```
import numpy as np
from numpy import linalg as la
import matplotlib
from matplotlib import pyplot as plt

b=np.zeros((35,5))
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(0,35):
			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()
```