```
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

```
# Generate some sample data (e.g., 100 samples, 10 features)
X = np.random.rand(100, 10)
```

```
# Initialize PCA with the desired number of components (e.g., 2 for visualization)
n_components = 2
pca = PCA(n_components=n_components)
# or pca = PCA(n_components=2)
```

```
# Fit PCA to the data and transform it
X_pca = pca.fit_transform(X)
```

```
print(X_pca)
```

```
[[ 0.52587834 -0.162335  ]
 [-0.1579774  -0.26269173]
 [ 0.25723409 -0.31206364]
 [-0.3893992  -0.07742533]
 [-0.26688674 -0.14809978]
 [-0.27788297  0.12088245]
 [-0.31865065 -0.19356261]
 [-0.32027    -0.49093855]
 [-0.11403983  0.26621222]
 [ 0.20958256  0.36039379]
 [-0.72220452 -0.31409899]
 [-0.24280625  0.55318537]
 [ 0.65268074  0.27926658]
 [ 0.02779116  0.02881118]
 [-0.14784294  0.11161389]
 [ 0.18932167 -0.39928842]
 [-0.21451674 -0.24193304]
 [ 0.19418146  0.00685986]
 [ 0.88740949 -0.16261538]
 [-0.25326629  0.22620139]
 [ 0.58966072  0.17856013]
 [-0.47118856 -0.33891677]
 [ 0.0629459   0.27821911]
 [-0.30817213  0.31963077]
 [ 0.09174778 -0.26779476]
 [-0.68210827  0.41230736]
 [ 0.06893602 -0.58863926]
 [-0.2234702   0.25386328]
 [ 0.27284117 -0.22915435]
 [ 0.26707641  0.26482312]
 [-0.02997546 -0.85496334]
 [-0.49378703  0.20688771]
 [-0.21371881  0.15988706]
 [-0.04682025 -0.39740594]
 [-0.31441305  0.0245623 ]
 [ 0.31718075 -0.45007786]
 [ 0.14752798  0.03220472]
 [ 0.01654108  0.10671293]
 [-0.09490503 -0.37109572]
 [-0.38045407  0.04402786]
 [ 0.06819046 -0.02385919]
 [ 0.39549458 -0.26586791]
 [ 0.27419474 -0.15804472]
 [ 0.46020281  0.0346244 ]
 [-0.1189558  -0.35104101]
 [-0.06382188  0.38582889]
 [ 0.48448898  0.06124753]
 [ 0.11510512  0.0244994 ]
 [-0.52469575  0.46890131]
 [ 0.15259544  0.04912924]
 [-0.12029033 -0.27587444]
 [ 0.25484605 -0.41619215]
 [ 0.48634504  0.39094517]
 [-0.18583428 -0.13798935]
 [-0.26663657 -0.32586264]
 [-0.49285374 -0.05967515]
 [ 0.29624128  0.44904586]
 [-0.28942239  0.09399483]
```

```
pca_one_component = PCA(n_components=1)
X_pca_one = pca_one_component.fit_transform(X)
```

```
print("\nTransformed data with 1 component:")
print(X_pca_one)
```

```
Transformed data with 1 component:
[[ 0.52587834]
 [-0.1579774 ]
 [ 0.25723409]
 [-0.3893992 ]
 [-0.26688674]
 [-0.27788297]
 [-0.31865065]
 [-0.32027   ]
 [-0.11403983]
 [ 0.20958256]
 [-0.72220452]
 [-0.24280625]
 [ 0.65268074]
 [ 0.02779116]
 [-0.14784294]
 [ 0.18932167]
 [-0.21451674]
 [ 0.19418146]
 [ 0.88740949]
 [-0.25326629]
 [ 0.58966072]
 [-0.47118856]
 [ 0.0629459 ]
 [-0.30817213]
 [ 0.09174778]
 [-0.68210827]
 [ 0.06893602]
 [-0.2234702 ]
 [ 0.27284117]
 [ 0.26707641]
 [-0.02997546]
 [-0.49378703]
 [-0.21371881]
 [-0.04682025]
 [-0.31441305]
 [ 0.31718075]
 [ 0.14752798]
 [ 0.01654108]
 [-0.09490503]
 [-0.38045407]
 [ 0.06819046]
 [ 0.39549458]
 [ 0.27419474]
 [ 0.46020281]
 [-0.1189558 ]
 [-0.06382188]
 [ 0.48448898]
 [ 0.11510512]
 [-0.52469575]
 [ 0.15259544]
 [-0.12029033]
 [ 0.25484605]
 [ 0.48634504]
 [-0.18583428]
 [-0.26663657]
 [-0.49285374]
```

```
# Print the shape of the original and transformed data
print("Original data shape:", X.shape)
print("Transformed data shape:", X_pca.shape)
```

```
Original data shape: (100, 10)
Transformed data shape: (100, 2)
```