import numpy as np

import matplotlib.pyplot as plt

class PCA:

def \_\_init\_\_(self, n\_components=2):

self.n\_components = n\_components

def fit(self, X):

self.mean = np.mean(X, axis=0)

X\_centered = X - self.mean

covariance\_matrix = np.cov(X\_centered.T)

eigenvalues, eigenvectors = np.linalg.eig(covariance\_matrix)

sorted\_indices = np.argsort(eigenvalues)[::-1]

self.eigenvectors = eigenvectors[:, sorted\_indices]

self.eigenvalues = eigenvalues[sorted\_indices]

self.components = self.eigenvectors[:, :self.n\_components]

def transform(self, X):

X\_centered = X - self.mean

return np.dot(X\_centered, self.components)

def fit\_transform(self, X):

self.fit(X)

return self.transform(X)

np.random.seed(0)

X = np.random.randn(100, 3)

pca = PCA(n\_components=2)

X\_reduced = pca.fit\_transform(X)

plt.scatter(X\_reduced[:, 0], X\_reduced[:, 1])

plt.title('PCA Result (2D Projection of 3D data)')

plt.xlabel('First Principal Component')

plt.ylabel('Second Principal Component')

plt.show()

print("Original data shape:", X.shape)

print("Reduced data shape:", X\_reduced.shape)