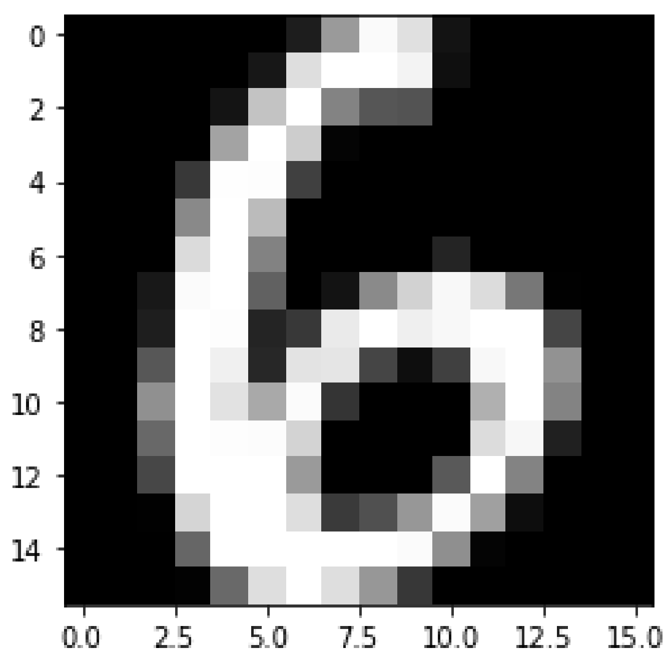
```
import pandas as pd
import io
import requests
import numpy as np
%matplotlib inline

url =
"https://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/zip.digits/
train.6"
s = requests.get(url).content
c = pd.read_csv(io.StringIO(s.decode('utf-8')))
data = np.array(c,dtype='float32');
data.shape
```

```
(663, 256)
```

```
#Part(a)
X=np.transpose(data)
X.shape
```

```
(256, 663)
```

```
import matplotlib.pyplot as plt
img1 = np.reshape(X[:,5],(16,16));
imgshow = plt.imshow(img1,cmap='gray')
```




```
mu = np.mean(X, axis=1);
mu.shape
```

```
(256,)
```

```
#Part(b)
mu=np.reshape(mu,(256,1))
X_centered=X-mu
X_centered.shape
```

```
(256, 663)
```

```
U, s, V = np.linalg.svd(X_centered)
U.shape
```

```
(256, 256)
```

```
s.shape
```

```
(256,)
```

```
s=np.reshape(s,(256,1))
s.shape
```

```
(256, 1)
```

```
#part(c)
k=10
s1=s[0:k,:]
```

```
s1.shape
```

```
(10, 1)
```

```
#part(d)
Sigma1=(1/X_centered.shape[1])*(np.matmul(X_centered,np.transpose(X_ce
ntered)))
Sigma1.shape
```

```
(256, 256)
```

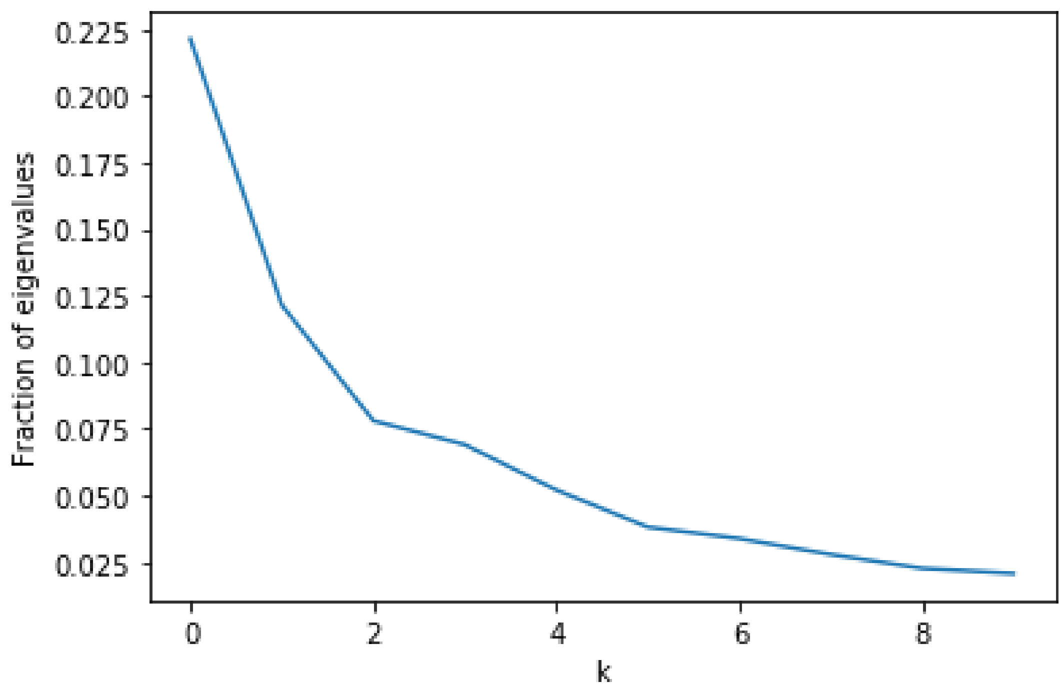
```
ei,_=np.linalg.eig(Sigma1)
ei1=-np.sort(-ei)
ei1.shape
```

```
(256,)
```

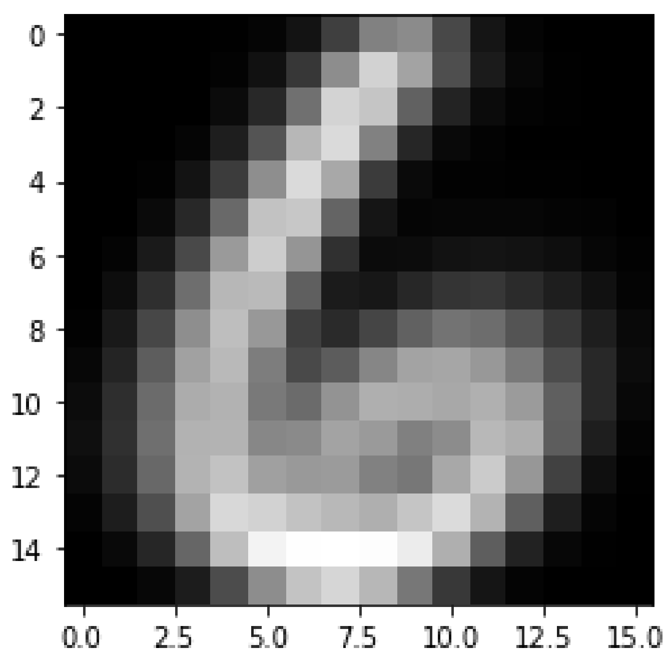
```
ei1=np.reshape(ei1,(256,1))
ei1.shape
```

```
(256, 1)
```

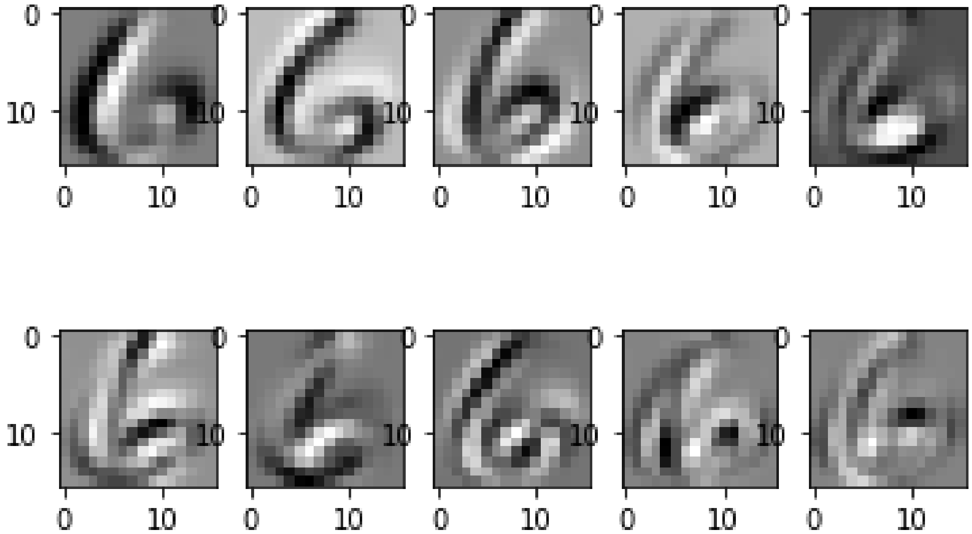
```
#part(d)
plt.plot(ei1[0:k]/np.trace(Sigma1))
plt.ylabel('Fraction of eigenvalues')
plt.xlabel('k')
plt.show()
```




```
#part(e)
img3 = np.reshape(mu,(16,16))
imgshow = plt.imshow(img3,cmap='gray')
```



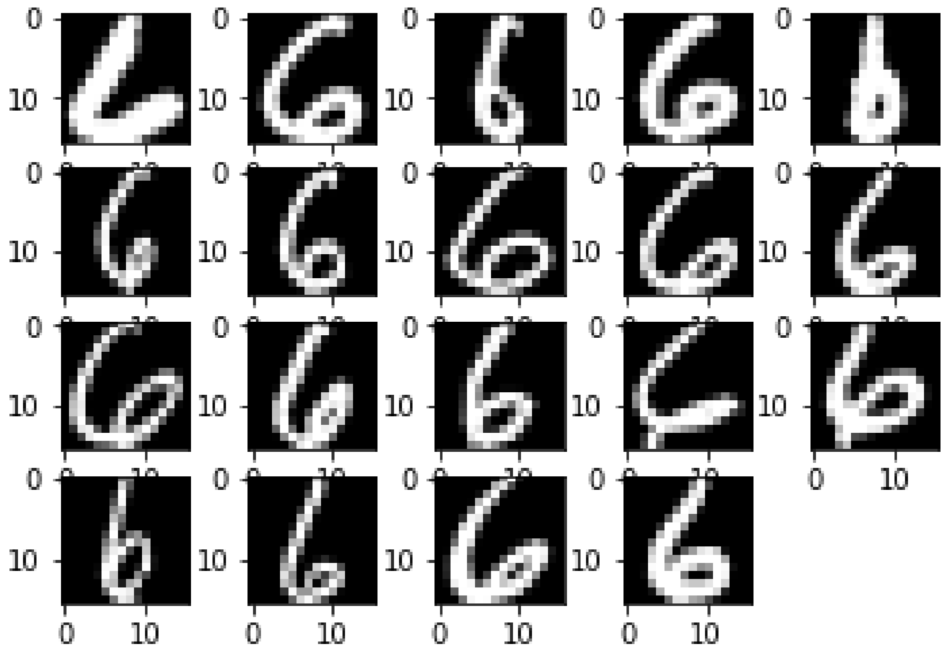

```
for i in range(1, k+1):
    plt.subplot(2, 5, i)
    plt.imshow(np.reshape(U[:,i-1],(16,16)),cmap='gray')
```



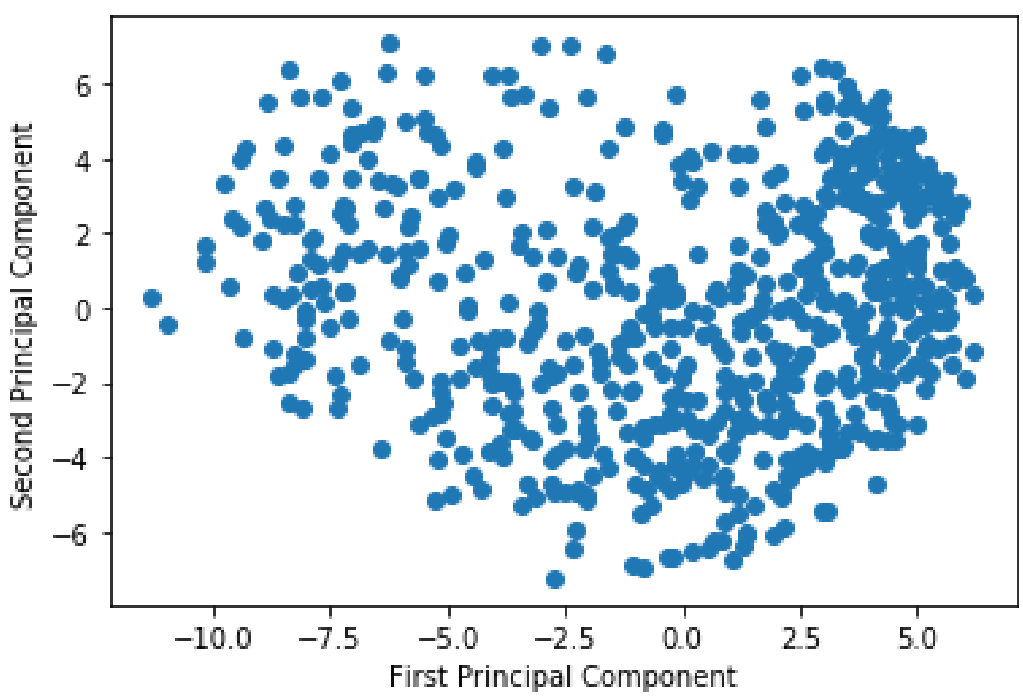
```
#part(f)
V.shape
```

```
(663, 663)
```

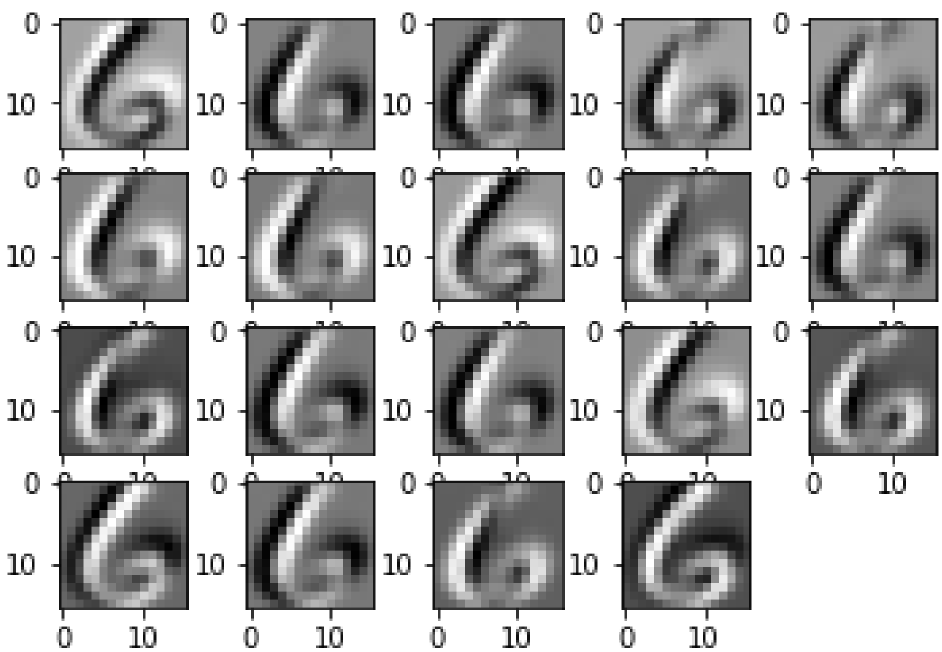
```
V1=np.transpose(V)

idx=np.argsort(V1[0,:])

X_sorted=X[:,idx]

#plotting only the first 20 sorted data points due to space
limitations
for i in range(1, 20):
    plt.subplot(4, 5, i)
    plt.imshow(np.reshape(X_sorted[:,i-1],(16,16)),cmap='gray')
```



```python
#(g) part
D=np.diag(np.reshape(s,[s.shape[0]]))
A=np.matmul(D,V[0:256,:])
plt.scatter(A[0,:],A[1,:])
plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
plt.show()
```



```
X_new=np.matmul(U[:,0:2],A[0:2,:])

#plotting only the first 20 sorted data points due to space
limitations
for i in range(1, 20):
    plt.subplot(4, 5, i)
    plt.imshow(np.reshape(X_new[:,i-1],(16,16)),cmap='gray')
```



```
%     Delhi Kolkata     Chennai     Mumbai      Bhopal      Bengaluru   Hyderabad
      Agra
% Delhi     0     1305.6      1758.9      1157.3      596.9 1742  1256.5      179.2
% Kolkata   1305.6      0     1360.6      1657.6      1125.4      1559.8      1181.1
      1163.6
% Chennai   1758.9      1360.6      0     1027.7      1172.9      286.8 516.5 1586.7
% Mumbai    1157.3      1657.6      1027.7      0     667.1 840.1 619.6 1049.9
% Bhopal    596.9 1125.4      1172.9      667.1 0     1145.1      662.8 439.8
% Bengaluru 1742  1559.8      286.8 840.1 1145.1      0     500.9 1581.1
% Hyderabad 1256.5      1181.1      516.5 619.6 662.8 500.9 0     1089.8
% Agra      179.2 1163.6      1586.7      1049.9      439.8 1581.1      1089.8
      0
clear; clc;
f1=load('cities.mat');
D=f1.D.^2;   %Squared Distance Matrix
n=length(D);
k=3; %Euclidean space of dimensionm 2
city
={'Delhi','Kolkata','Chennai','Mumbai','Bhopal','Bengaluru','Hyderabad','Agra'};

H=eye(n)-(1/n)*ones(n,1)*ones(1,n);

B=(-1/2)*H*D*H';

[U,Lambdam]=eig(B);

lamdas=diag(Lambdam);
[lamdas_sorted, idx]=sort(lamdas,'descend');

U1=U(:,idx);
U2=U1(:,1:k);
Xk=U2*sqrtm(diag(lamdas_sorted(1:k)));
c = linspace(1,20,n);
sz=40;

figure(1);
plot(lamdas_sorted/sum(lamdas));
title('Normalized Eigenvalues of B')

figure(2);
scatter3(Xk(:,1),Xk(:,2),Xk(:,3),sz,c,'filled');
text(Xk(:,1)+25,Xk(:,2),Xk(:,3),city)
% xlabel('Km')
% ylabel('Km')
```

Q) 3) a) $K \succeq 0$ iff $\lambda_i \geq 0$

$K v_i = \lambda_i v_i$

multiply both sides by $v_i^T$

$v_i^T K v_i = \lambda_i \|v_i\|^2$

Since $\|v_i\|^2 > 0$

for $v_i^T K v_i \geq 0 \Rightarrow \lambda_i \geq 0 \ \forall i$

$\Rightarrow$ All the eigen values of $K$ should be non-negative for $K$ to be p.s.d

b) $d_{ij} = K_{ii} + K_{jj} - 2K_{ij}$

To show: $\exists u_i, u_j \in \mathbb{R}^n$ $K$ is p-d

s.t $d_{ij} = \|u_i - u_j\|^2$

Since $d_{ij} = d_{ji} \Rightarrow D = [d_{ij}]_{i,j=1}^n$

is a real symmetric matrix

Sim $d_{ii} = 0 \rightarrow$ Diagonal elements are zeros.

To show, $D$ is c.n.d matrix

$\sum_{i=1}^n v_i = 0$, $v^T D v \leq 0$

$R = \text{diag}(K)$

$D = R \cdot 1^T + 1 R^T - 2K$

$v^T D v = v^T R \cdot 1^T v + v^T \cdot 1 \cdot K^T v - 2 v^T K v$

if $v^T 1 = 1^T v = 0$

then $v^T D v = -2 v^T K v \leq 0$

because $K$ is p.s.d

$\Rightarrow D$ is c.n.d

$\Rightarrow$ D is a squared distance matrix using classical MDS theory.

consider $B_\alpha = -\frac{1}{2} H_\alpha D H_\alpha^T$ D is c.n.d.

$H_\alpha = \cancel{I - 1} \quad I - 1 \cdot \tau$

$x^T B_\alpha x = -\frac{1}{2} x^T H_\alpha D H_\alpha^T x$

$= -\frac{1}{2} (H_\alpha^T x)^T D (H_\alpha^T x)$

let $y = H_\alpha^T x$

$= -\frac{1}{2} y^T D y^T$

claim: $1^T y = 0$

$1^T y = 1^T H_\alpha^T x = 1^T (I - \alpha 1^T) x$

$= (1 - 1^T \alpha) 1^T x = 0$

since $1^T \alpha = 0 \rightarrow$ for signed probability

$\Rightarrow x^T B_\alpha x = -\frac{1}{2} y^T D y^T \geq 0$

because D is c.n.d

$\Rightarrow y^T D y^T \leq 0$

$\Rightarrow B_\alpha$ is p.s.d

d) Given $A \geq 0$ & $B \geq 0$

To show : $A + B \geq 0$

(i) Consider $x^T(A+B)x$

$$\Rightarrow x^T A x + x^T B x$$

$$\geq 0 \qquad \geq 0$$

$$\Rightarrow x^T(A+B)x \geq 0 \quad \forall x$$

$\Rightarrow A + B$ is p.s.d.

(ii) Consider : $A \circ B = [A_{ij} B_{ij}]_{ij}$

$x^T(A \circ B)x$ $\qquad \sum_{ij} x_i A_{ij} x_j \geq 0$

$\sum_{ij} x_i A_{ij} B_{ij} x_j$ $\qquad \sum_{ij} x_i B_{ij} x_j \geq 0$

using EVD of $A$ & $B$

$A = \sum_i \lambda_i u_i u_i^T$ , $B = \sum_i \mu_i v_i v_i^T$

$$A \circ B = \sum_{ij} \lambda_i \mu_j (u_i u_i)^T \circ (v_j v_j)^T$$

$$= \sum_{ij} \lambda_i \mu_j (u_i \circ v_j)(u_i \circ v_j)^T$$

Each $(u_i \circ v_j)(v_i \circ v_j)^T$ is positive semi definite

& $\lambda_i \mu_j > 0$

$\Rightarrow A \circ B$ is also p.s.d