```
In [1]: import numpy as np
```

```
In [2]: from numpy import linalg as la
```

```
In [3]: r=64.
        x=np.array([2.5,0.5,2.2,1.9,3.1,2.3,2.0,1.0,1.5,1.2])
        y=np.array([2.4,0.7,2.9,2.2,3.0,2.7,1.6,1.1,1.6,0.9])
        data = np.array([x, y])
        print(data)
        xMean = np.mean(x)
        yMean = np.mean(y)
        print(xMean)
        print(yMean)
```

```
[[2.5 0.5 2.2 1.9 3.1 2.3 2.  1.  1.5 1.2]
 [2.4 0.7 2.9 2.2 3.  2.7 1.6 1.1 1.6 0.9]]
1.8199999999999998
1.9100000000000001
```

```
In [4]: meanAdjusted = np.zeros((2, 10))
        for i in range(len(data[0])):
            meanAdjusted[0][i] = data[0][i] - xMean
        for i in range(len(data[1])):
            meanAdjusted[1][i] = data[1][i] - yMean
        print(meanAdjusted)
```

```
[[ 0.68 -1.32  0.38  0.08  1.28  0.48  0.18 -0.82 -0.32 -0.62]
 [ 0.49 -1.21  0.99  0.29  1.09  0.79 -0.31 -0.81 -0.31 -1.01]]
```

```
In [5]: cov_mat = np.cov(data)
        print(cov_mat)
        eig_vals, eig_vecs = np.linalg.eig(cov_mat)
        print('Eigenvectors \n%s' %eig_vecs)
        print('\nEigenvalues \n%s' %eig_vals)
```

```
[[0.60177778 0.60422222]
 [0.60422222 0.71655556]]
Eigenvectors
[[-0.7397818  -0.67284685]
 [ 0.67284685 -0.7397818 ]]

Eigenvalues
[0.05222517 1.26610816]
```

```
In [6]: eig_pairs = [(np.abs(eig_vals[i]), eig_vecs[:,i]) for i in range(len(eig_vals
        ))]
```

```
In [7]: eig_pairs.sort()
        eig_pairs.reverse()
```

```
In [8]:  print('Eigenvalues in descending order:')
         for i in eig_pairs:
             print(i[0])

         Eigenvalues in descending order:
         1.2661081617014531
         0.05222517163188023
```

```
In [9]:  print('Eigenvectors in descending order:')
         for i in eig_pairs:
             print(i[1])

         Eigenvectors in descending order:
         [-0.67284685 -0.7397818 ]
         [-0.7397818   0.67284685]
```

```
In [10]: eig_pairs[0][1]
```

```
Out[10]: array([-0.67284685, -0.7397818 ])
```

```
In [11]: transformedData1 = np.matmul(meanAdjusted.T, eig_pairs[0][1])
         transformedData2 = np.matmul(meanAdjusted.T, eig_pairs[1][1])
         transformedData =[transformedData1,transformedData2]
         transformedData =np.transpose(transformedData)
         print(transformedData)

         [[-0.82002894 -0.17335667]
          [ 1.78329382  0.16236729]
          [-0.98806579  0.3850013 ]
          [-0.26836447  0.13594304]
          [-1.66760613 -0.21351764]
          [-0.90739411  0.17645374]
          [ 0.10821993 -0.34174325]
          [ 1.15095768  0.06161513]
          [ 0.44464335  0.02814765]
          [ 1.16434467 -0.2209106 ]]
```

```
In [ ]:
```