In [1]:
```python
# Writing HW Q3
import numpy as np
import numpy.linalg as la
```

In [2]:
```python
M = np.array([[1,2,3],[3,4,5],[5,4,3],[0,2,4],[1,3,5]])
M
```

Out[2]:
```
array([[1, 2, 3],
       [3, 4, 5],
       [5, 4, 3],
       [0, 2, 4],
       [1, 3, 5]])
```

In [3]:
```python
MT = M.transpose()
MT
```

Out[3]:
```
array([[1, 3, 5, 0, 1],
       [2, 4, 4, 2, 3],
       [3, 5, 3, 4, 5]])
```

In [4]:
```python
#part a) MT  M
MTM = MT.dot(M)
MTM
```

Out[4]:
```
array([[36, 37, 38],
       [37, 49, 61],
       [38, 61, 84]])
```

In [5]:
```python
# M MT
MMT = M.dot(MT)
MMT
```

Out[5]:
```
array([[14, 26, 22, 16, 22],
       [26, 50, 46, 28, 40],
       [22, 46, 50, 20, 32],
       [16, 28, 20, 20, 26],
       [22, 40, 32, 26, 35]])
```

In [6]:
```python
# part B
# eigenvector and eigenvalue of MTM
egv_MTM, egvt_MTM = la.eig(MTM)
print (egv_MTM[0].real)
print (egv_MTM[1].real)
MTM_V1 = -1*egvt_MTM[:,0].real
print (MTM_V1)
MTM_V2 = egvt_MTM[:,1].real
print (MTM_V2)
```

```
153.56699646
15.43300354
[ 0.40928285  0.56345932  0.7176358 ]
[-0.81597848 -0.12588456  0.56420935]
```

In [7]:
```
# egienvector and eigenvalue of MMT, only get index 0 and 2
# part C
egv_MMT, egvt_MMT = la.eig(MMT)
print (egv_MMT[0].real)
print (egv_MMT[2].real)
MMT_V1 =  egvt_MMT[:,0].real
print (MMT_V1)
MMT_V2 =  egvt_MMT[:,2].real
print (MMT_V2)
```

```
153.56699646
15.43300354
[ 0.29769568  0.57050856  0.52074297  0.32257847  0.45898491]
[-0.15906393  0.0332003   0.73585663 -0.5103921  -0.41425998]
```

In [8]:
```
# part D
# The left-singular vectors of M are a set of orthonormal eigenvectors of MI

U = np.vstack((egvt_MMT[:,0].real, egvt_MMT[:,2].real)).T
print (U)
```

```
[[ 0.29769568 -0.15906393]
 [ 0.57050856  0.0332003 ]
 [ 0.52074297  0.73585663]
 [ 0.32257847 -0.5103921 ]
 [ 0.45898491 -0.41425998]]
```

In [9]:
```
# using part b and part c to calculate U, sigma, Vt
# sigma = the square roots of the eigenvalues for MTM
sigma = np.zeros(shape = (2,2))
import math
for i in range(2):
    sigma[i][i] = math.sqrt(egv_MTM[i])
print (sigma)
```

```
[[ 12.39221516   0.        ]
 [  0.           3.92848616]]
```

In [10]:
```
# V is eigenvectors of MMT
# http://stackoverflow.com/questions/17710672/create-2-dimensional-array-wit
V = np.vstack((-1*egvt_MTM[:,0].real, egvt_MTM[:,1].real)).T
print (V)
```

```
[[ 0.40928285 -0.81597848]
 [ 0.56345932 -0.12588456]
 [ 0.7176358   0.56420935]]
```

In [11]:
```
# change smaller singular value to zero
new_sigma = sigma.copy()
new_sigma[1][1] = 0
print (sigma)
```

```
[[ 12.39221516   0.        ]
 [  0.           3.92848616]]
```

In [12]:

```
# part e
new_M = (U.dot(new_sigma)).dot(V.T)
print (new_M)
```

```
[[ 1.509889    2.0786628   2.64743661]
 [ 2.89357443  3.98358126  5.0735881 ]
 [ 2.64116728  3.63609257  4.63101787]
 [ 1.63609257  2.25240715  2.86872172]
 [ 2.32793529  3.20486638  4.08179747]]
```

In [13]:

```
# part f
sum_energy = 0
for i in range(len(sigma)):
    sum_energy += sigma[i][i] * sigma[i][i]
energy = new_sigma[0][0] * new_sigma[0][0] / sum_energy
print (energy)
```

```
0.908680452426
```

In [ ]:

In [ ]: