020e+07  2.508512e+07  2.888665e+07  4.188498e+07  ...  5.809092e+07  5.319205e+07  4.809768e+07  4.532151e+07
day4_rep2    2.038561e+07  2.418930e+07  2.783159e+07  4.034423e+07  ...  5.594839e+07  5.120910e+07  4.630071e+07  4.363317e+07
day4_rep3    1.616418e+07  1.920486e+07  2.206682e+07  3.184810e+07  ...  4.414090e+07  4.048299e+07  3.659934e+07  3.449738e+07
day5_rep1    3.396913e+07  4.020812e+07  4.641450e+07  6.557653e+07  ...  9.163029e+07  8.260056e+07  7.465023e+07  7.040829e+07
day5_rep2    1.683035e+07  1.978353e+07  2.300536e+07  3.239301e+07  ...  4.541745e+07  4.072833e+07  3.681935e+07  3.470787e+07
day11_rep1   2.095643e+07  2.470337e+07  2.870200e+07  4.381562e+07  ...  6.356685e+07  5.044369e+07  4.548516e+07  4.292483e+07
day11_rep2   2.347577e+07  2.755303e+07  3.209892e+07  4.881837e+07  ...  7.108898e+07  5.768343e+07  5.213014e+07  4.911103e+07
day11_rep3   2.455060e+07  2.887978e+07  3.361436e+07  5.123880e+07  ...  7.432631e+07  5.923881e+07  5.344634e+07  5.041385e+07
day18_rep1   2.358572e+07  2.751102e+07  3.226632e+07  4.245728e+07  ...  5.923881e+07  5.632626e+07  5.095365e+07  4.797704e+07
day18_rep2   2.133558e+07  2.483155e+07  2.917755e+07  3.829737e+07  ...  5.344634e+07  5.095365e+07  4.614159e+07  4.341783e+07
day18_rep3   2.009956e+07  2.344860e+07  2.749037e+07  3.612157e+07  ...  5.041385e+07  4.797704e+07  4.341783e+07  4.089478e+07

[20 rows x 20 columns]
```

The cumulative variance for each principal component rapidly increases, reaching 100% after principal component 19. Standard deviations of the principal components vary significantly. That being said, the dataset is highly correlated, with the first principal component explaining a substantial portion of the variance.

```
Accumulated total variance in each principal component:
[0.95352686 0.97962999 0.99169065 0.99606767 0.99792416 0.9988168
 0.99940518 0.99968867 0.9998484  0.99989765 0.99991693 0.99993351
 0.99994944 0.99996224 0.99997279 0.99998013 0.99998659 0.99999187
 0.99999626 1.        ]

Standard deviation of each principal component:
[27902.61842833  4616.62680247  3138.08000428  1890.46241906
  1231.18942039   853.72231007   693.11732648   481.11161197
   361.13507681   200.54252743   125.47574231   116.32045576
   114.06551082   102.21440195    92.84566217    77.40674377
    72.60033709    65.69875384    59.84414972    55.25869274]
```

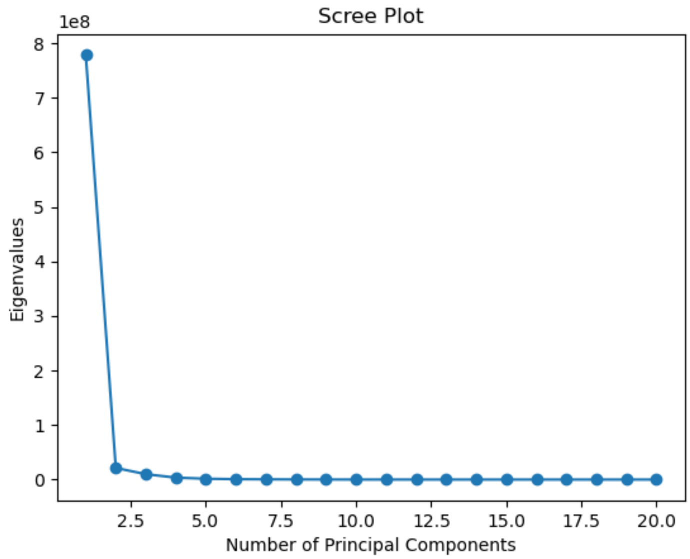
The PCA coordinates provide a transformed representation of the original dataset, emphasizing the importance of each observation along the principal components.

```
PCA coordinates of original dataset:
[[-5.27193339e+03 -1.27775394e+02  2.24088352e+02 ...  1.89428034e+00
   4.58236396e-01 -2.74217549e+00]
 [ 5.76333711e+03 -4.81958708e+02  4.86369939e+02 ...  9.79388343e+01
   3.59456885e+01  2.94155724e+01]
 [-5.16952326e+03 -1.22777676e+02  2.16933230e+02 ... -2.77609299e+00
  -2.71384959e+00 -5.94492197e-01]
 ...
 [-5.27362185e+03 -1.28210280e+02  2.24424262e+02 ...  1.81815938e+00
   5.44507081e-01 -2.41785540e+00]
 [-5.27362185e+03 -1.28210280e+02  2.24424262e+02 ...  1.81815938e+00
   5.44507081e-01 -2.41785540e+00]
 [-5.27362185e+03 -1.28210280e+02  2.24424262e+02 ...  1.81815938e+00
   5.44507081e-01 -2.41785540e+00]]
```

Finally, looking at the scree plot, we see a rapid drop-off in eigenvalues, confirming that a few principal components (9) explain most of the variance. As for the Kaiser Rule, it suggests retaining all components while the 3/4 rule recommends keeping only the first component.

```
Scree Plot:
PC1: 778556115.1567
PC2: 21313243.0333
PC3: 9847546.1132
PC4: 3573848.1579
PC5: 1515827.3889
PC6: 728841.7827
PC7: 480411.6283
PC8: 231468.3832
PC9: 130418.5437
PC10: 40217.3053
PC11: 15744.1619
PC12: 13010.9408
PC13: 13530.4484
PC14: 10447.7840
PC15: 8620.3170
PC16: 3053.5231
PC17: 3581.3223
PC18: 4316.3263
PC19: 5991.8040
PC20: 5270.8089

Kaiser rule: 20

3/4 rule: 1
```

From the scree plot as well that is generated, the goal is to choose a quantity that balances the amount of information kept and the reduction of dimensionality that is accomplished.




Conclusion -

PCA effectively captures the main components of datasets, revealing patterns and reducing dimensionality. Interpretation of results should consider both cumulative variance and specific methods for deciding the number of retained components.