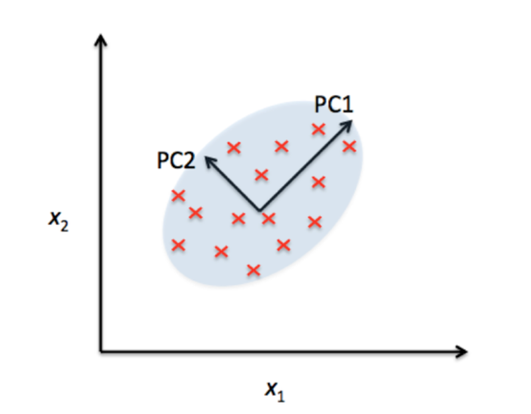
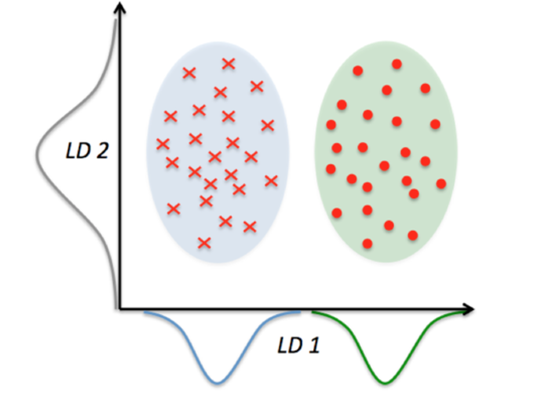
**BASICS OF DIMENSIONALITY REDUCTION**

* “Curse of dimensionality” is one of the major problems in the field of Machine Learning. Data scientists need to be careful of this curse and have to be smart enough to avoid this curse.
* What exactly is this curse? In short, as the number of features or dimensions grows, the amount of data we need to generalize accurately, grows exponentially. We generally tend to think, the more information we have about a particular data point, the better. This is true to a certain degree, so long as we're not adding redundant data or features that aren't useful for separating what's important and what's unimportant. This is what's known as the curse of dimensionality: as you increase the number of dimensions, the need for data points increases as well.
* Let’s take an example to explain it better:
  + Let there be a set of features that describe a data point. Let’s look at the weather data. That set of features might include things like temperature, humidity, time of day, etc.
  + So each data point might have one feature (if we’re only looking at temperature) or it might have 2 features (if we’re looking at temperature and humidity) and so on.
  + The point being, based on the number of dimensions the data has (how many features it has), the more difficult it is to make an estimator. This is because if you simply have one feature of data, or 1-dimensional data, then when you go to graph this data, you get a line graph, and imagining a line graph between let's say 0-50 degrees-C, it only takes 50 random points before each data point is about 1 degree from any other data point.
  + Now let's think about 2 dimensions, talking about humidity and temperature, now it's trickier to find that d such that all the points are within "d" units of each other. Imagine temperature is still between 0-50 but now humidity is also between 0-100%. Now we have a space with close to 100 \* 50 = 5,000 data points. From 50 data points in1-D, now we have 5000 data points in 2-D.
  + Similarly, let’s take 3 dimensions, the amount of data point will increase exponentially.
* The problem, the so-called “curse”, is that while we want to get more features to look at in order to figure out the difference between a “yes” and a “no”, doing so can get expensive.
* More and more data is required as we add each new dimension, and data is expensive. Even then, if a data scientist is lucky enough to have plenty of data, it's still expensive to process.
* Hence, data scientists must develop skills that allow them to analyze data with more features while strategically looking only at the dimensions that are most rich with information, or are most important to the questions that they are trying to ask.
* Dimensionality reduction uses linear algebra to reduce the dimensions you're looking at to the part of the data space where your data is most dense, ignoring parts where the data is sparse. PCA and SVD are two such techniques for accomplishing this.

2. ***Linear Discriminant Analysis vs PCA***

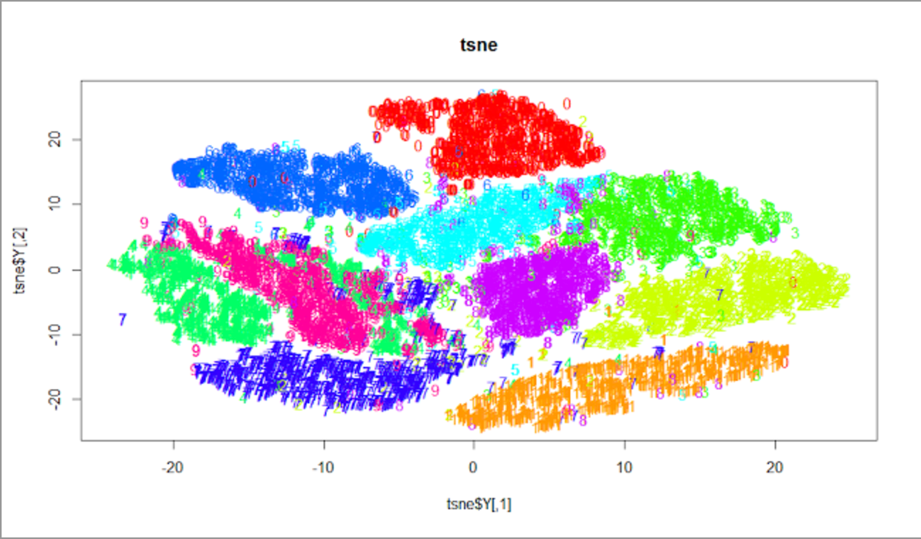
* Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are linear transformation techniques that are commonly used for dimensionality reduction.
* First, let’s understand PCA. Consider a dataset with only two dimensions. This dataset can be plotted as points in a plane. But if we want to tease out variation, PCA finds a new coordinate system in which every point has a new (x,y) value. The axes don't actually mean anything physical; they're combinations of height and weight called "principal components" that are chosen to give one axes lots of variation.
* Similarly, as the number of dimensions grows it firstly becomes very hard to visualize data and hence it is a huge problem to make sense of that data. Hence, we apply PCA to the data to reduce its dimensions and hence go on the procedure to make sense.
* What PCA does is that, it finds new axes in which PC1(1st principal component) will have the maximum variance of the data.
* On the other hand, LDA is more about maintaining the seperatibility between the classes of the data. The goal is to project a dataset onto a lower-dimensional space with good class-seperatibility in order avoid overfitting and also reduce computational costs.
* PCA can be described as an unsupervised algorithm, since it ignores class labels and its goal is to find the directions that maximize the variance in a dataset. In contrast to PCA, LDA is supervised and computes the directions that will represent the axes that that maximize the separation between multiple classes.
* Below Shown figure will tell more about the difference between PCA and LDA:

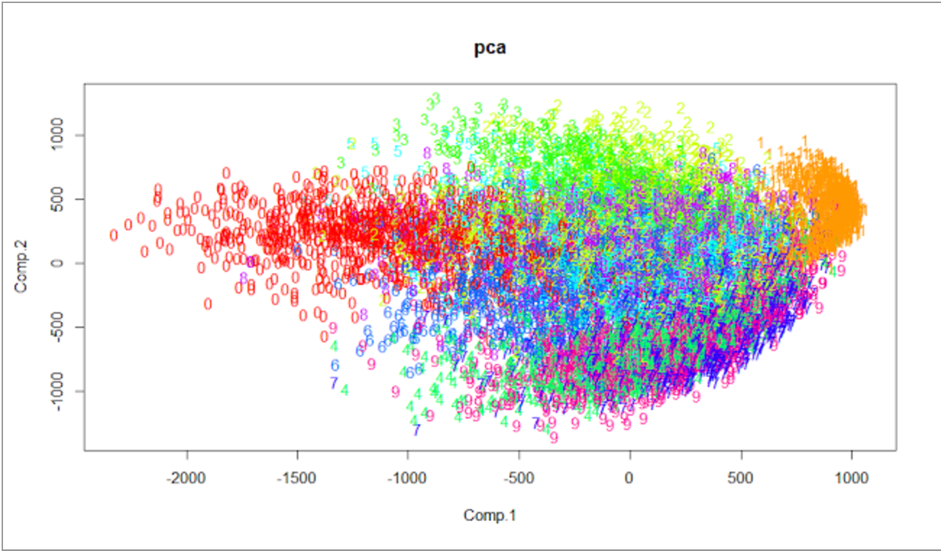
 

Reference: <http://sebastianraschka.com/faq/docs/lda-vs-pca.html>

1. ***t-Distributed Stochastic Neighbor Embedding (t-SNE) vs PCA***

* t-Distributed Stochastic Neighbor Embedding is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.
* t-SNE algorithm comprises two main stages. First, t-SNE constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked.
* Second, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the divergence between the two distributions with respect to the locations of the points in the map.
* PCA is a linear and parametric method. Whereas t-SNE can be used to embed high-dimensional data into low dimensions. For clustering analysis, t-SNE is better preferred as compared to PCA although it is not a clustering technique.
* Both PCA and t-SNE was applied to a high dimensional dataset and below are the visualizations:





* t-SNE appears lot better in visualization but according to the experiment done, PCA provided more accurate results.
* Reference: <http://noviceactuarialstudent.blogspot.com/2016/05/comparison-of-t-sne-vs-pca.html>

<https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding>

**THE SPECIFICS OF PRINCIPAL COMPONENT ANALYSIS**

1. ***Importance of Eigen Vectors and Eigen Values***

* A matrix can be thought of as a transformation that takes in a vector and spits out a new vector. Given an N x N matrix, there are special vectors that when multiplied to the matrix give you a constant multiple of that vector. Multiplying this special vector by the matrix is the same as multiplying the special vector by that corresponding special value. These special vectors are called eigenvectors and the constant multiple is called an eigenvalue.
* In mathematical terms, the above can be summarized. Let *v* be a vector and *A* be a square matrix with two columns a1 and a2. If we multiply *v* by *A*, then *A* sends vector *v* to a new vector *Av*. This new vector is the a constant λ multiplied to original vector *v*. Hence, ***Av*** = ***λv*** where*v*is the Eigen vector andλ is the Eigen value.
* Eigen vectors and Eigen values has great application in Machine learning as it is being used in one of the dimensionality reduction techniques called Principal component analysis. They are also used in Face detection algorithms.
* PCA is all about finding the eigen vectors and eigen values of co-variance or correlation matrix of the data and in the end reducing the dimensions of the data.

1. ***Can a non-square matrix have Eigen values?***

* By definition, eigen-vectors and eigen-values are defined for square matrices. So we cannot have eigen values for a non-square matrix.
* For non-square matrices, we can define singular values. The singular values of M x N matrix A are the positive square roots of the non-zero eigen-values of the corresponding matrix ***AT.A.*** The corresponding eigen-vectors are called singular vectors.

1. ***Does PCA maintain the interpretability of the features of the data. Explain your answer with an example.***

* No, PCA does maintain the interpretability of the features of the data. If interpretability of the features is important to the problem, then we should not use PCA.
* In the 6th part of this question, I have solved computed PCA for a 4 x 2 matrix and have reduced to 4 x 1 matrix.
* As you can see, the reduced matrix does not interpret into original data point.

1. ***What is PCA applied mainly applied to: Covariance or correlation matrix. And what is the difference between the both?***

* Typically, Covariance matrix is preferred for PCA. But there is a difference on when it can what be applied. So when the scales of the data is the same, then we apply covariance matrix and when the scales of data are not the same then we apply correlation matrix.

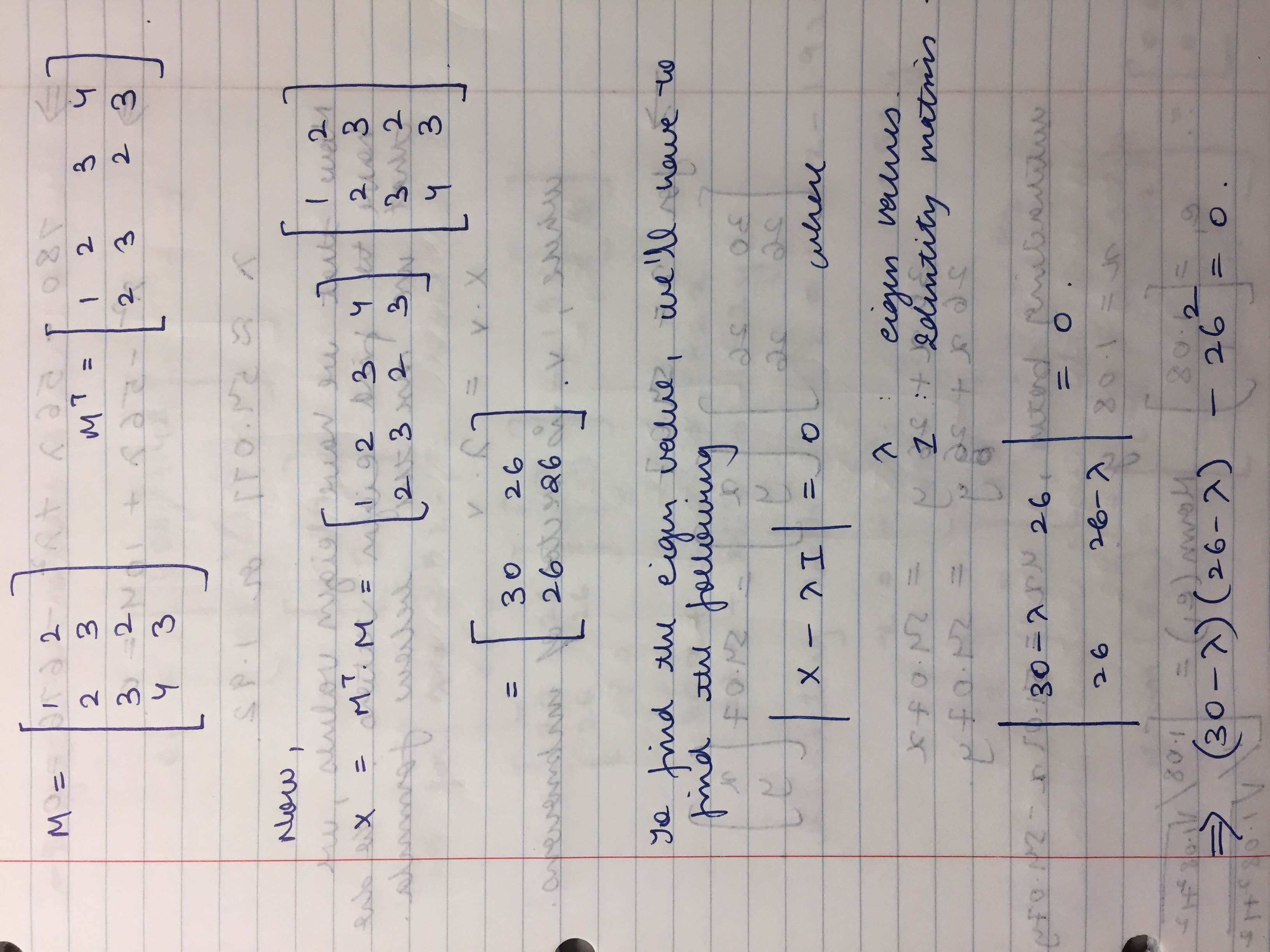
1. ***Lists of steps for PCA***

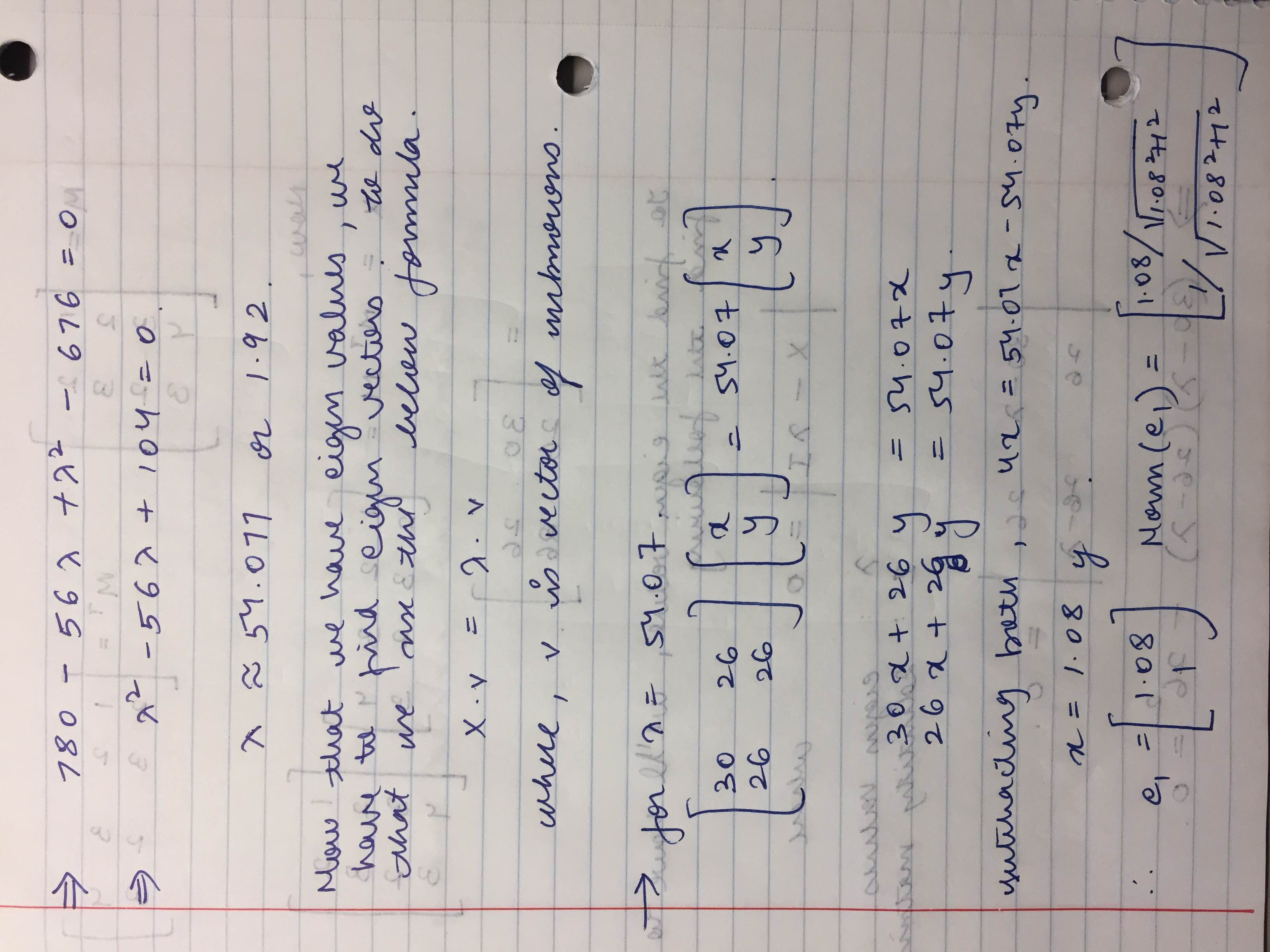
Let, we have a matrix A of size M x N.

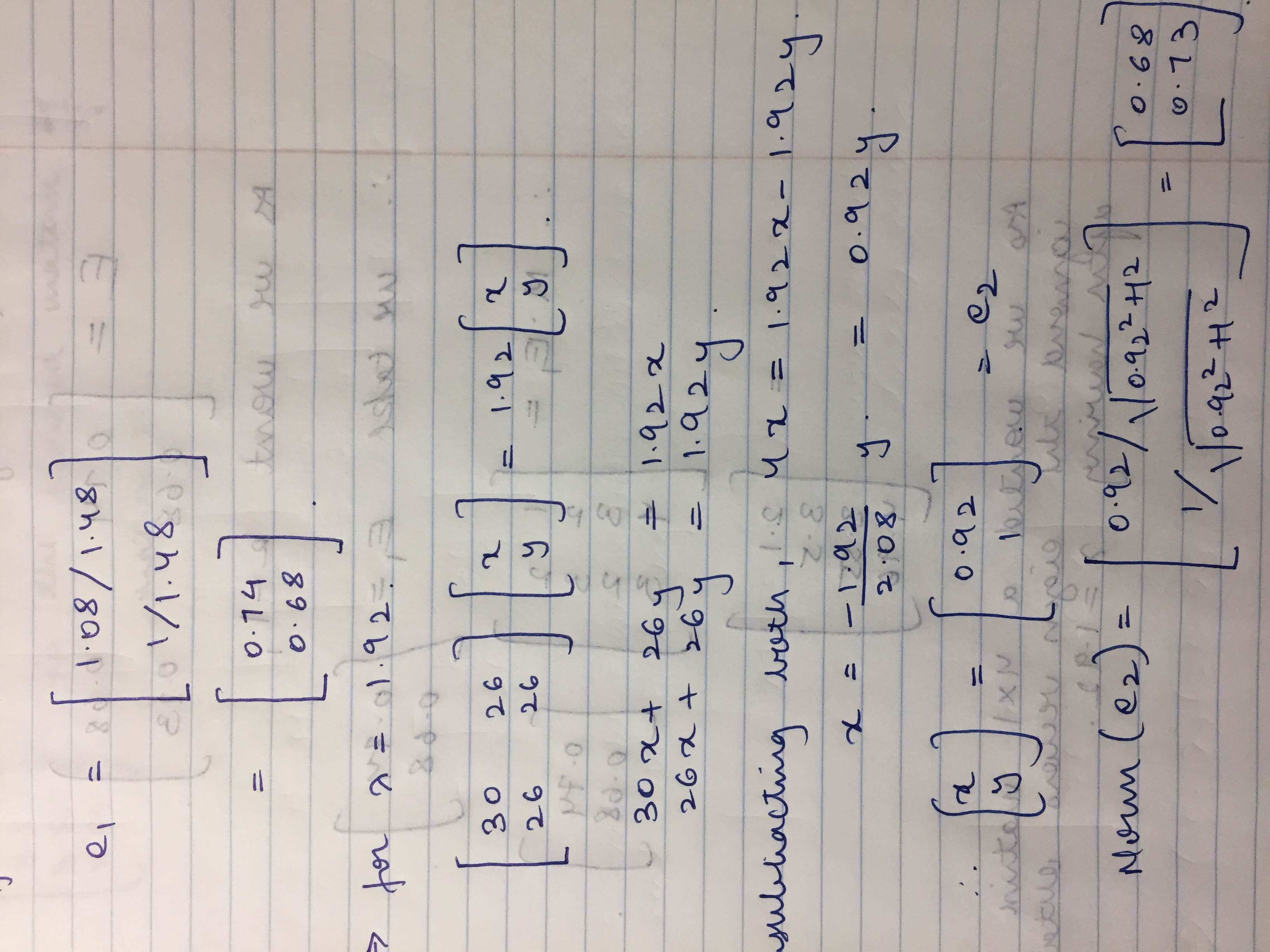
* Normalize A
* Calculate ***ATA*** or ***covariance of A*** or ***correlation of A***.
* Suppose, you find ***X = ATA.*** Now you calculate the Eigen Value by finding solving the equation ***|X - λI | = 0*** i.e. equalizing the ***determinant of X - λI*** to 0where, ***λ*** is the eigen value and I is the identity matrix of X.
* After knowing the eigen values, you multiply the matrix X with unit vector. So here in this step we do ***X.v = λ.v,*** where ***v*** is a vector of unknowns and ***λ*** is the eigen values. On solving this equality for both the eigen values, we will get the eigen vectors.
* Then we calculate the unit-eigen-vectors ***e*** for all the eigen vectors obtained, i.e. we normalize the vectors.
* The matrix formed by eigen vectors be ***E***. We multiply the original matrix ***A*** with matrix ***Ek*** (where k is the number of dimensions to be retained) to get the dimensionality reduced matrix.

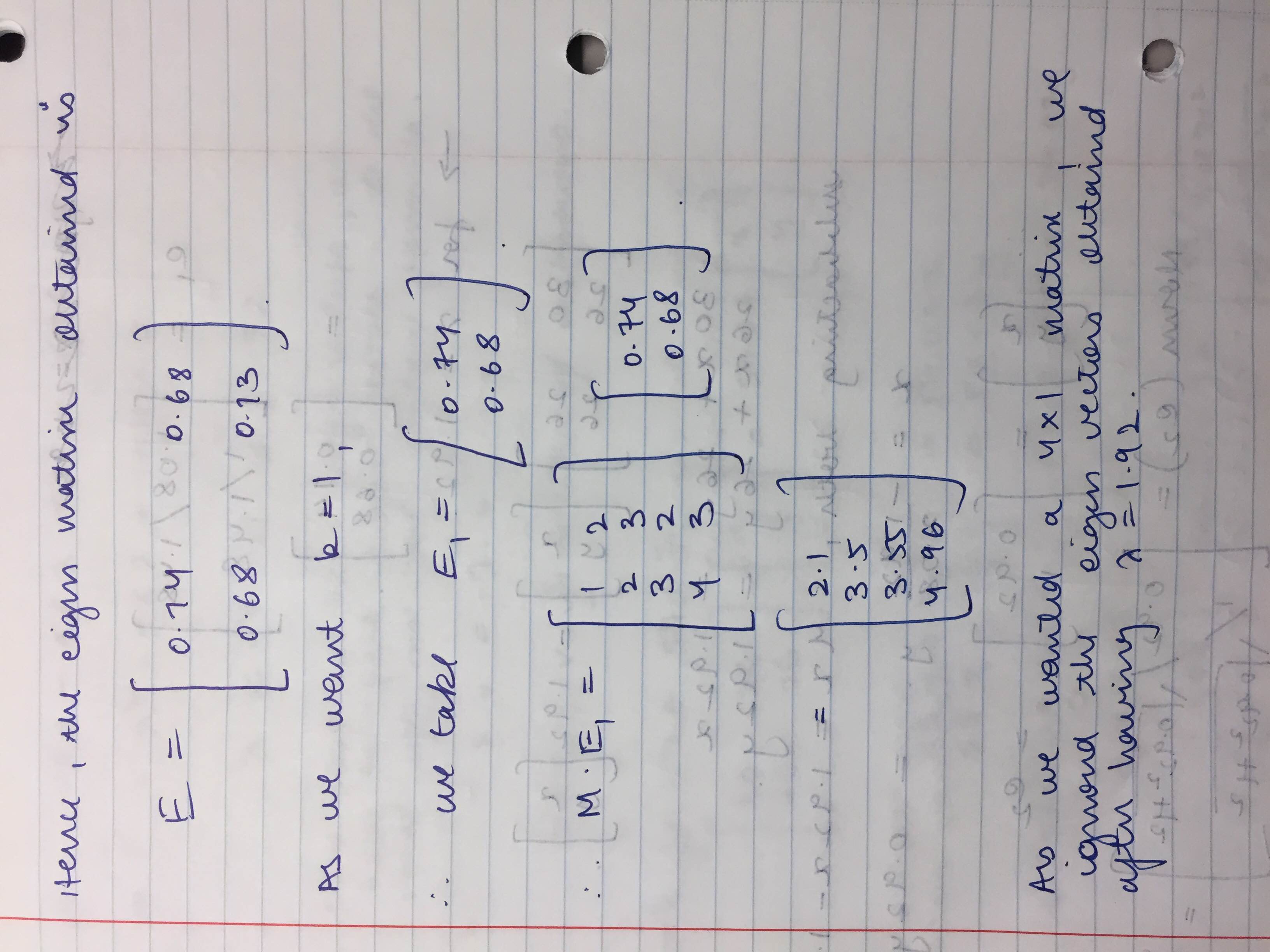
1. ***Calculation of PCA for the given matrix M***

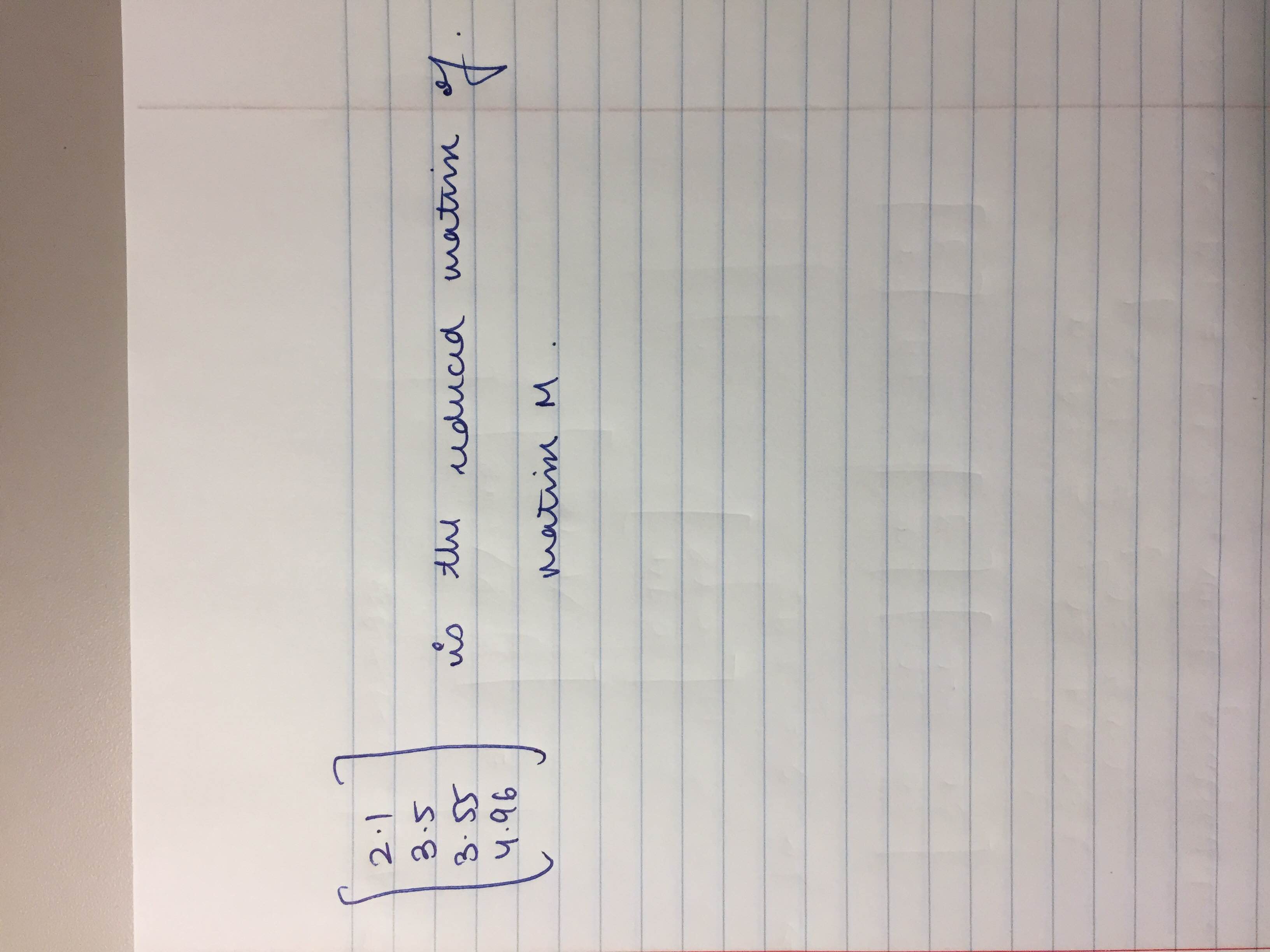
\*\*\*Calculation started from the next page\*\*\*











1. ***Explain the rotation of original axis to principal component analysis’s axis of the new Euclidean space.***

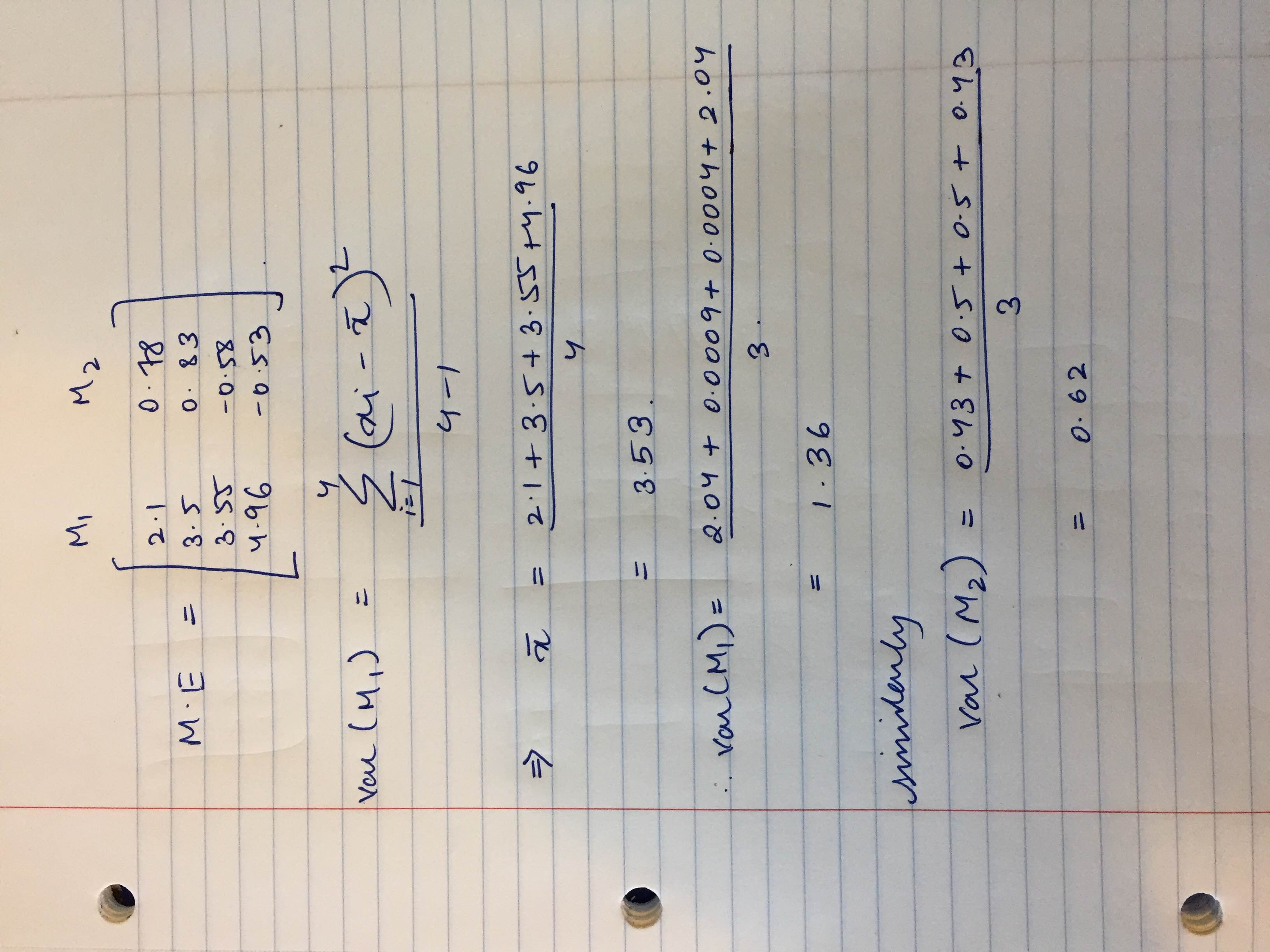
As you can see in the above PCA calculation, we get two new eigen vectors. These two eigen vectors are unit vectors which are orthonormal to each other. These two together form the new principal component analysis’s axis which are different from the original axis. The axis which is the 1st Principal component(PC1) has the max variance and the 2nd principal component(PC2) is orthonormal to PC2. These two new axis are the rotation from the original axis either in the clock-wise or counter-clockwise direction. Also, any matrix of orthonormal vectors (unit vectors that are orthogonal to one another) represents a rotation of the axes of a Euclidean space.

In the above example, in the original matrix M we had the 1st column as [1 2 3 4] which has been transformed into [2.1 3.5 3.55 4.96] . Similarly, we can calculate for the 2nd column where the new points are different from the original ones. Hence, we can see that there is rotation in the axis formed from PCA from the original axis.

1. ***Report and comment on the variance of the resulting matrix of PCA1, PCA2 (of the eigen vectors) from the matrix in question 6.***

* Variance is the average of squared differences from mean.

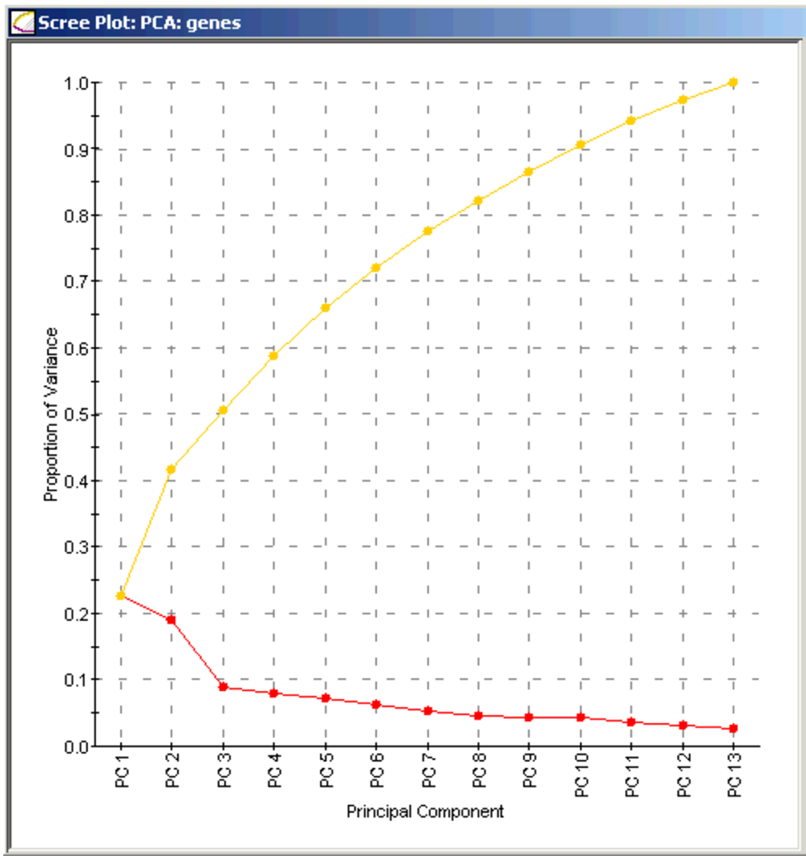
Computing the Variance for both the features PC1 and PC2.



* The variance for PC1 is 1.36 and PC2 is 0.62
* PC1, is the most significant axis as it has the highest variance.
* PC2, the next most significant axis has the 2nd most-highest variance.

1. ***When you apply PCA to an input dataset that has n attributes (dimensions), you aim to reduce the data into m dimension such that m is less than n. Provide an algorithm that allows you to find the optimal m (number of components) for the reduce data.***

* There are several techniques which can be used to find the optimal m. They are
  + Scree Plot: A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each PC. The PCs are ordered, and by definition are therefore assigned a number label, by decreasing order of contribution to total variance. Such a plot when read left-to-right across the abscissa can often show a clear separation in fraction of total variance where the 'most important' components cease and the 'least important' components begin. The point of separation is often called the 'elbow'.



In the above plot, at PC3 is where the elbow formed. So we can consider the first 3 dimensions.

* Eigenvalues > 1 rule: This rules states that there are as many reliable factors as there are eigenvalues greater than one. The reasoning is that an eigenvalue less than one implies that the scores on the component would have negative reliability.
* Percentage of variance explained criterion: The eigenvalues in PCA tell you how much variance can be explained by its associated eigenvector. Therefore, the highest eigenvalue indicates the highest variance in the data was observed in the direction of its eigenvector. We can also get relative numbers by first summing up all eigenvalues and then divide an eigenvalue with this sum. This way you end up with a "percentage of variance" for each eigenvector. Based on the percentage of contribution of each eigen value, we can find out the dimensions to retain.
* Reference: <http://www.improvedoutcomes.com/docs/WebSiteDocs/PCA/Creating_a_Scree_Plot.htm>

<http://www.chegg.com/homework-help/questions-and-answers/according-eigenvalue-criterion-components-eigenvalues-greater-1-selected-component-explain-q2254945>

1. ***Define the concept of “Stylometry” -- Explain how can PCA be used in Stylometry.***

* Stylometry is the study of linguistic, primarily written, and other expressive styles such as music and painting.    One of the main motivations for stylometry arises from the need to analyze texts for authenticity and authorship.
* Several machine learning techniques has been applied in stylometry in which PCA is also one.
* A PCA analysis of a multi-dimensional data set will attempt to find natural axes through the data points where the principal axes or components are the ones which have the greatest variance in values along those axis.
* That means that the data is essentially spread out along that axis, as an oval is spread out along its major and minor axes. We know that an N-dimensional space always has N axes, the PCA analysis will return the N natural axes it finds in order of the variance of the data along each axis.
* We can thus reduce our attention to only the few principal components that have a significant data variation along them, in effect, reducing our problem from thousands of dimensions to just a few.
* In the field of stylometry, we can take a new unknown text and compare it against the stylometrics we analyzed from known text sources. PCA analysis would reduce our known texts to statistical measures of variation along a few principal components.
* Any given piece of text is represented as a single point in our thousand-dimension space. If we measure the unknown text source's data point in terms of if its measure along those principal components, we can see whether or not the unknown text's measure is within the expected statistical bounds for those principal components.
* Mathematically, we are taking the [projection](http://en.wikipedia.org/wiki/Projection_(linear_algebra)) of the unknown's data point onto the principal component axes. Once we get those measures, we can make some conjectures as to whether or not the unknown text had the same authorship as the PCA-analyzed texts.
* Reference: <https://en.wikipedia.org/wiki/Stylometry>

<http://journals.sfu.ca/jmde/index.php/jmde_1/article/viewFile/352/352>