University of Dayton

Lab 12

CPS 563 – Data Visualization

Dr. Tam Nguyen

tamnguyen@udayton.edu

Outline

- Create PCA function
- Generate eigenfaces

Recall: Practical Computation of PCA

- In practice, we compute the PCs via singular value decomposition (SVD) on the centered data matrix.

- Form the centered data matrix:

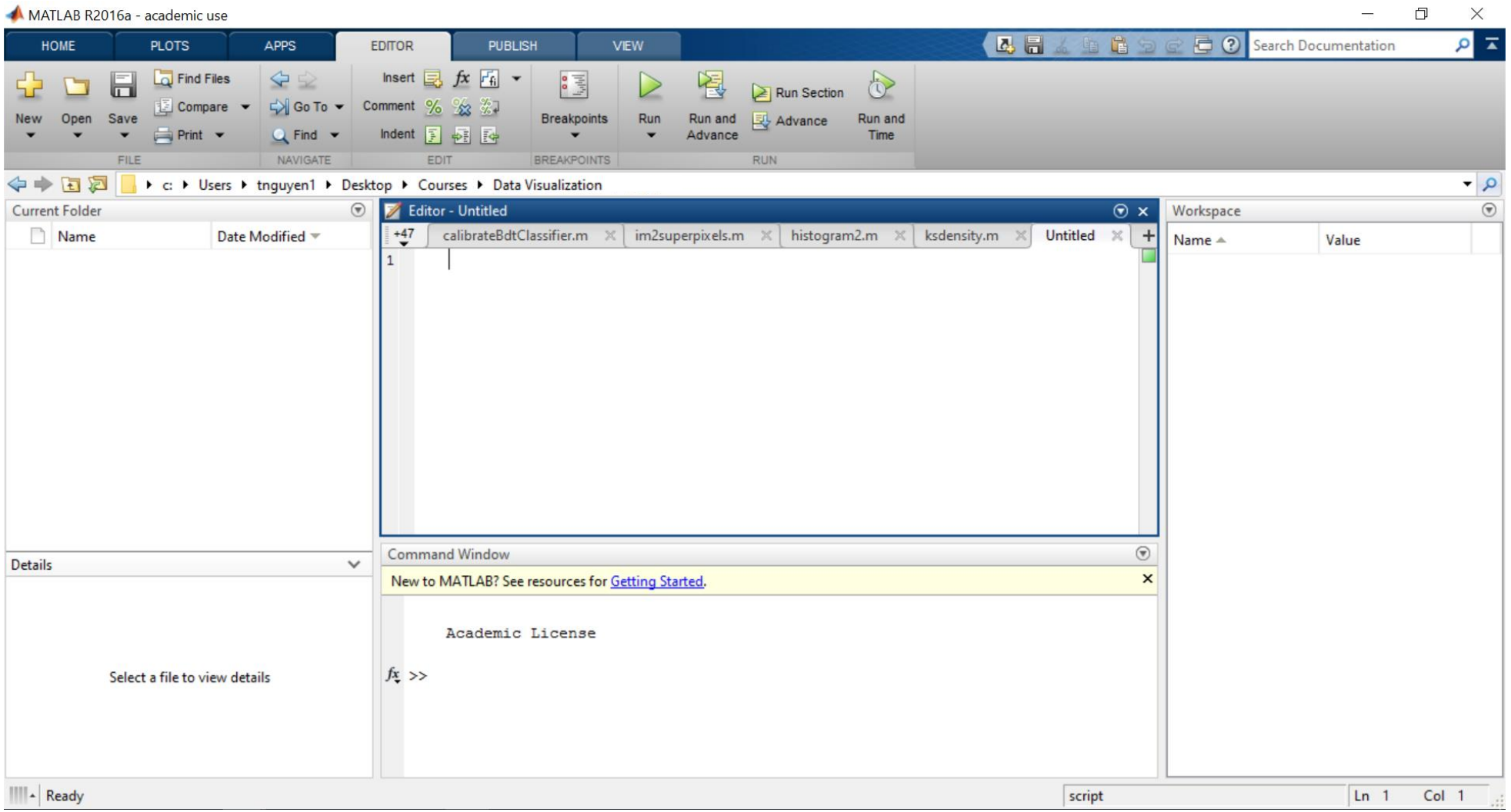
$$X_{d,n} = [(x_1 - \bar{x}) \dots (x_n - \bar{x})]$$

- Compute its SVD:

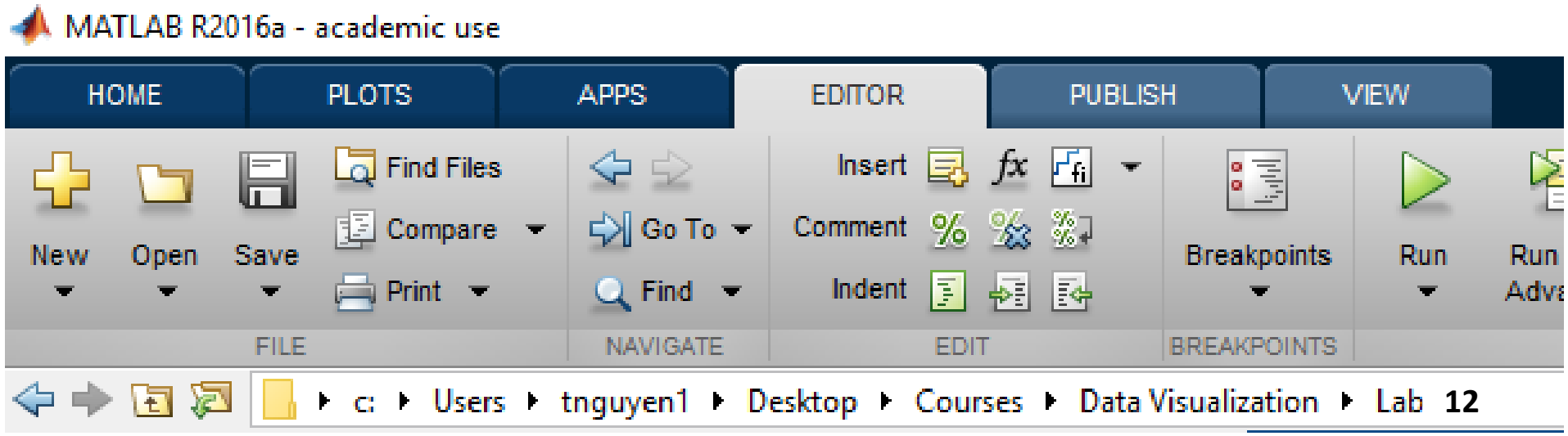
$$X = U_{d,d} D_{d,n} (V_{n,n})^T$$

- *U* and *V* are orthogonal matrices, *D* is a diagonal matrix

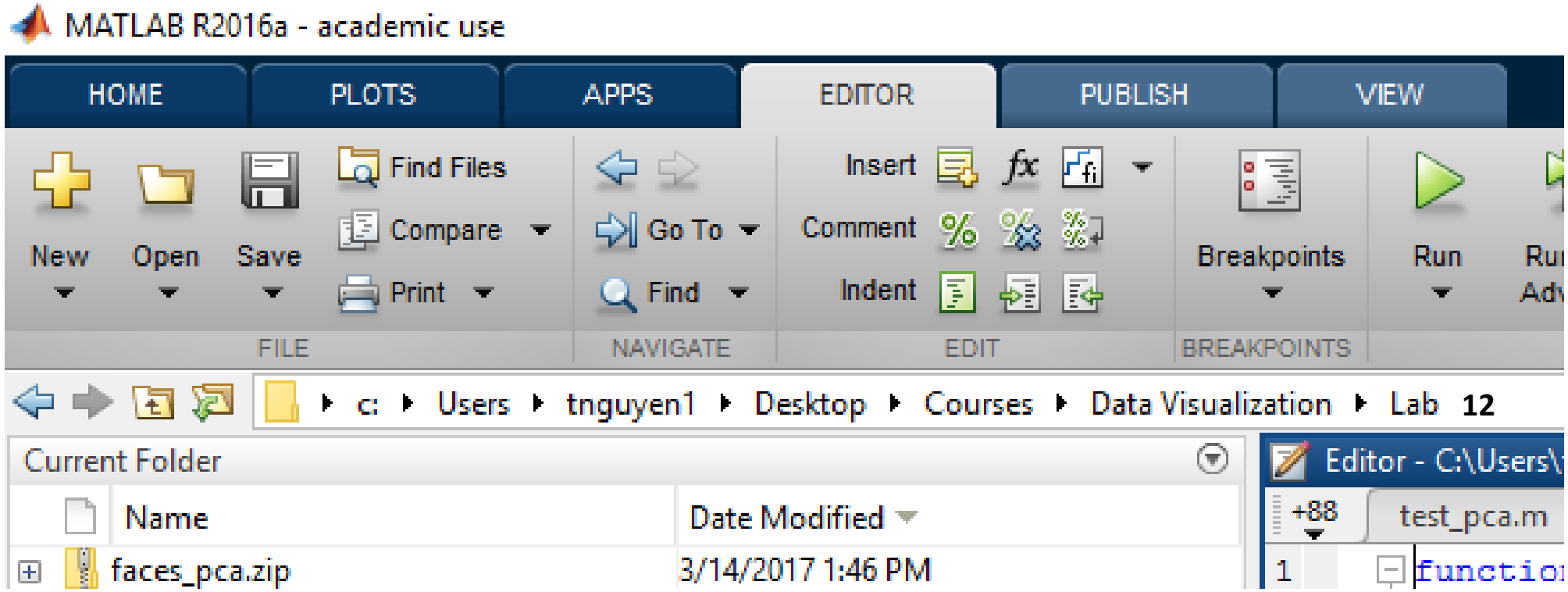
Start MATLAB



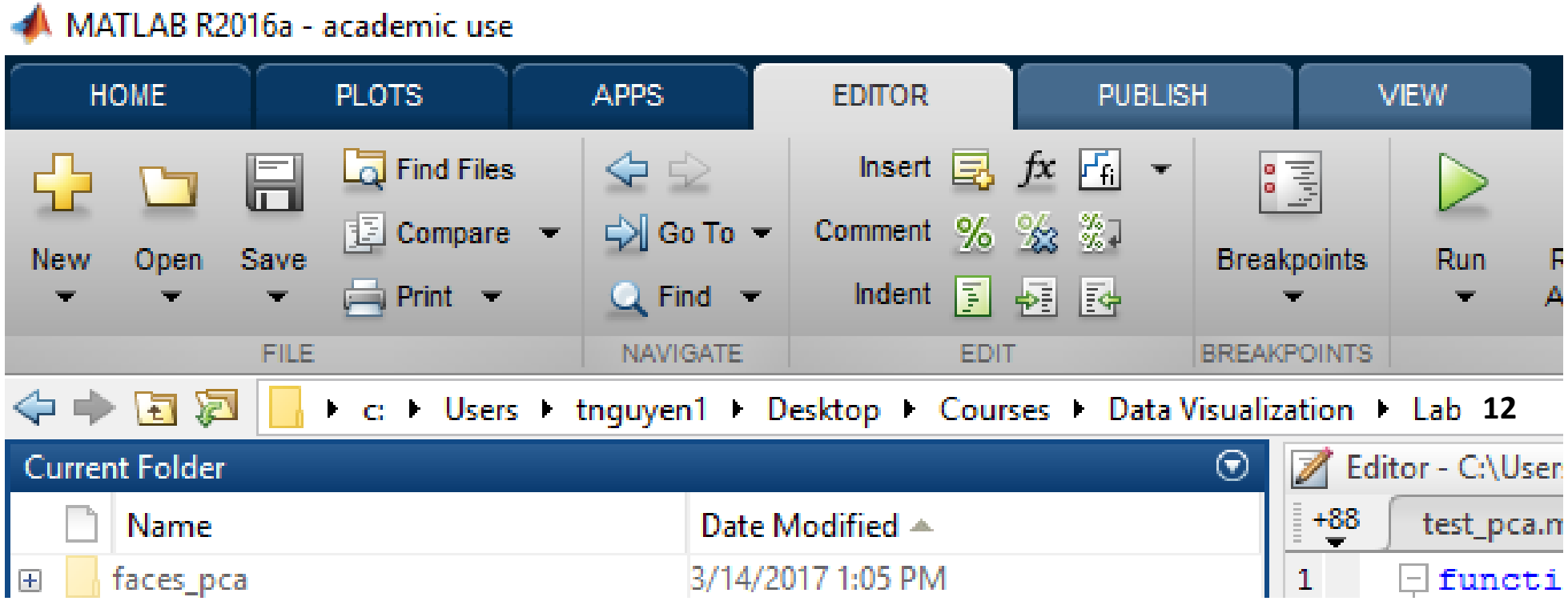
Create Lab 12 folder



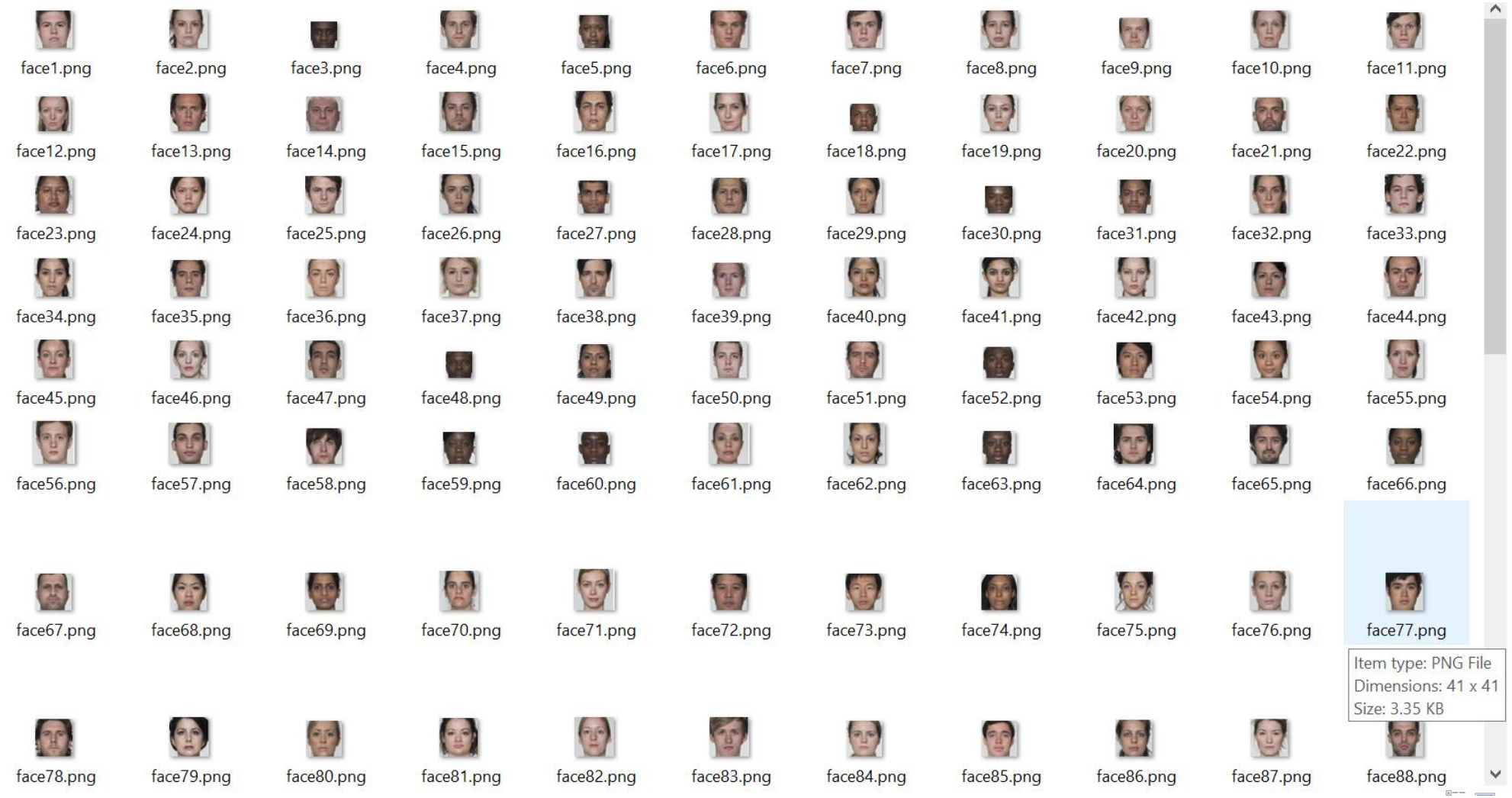
Copy faces_pca.zip from isidore to Lab 12 folder



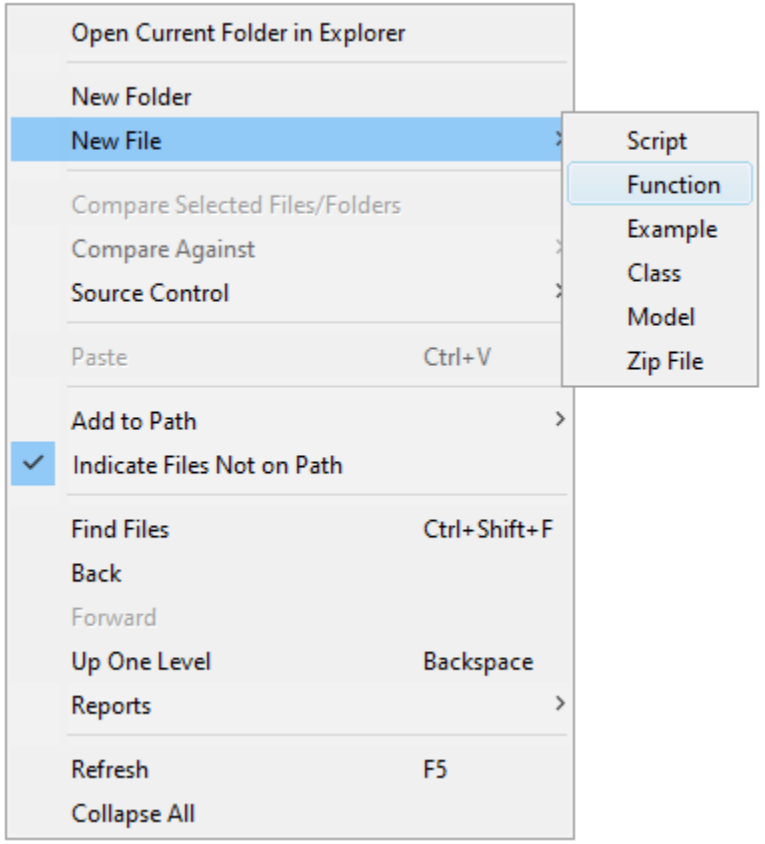
Unzip the .zip file to faces_pca folder



Unzip the .zip file to faces_pca folder



Create new function file: pca.m



pca.m

```
function [ output_args ] = pca( input_args )
%PCA Summary of this function goes here
%   Detailed explanation goes here


end
```

pca.m

```
function [PC,V] = pca(data)
%PCA Summary of this function goes here
%   Detailed explanation goes here


end
```

pca.m

```
function  [PC,V] = pca(data)
% data - MxN matrix of input data
% (M dimensions, N trials)
% PC - each column is a PC
% V - Mx1 matrix of variances
[M,N] = size(data);

end
```

pca.m

```
function  [PC,V] = pca(data)
% data - MxN matrix of input data
[M,N] = size(data);

% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

end
```

pca.m

```
function  [PC,V] = pca(data)
% data - MxN matrix of input data
[M,N] = size(data);

% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% construct the matrix Y
Y = data' / sqrt(N-1);
end
```

pca.m

```
function  [PC,V] = pca(data)
% data - MxN matrix of input data
[M,N] = size(data);

% subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% construct the matrix Y
Y = data' / sqrt(N-1);

% SVD does it all
[u,S,PC] = svd(Y);
end
```

pca.m

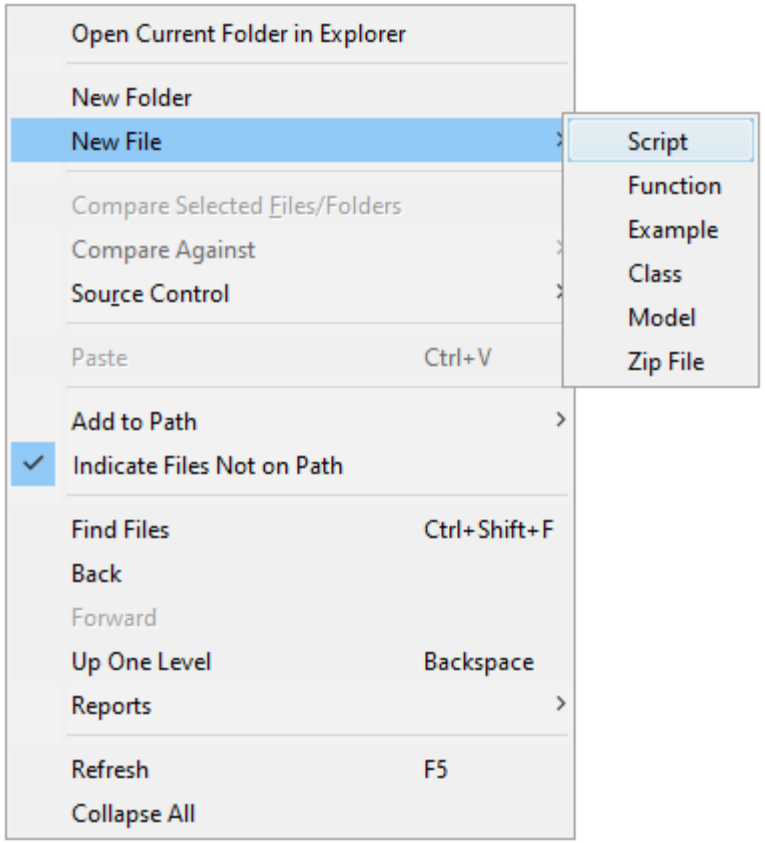
```
function  [PC,V] = pca(data)
[M,N] = size(data);
mn = mean(data,2);

data = data - repmat(mn,1,N);
Y = data' / sqrt(N-1);

% SVD does it all
[u,S,PC] = svd(Y);

% calculate the variances
S = diag(S);
V = S .* S;
end
```

Create new script file: Lab12.m



Lab12.m

```
close all;
clear all;
clc;
```

Lab12.m

```
close all;
clear all;
clc;
```

```
%% prepare data
data = zeros(64 * 64, 130);
for i = 1:130
    im = imread(['./faces_pca/face' num2str(i) '.png']);
    im = rgb2gray(im);
    im = im2double(im);
    im = imresize(im,[64,64]);
    data(:,i) = im(:);
end
```

Lab12.m

```
%% prepare data
data = zeros(64 * 64, 130);
for i = 1:130
    im = imread(['./faces_pca/face' num2str(i) '.png']);
    im = rgb2gray(im);
    im = im2double(im);
    im = imresize(im,[64,64]);
    data(:,i) = im(:);
end
 %% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:4096);
```

Lab12.m

```
%% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:4096);
```

```
%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

Lab12.m

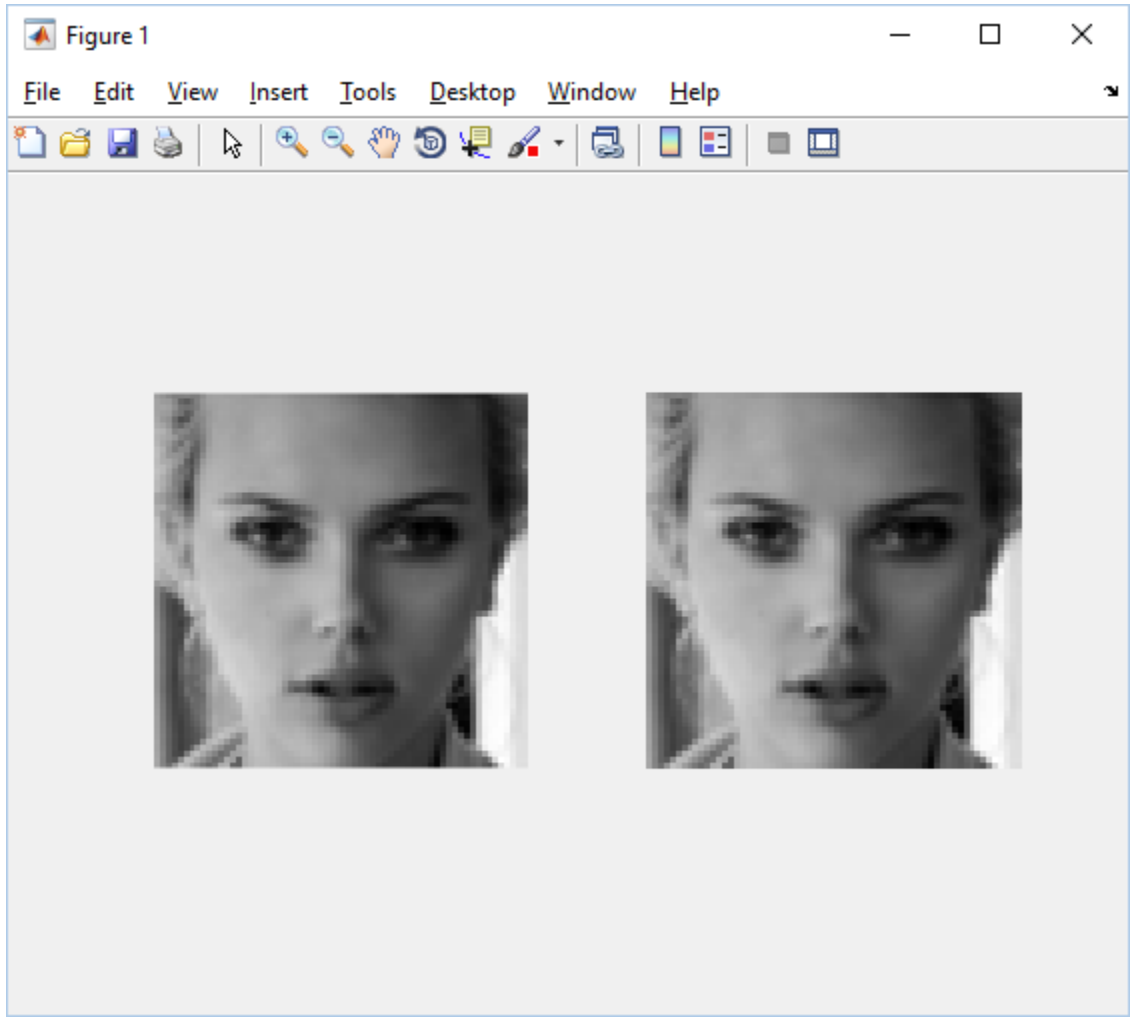
```
%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

```
%% perform PCA and recover the test image
im_pca = im_test*PC;
im_recover = im_pca*PC';
im_recover = reshape(im_recover,[64 64]);
im_test = reshape(im_test,[64 64]);
figure,subplot(1,2,1);imshow(im_test,[]);
```

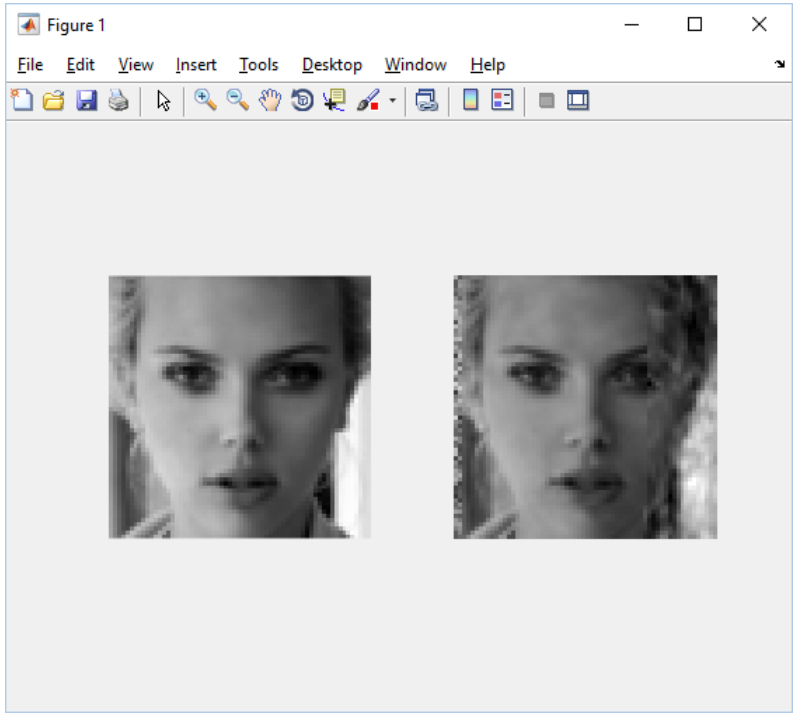
Lab12.m

```
%% perform PCA and recover the test image
im_pca = im_test*PC;
im_recover = im_pca*PC';
im_recover = reshape(im_recover,[64 64]);
im_test = reshape(im_test,[64 64]);
```

```
%% plot before and after images
figure,subplot(1,2,1);imshow(im_test,[]);
subplot(1,2,2);imshow(im_recover,[]);
```



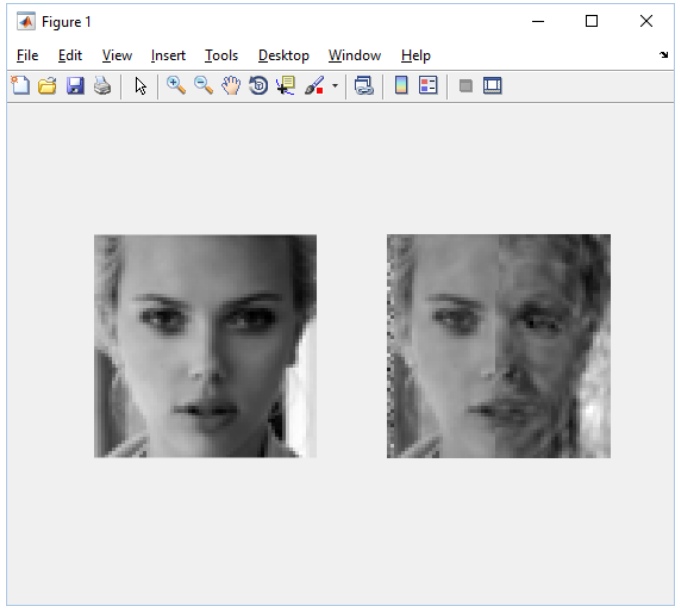
Change the PCs




```
%% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:3000);

%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

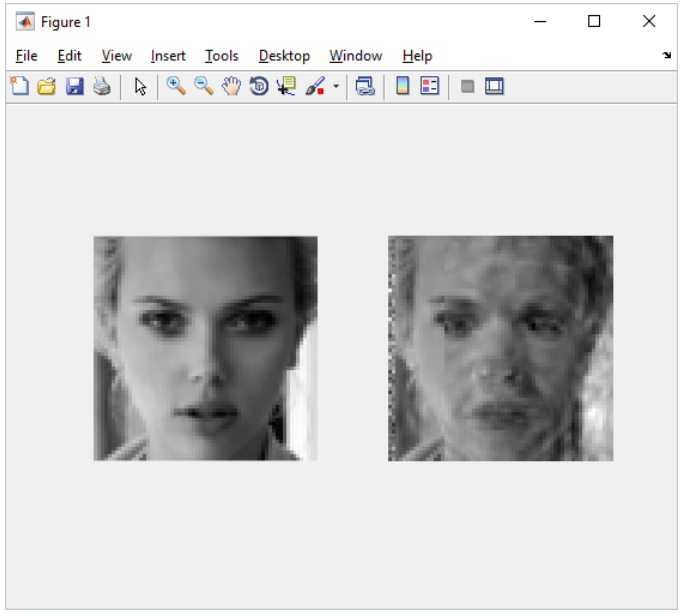
Change the PCs




```
%% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:2000);

%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

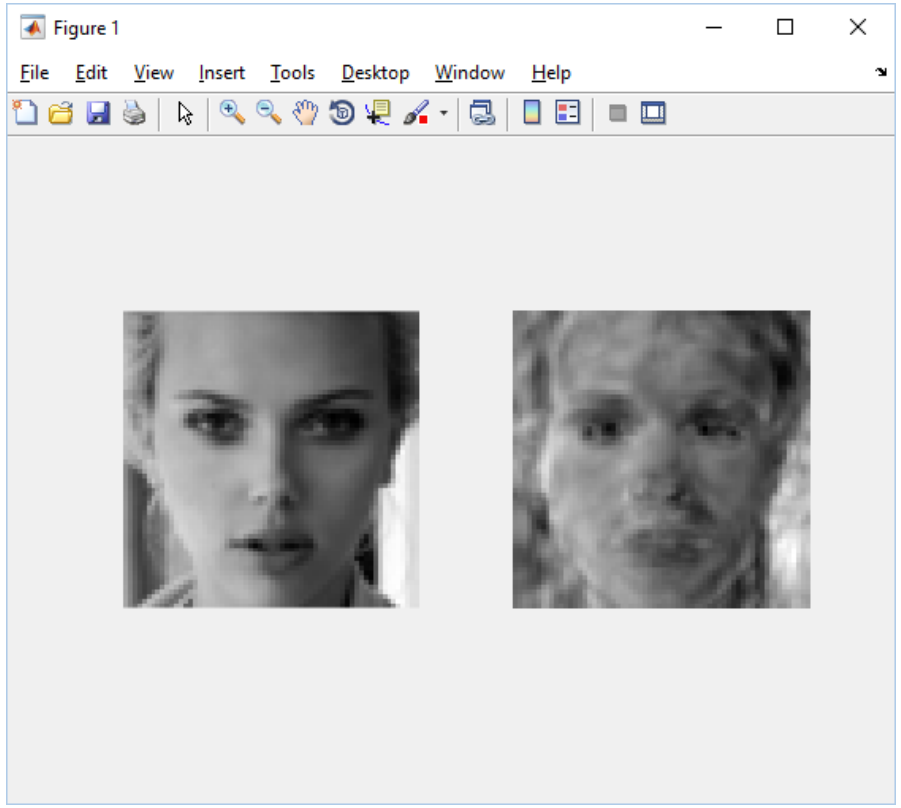
Change the PCs




```
%% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:1000);

%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

Change the PCs




```
%% perform PCA
[PC, V] = pca(data);
PC = PC(:,1:100);

%% prepare test data
im_test = imread(['./faces_pca/face' num2str(131) '.png']);
im_test = rgb2gray(im_test);
im_test = im2double(im_test);
im_test = imresize(im_test,[64,64]);
im_test = im_test(:)';
```

Display eigenfaces

```
%% plot before and after images
subplot(1,2,2);imshow(im_recover,[]);

%% plot eigenfaces
figure,
for i = 1:10
end
```

Display eigenfaces

```
%% plot eigenfaces
figure,
for i = 1:10
    subplot(1,10,i);
    temp = reshape(PC(:,i),[64 64]);
    imshow(temp,[]);
end
```

Q&A