```
from sklearn.decomposition import PCA
import numpy as np

import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
style.use("fivethirtyeight")


# target dimension(s)
k = 1
# Create a new PCA instance
pca = PCA(k)
```

dataset: [2.5, 2.4], [0.5, 0.7],[2.2, 2.9],[1.9, 2.2],[3.1, 3.0], [2.3, 2.7],[2, 1.6],[1.0, 1.1],[1.5,1.6], [1.1,0.9]

```
# 4x2 data matrix
data = np.array([[2.5, 2.4], [0.5, 0.7], [2.2, 2.9], [1.9, 2.2], [3.1, 3.0], [
print("Data: \n", data)
```

```
Data:
 [[2.5 2.4]
 [0.5 0.7]
 [2.2 2.9]
 [1.9 2.2]
 [1.  1.1]
 [1.5 1.6]
 [1.1 0.9]]
```

Saved successfully!

```
print("Reduced: \n", pca.fit_transform(data))
```

```
Reduced:
 [[-0.82797019]
 [ 1.77758033]
 [-0.99219749]
 [-0.27421042]
 [-1.67580142]
 [-0.9129491 ]
```

✓ 0s completed at 3:23 PM

```
M = mean(data.T, axis=1)
print(M)
# center columns by subtracting column means
C = data - M
print(C)
# calculate covariance matrix of centered matrix
```

```
[1.81 1.91]
[[ 0.69  0.49]
 [-1.31 -1.21]
 [ 0.39  0.99]
 [ 0.09  0.29]
 [ 1.29  1.09]
 [ 0.49  0.79]
 [ 0.19 -0.31]
 [-0.81 -0.81]
 [-0.31 -0.31]
 [-0.71 -1.01]]
```

Get covariance matrix

```
from numpy import cov
from numpy.linalg import eig
V = cov(C.T)
print(V)
```

```
[[0.61655556 0.61544444]
 [0.61544444 0.71655556]]
```

Saved successfully!

eigendecomposition of covariance matrix

```
 [ 0.01776463  0.43804614]
 [-0.16267529  1.22382056]]
```

```
print('\nLargest eigen value column index')
col = np.argmax(np.max(P, axis=1))
print(col)
print('\nfinal 1-D Space Data:')
print(P.T[:, col])
```

```
Largest eigen value column index
1

final 1-D Space Data:
[-0.82797019  1.77758033 -0.99219749 -0.27421042 -1.67580142 -0.912949
  0.09910944  1.14457216  0.43804614  1.22382056]
```

Saved successfully!

Saved successfully!