```
import numpy as np
import matplotlib.pyplot as plt

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

num_points = 50
x = np.linspace(-10,10,num_points)

#Dataset 1
X_1 = np.vstack((x,np.zeros(num_points))).T
#Dataset 2
X_2 = np.vstack((x,0.3*x)).T
#Dataset 3
X_3 = np.vstack((x,0.6*x)).T
#Dataset 4
X_4 = np.vstack((x,x)).T + np.random.randn(num_points,2)
#Dataset 5
x_abs = abs(x)
X_5 = np.vstack((x_abs*np.cos(4*x_abs),x_abs*np.sin(4*x_abs))).T
#Dataset 6
t = np.linspace(0,359,num_points) * np.pi/180
X_6 = np.vstack((10*np.cos(t),5*np.sin(t))).T
cs = np.cos(-np.pi/4)
ss = np.sin(-np.pi/4)
X_6 = X_6 @ np.asarray([[cs,-ss],[ss,cs]])
X_6 = X_6 + np.random.randn(num_points,2) * 0.5

#Correlation Coefficient calculation and Dataset plot function
def CorrCoeff(data):
    x = data[:, 0].T
    y = data[:, 1].T
    mean_x = np.mean(x)
    mean_y = np.mean(y)
    var_x = np.var(x)
    var_y = np.var(y)
    plt.scatter(x, y)
    plt.show()
    if var_x == 0 or var_y == 0:
        return "N/A"
    return
     np.sum((x-mean_x)*(y-mean_y))/np.sqrt(np.sum((x-mean_x)**2)*np
     .sum((y-mean_y)**2))

#Dataset 1 result
x_1 = CorrCoeff(X_1)
print(x_1)

#Dataset 2 result
x_2 = CorrCoeff(X_2)
print(x_2)

#Dataset 3 result
x_3 = CorrCoeff(X_3)
```

```
print(x_3)

#Dataset 4 result
x_4 = CorrCoeff(X_4)
print(x_4)

#Dataset 5 result
x_5 = CorrCoeff(X_5)
print(x_5)

#Dataset 6 result
x_6 = CorrCoeff(X_6)
print(x_6)
```