

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import fractional_matrix_power

#####
# Generate Data #
#####

num_points = 100

sigH = 3.5
sigNx = 2
sigNy = 3

Hx = np.sort(np.random.randn(num_points,1)) * sigH
Hy = Hx

Nx = np.random.randn(num_points,1) * sigNx
Ny = np.random.randn(num_points,1) * sigNy

t = np.asarray([[1,0.5],[0.5,1]])
X = np.hstack((Hx,Nx)) @ t
t = np.asarray([[1,-0.5],[-0.5,1]])
Y = np.hstack((Hy,Ny)) @ t

c_xx = X.T@X
c_yy = Y.T@Y
c_xy = X.T@Y
X_w = X@(fractional_matrix_power(c_xx, -0.5))
Y_w = Y@(fractional_matrix_power(c_yy, -0.5))
u_t, s, v_t_t = np.linalg.svd(fractional_matrix_power(c_xx,
    -0.5)@c_xy@fractional_matrix_power(c_yy, -0.5))
v_t = v_t_t.T
u = fractional_matrix_power(c_xx, -0.5)@u_t
v = fractional_matrix_power(c_yy, -0.5)@v_t
u_t1 = u_t[:,0]
u_t2 = u_t[:,1]
v_t1 = v_t[:,0]
v_t2 = v_t[:,1]
u1 = u[:,0]
u2 = u[:,1]
v1 = v[:,0]
v2 = v[:,1]
#Original X
plt.figure(figsize=(6,6))
plt.scatter(X[:,0], X[:,1], c=(X@(u1.T)).T)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.grid()
plt.show()
#Original Y
plt.figure(figsize=(6,6))
plt.scatter(Y[:,0], Y[:,1], c=(Y@(v1.T)).T)
plt.xlim(-10, 10)
plt.ylim(-10, 10)

```

```

plt.grid()
plt.show()
#Whitened X
plt.figure(figsize=(6,6))
plt.scatter(X_w[:,0], X_w[:,1], c=(X_w@(u_t1.T)).T)
plt.xlim(-0.3, 0.3)
plt.ylim(-0.3, 0.3)
plt.grid()
plt.show()
#Whitened Y
plt.figure(figsize=(6,6))
plt.scatter(Y_w[:,0], Y_w[:,1], c=(Y_w@(v_t1.T)).T)
plt.xlim(-0.3, 0.3)
plt.ylim(-0.3, 0.3)
plt.grid()
plt.show()
#CCA on whitened X
plt.figure(figsize=(6,6))
plt.scatter(X_w[:,0], X_w[:,1], c=(X_w@(u_t1.T)).T)
plt.xlim(-0.3, 0.3)
plt.ylim(-0.3, 0.3)
plt.grid()
plt.arrow(-u_t1[0]*0.2,-u_t1[1]*0.2,u_t1[0]*0.4,u_t1[1]*0.4, width=0.005,
          color='g')
plt.arrow(-u_t2[0]*0.2,-u_t2[1]*0.2,u_t2[0]*0.4,u_t2[1]*0.4, width=0.001,
          color='g')
plt.show()
#CCA on whitened Y
plt.figure(figsize=(6,6))
plt.scatter(Y_w[:,0], Y_w[:,1], c=(Y_w@(v_t1.T)).T)
plt.xlim(-0.3, 0.3)
plt.ylim(-0.3, 0.3)
plt.grid()
plt.arrow(-v_t1[0]*0.2,-v_t1[1]*0.2,v_t1[0]*0.4,v_t1[1]*0.4, width=0.005,
          color='g')
plt.arrow(-v_t2[0]*0.2,-v_t2[1]*0.2,v_t2[0]*0.4,v_t2[1]*0.4, width=0.001,
          color='g')
plt.show()
#CCA on X
plt.figure(figsize=(6,6))
plt.scatter(X[:,0], X[:,1], c=(X@(u1.T)).T)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.grid()
plt.arrow(-u1[0]*150,-u1[1]*150,u1[0]*300,u1[1]*300, width=0.1, color='g')
plt.arrow(-u2[0]*80,-u2[1]*80,u2[0]*160,u2[1]*160, width=0.05, color='g')
plt.show()
#CCA on Y
plt.figure(figsize=(6,6))
plt.scatter(Y[:,0], Y[:,1], c=(Y@(v1.T)).T)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.grid()
plt.arrow(-v1[0]*150,-v1[1]*150,v1[0]*300,v1[1]*300, width=0.1, color='g')
plt.arrow(-v2[0]*160,-v2[1]*160,v2[0]*320,v2[1]*320, width=0.05, color='g')

```

```
plt.show()
```