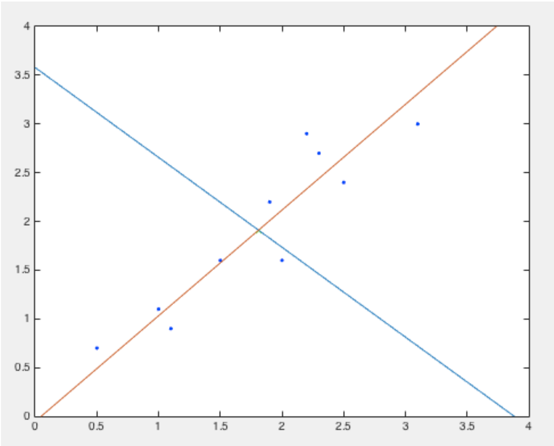
1.

(a)

e1=[0.6779 , 0.7352]

e2=[-0.7352 , 0.6779]



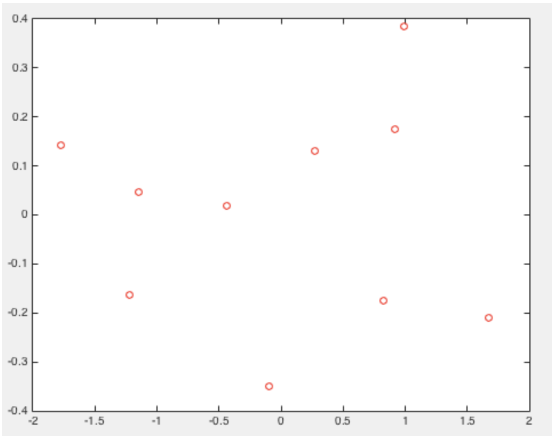
```
clear
clc
X=[2.5 0.5 2.2 1.9 3.1 2.3 2.0 1.0 1.5 1.1;
    2.4 0.7 2.9 2.2 3.0 2.7 1.6 1.1 1.6 0.9];
X=X';
mx=sum(X(:,1))/10;
my=sum(X(:,2))/10;
plot(mx,my,'Xg');
hold on;
D=[X(:,1)-mx X(:,2)-my];
cov=D'*D/10;
[evector,evalue]=eig(cov);
x=X(:,1);
y=X(:,2);
plot(x,y,'.b');
hold on;
a=evector(1,1);
b=evector(1,2);
x=[0:4];
y=1/a*(b*x-(b*mx-a*my));
plot(x,y);
hold on;
a=evector(2,1);
b=evector(2,2);
y=1/a*(b*x-(b*mx-a*my));
plot(x,y);
axis([0 4 0 4]);
```

I subtracted mean values from the original data and used “eig” to calculate eigenvectors, which are e1 and e2 shown above.

(b)

[a1,a2]=

0.8280	-0.1751
-1.7776	0.1429
0.9922	0.3844
0.2742	0.1304
1.6758	-0.2095
0.9129	0.1753
-0.0991	-0.3498
-1.1446	0.0464
-0.4380	0.0178
-1.2238	-0.1627



(c) optimal 1D representation=

0.8280
-1.7776
0.9922
0.2742
1.6758
0.9129
-0.0991
-1.1446
-0.4380
-1.2238

Range = 3.4534 (max - min)

2.

(a)

4x4 inner product matrix =

69.8750	-18.8750	-26.3750	-24.6250
-18.8750	121.3750	-53.1250	-49.3750
-26.3750	-53.1250	98.3750	-18.8750
-24.6250	-49.3750	-18.8750	92.8750

minimum square error= 1.4329×10^{-30}

I subtracted the mean value from original data and took inner product to get 4x4 scatter matrix. Then I took the inner product of the original data and the scatter matrix to get 4 eigenfaces. I got rid of the less important eigenface and reconstructed the sample data with three eigenfaces and calculated the minimum square error.

(b)

weights for reconstruct the second sample = -3.3429 0.7295 -10.4722

(c)

mean-squared error (3D space) = 2.5216×10^{-29}

code for the questions above

```
clear
clc
data=[-2 1 2 -3 4 1 0 3 0 2 1 1 2 3 -2 -3 2 1 0;
    1 2 -4 2 -4 2 5 2 2 1 -3 0 0 1 -2 1 1 -3 -2;
    1 -3 2 1 0 -3 -5 -1 3 3 -2 -3 -2 -1 1 0 5 4 2;
    3 -1 0 2 2 -5 -4 -1 2 -1 3 4 4 2 1 2 -2 1 -1];
for i=1:19
    Mean(i)=sum(data(:,i))/4;
    A(:,i)=data(:,i)-Mean(i);
end

D=A*A';
[evector,evalue]=eig(D);
evector(:,1)=[];
eface=A'*evector;
for i=1:3
    eface(:,i)=eface(:,i)/norm(eface(:,i));
end

weights=A*eface;

recon=zeros(19,4);
for i=1:4
    for j=1:3
        recon(:,i)=recon(:,i)+eface(:,j)*weights(i,j);
    end
    recon(:,i)=recon(:,i)+Mean';
end
SquareError=sum((recon-data').^2);
MinSquareError=min(SquareError);
MeanSquareError=sum(sum((recon-data').^2));
```

(d)

mean-squared error (2D space) = 23.1579

(e)

Euclidean distance for each sample = 12.9228 5.6569 16.7929 15.7797

The 2nd sample is the most similar one.

(f)

Euclidean distance for each sample = 12.9228 5.6569 16.7929 15.7797

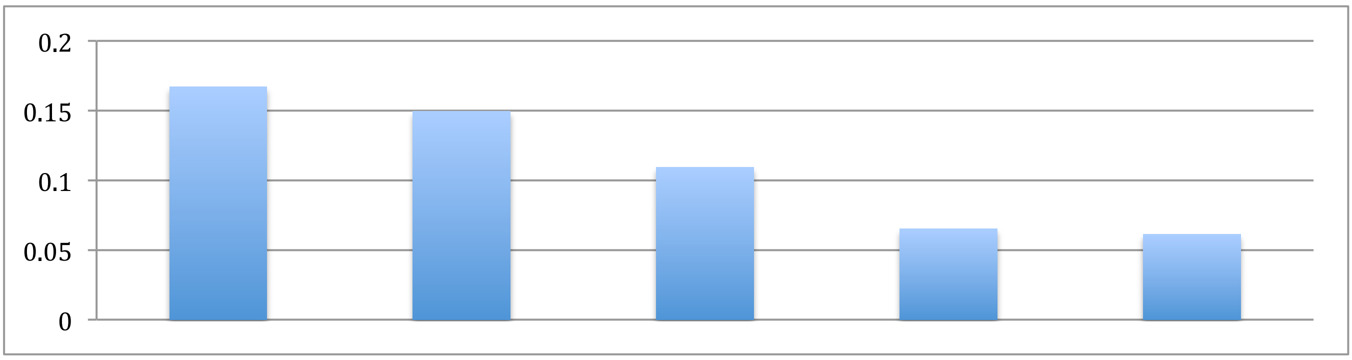
The 2st sample is the most similar one.

The results match and I think it make intuitive sense. Because I used three of the four eigenvectors to reconstruct the data, in which the eigenvalue corresponding to the fourth eigenvector is really small, meaning that it's not important and can be disposed. I think in the case we can almost perfectly reconstruct the data with three important eigenvectors so the fact that reconstructed version matches the original data really make sense.

3.

(a)

Number of dimension: 35



Bar Graph(variance)

(b)

first reconstructed sample (from left to right: original,10,20,30,40,50):

(c)

overall test accuracy: 83.33%

(d)

overall test accuracy: 83.33%

The accuracy calculated from reduced space and original space are quite the same. Originally I thought reduced space would have had lower accuracy because of less amount of calculation. But it turns out that they have almost the same results, which means the reduced dimension method would have good result with better efficiency.