24-703 Numerical Methods Project Report: Principal Component Analysis on Paderborn University Bearing Data Set

Doshi Ruchit, andrewID: ruchitsd
Ghule Lalit, andrewID: lghule

Fall 2019

Contents

1 Objective

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

PCA was first put forth in 1901 by Karl Pearson, as an analogue of the principal axis theorem in mechanics. It was also independently devicedby Harold Hotelling in the 1930s. Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT) in signal processing, the Hotelling transform in multivariate quality control, proper orthogonal decomposition (POD) in mechanical engineering, singular value decomposition (SVD) of X (Golub and Van Loan, 1983), eigenvalue decomposition (EVD) of XTX in linear algebra. It is often used to visualize genetic distance and relatedness between populations. PCA is mostly used as a tool in exploratory data analysis and for making predictive models.

2 Motivation

There are times, when a data-set has millions of observations (data entries) and every data entry has thousands of features associated with it. For example, 'Housing Price Data-set'. In such a kind of data-sets, features may range from a few hundred to a few thousand. Features like number of bedrooms, number of washrooms, total area, locality, distance form the nearest airport or nearest school, etc. In such cases, it becomes difficult to deal with such a huge number of dimensions. It is very important to reduce the number of dimensions without loosing much of information. With minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it.

Another reason behind the requirement of dimensionality reduction is the limited computation power. If we try to use the complete data-set for building a prediction model. It takes a huge amount of computation effort. This essentially, slows the overall process. It is advisable to keep the relevant features rather than having redundant features.

In summary, dimensionality reduction techniques such as Principal Component Analysis are required for the resolution of problems like huge data-sets, curse of dimensionality, limited computation power and data visualization.

3 Engineering Application

To demonstrate the Principal Component Analysis, the engineering application chosen is an open source dataset of Paderborn University, Germany. The data is generated using 32 bearings.The bearing type used for this data set generation was 6203, roller ball bearing. Out of 32, 6 bearings are healthy ones. 12 bearings have artificially created defect. Other 14 damaged bearings samples for natural damage were produced by accelerated lifetime tests.

Further, the bearings' samples can be divided intone of 3 classes, healthy, inner race fault and outer race fault. By this classification, there are 6 healthy bearings, 11 inner race fault bearings and 12 outer race fault bearings. This amounts 29 bearings in total. Remaining 3 bearings are omitted due to their nature of fault. The 3 bearings have inner as well as outer race fault. In real study conducted by Paderborn University, researcher classified these bearings on the grounds of the maximum contributing fault. If the inner race damage is more compared to outer race, the bearing is classified as inner race fault bearing.

The data set is generated with multiple testing conditions' combinations of rpm, torque, and load. There are 4 different combinations used for data generation. Each bearing is used 20 times to generate 20 signals with one fixed combination. This results in 4 times 20 signals for each bearing, i.e 80 signals (4 different combinations and 20 times repetition). The signal generated is a vibration signal for 4 sec with sampling frequency of 64kHz. That means, in a signal there are 2,56,000 data points. To avoid initial and ending noise and disturbance, the sample signal is clipped for the first $1/16^{th}$ part and the last $1/16^{th}$ part. Eventually, signal used, has 2,24,000 data points which are used further for featurization.In total 2320 signals have been used for classification.

3.1 Featurization

In featurization, it's important to get meaningful features that help in classification. Featurization includes deriving different domains' features from raw signals such as time domain, time-frequency domain, etc. For this study, only time domain features have been extracted and used for Principal Component Analysis. The vibration signals from machinery components are considered non-stationary. The non-stationary signals mean that, the signals the frequency varies with time. Hence, it is important to extract features from time domain.

For bearing fault detection, statistical time domain features such as mean, variance, standard deviation, root mean square (RMS) are used. More advanced features such as kurtosis and skewness have also been extracted out of raw vibrations signals. The approximate values of kurtosis and skewness for a normal bearing are 3 and 1 respectively. For bearings which are faulty are expected have values which are shifted. For faulty bearings, the bearing signal amplitude undergoes abrupt changes when rolling elements pass over defective region of the bearing. These abrupt changes disturb the overall distribution of signal. Generally, value of kurtosis increases, and skewness may change to negative or positive side for faulty bearings. Besides these features, dimensionless features such as crest factor, shape factor, impulse factor are also extracted.

In total, 9 features were extracted from raw signal data. Following are the name of features extracted. ***1.Kurtosis-*** Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. ***2.Mean-*** The mean (or average) of a set of data values is the sum of all of the data values divided by the number of data values. ***3.Maximum-*** The maximum value in the data set is the largest mathematical value in the data set. ***4.Minimum-*** The minimum value in the data set is the smallest mathematical value in the data set. ***5.Peak to peak-*** It is a difference between maximum and minimum value ***6.Variance-*** Variance describes how much a random variable differs from its expected value. The variance is defined as the average of the squares of the differences between the individual (observed) and the expected value. ***7.RMS-*** Root mean square value of the data set ***8.Impulse Factor-*** Its a ratio of Maximum value and mean of the data. ***9.Crest Factor-*** Crest factor is a measure of an impact when a rolling element comes in contact with raceway.

4 Assumptions of Principal Component Analysis

1.Features are related-
To reduce the number of features without losing much of information needs features to be related. Considering, features are not related, that mean, they cannot be expressed as combination of other feature or features, then removing any of the feature essentially means that part of the total information is lost. To avoid this, the features which are reduced must be able to represented with the help other features combination so that, the data represented by that feature is not completely lost.

2.Mean and Variance are the sufficient statistical Parameters-
The formalism of enough statistics captures the notion that the mean and the variance entirely describe a probability distribution. The only zero-mean probability distribution that is fully described by the variance is the Gaussian distribution. A dataset can be represented using other statistical properties such as Entropy, Kurtosis, RMS, etc. It is assumed that, whatever the nature of the data is, it can be completely represented using mean and variance (covariance, in case of variance between two different features) only. There is no need of other statistics.

3.Large variances have important Dynamics-
This assumption also encompasses the that the data has a high signal to noise ratio. Hence, principal components with larger associated variances represent interesting dynamics, while those with lower variances represent noise.

5 Mathematics behind Principal Component Analysis

5.1 Understanding Data Variance and the Goal:

Principal Component Analysis(PCA) is a very simple yet effective algorithm to re-express the given m-dimensional(m is the number of features) data into n directions (n<m) such that these n dominant dimensions (i.e. features) can best express most of the data.

As discussed in 'Motivation' section, there are data-sets where number of features are large(m) and it is necessary to perform some kind of dimensionality reduction in order to save computation effort.That is, reduce the amount of features, in order to fit machine learning algorithms. It is known that reducing the dimensionality of the problem has one drawback that might result in loss of important information. To overcome this problem, the variance of the covariate seems to be a good measure of information. The larger the variance the larger the amount of information the feature contains [One of the assumptions as well].

Thus, PCA is a mathematical approach that transforms the matrix of given features X into another one of the same dimension, call it Y, such that
1.The covariance matrix of Y is diagonal, meaning that all the transformed features are uncorrelated, and
2.The transformed features are sorted by a decreasing amount of information, meaning that the diagonal entries of the covariance matrix of the transformed features, which contain their variances (amount of information) as explained earlier, decrease as we move to the right (or down) throughout the matrix.

5.2 Steps to Perform Principal Component Analysis:

Let X be the matrix of the given data points of dimension p x m. Here, m is the number of features and p is the more of observations or data points. Thus, each column of the X matrix represents a set of measurements for a particular feature.
Let Y be a matrix with same dimensions and is related to X by a linear transformation matrix P. Thus, X is the original recorded data set matrix and Y is the re-representation of the data-set.

$$Y = PX$$

Step 1: Normalized the given data:

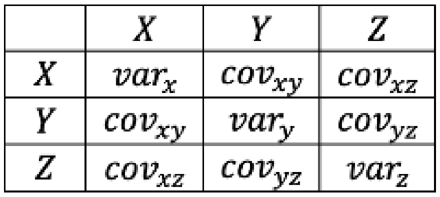
$$\vec{\mu} = \frac{1}{n}\left(\vec{x}_1 + \ldots + \vec{x}_n\right). \qquad B = [\vec{x}_1 - \vec{\mu} \mid \ldots \mid \vec{x}_n - \vec{\mu}].$$

Figure 1: Normalizing the given data set

The normalization of each feature consists of mean centering– subtracting each data value from its variable's measured mean so that its empirical mean (average) is zero – and, possibly, normalizing each variable's variance to make it equal to 1. If the mean centering is not performed on each feature, then there is a possibility of biased results. One of the most common reason for this is the range of values of each feature. For example, a feature has values ranging from 1 to 1000 and the other feature has 1 to 10. To overcome this inherent values' difference, it is recommended to perform mean centering to avoid falsified results.

Step 2: Compute the Covariance matrix:
The Covariance matrix of X is given by:

$$S_x = \frac{1}{p-1} X X^T$$



	X	Y	Z
X	var_x	cov_{xy}	cov_{xz}
Y	cov_{xy}	var_y	cov_{yz}
Z	cov_{xz}	cov_{yz}	var_z

Figure 2: An example of 3 featured Covariance Matrix

If the goal is to reduce the number of dimensions, then, it is important that, each variable co-varies as little as possible with other variables. More precisely, to remove redundancy, covariances between separate measurements need to be zero. Evidently, in an "optimized" matrix, all off-diagonal terms in S_Y are zero.

The Covariance matrix is a special Matrix. It is symmetric, meaning that the covariance of 'a' with respect to 'b' and covariance of 'b' with respect to 'a' is same.

Thus, using the Diagonalization Theorem, a symmetric matrix can be written as:

$$S = PDP^{-1}$$

Here, D is a diagonal matrix of size m x m and the P matrix represents the same linear transformation matrix explained above.

An important property of P matrix is that it is orthogonal matrix i.e.

$$P^{-1} = P^T$$

Thus, Convariance matrix of Y, S_Y can be return as:

$$Cov(Y) = Cov(XP)$$

$$Cov(Y) = P^T Cov(X) P$$
$$S_Y = P^T S_X P$$

Here, the aim is to diagonalize the S_Y matrix. The proof is as follows:

$$\begin{aligned}
\mathbf{S} &= \mathbf{PDP}^{-1} && \text{(Diagonalization Theorem)} \\
\mathbf{S} &= \mathbf{PDP}^{T} && \text{(P is orthonormal)} \\
\mathbf{P}^{-1}\mathbf{S} &= \cancel{\mathbf{P}^{-1}\mathbf{P}}\mathbf{DP}^{T} && \text{(Multiply both sides by inverse of P)} \\
\mathbf{P}^{-1}\mathbf{S} &= \mathbf{DP}^{T} && \\
\mathbf{P}^{-1}\mathbf{S}(\mathbf{P}^{T})^{-1} &= \mathbf{D}\cancel{\mathbf{P}^{T}(\mathbf{P}^{T})^{-1}} && \text{(Multiply both sides by inverse of transpose of P)} \\
\mathbf{P}^{-1}\mathbf{S}(\mathbf{P}^{T})^{-1} &= \mathbf{D} && \\
\mathbf{P}^{T}\mathbf{S}(\mathbf{P}^{T})^{T} &= \mathbf{D} && \text{(P is orthonormal)} \\
\mathbf{P}^{T}\mathbf{SP} &= \mathbf{D} && \text{(Transpose of transpose is equal to original)} \\
\mathbf{Cov(Y)} &= \mathbf{D} && \text{(Proven above)}
\end{aligned}$$

Figure 3: Proof to Diagonalize the covariance matrix of Y

Thus, X can be written as, $X = YP^{-1}$ and since P is orthogonal,

$$X = YP^T$$

This is a special case called orthogonal change of variable, which allows the total variance of the data to be kept unchanged.

$$\text{Total variance of X} = \text{Total variance of Y} = trace(D)$$

where trace is simply the sum of the diagonal entries of a given matrix. Given that D contains all the variances of the principal components, their sum measures the total amount of information contained in X.

Step 3: Eigenvalues and EigenVectors of Covariance Matrix:
So, far we do have much information about the matrix P. Only one property is known that, P is orthogonal matrix and it transforms X into Y.

Another property of symmetric matrices is that, it is diagonalized by a matrix of its orthonormal eigenvectors.

Mathematically, the covariance matrix can be represented as:

$$S = EDE^{-1}$$

Therefore, by taking linear transformation matrix P equal to the E Matrix (i.e.eigenvectors of the covariance matrix S_X), then the given matrix X can be transformed into a matrix Y such that the covariance matrix S_Y is a diagonal matrix D and the diagoanal elements are the eigenvalues of the S_X matrix.

The new axes, after transformation, are known as the principal components whose directions are given by the eigenvectors and the eigenvalues represent the amount of information each principal component can represent. Thus, these principle components are sorted in decreasing order in order to select the most dominant features.

The amount of information represented by each feature can be given by:

$$i^{th} \text{ feature represents} = \frac{\lambda_i}{\lambda_1 + \lambda_2 + ...\lambda_m}$$

6 Results of Principal Component Analysis

As discussed earlier in section 'Engineering Application', 9 feature have been extracted. Those features are processed in a python code script (refer Appendix for detailed code). The output of code gives a features in descending order of the magnitude of data being presented that feature. Figure 4, shows the amount of data represented by each Principal Component. First 3 feature together represent nearly 95% of the data. Figure 5, shows the cumulative data representation.

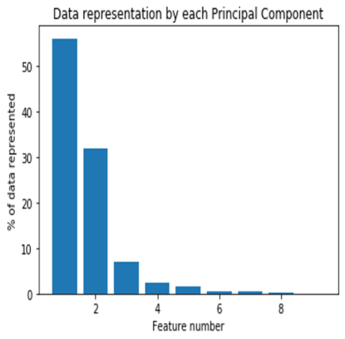


Figure 4: Data representation by each Principal Component

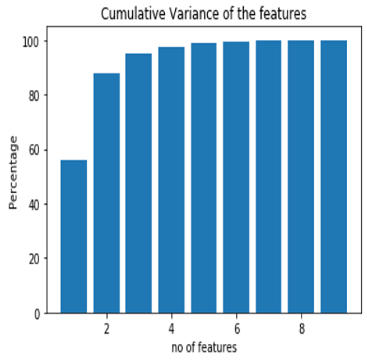


Figure 5: Cumulative Data Representation

To demonstrate dimensionality reduction, here 2 graphs have been plotted. For plotting, 2 features, Kurtosis and Crest Factor have been selected. Referring figure 6 , two features have been plotted before performing Principal Component Analysis. Figure 7 shows the same features but after performing Principal Component Analysis. Its clearly evident from figure 6, that the feature, Crest Factor can be expressed as linear combination of Kurtosis. Same result is shown in figure 7 Hence, it can be neglected. This is known as 'Dimesionality Reduction'

The plot in figure 8, shows the PC1 vs PC2 (Kurtosis vs Mean). These features are plotted after Principal Component Analysis. These 2 features together represent nearly

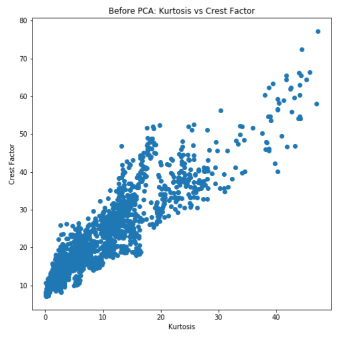


Figure 6: Before PCA: Kurtosis vs Crest Factor

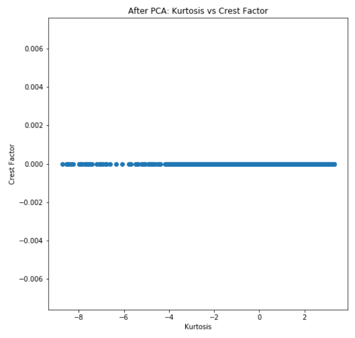


Figure 7: After PCA: Kurtosis vs Crest Factor

88% of the variance of the data. As evident from the plot, PC2 cannot be expressed in terms of PC1 in order to reduce the dimension. If PC2 neglected, significant amount of data representation by PC2 would be lost.

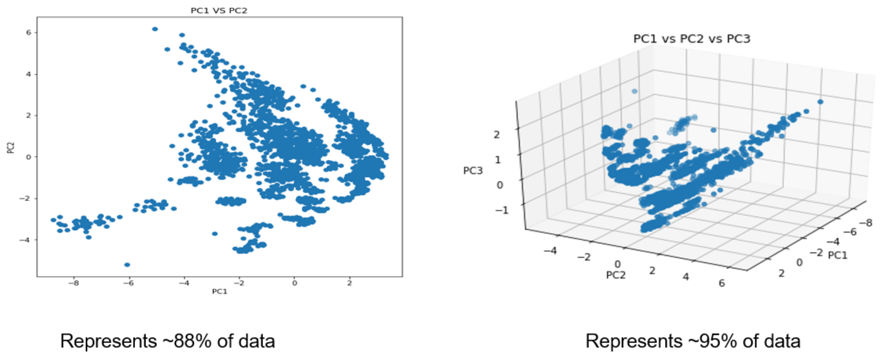


Figure 8: Data represented by first 2 and first 3 Principal Components

7 Limitations of Principal Component Analysis

- PCA is not optimal for classification
- It heavily depends upon the data-set. If the features data-set is impossible to uncorrelate, PCA may fail. Especially, if the data is non-linear in nature, PCA does not work well.

In such cases, Independent Component Analysis (ICA) can be useful.

References

[1] Amir Barati Farimani Class Notes, 24787,Prof.Carnegie Mellon University.

[2] Andrew Ng, Cs229 Lecture Notes, Stanford University.

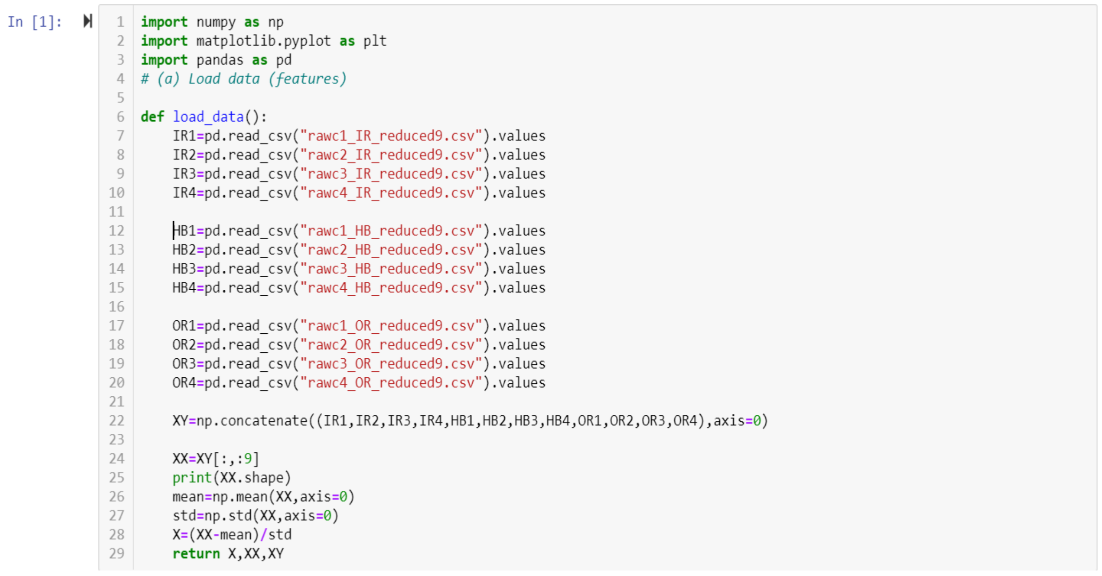
[3] https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf

[4] https://en.wikipedia.org/wiki/Principal_component_analysis

[5] https://mb.uni-paderborn.de/en/kat/main-research/ data center/bearing-data center [toc,page]appendix

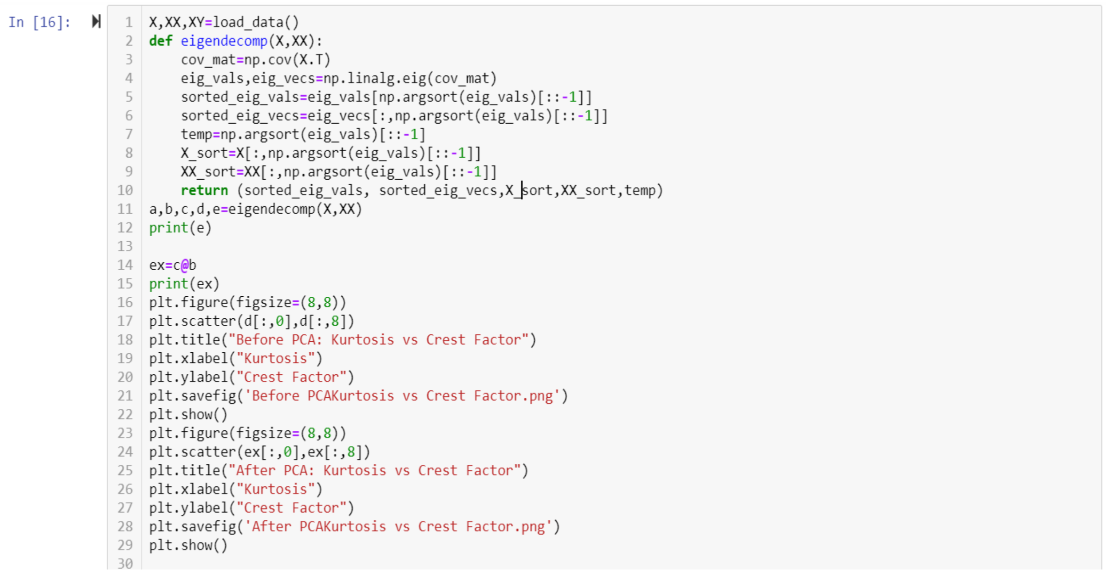
8 Appendix

8.1 Code of the Program



```
In [1]:
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# (a) Load data (features)

def load_data():
    IR1=pd.read_csv("rawc1_IR_reduced9.csv").values
    IR2=pd.read_csv("rawc2_IR_reduced9.csv").values
    IR3=pd.read_csv("rawc3_IR_reduced9.csv").values
    IR4=pd.read_csv("rawc4_IR_reduced9.csv").values

    HB1=pd.read_csv("rawc1_HB_reduced9.csv").values
    HB2=pd.read_csv("rawc2_HB_reduced9.csv").values
    HB3=pd.read_csv("rawc3_HB_reduced9.csv").values
    HB4=pd.read_csv("rawc4_HB_reduced9.csv").values

    OR1=pd.read_csv("rawc1_OR_reduced9.csv").values
    OR2=pd.read_csv("rawc2_OR_reduced9.csv").values
    OR3=pd.read_csv("rawc3_OR_reduced9.csv").values
    OR4=pd.read_csv("rawc4_OR_reduced9.csv").values

    XY=np.concatenate((IR1,IR2,IR3,IR4,HB1,HB2,HB3,HB4,OR1,OR2,OR3,OR4),axis=0)

    XX=XY[:,:9]
    print(XX.shape)
    mean=np.mean(XX,axis=0)
    std=np.std(XX,axis=0)
    X=(XX-mean)/std
    return X,XX,XY
```

Figure 9: Data loading and Normalization (Data centering)



```
In [16]:
X,XX,XY=load_data()
def eigendecomp(X,XX):
    cov_mat=np.cov(X.T)
    eig_vals,eig_vecs=np.linalg.eig(cov_mat)
    sorted_eig_vals=eig_vals[np.argsort(eig_vals)[::-1]]
    sorted_eig_vecs=eig_vecs[:,np.argsort(eig_vals)[::-1]]
    temp=np.argsort(eig_vals)[::-1]
    X_sort=X[:,np.argsort(eig_vals)[::-1]]
    XX_sort=XX[:,np.argsort(eig_vals)[::-1]]
    return (sorted_eig_vals, sorted_eig_vecs,X_sort,XX_sort,temp)
a,b,c,d,e=eigendecomp(X,XX)
print(e)

ex=c@b
print(ex)
plt.figure(figsize=(8,8))
plt.scatter(d[:,0],d[:,8])
plt.title("Before PCA: Kurtosis vs Crest Factor")
plt.xlabel("Kurtosis")
plt.ylabel("Crest Factor")
plt.savefig('Before PCAKurtosis vs Crest Factor.png')
plt.show()
plt.figure(figsize=(8,8))
plt.scatter(ex[:,0],ex[:,8])
plt.title("After PCA: Kurtosis vs Crest Factor")
plt.xlabel("Kurtosis")
plt.ylabel("Crest Factor")
plt.savefig('After PCAKurtosis vs Crest Factor.png')
plt.show()
```

Figure 10: Calculating Co-variance Matrix and main PCA code