import numpy as np

from numpy import linalg as la

import matplotlib

from matplotlib import pyplot as plt

b=np.zeros((10,5))

def pca\_imp(a):

print ("\n\n")

a1= np.mean(a, axis = 0)

print ("the mean of each row \n ")

print (a1)

print ("\n Matrix 2 \n")

for j in range(0,5):

for i in range(0,10):

b[i,j]=a[i,j]-a1.item(j)

print (b)

print ("\n The transpose \n")

d= b.transpose()

print (d)

#b[0,0]=a.item(0)-xdash

#b[0,1]=a.item(1)-ydash

#b[1,0]=a.item(2)-xdash

#b[1,1]=a.item(3)-ydash

#print b

print ("\n covariance Matrix\n")

c = np.cov(d)

print (c)

eig\_vals, eig\_vecs = la.eig(c)

print ("The eigen value \n")

print (eig\_vals)

print ("The eigen vector \n")

print (eig\_vecs)

eig\_pairs = [(np.abs(eig\_vals[i]), eig\_vecs[:,i]) for i in range(len(eig\_vals))]

# Sort the (eigenvalue, eigenvector) tuples from high to low

eig\_pairs.sort(key=lambda x: x[0])

eig\_pairs.reverse()

# Visually confirm that the list is correctly sorted by decreasing eigenvalues

ind = list(reversed(eig\_vals.argsort()))

top\_eig\_vecs = eig\_vecs[:,ind]

print ('Eigenvalues in descending order:')

for i in eig\_pairs:

print (i[0])

red\_data = np.mat(b) \* np.mat(top\_eig\_vecs)

print (red\_data)

return red\_data

def plot(red\_data):

plt.plot(red\_data,b,'ro')

plt.show()