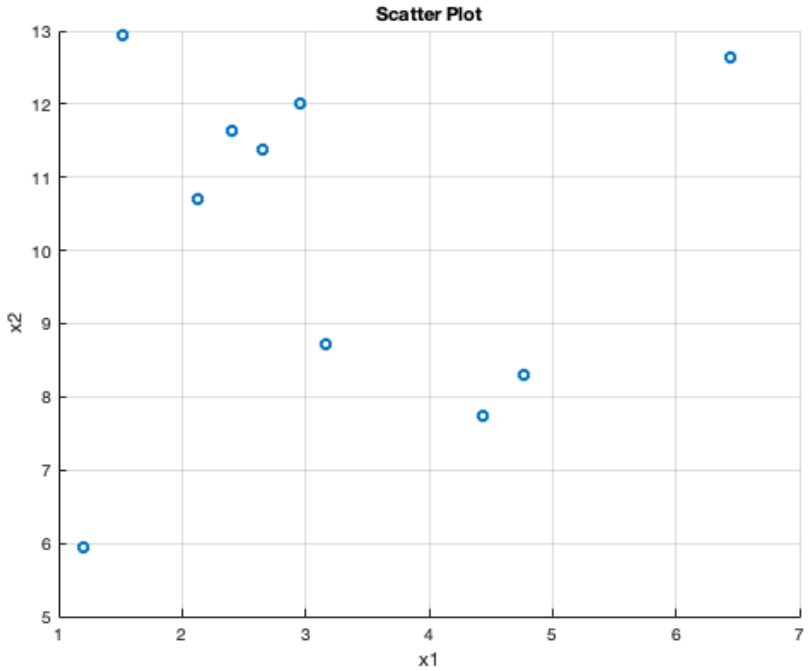
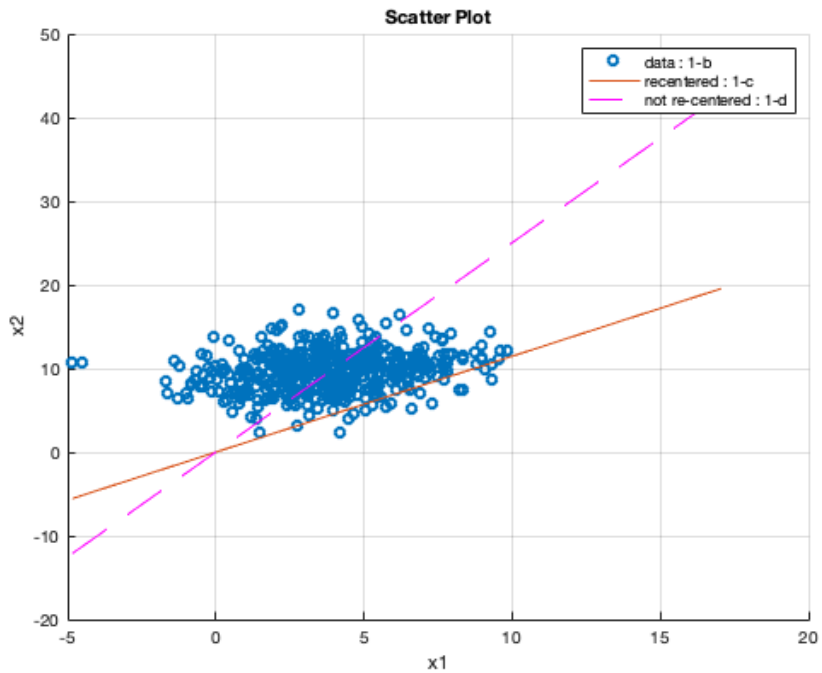
```
% Homework 5 - Question 1

disp('1a)')
mu = [4 ;10];
W = [1; 1];
sigma_sq = 4;
zi = 0;
M = 10;
X = mvnrnd(W*zi+mu, sigma_sq*eye(2), M);
figure(1);
scatter(X(:,1), X(:,2));
xlabel('x1');
ylabel('x2');
title('Scatter Plot');
grid on
snapnow;

disp('1b)')
N = 50;
z = normrnd(0, 1, [N,1]);
X = [];
for i=1:N
    mu = [4 ;10];
    W = [1; 1];
    sigma_sq = 4;
    zi = z(i);
    M = 10;
    X = [X; mvnrnd(W*zi+mu, sigma_sq*eye(2), M)];
end
figure(2);
scatter(X(:,1), X(:,2));
grid on
xlabel('x1');
ylabel('x2');
title('Scatter Plot');


disp('1c)')
X_new= X - mean(X);
n = size(X_new,1);
S = 1/n * (X_new' * X_new);
[V, ~] = eigs(S, 1);
hold on
tx = [min(X(:,1)) max(X(:,2))];
ty = V(2)*tx/V(1);
plot(tx, ty);
grid on

disp('1d)')
n = size(X,1);
S = 1/n * (X' * X);
[V, ~] = eigs(S, 1);
tx = [min(X(:,1)) max(X(:,2))];
ty = V(2)*tx/V(1);
plot(tx, ty, '--m');

legend('data : 1-b', 'recentered : 1-c', 'not re-centered : 1-d')
snapnow;
disp(" The variance is not captured properly if data is not re-centered.")

disp('1e)')
mu = [4 ;10];
W = [1; 1];
sigma_sq = 4;
N = 500;
X = zeros(N,2);
for i=1:N
    zi = normrnd(0, 1);
    X(i,:)=mvnrnd(W*zi+mu, sigma_sq*eye(2));
end
figure(5);
scatter(X(:,1), X(:,2));
grid on
xlabel('x1');
```

```
ylabel('x2');
title('Scatter Plot');
snapnow;
disp("In PPCA, we generate zi and then generate Xi - repeat this 500 times. This is different from 1d, where");
disp("zi is randomly generated 50 times and then used to generate Xi. Here zi's are randomly picked 500 times.");
```

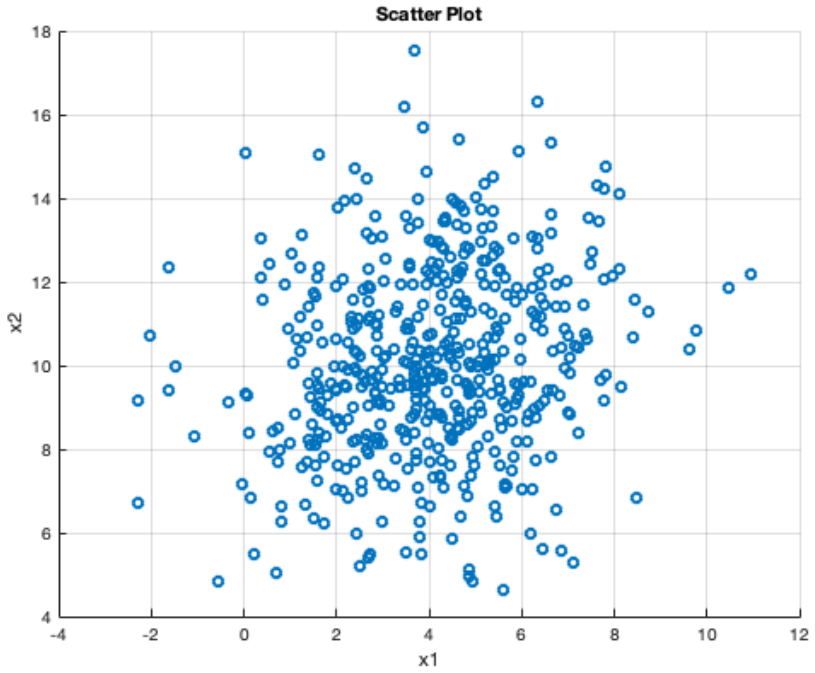
1a)



1b)
1c)
1d)



The variance is not captured properly if data is not re-centered.
1e)



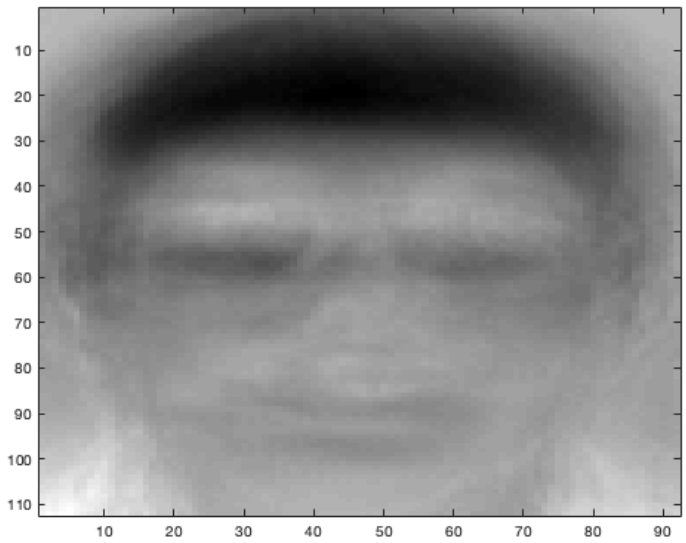

```
In PPCA, we generate zi and then generate Xi - repeat this 500 times. This is different from 1d, where
zi is randomly generated 50 times and then used to generate Xi. Here zi's are randomly picked 500 times.
```

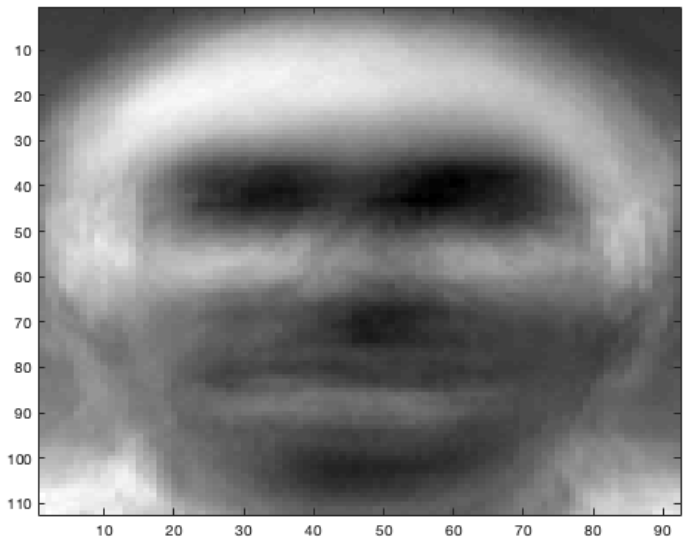
Published with MATLAB® R2018a

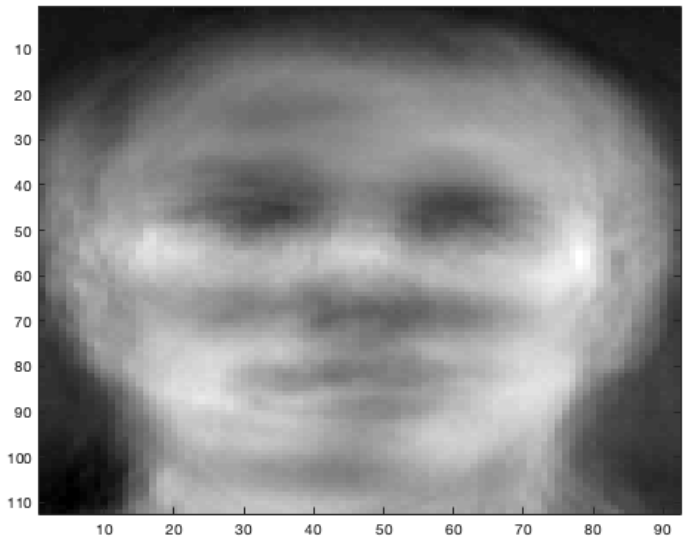
```
% Homework 5 - Question 2
X_train = csvread('X_train.csv');
Y_train = csvread('Y_train.csv');
%figure(1);colormap(gray);
%imagesc(reshape(X(10,:),112,92));
disp('2a)')
M = 112;
N = 92;
X_train_new = X_train - mean(X_train);
n = size(X_train_new,1);
S = 1/n * (X_train_new' * X_train_new);
[V, ~] = eigs(S, 10);

for i=1:10
    figure();colormap(gray);
    imagesc(reshape(V(:,i), M, N));
    snapnow;
end
disp("Characteristics captured by eigenfaces :-");
fprintf("1 - Most of the Hair \n 2 - eyes and nose \n 3 - eyes \n 4 - Hair \n 5 - Right side of face");
fprintf("6 - Hair, eyes and mouth \n 7 - Cheeks \n \n 8 - One eye \n 9 - Chin \n 10 - eyebrows, nose and mustache, ears, chin \n");

disp('2b)');
D = eigs(S, M*N);
plot(D);xlim([0 100]);
title('Elbow Plot');
snapnow;
elbow = 20;
fprintf('Elbow is approximately found at q=%d \n', elbow);
var = (sum(D(1:elbow))*100)/sum(D);
fprintf("Percentage of variance explained = %f \n", var);

disp('2c)');
X_test = csvread('X_test.csv');
Y_test = csvread('Y_test.csv');
[V, ~] = eigs(S, 50);
X_P_train = X_train_new*V;
X_test_new = X_test - mean(X_test);
X_P_test = X_test_new*V;
correctly_classified = 0;
Y_pred=zeros(size(X_P_test,1));
for i=1:size(X_P_test,1)
    dist = zeros(size(X_train,1),1);
    for j=1:size(X_train,1)
        dist(j)=norm(X_P_train(j,:)'-X_P_test(i,:)');
    end
    [~,ind] = min(dist);
    Y_pred(i) = Y_train(ind);
    if Y_test(i)==Y_pred(i)
        correctly_classified = correctly_classified+1;
    else
        figure
        colormap(gray);
        imagesc(reshape(X_test(i,:), M, N));
        title('Test Image')
        snapnow;
        figure
        colormap(gray);
        imagesc(reshape(X_train(ind,:), M, N));
        title('Train Image')
        snapnow;
    end
end

fprintf("Correctly classified Fraction with PCA = %.2f%% \n", (correctly_classified*100)/size(X_test,1));
correctly_classified = 0;
disp('2d)');
for i=1:size(X_test,1)
    dist = zeros(size(X_train,1),1);
    for j=1:size(X_train,1)
        dist(j)=norm(X_train(j,:)'-X_test(i,:)');
    end
    [~,ind] = min(dist);
    Y_pred(i) = Y_train(ind);
    if Y_test(i)==Y_pred(i)
        correctly_classified = correctly_classified+1;
    else
        figure
        colormap(gray);
        imagesc(reshape(X_test(i,:), M, N));
        title('Test Image')
        snapnow;
        figure
        colormap(gray);
        imagesc(reshape(X_train(ind,:), M, N));
        title('Train Image')
```

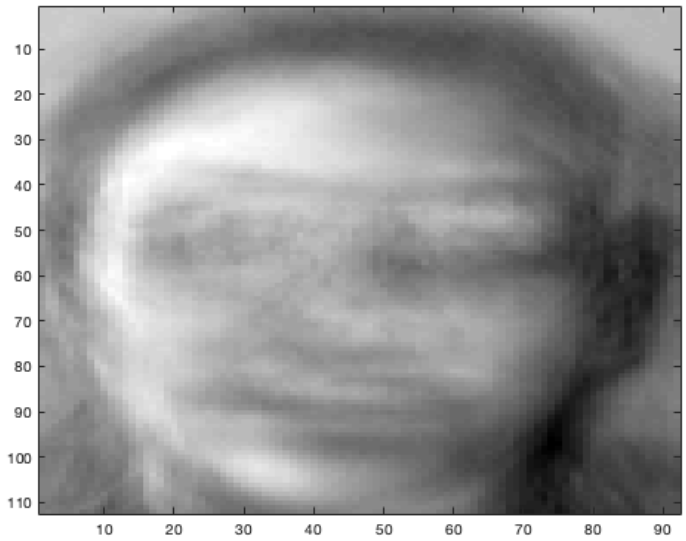
```
        snapnow;
    end
end
fprintf("Correctly classified Fraction without PCA = %.2f%% \n", (correctly_classified*100)/size(X_test,1));
```

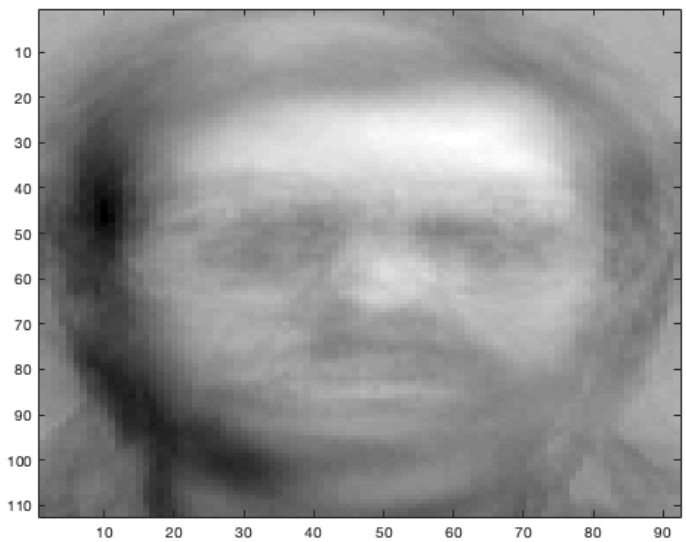
2a)

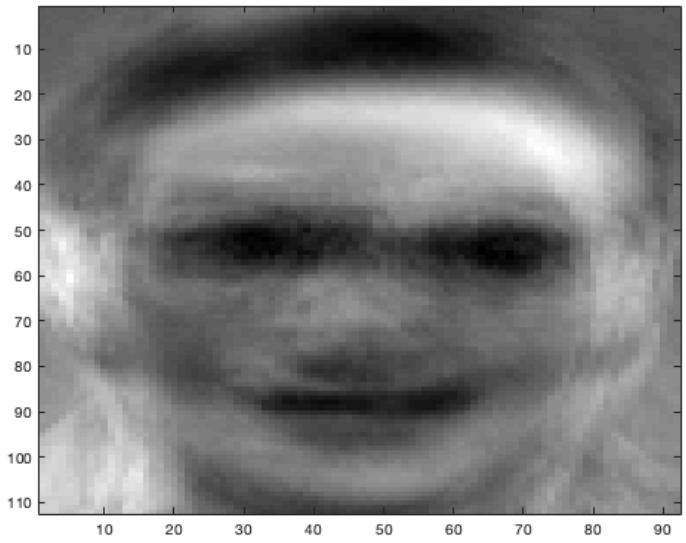


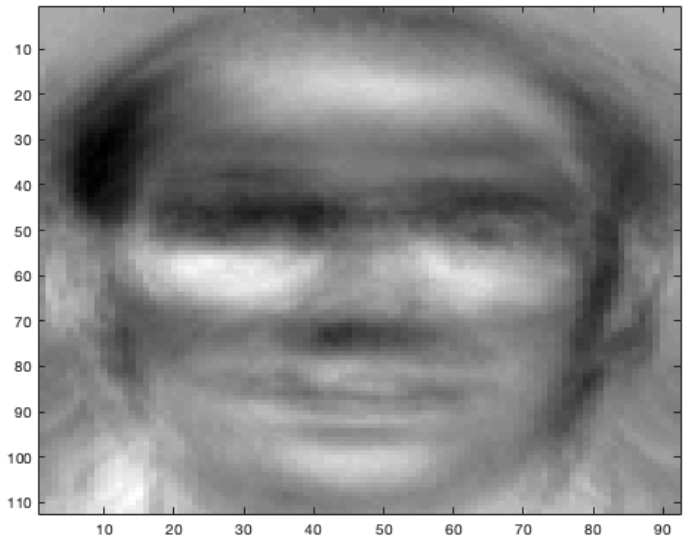


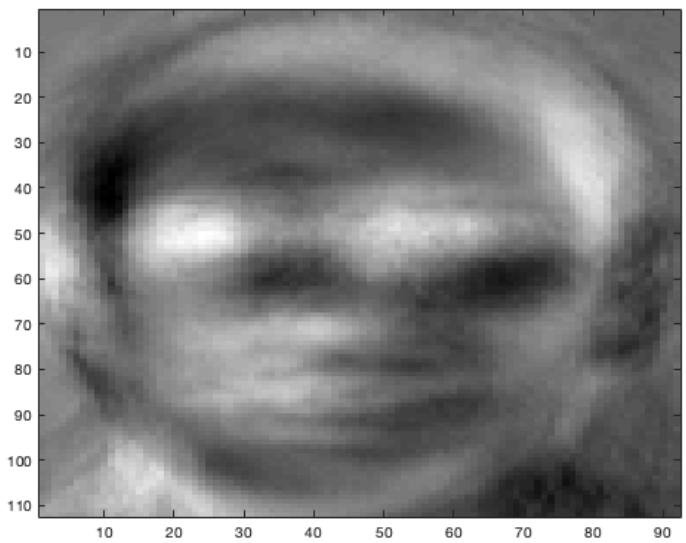

10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90

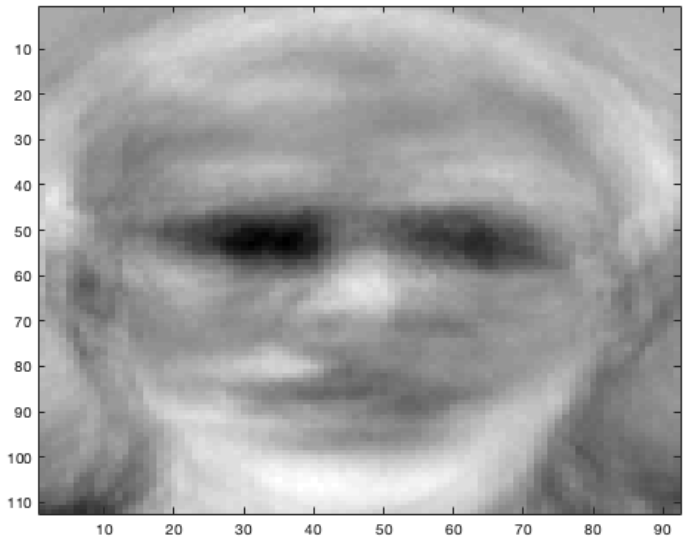

10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90

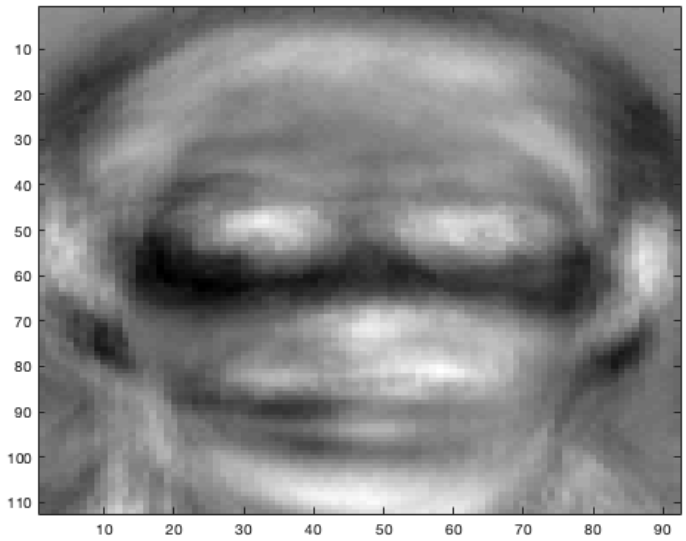

10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90


10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90

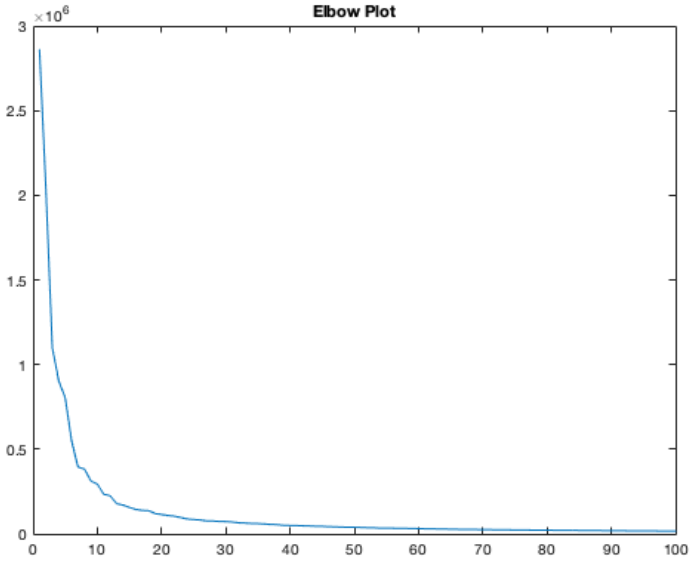

10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90


10
20
30
40
50
60
70
80
90
100
110
10 20 30 40 50 60 70 80 90

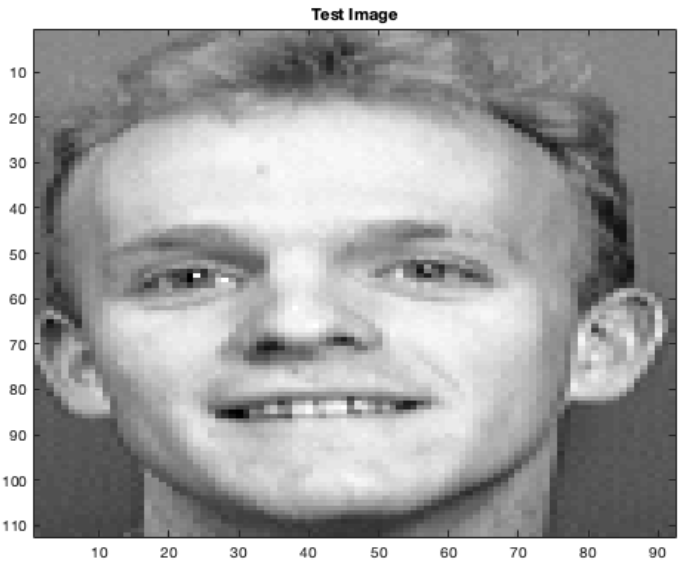


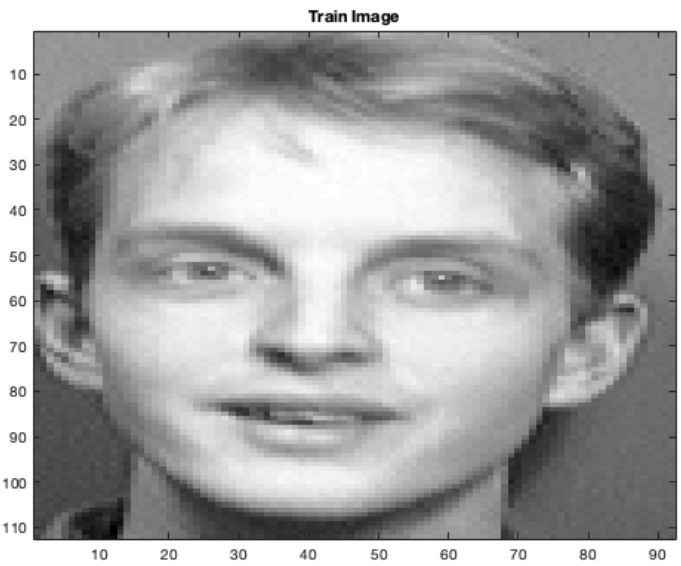


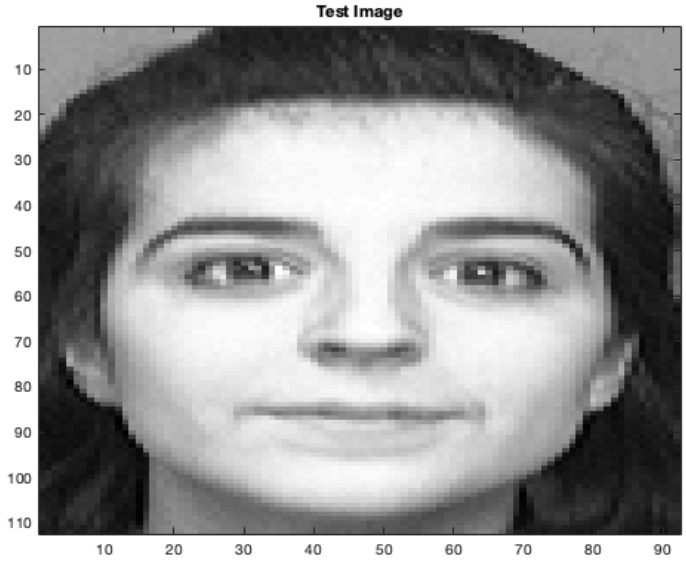

```
Characteristics captured by eigenfaces :-
1 - Most of the Hair
 2 - eyes and nose
 3 - eyes
 4 - Hair
 5 - Right side of face6 - Hair, eyes and mouth
 7 - Cheeks

 8 - One eye
 9 - Chin
 10 - eyebrows, nose and mustache, ears, chin
2b)
```

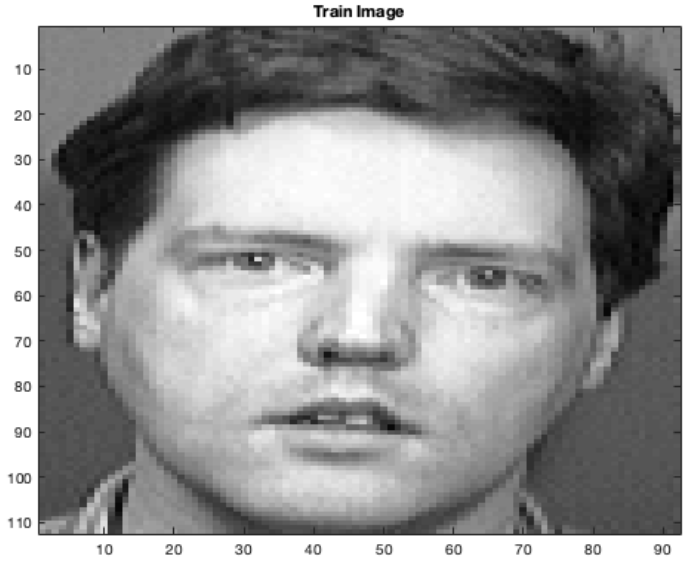



```
Elbow is approximately found at q=20
Percentage of variance explained = 70.266018
2c)
```

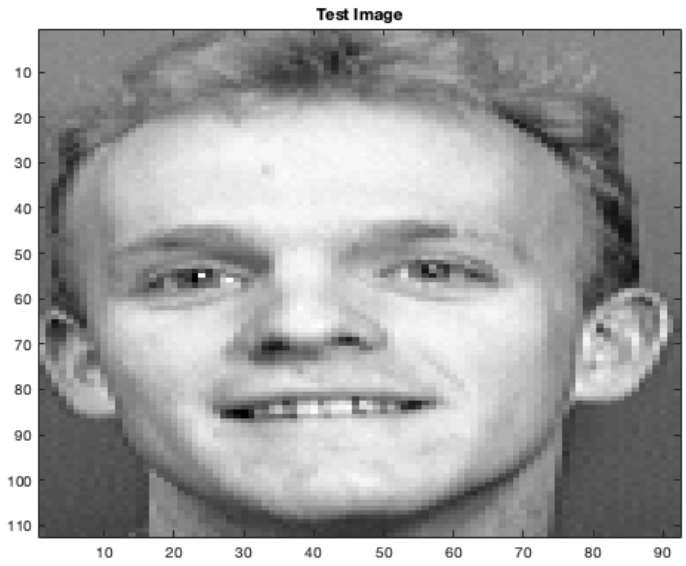


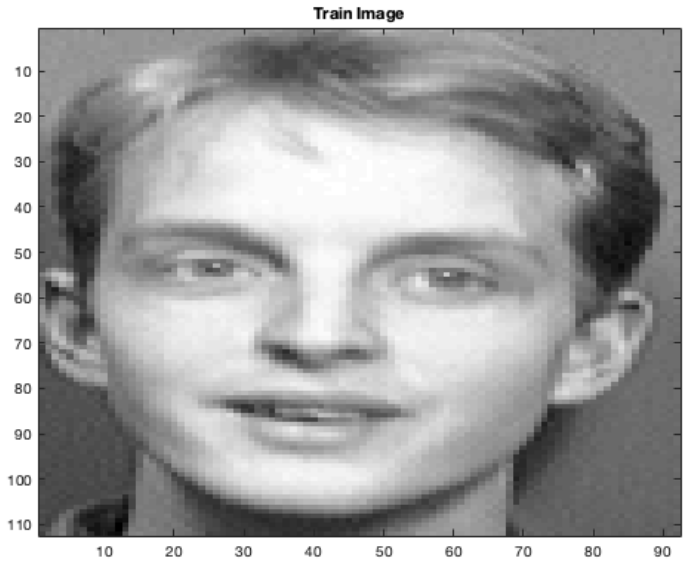


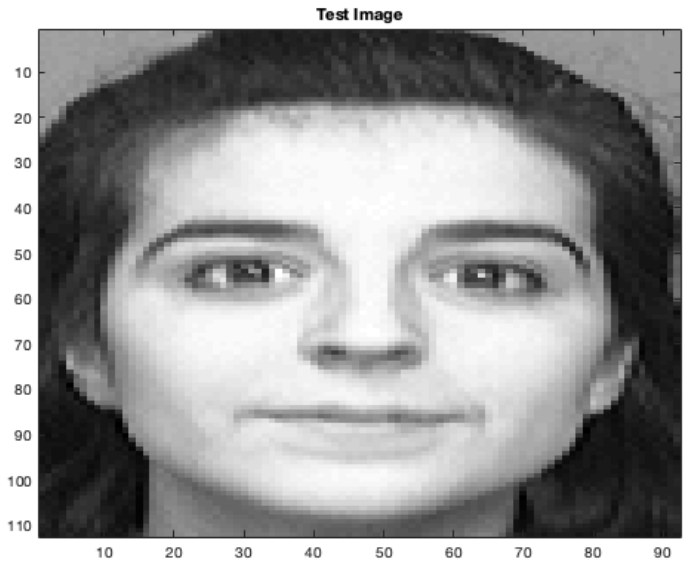


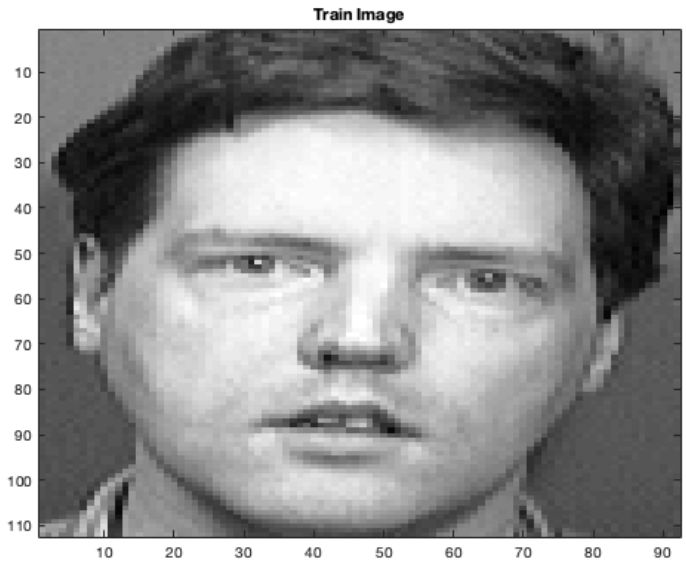


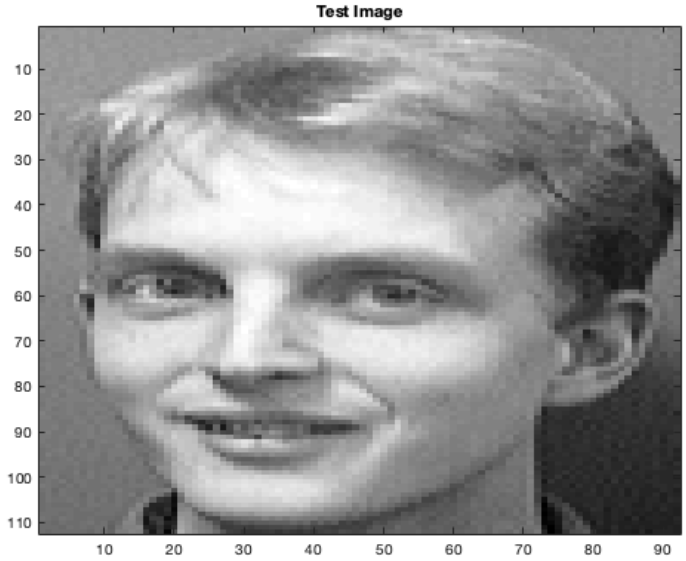

```
Correctly classified Fraction with PCA = 95.00%
2d)
```

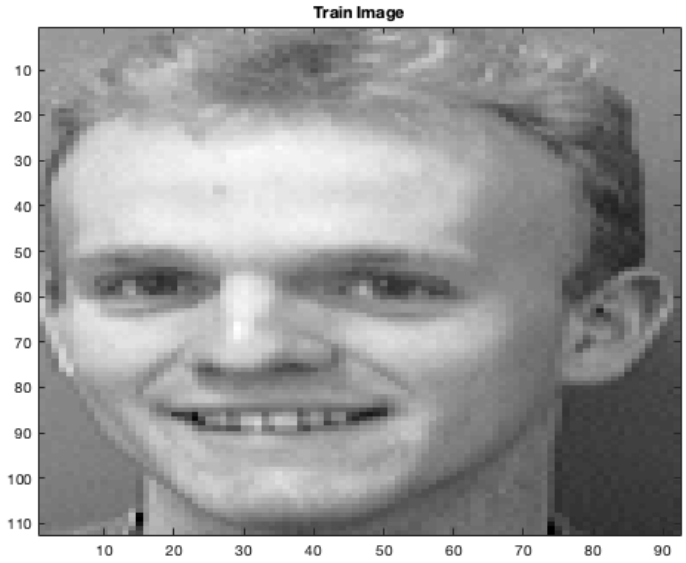

Test Image
10
20
30
40
50
60
70
80
90
100
110
10
20
30
40
50
60
70
80
90


Train Image
10
20
30
40
50
60
70
80
90
100
110
10
20
30
40
50
60
70
80
90


Test Image
10
20
30
40
50
60
70
80
90
100
110
10
20
30
40
50
60
70
80
90







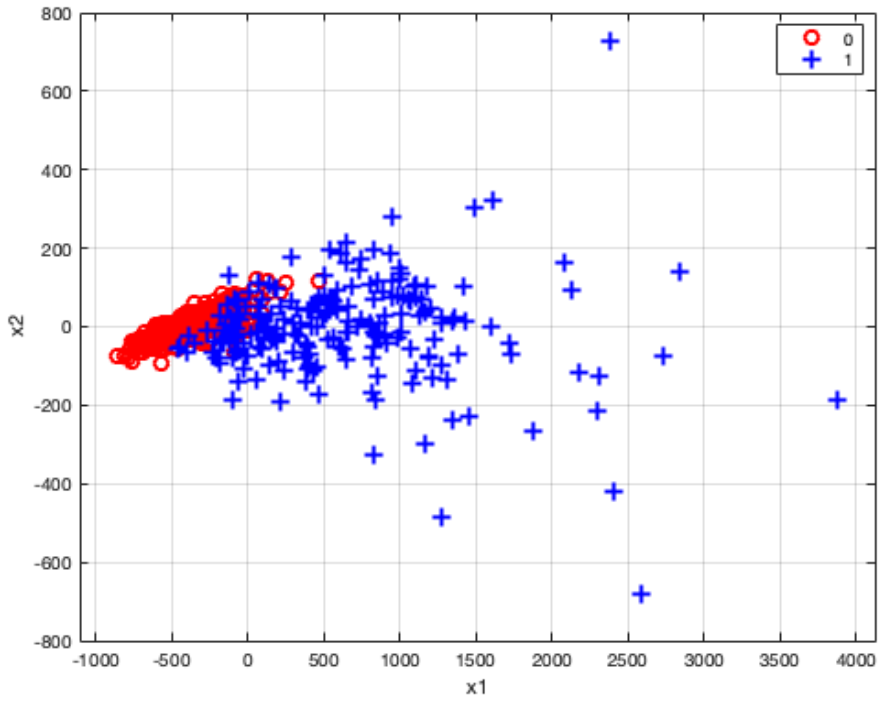

```
Correctly classified Fraction without PCA = 92.50%
```

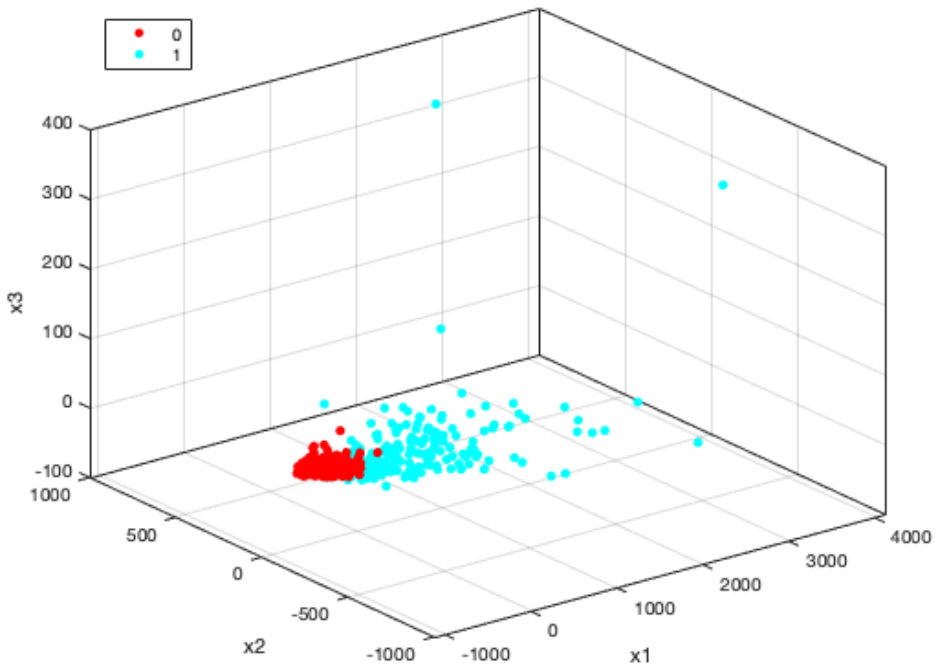
Published with MATLAB® R2018a

```
% Homework 5 Question 3
data = readtable('bc_wisc.csv');
data = data.Variables;
y = data(:, 2);
X = data(:,3:end);
disp('3a)')
B_glm = glmfit(X,y,'binomial');
disp('Maximum iteration exceeded - failed to converge - due to high dimension')
disp('3b)')
X_new= X - mean(X);
n = size(X_new,1);
S = 1/n * (X_new' * X_new);
[V, D] = eigs(S, 30);
D = diag(D);
lambda = [];
for q=2:10
    fprintf("q=%d , Percentage of variance explained =%.2f%% \n", q, (sum(D(1:q))*100)/sum(D));
end

X_P = X_new*V;
disp('3c)')
figure(1);
gscatter(X_P(:,1), X_P(:,2), y, 'rb', 'o+', 8, 'on', 'x1', 'x2');
grid on
xlabel('x1');
ylabel('x2');
snapnow;

figure(2);
h = gscatter(X_P(:,1), X_P(:,2), y);
grid on
xlabel('x1');
ylabel('x2');
zlabel('x3');

gu = unique(y);
x3 = X_P(:,3);
for k = 1:numel(gu)
      set(h(k), 'ZData', x3( y == gu(k) ));
end
view(3);
snapnow;

ind = 1:size(X,1);
disp('Row indices of outliers');
disp(ind(X_P(:,3)>250));

disp('3d)')
for q=2:10
    X_new = X_P(:,1:q);
    B_glm = glmfit(X_new,y,'binomial');
    X_test = [ones(size(y)), X_new];
    y_pred = X_test*B_glm>=0;
    num_correct_predictions = (sum(y_pred==y)*100)/size(y_pred,1);
    fprintf("q=%d, Percentage of correct predictions=%.2f%% \n", q, num_correct_predictions);
end
disp("Percentage of correct predictions increases slightly when we increase q. This is expected");
disp("Most of the variance is captures by q=2. However, when we capture more variance by increasing q,");
disp(" accuracy slightly increases.");
```

```
3a)
Warning: Iteration limit reached.
Maximum iteration exceeded - failed to converge - due to high dimension
3b)
q=2 , Percentage of variance explained =99.82%
q=3 , Percentage of variance explained =99.98%
q=4 , Percentage of variance explained =99.99%
q=5 , Percentage of variance explained =100.00%
```

```
q=6 , Percentage of variance explained =100.00%
q=7 , Percentage of variance explained =100.00%
q=8 , Percentage of variance explained =100.00%
q=9 , Percentage of variance explained =100.00%
q=10 , Percentage of variance explained =100.00%
3c)
```






```
Row indices of outliers
    68   256

3d)
q=2, Percentage of correct predictions=93.21%
q=3, Percentage of correct predictions=93.04%
q=4, Percentage of correct predictions=95.54%
```

```
q=5, Percentage of correct predictions=95.36%
q=6, Percentage of correct predictions=95.36%
q=7, Percentage of correct predictions=94.64%
q=8, Percentage of correct predictions=95.36%
q=9, Percentage of correct predictions=95.71%
q=10, Percentage of correct predictions=96.07%
Percentage of correct predictions increases slightly when we increase q. This is expected
Most of the variance is captures by q=2. However, when we capture more variance by increasing q,
 accuracy slightly increases.
```

Published with MATLAB® R2018a