PROJECT IN UNSUPERVISED MACHINE LEARNING

06.04.2012

Ajay Anand Kumar: 013711933

Instructions to use the code:
File: Exercise1.r, visual.r
Platform: R in Windows/Linux
Place the Exercise1.r source code in same directory that contains following files:
1. digits.txt
2. noisyDigits.txt
3. visual.r

I am attaching the visual.r code also with this . Please use this version because the original file downloaded from the course webpage gave some errors while plotting. I am not sure about the reason behind it.

>source('Exercise1.r')
>source("visual.r")

1. Creating artificial data to produce the desired plot.
   - Initially two dimensional data was created using the random Gaussian distribution having mean = 0 and variance = 1.
   - Then a unit vectors are created making an angle = pi/4 with the X axis. Another unit vector is orthogonal to this vector. The data will be projected on these unit vectors to get the desired plot(s).
   - A 2×2 diagonal matrix with value 4 and 1 as its diagonal elements is created. These values are the corresponding eigen values of the covariance matrix. These values are the respective variances chosen by default.
   - The data can be elongated based on these diagonal elements of the matrix. For e.g if first diagonal element =4 is changed to 16 the data gets elongated along the first principal component axis. If second diagonal element is changed to 4 from 1 the data gets elongated along the other principal component direction.
   - The direction of the data is controlled by changing the direction of unit vectors along which the artificial data is projected, i.e by changing u1 and u2 vectors.

   Usage of the code:
   >source("Exercise1.r")
   >ex1.1

The output is:
Plot 1: When angle between first uniy vector u1 and X axis is -pi/3 and elements of diagonal matrix is 4,0,0,1
Plot 2: When angle between first unit vector u1 and X axis is 0 and elements of diagonal matrix is 4,0,0,1
Plot 3: When angle between first unit vector u1 and X axis is pi/3 and elements of diagonal matrix is 4,0,0,1
Plot 4: When angle between first unit vector u1 and X axis is pi/3 and elements of diagonal matrix is 9,0,0,1

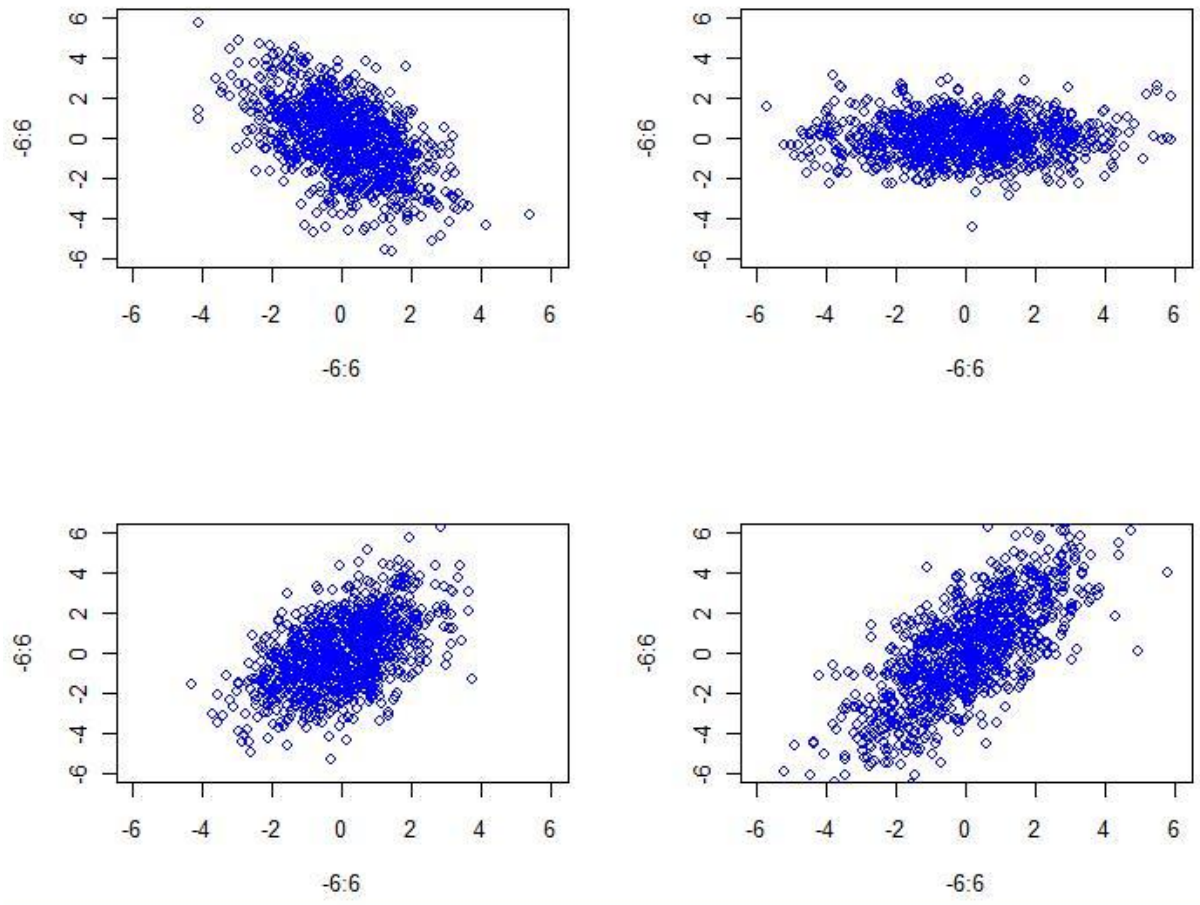


Figure1.1: Plots of Gaussian data projected along the subspace spanned by unit vectors.

Result: The desired plots are produced. The artificial data can be projected along the subspace spanned by the unit vectors. One can clearly see that if variance is varied the data points get elongated towards corresponding principal component axis. For example, in plot 4 of Figure 1,

by changing the variances that are given by the diagonal elements of the covariance matrix, the shape of the projected data can be controlled.

Exercise 1.2
PCA analysis for two of the scatter plots.
Plots chosen: 1 and 3 from Figure 1.1
Commands:
N.B : Close all windows before proceeding.
>source("Exercise1.r")
>ex1.2()

Ouput:

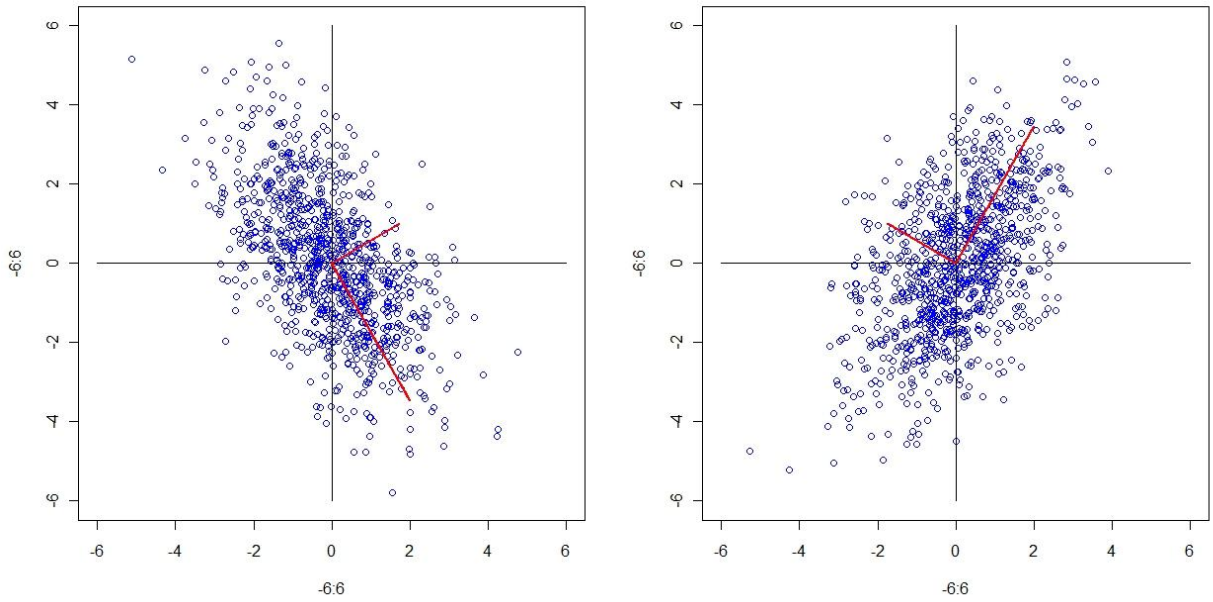


Figure 1.2 : Describes the scatter plot of Gaussian data over the subspace spanned by the unit vectors making an angle of –pi/3 in case of first and pi/6 in case of second plot. It also present the corresponding unit vectors on top of the scatterplot.

Plot corresponding to Figure4.1 p.30

Although this plot is not quantitively similar to the corresponding plot in *Lecture notes*, but idea is clear regarding plotting the data projected to eigen vectors. The principal axes have phase lag of Π radians. As such they are plotted as an axis making an angle of pi/3 radian to X axis.

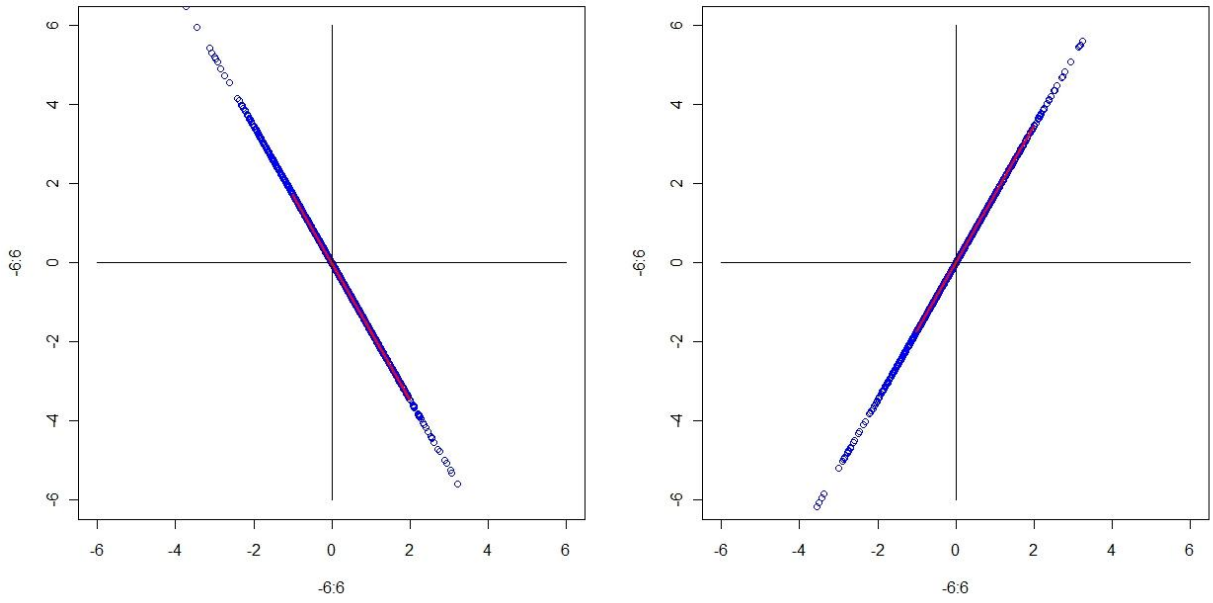


Figure 1.2.2: Qualitative similar plot to Figure 4.1 p30 of *Lecture notes*.

Exercise1.3

Solns:

The chosen plot is ‘plot 3’ from Figure 1.1. The plot has input Gaussian data (1000 observations) with mean =0 and variance =1. The data is projected in the unit vector u1 and u2 where u1 makes an angle of pi/3 radians to the X axis.

Commands:

>source(“Exercise1.r”)

>ex1.3()

The output is:

Printing histogram with 20 bins
printing dimension of projected data
2 1000
The variance of the data
[1] 1.280352
[1] 3.841056

Histogram plot:

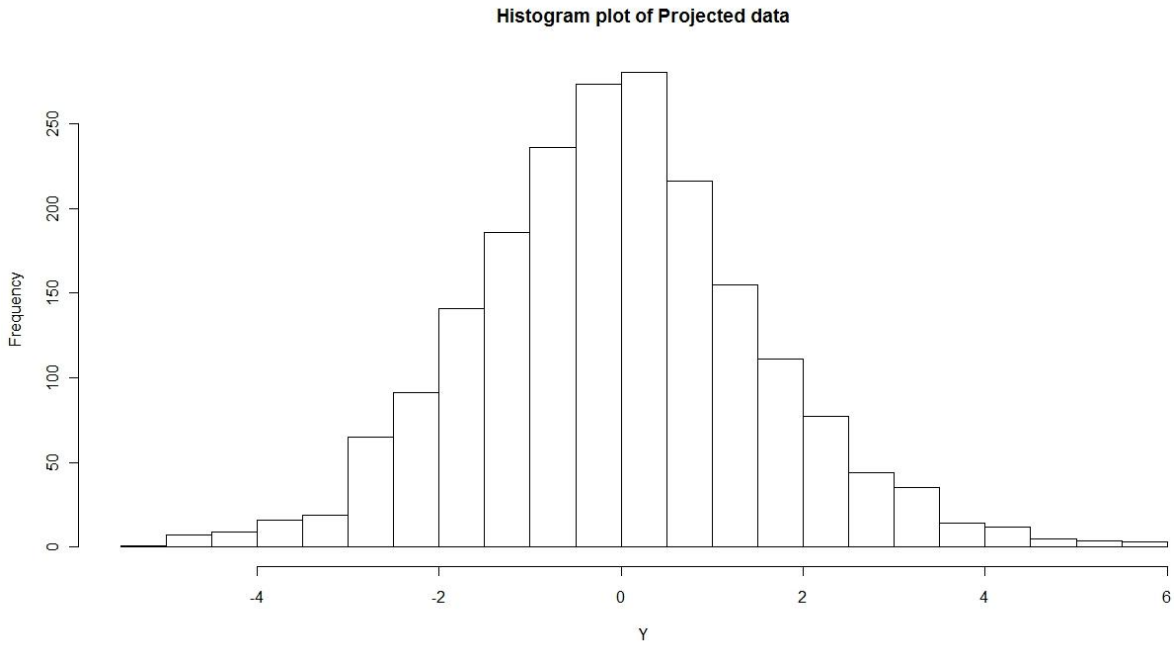


Figure 1.3: The corresponding variance of the projected data is: **1.19 and 3.84**. The covariance matrix used in the code was diagonal with diagonal elements as 4 and 1. We know from the

theory that diagonal elements constitute the variance of the data. And here the calculated variance are approximately equal to diagonal elements. For larger observation >1000 random Gaussian data, these variances can be realized close to the desired value.

Exercise 1.4

Commands:
>source("Exercise1.r")
>ex1.4()
The output is:

covariance matrix of Y

```
     [,1] [,2]
[1,]  2   -1
[2,] -1    2
```

eigen values and vectors

```
$values
[1] 3 1
$vectors
          [,1]       [,2]
[1,] -0.7071068 -0.7071068
[2,]  0.7071068 -0.7071068
```

The corresponding scatter plot is:

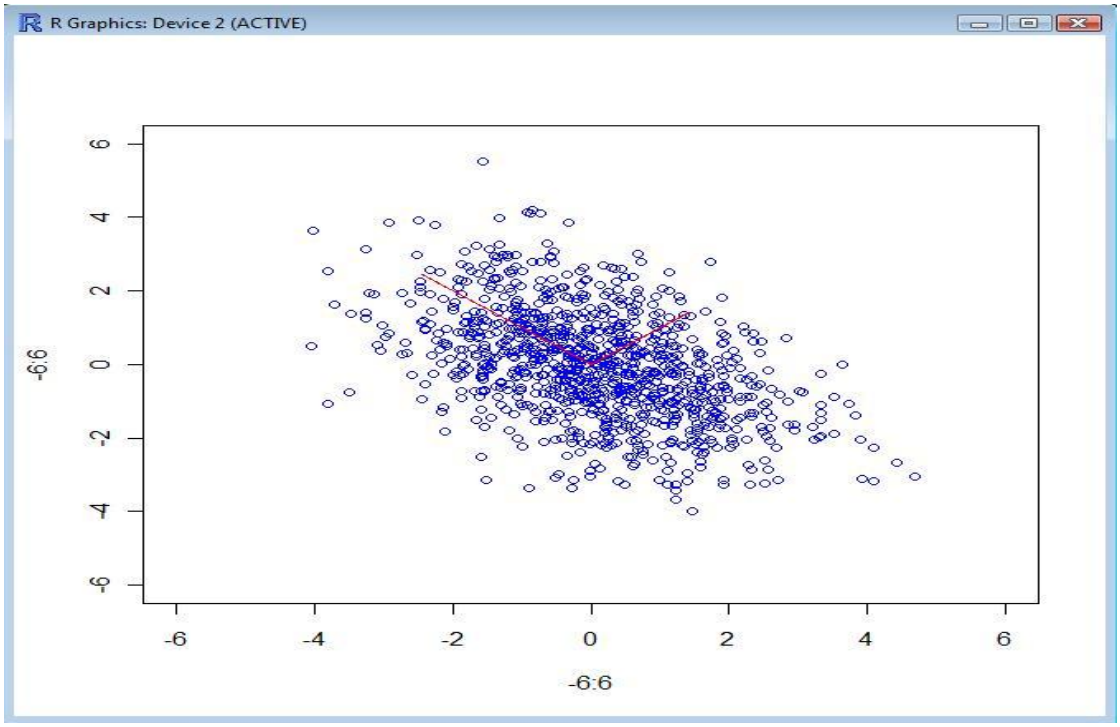


Figure 1.4: Scatter plot of the data along the principal component directions

Exercise1.5
Dimensionality reduction using PCA

Command:
>source("Exercise1.r")
>ex1.5()
The output is:
The proportion of variance explained is:
[1] 0.75 1.00
The corresponding plot is:

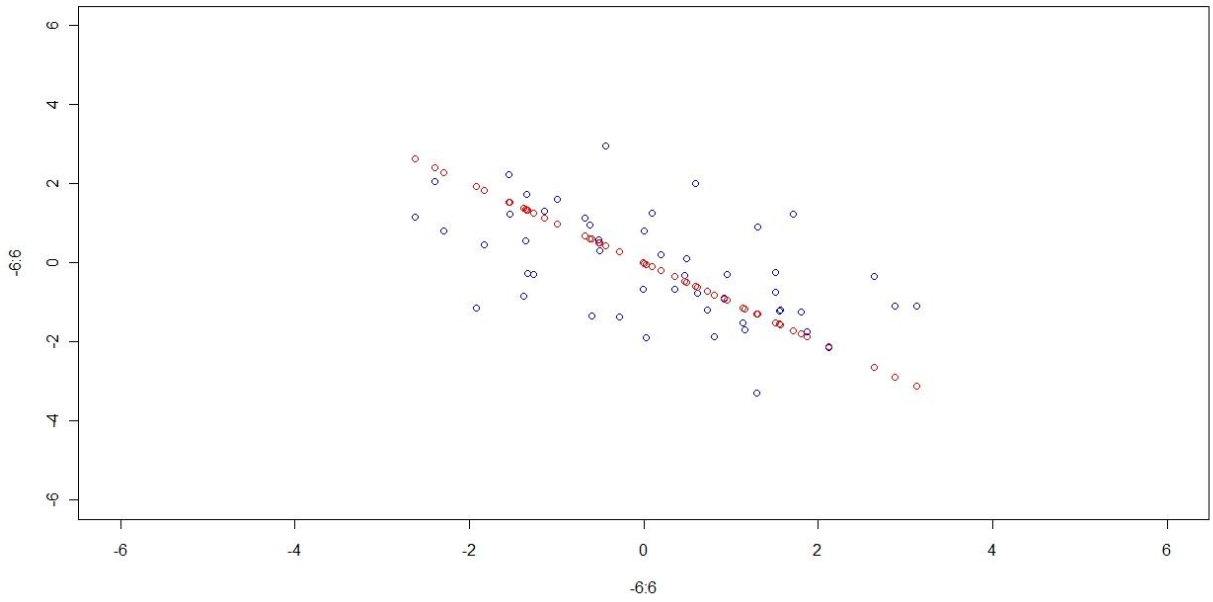


Figure1.5: Scatterplot of original data (in blue) and reduced dimension of data(in red).

Results: The corresponding proportion of variance explained by first or largest principal component i.e the eigen value is 0.75 or 75 %.

Exercise2.1
Reproduction of Figure 4.2 on p.34 of *Lecture notes*
Commands:
>source("Exercise2.1.r")
>ex2.1()

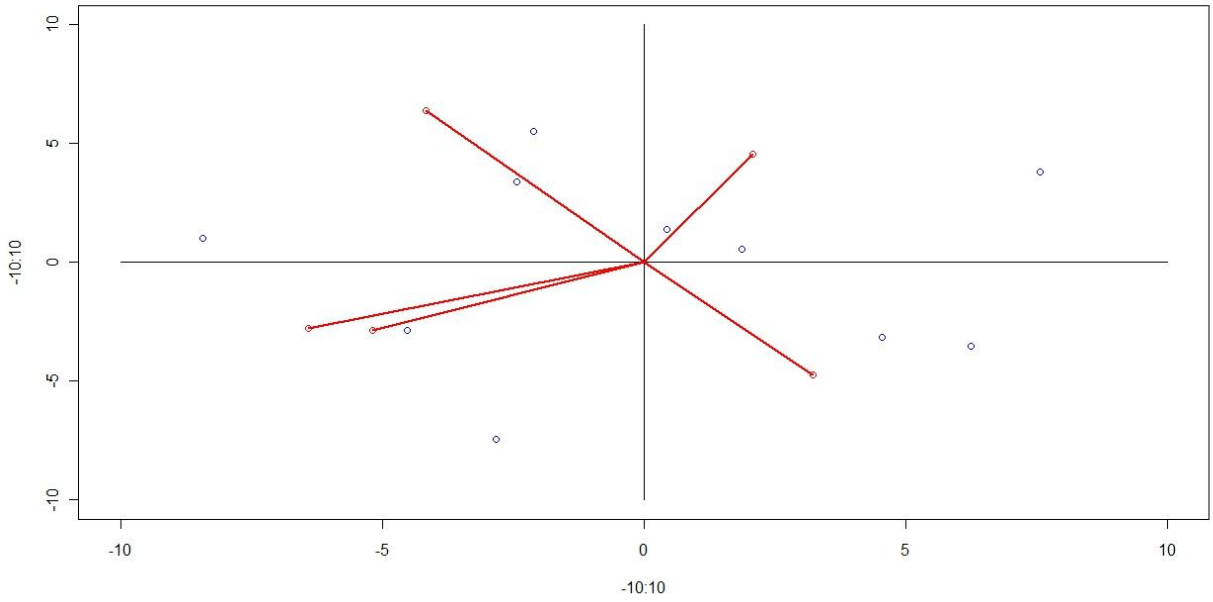


Figure 2.1: This explains projection of data points to each axis of principal components.

Explanation : Given the data matrix of size 5 × 10 similar to *Lecture notes* section 4.4 p.33 where each row is one variable and each column is one observation. Each data point in (blue) is projected to coordinate axes given by the **u1** and **u2** vector. The coordinate axes are plotted based on the values of **u1** and **u2** vectors. The original data variables are marked blue. The idea here is to project these data points to these coordinate axes and inspect which observations are close to each other, or in fact see which are more similar to each other.

The length of red lines are taken in proportion to number of observations. The length of data points of **u1** and **u2** from origin were scaled in proportion to number of observations.

Exercise2.2
Proportion of Variance as function of number of principal components:
>source("Exercise1.r")
>ex2.2()
The output is:

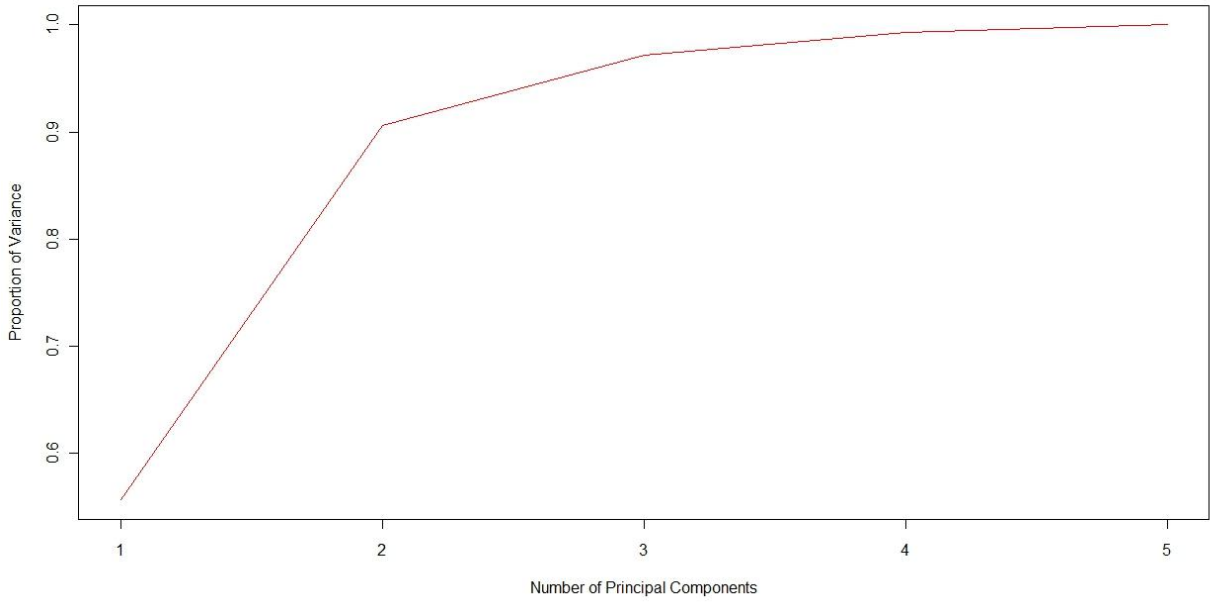


Figure 2.2: The plot of proportion of variance vs number of principal components

Results:
We can see that as number of PC increases the proportion of variance almost reaches 1. This graph infact help us in choosing minimum number of principal components in order to explain the variance of the data. For e.g in this case with the choice of 2 PC's the variance of data explained is almost 90%. This helps in reducing the dimensionality of the data. If given data of 10 dimension is reduced such that its content is preserved, theb optimal choice would be 2 dimension be seeing the proportion of variance.

Exercise2.3
--NA

Exercise2.4
--NA

Exercise3.1
Commands:
>source("Exercise1.r")
>ex3.1()
The output is:

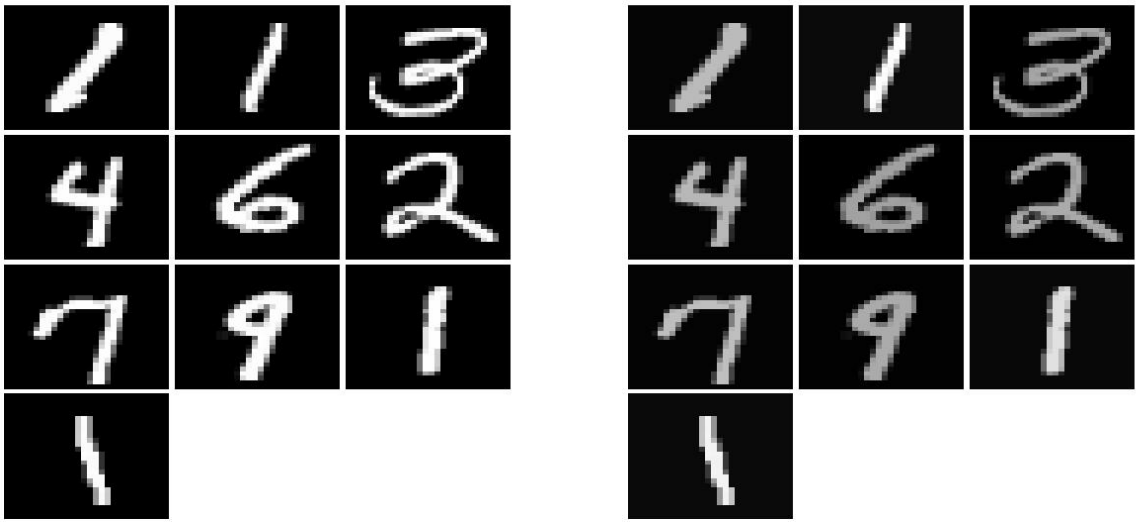


Figure3.1: The corresponding plot of digits (first 10 digits) before and after preprocessing. The preprocessing was done to remove the mean and normalizing the data to unit norm.

Results:
It can be seen that after reducing the mean and normalization the intensity of digits has been reduced.

Exercise3.2
Perform PCA on preprocessed data.
>source("Exercise1.r")
>ex3.2()
The output is:
Percentage of variance explained by first 20 PCs
65.8001%

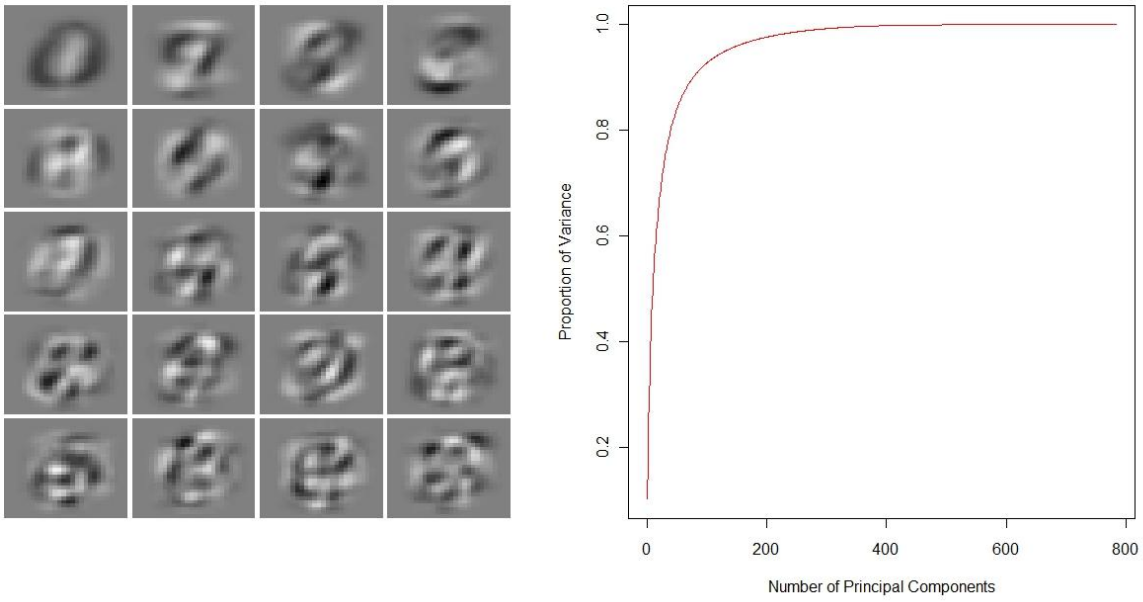


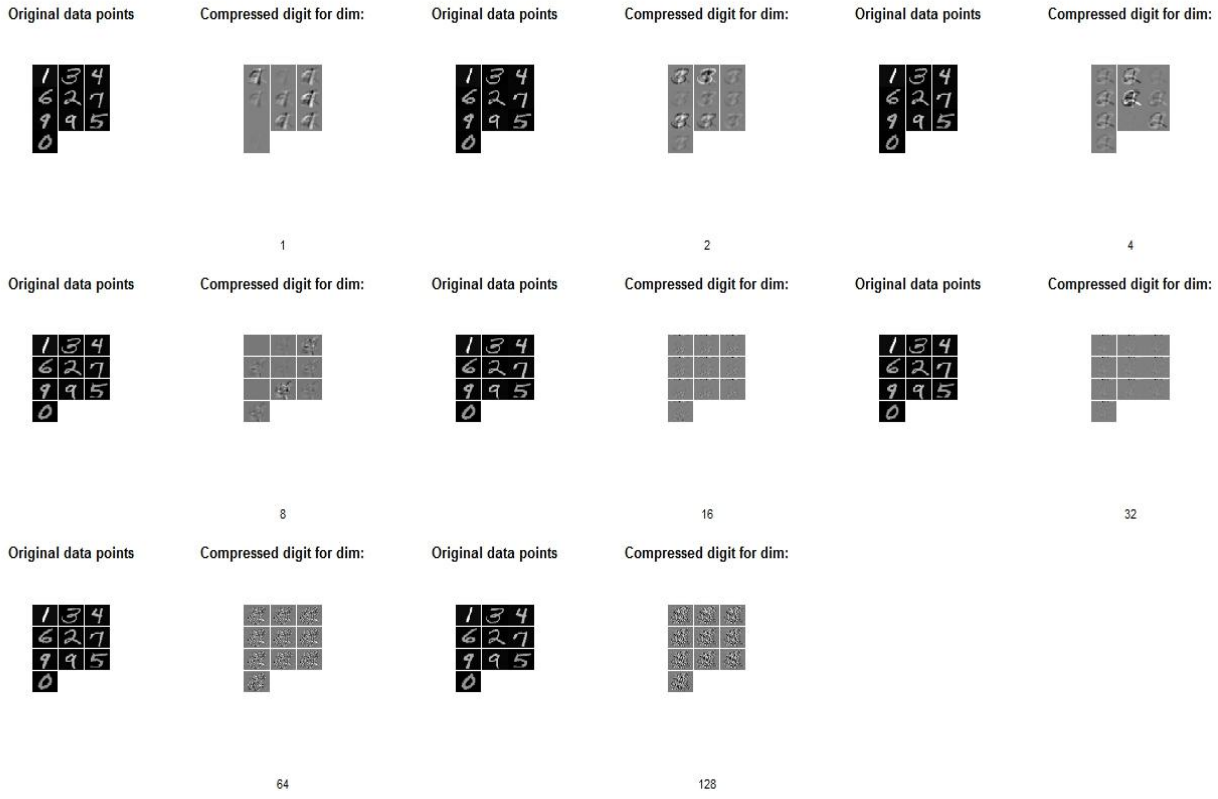
Figure 3.2: This explains the first 20 principal component directions. The other part explains the plot of proportion of variance vs number of principal components.

Results: The figure in plot 1 explains the first principal component directions. The proportion of variance explained by these PCs is 65.8%. The figure in plot 2 explains the proportion of variance vs number of principal components. As number of PCs increases the proportion of variance also increases. The rate of increment is high during beginning but reaches a point where it is maximum and then it reaches the plateau as number of PC is increased. Hence, we can choose the point where rate of increment of proportion of variance is maximum for further analysis.

Exercise3.3
Commands:
>source(“Exercise3.3”)
>ex3.3()
The output is:
the mean error

1	0.3785539
2	0.3789369
4	0.3809568
8	0.3876195
16	0.3880918
32	0.3864755
64	0.3880976
128	0.3875803

The corresponding original digits and compressed digits for each dimension:



Exercise3.4
Commands:
>source("Exercise1.r")
>ex3.4()
The output is:

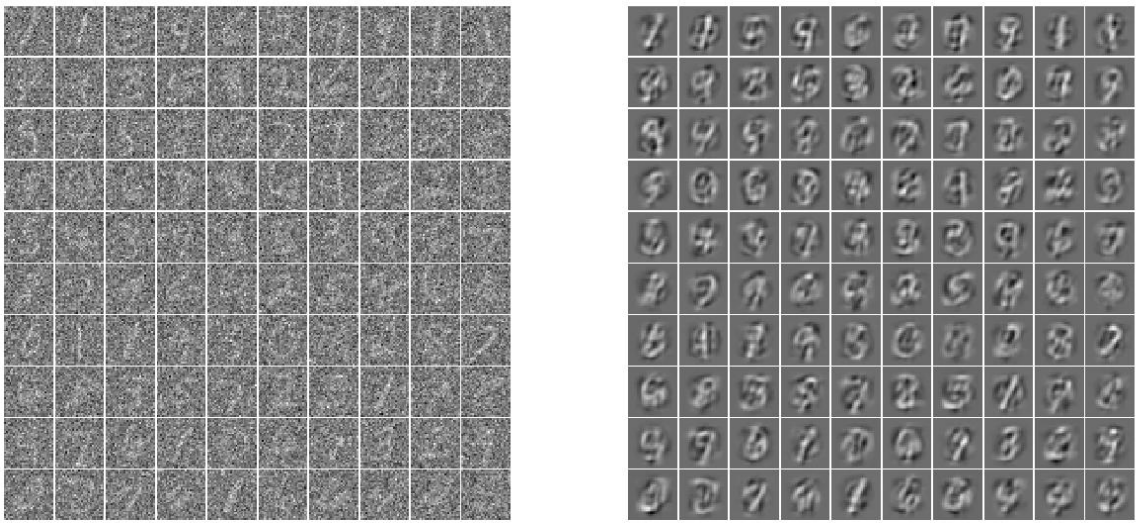


Figure 3.4: The plot in left is the original noise data. The right plot is digit data reduced with noise.

Explanation: The noise digit data was projected to the eigen space of normalized original digit data from digit.txt. The noised data was projected to subspace of first 30 principal components. The choice of this number was just by random. Then projected data was re-projected to same subspace to get back the noise reduced data as explained in the theory:

$$\mathbf{x.hat} = \mathbf{A}\mathbf{A}^{\mathbf{T}}\mathbf{x}$$

where, **A** has orthogonal eigen vectors of original digit data (normalised one from "digits.txt"). The eigen vectors were estimated in this exercise as well.

Result: Thus we can see that the noise has been reduced from given data.