findClosestCentroids.m

```
M = size(X, 1);
for i = 1:M
        distance = zeros(K, 1);

        for j = 1:K
            temp = X(i, :) - centroids(j, :);
            distance(j) = temp * temp';
        end

        [x ix] = min(distance);
        idx(i) = ix;
end
```

computeCentroids.m

```
for i = 1:K
        data = X(idx == i, :);

        % if data contains only one row
        if size(data, 1) == 1
            centroids(i, :) = data;
        else
            centroids(i, :) = mean(data);
        end
end
```

kMeansInitCentroids.m

```
% Initialize the centroids to be random examples
% Randomly reorder the indices of examples
randidx = randperm(size(X, 1));
% Take the first K examples as centroids
centroids = X(randidx(1:K), :);
```

pca.m

```
%

CorrX = X'*X;
[U,S,V] = svd(CorrX/m);
```

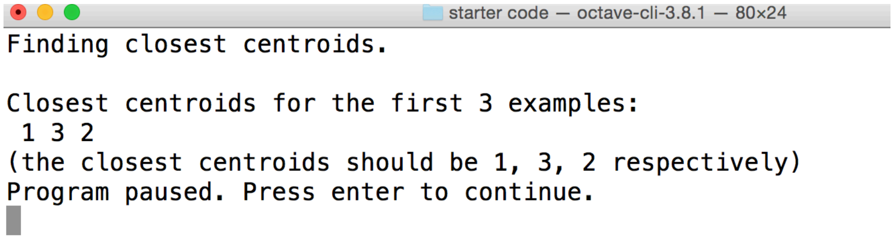
projectData.m

```
Z = X * U(:, 1:K);
```

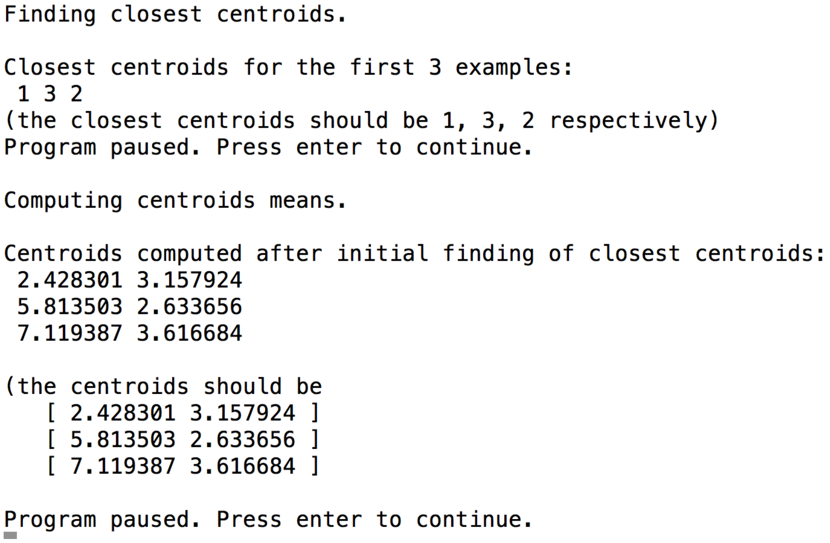
recoverData.m

```
X_rec = Z * U(:, 1:K)';
```

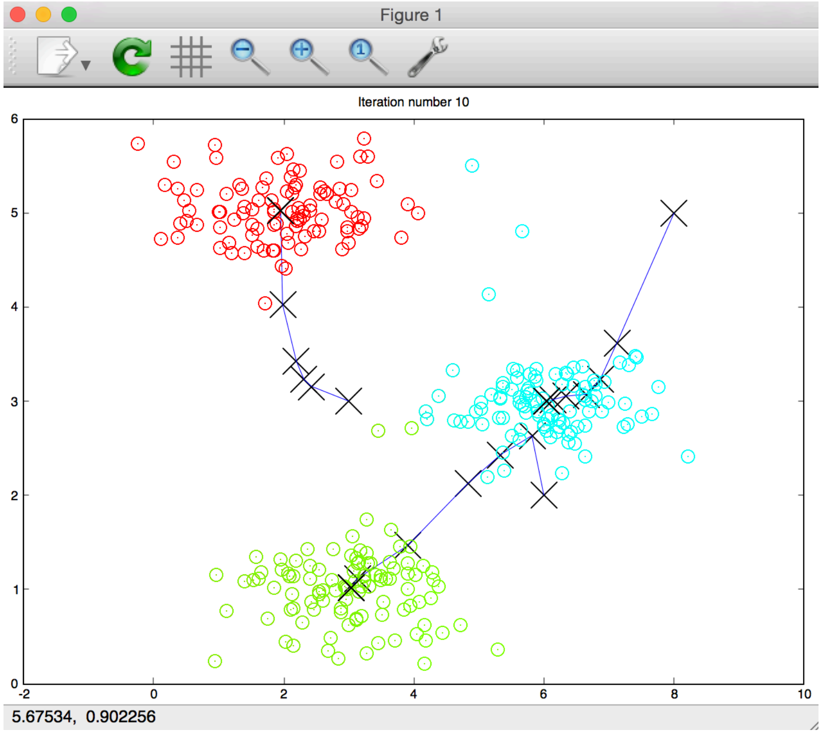
ScreenShoot:

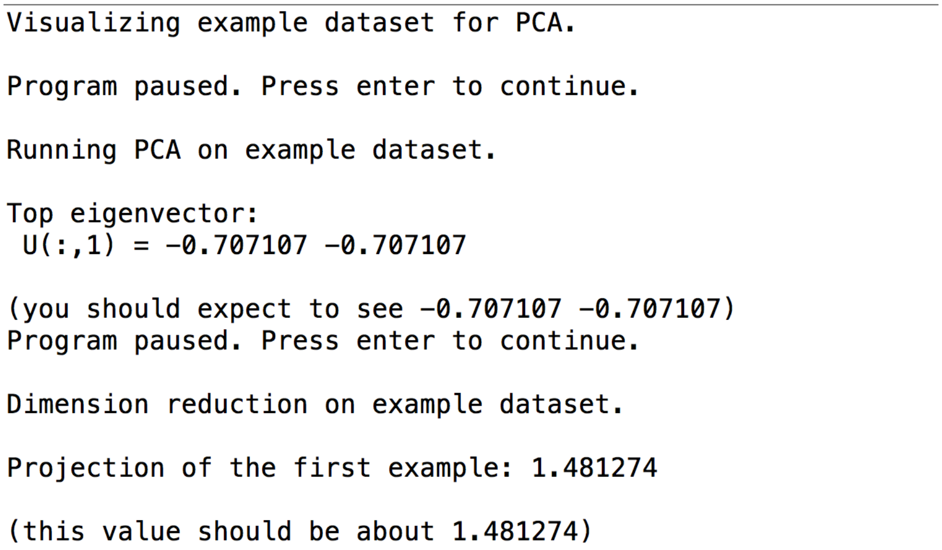


```
Finding closest centroids.

Closest centroids for the first 3 examples:
 1 3 2
(the closest centroids should be 1, 3, 2 respectively)
Program paused. Press enter to continue.
```

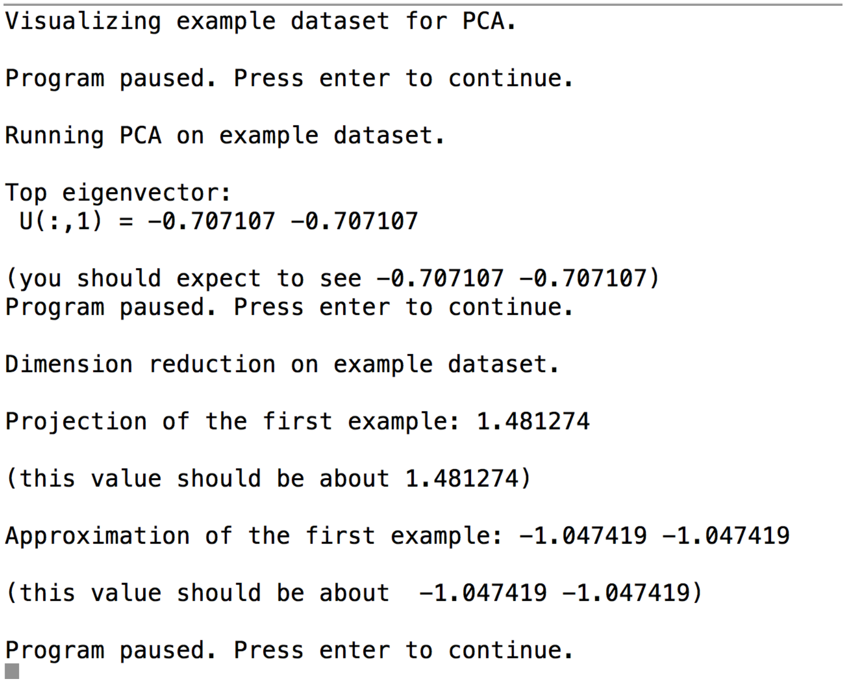


```
Finding closest centroids.

Closest centroids for the first 3 examples:
 1 3 2
(the closest centroids should be 1, 3, 2 respectively)
Program paused. Press enter to continue.

Computing centroids means.

Centroids computed after initial finding of closest centroids:
 2.428301 3.157924
 5.813503 2.633656
 7.119387 3.616684

(the centroids should be
   [ 2.428301 3.157924 ]
   [ 5.813503 2.633656 ]
   [ 7.119387 3.616684 ]

Program paused. Press enter to continue.
```

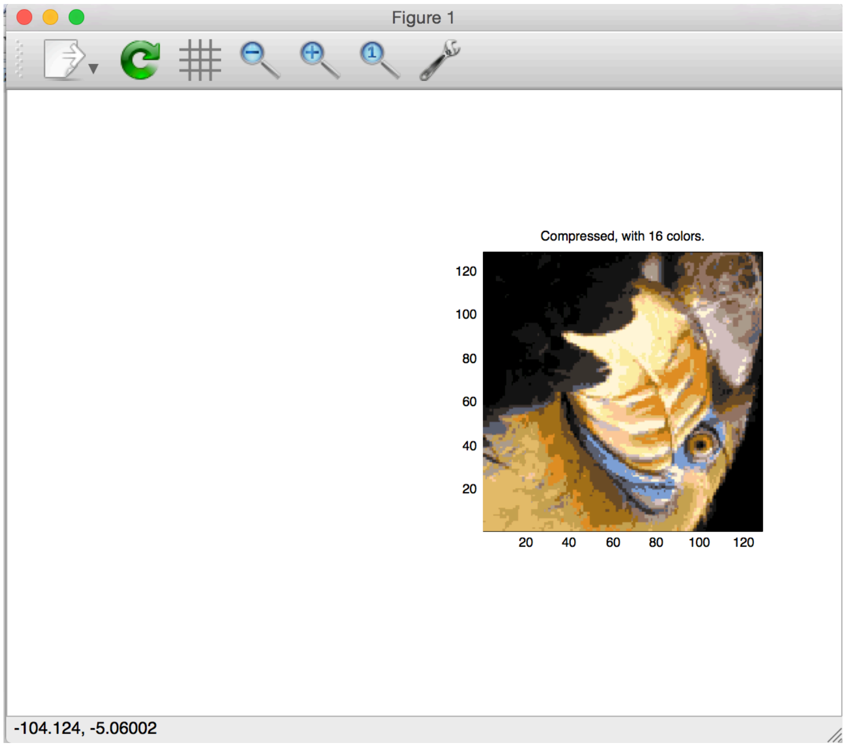


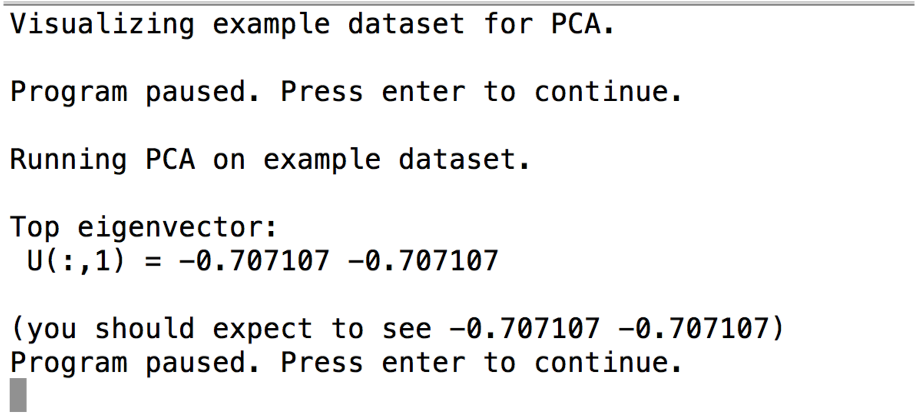


```
Visualizing example dataset for PCA.

Program paused. Press enter to continue.

Running PCA on example dataset.

Top eigenvector:
 U(:,1) = -0.707107 -0.707107

(you should expect to see -0.707107 -0.707107)
Program paused. Press enter to continue.

Dimension reduction on example dataset.

Projection of the first example: 1.481274

(this value should be about 1.481274)
```

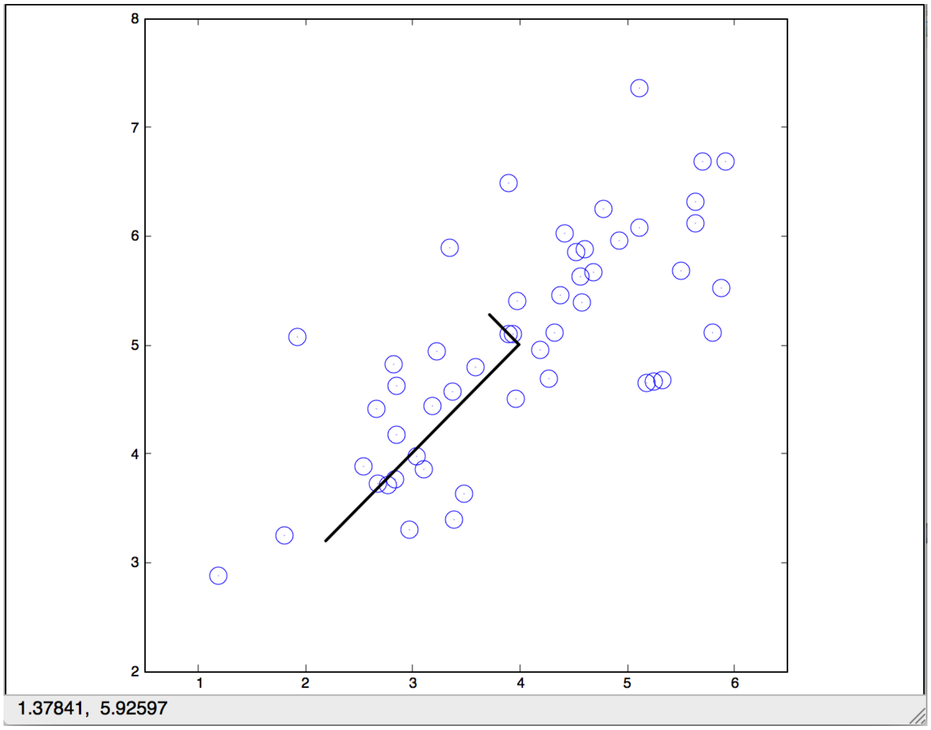


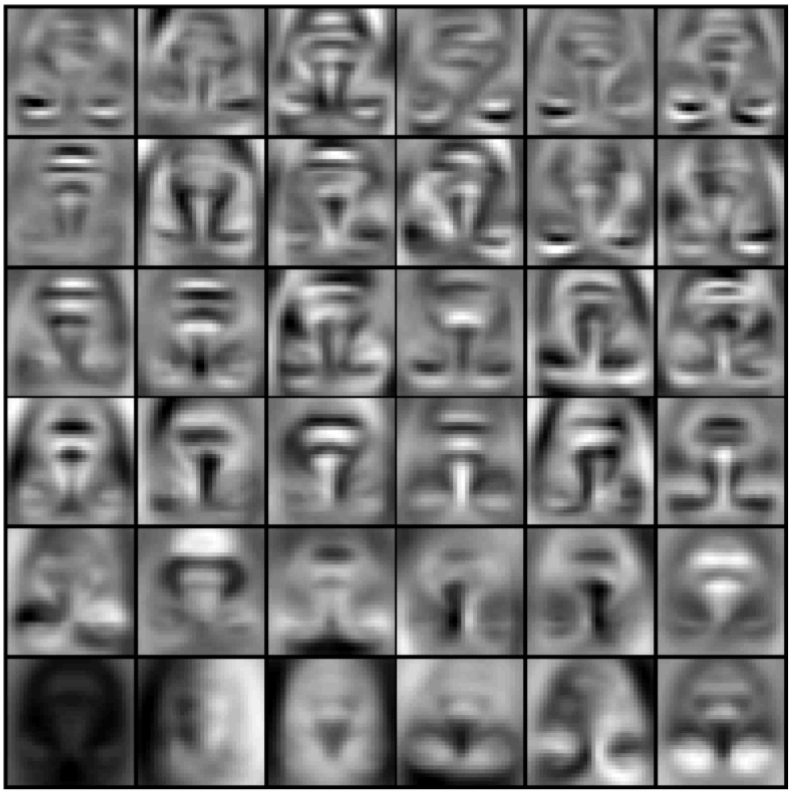
```
Visualizing example dataset for PCA.

Program paused. Press enter to continue.

Running PCA on example dataset.

Top eigenvector:
 U(:,1) = -0.707107 -0.707107

(you should expect to see -0.707107 -0.707107)
Program paused. Press enter to continue.

Dimension reduction on example dataset.

Projection of the first example: 1.481274

(this value should be about 1.481274)

Approximation of the first example: -1.047419 -1.047419

(this value should be about  -1.047419 -1.047419)

Program paused. Press enter to continue.
```

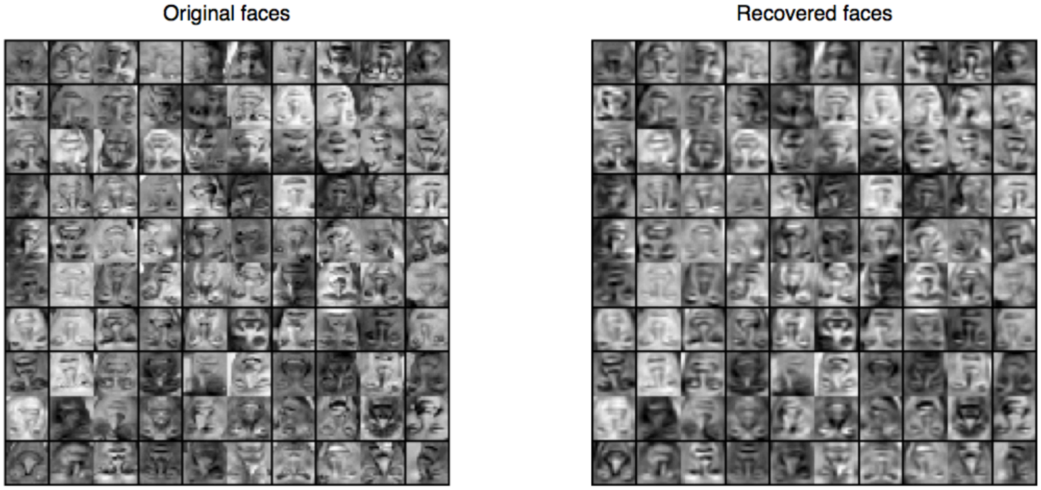


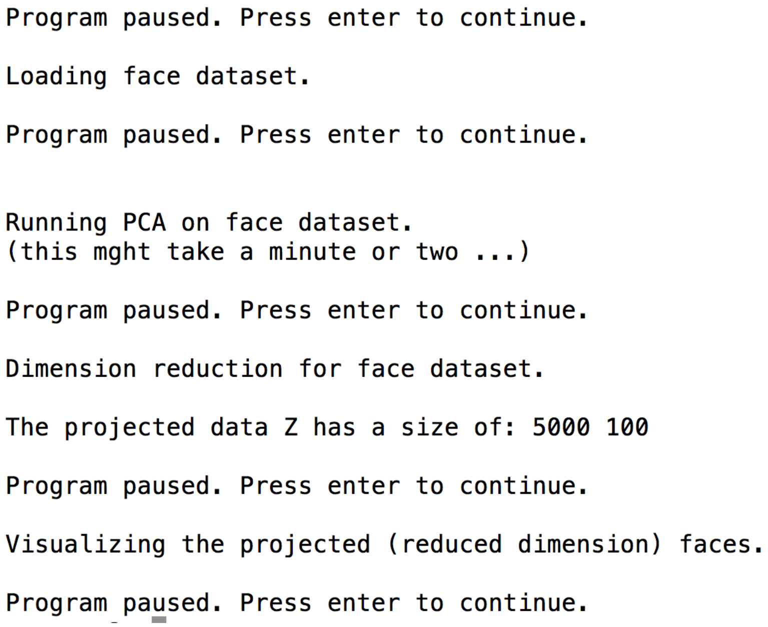


```
Visualizing example dataset for PCA.

Program paused. Press enter to continue.

Running PCA on example dataset.

Top eigenvector:
 U(:,1) = -0.707107 -0.707107

(you should expect to see -0.707107 -0.707107)
Program paused. Press enter to continue.
```









```
Program paused. Press enter to continue.

Loading face dataset.

Program paused. Press enter to continue.


Running PCA on face dataset.
(this mght take a minute or two ...)

Program paused. Press enter to continue.

Dimension reduction for face dataset.

The projected data Z has a size of: 5000 100

Program paused. Press enter to continue.

Visualizing the projected (reduced dimension) faces.

Program paused. Press enter to continue.
```