62130500048 นางสาวปฏิญญา ทองอ่วม

การทดลองจัดกลุ่มข้อมูล 2 มิติ ด้วยวิธี k-Means และ Hierarchical Clustering

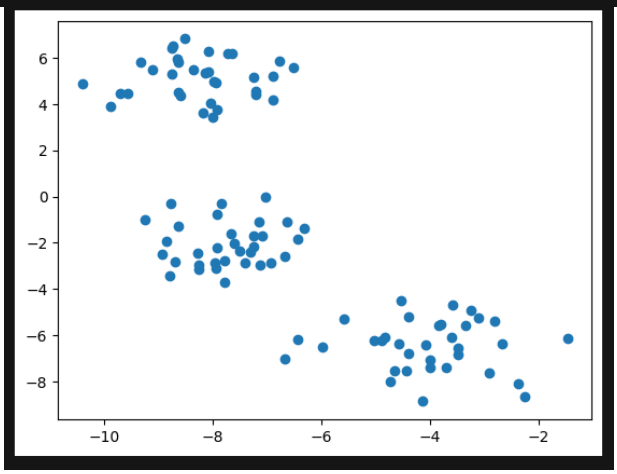
ชุดข้อมูลที่ 1

1. อ่านข้อมูล

```
data = pd.read_csv('data2Dset1.csv', header=None)
```

2. Plot จุดข้อมูล

```
x = data[0]
y = data[1]
plt.scatter(x,y)
plt.show()
```

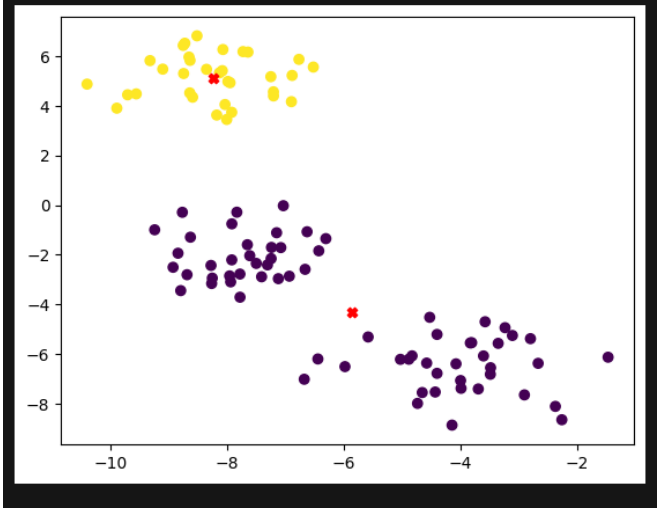


3. จัดกลุ่มข้อมูลด้วย K-Means

    a. k=2

```
model_kmeans = cluster.KMeans(n_clusters=2, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[-5.84938602 -4.29998479]
 [-8.24132694  5.12663729]]
```

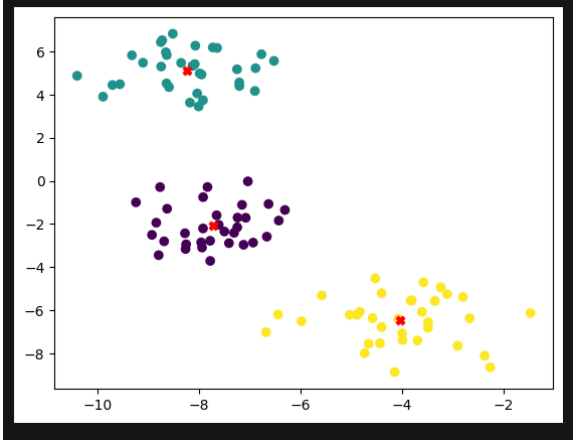


Scatter plot with k=2

b. k=3

```
model_kmeans = cluster.KMeans(n_clusters=3, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[-7.72806305e+00 -2.06698658e+00 -6.66133815e-16]
 [-8.24132694e+00  5.12663729e+00  1.00000000e+00]
 [-4.02596420e+00 -6.46730659e+00  2.58823529e+00]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

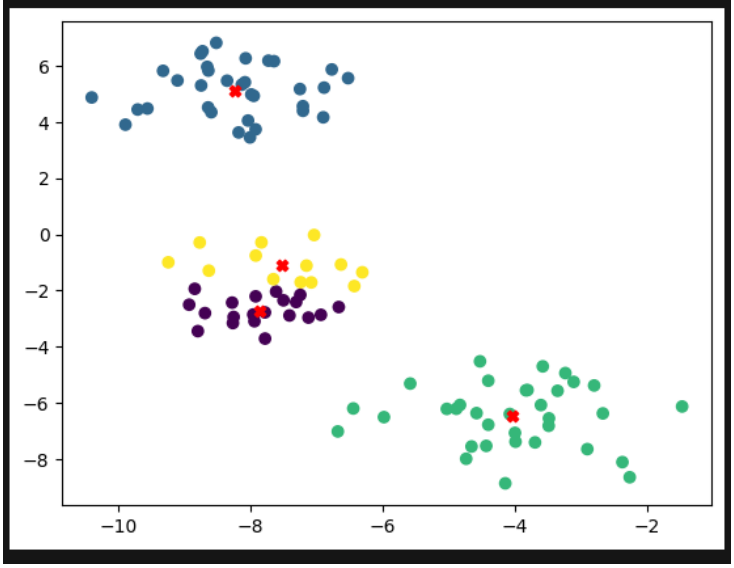


Scatter plot with k=3

c. k=4

```
model_kmeans = cluster.KMeans(n_clusters=4, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[-7.85763956e+00 -2.70745862e+00 -2.22044605e-16]
 [-8.24132694e+00  5.12663729e+00  1.00000000e+00]
 [-4.02596420e+00 -6.46730659e+00  2.82352941e+00]
 [-7.52871458e+00 -1.08164498e+00  3.00000000e+00]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```



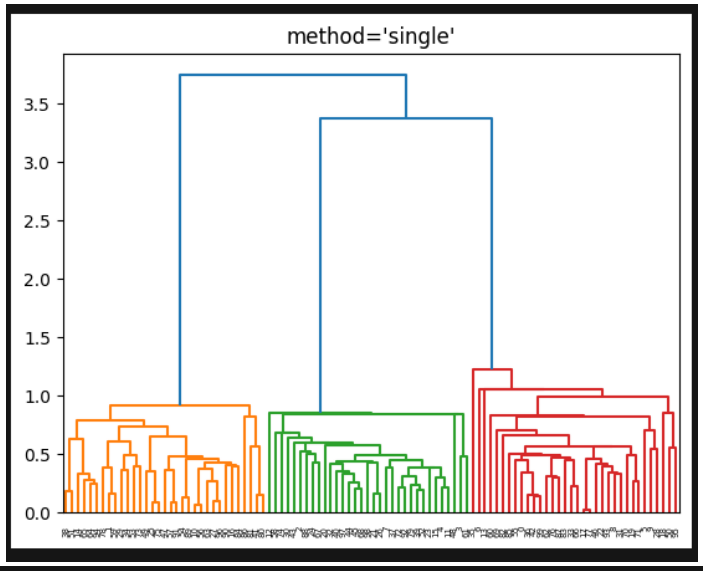
Scatter plot with k=4

โดยสรุปจาก k ทั้ง 3 ค่าที่ได้ทดลองใช้ พบว่าค่า k ที่ดีที่สุดคือ 3 เพราะจากข้อมูลที่ได้มาเมื่อนำมา plot แล้วจะเห็นได้ว่าข้อมูลมีการแบ่งออกเป็น 3 กลุ่มชัดเจน และเมื่อใช้สีในการแยก cluster ก็จะเห็นได้ว่าข้อมูลที่อยู่ในกราฟถูกแย่งแยกอย่างชัดเจนมากขึ้น

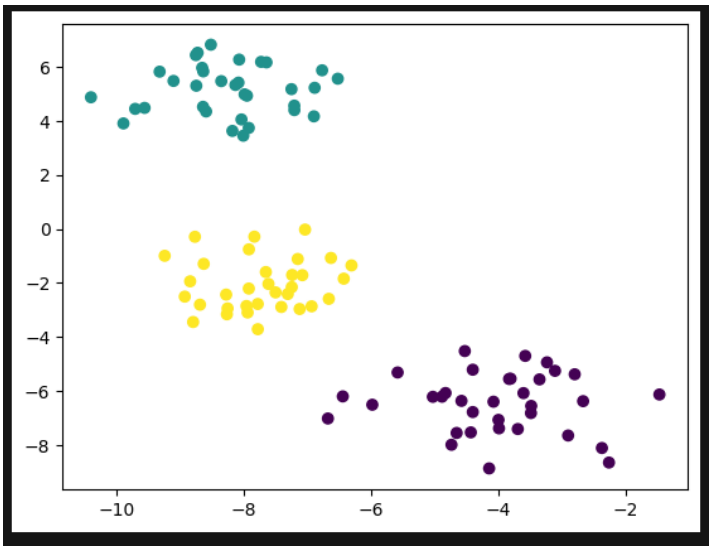
4. จัดกลุ่มข้อมูลด้วย Hierarchical Clustering

   a. Method single

```
linkage_data = linkage(data, method='single', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='single'")
plt.show()
```



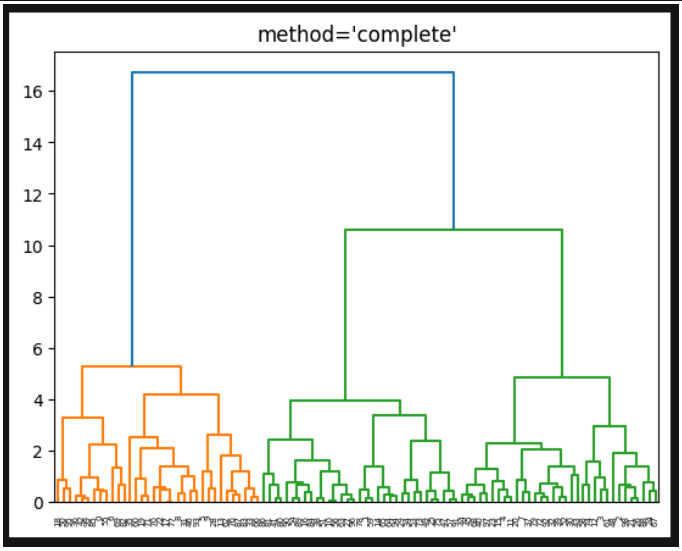

```
cluster_id = fcluster(linkage_data, t=1.5, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



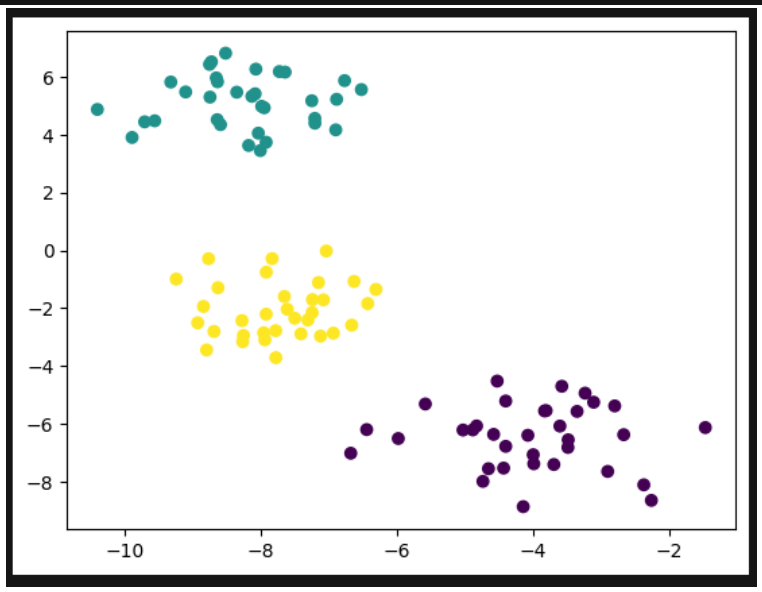
Cutoff โดยกำหนด t=1.5

b. Method complete

```
linkage_data = linkage(data, method='complete', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='complete'")
plt.show()
```



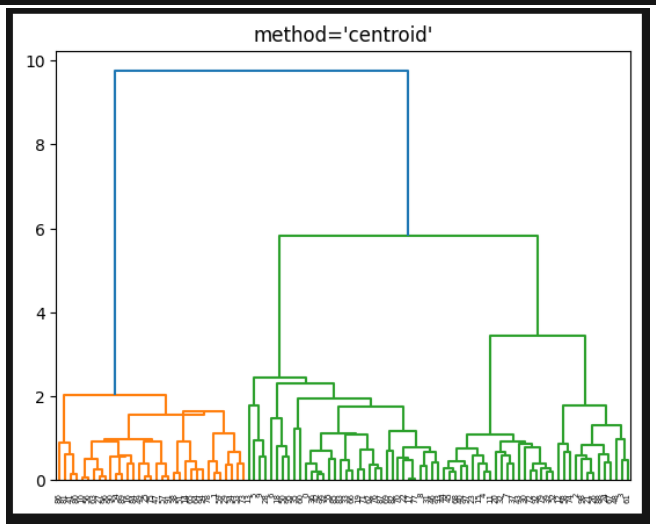

```
cluster_id = fcluster(linkage_data, t=6, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



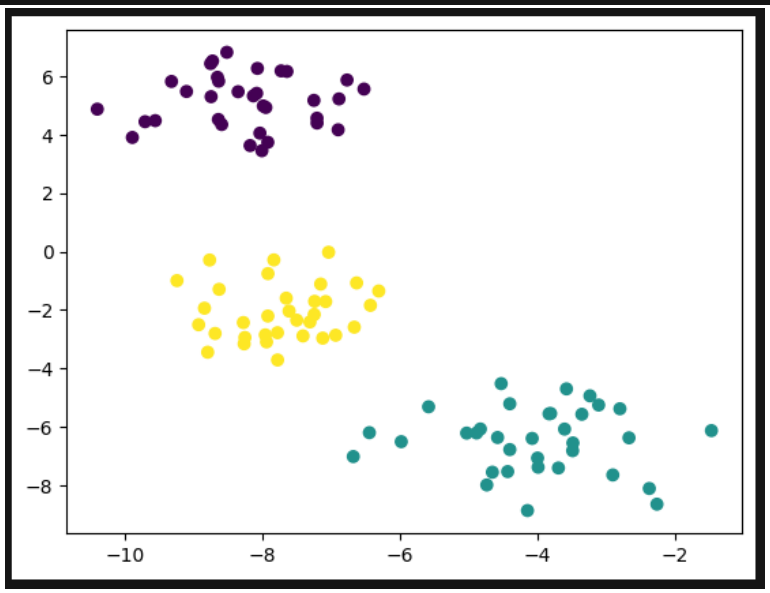
Cutoff โดยกำหนด t=6

c. Method centroid

```
linkage_data = linkage(data, method='centroid', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='centroid'")
plt.show()
```




```
cluster_id = fcluster(linkage_data, t=4, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



Cutoff โดยกำหนด t=4

62130500048 นางสาวปฏิญญา ทองอ่วม

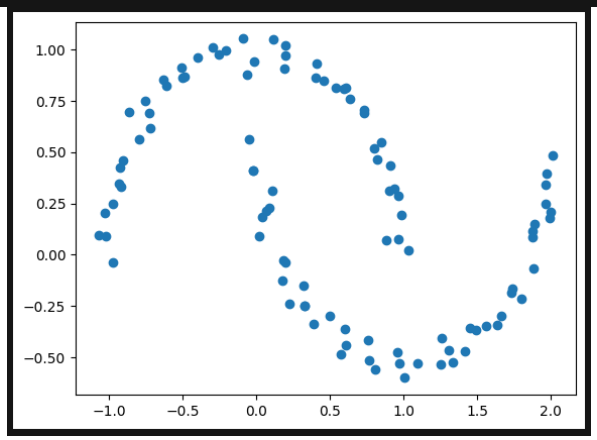
ชุดข้อมูลที่ 2

1. อ่านข้อมูล

```
data = pd.read_csv('data2Dset2.csv', header=None)
```

2. Plot จุดข้อมูล

```
x = data[0]
y = data[1]
plt.scatter(x,y)
plt.show()
```

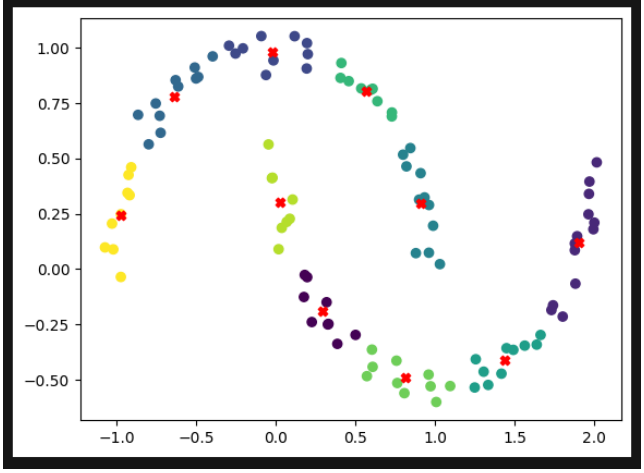


3. จัดกลุ่มข้อมูลด้วย K-Means

    a. k=10

```
model_kmeans = cluster.KMeans(n_clusters=10, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[ 0.29348978 -0.18971205]
 [ 1.90144855  0.12093266]
 [-0.02180567  0.97833667]
 [-0.63778976  0.77996668]
 [ 0.91166141  0.29507469]
 [ 1.43674707 -0.41001828]
 [ 0.56779473  0.80285646]
 [ 0.81444878 -0.49032176]
 [ 0.0275966   0.30164702]
 [-0.9733747   0.24049783]]
```

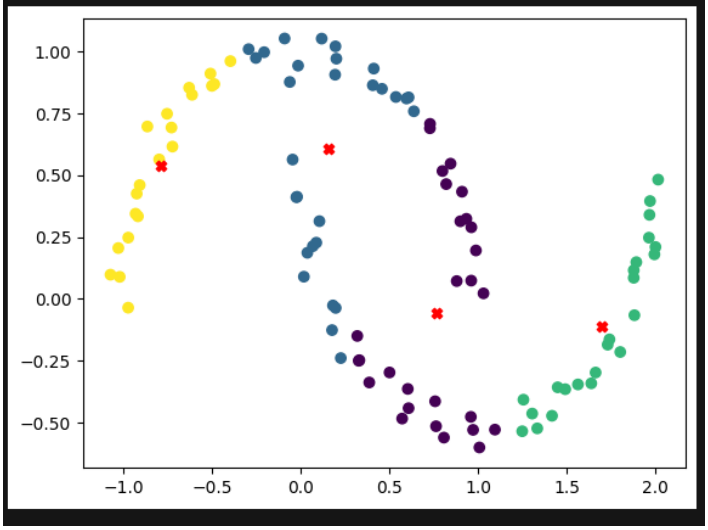


Scatter plot with k=10

b. k=4

```
model_kmeans = cluster.KMeans(n_clusters=4, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[ 7.67568428e-01 -5.51056610e-02 -1.11022302e-16]
 [ 1.52935308e-01  6.06910519e-01  1.00000000e+00]
 [ 1.69940443e+00 -1.09915574e-01 -1.11022302e-16]
 [-7.88802984e-01  5.37205698e-01  1.00000000e+00]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

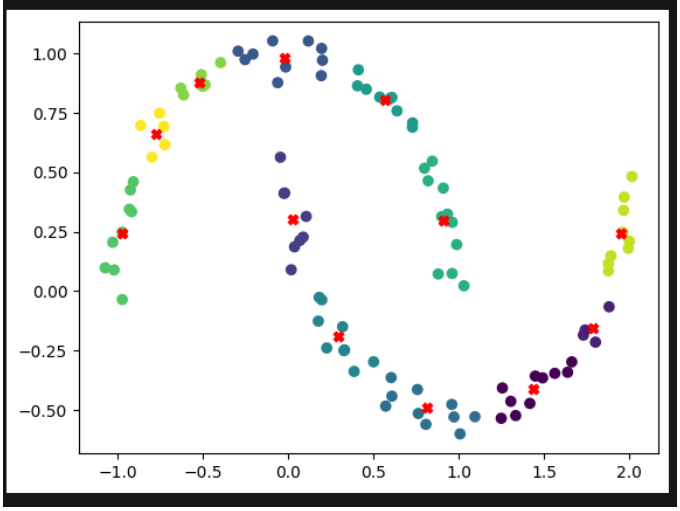


Scatter plot with k=4

c. k=12

```python
model_kmeans = cluster.KMeans(n_clusters=12, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[ 1.43674707 -0.41001828  5.        ]
 [ 1.78915463 -0.1569628   1.        ]
 [ 0.0275966   0.30164702  8.        ]
 [-0.02180567  0.97833667  2.        ]
 [ 0.81444878 -0.49032176  7.        ]
 [ 0.29348978 -0.18971205  0.        ]
 [ 0.56779473  0.80285646  6.        ]
 [ 0.91166141  0.29507469  4.        ]
 [-0.9733747   0.24049783  9.        ]
 [-0.52352643  0.87817123  3.        ]
 [ 1.95135696  0.24444176  1.        ]
 [-0.77490576  0.66212123  3.        ]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

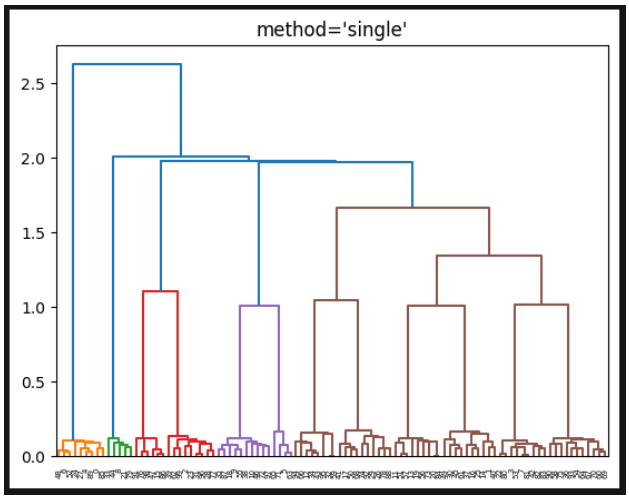


Scatter plot with k=12

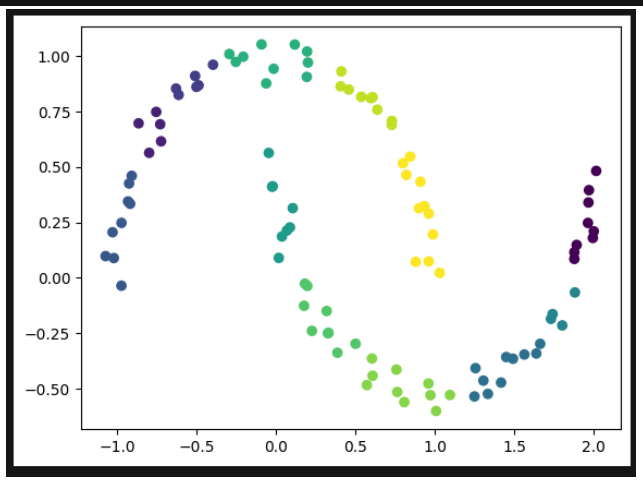
โดยสรุปแล้ว k ที่ทดลองนำมาใช้แล้วคิดว่าดีที่สุดคือ k=10 เนื่องจากหากมองที่ k=4 จะเห็นว่าถึงแม้จะสามารถแยกข้อมูลตามสีได้ชัด แต่กลุ่มแต่ละกลุ่มนั้นกว้างเกินไป และหากเป็น k=12 การแบ่งกลุ่มนั้นละเอียดเกินไปจนมองยาก ดังนั้น k=10 จึงเป็นตัวเลือกที่ดีที่สุดเพราะนอกจากจะแยกข้อมูลได้ไม่กว้างเกินไปแล้วก็ยังไม่ละเอียดเกินไปอีกด้วย

4. จัดกลุ่มข้อมูลด้วย Hierarchical Clustering
    a. Method single

```python
linkage_data = linkage(data, method='single', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='single'")
plt.show()
```



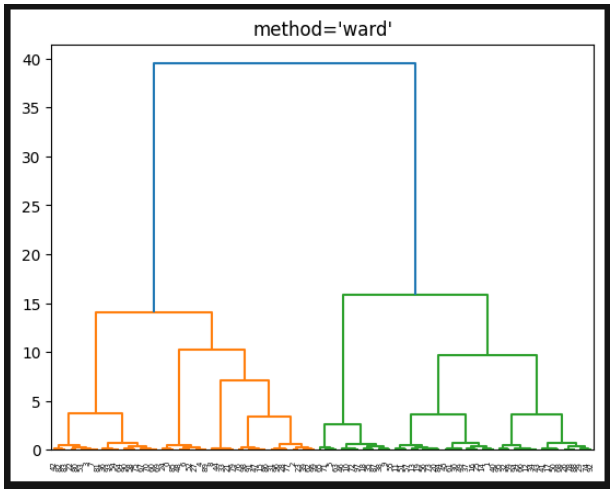

```
cluster_id = fcluster(linkage_data, t=1.0, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



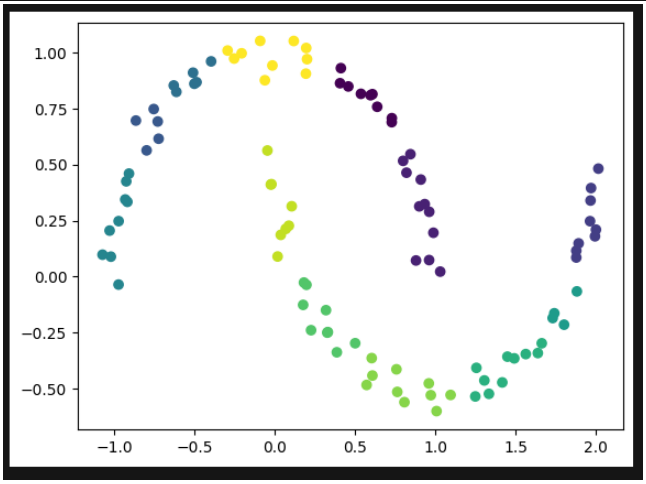
Cutoff ที่ t=1

b. Method ward

```
linkage_data = linkage(data, method='ward', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='ward'")
plt.show()
```




```
cluster_id = fcluster(linkage_data, t=2.5, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```

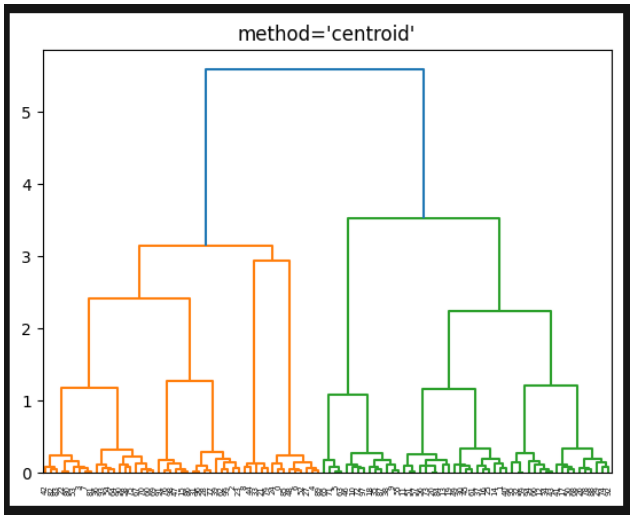


Cutoff ที่ t=2.5

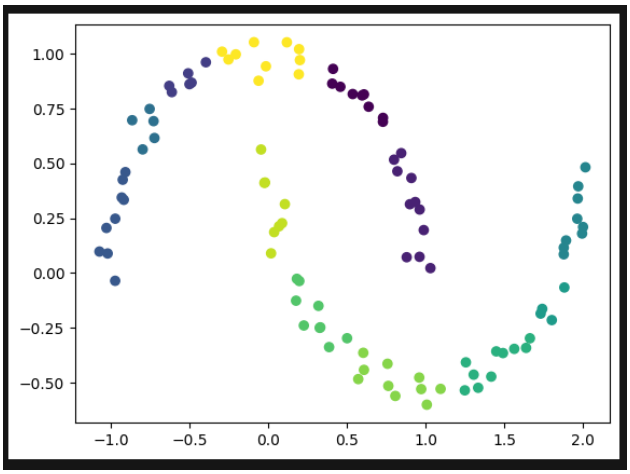
c. Method centroid

```
linkage_data = linkage(data, method='centroid', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='centroid'")
plt.show()
```

62130500048 นางสาวปฏิญญา ทองอ่วม




```
cluster_id = fcluster(linkage_data, t=1, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



Cutoff ที่ t=1

62130500048 นางสาวปฏิญญา ทองอ่วม

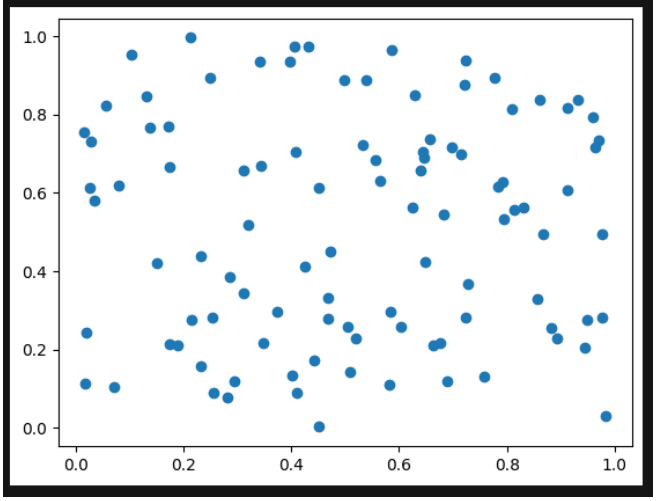
ชุดข้อมูลที่ 3

1. อ่านข้อมูล

```
data = pd.read_csv('data2Dset3.csv', header=None)
```

2. Plot จุดข้อมูล

```
x = data[0]
y = data[1]
plt.scatter(x,y)
plt.show()
```

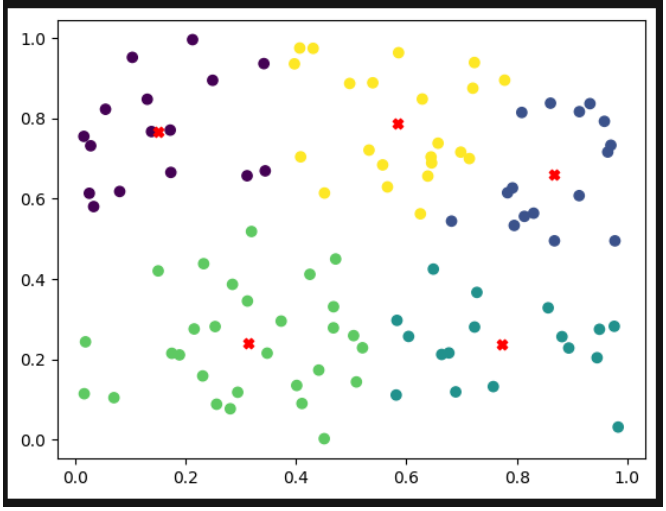


3. จัดกลุ่มข้อมูลด้วย K-Means

    a. k=5

```
model_kmeans = cluster.KMeans(n_clusters=5, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[0.15123791 0.76708817]
 [0.8664977  0.66124364]
 [0.77315914 0.23617367]
 [0.3137367  0.24132663]
 [0.58401396 0.78610906]]
```

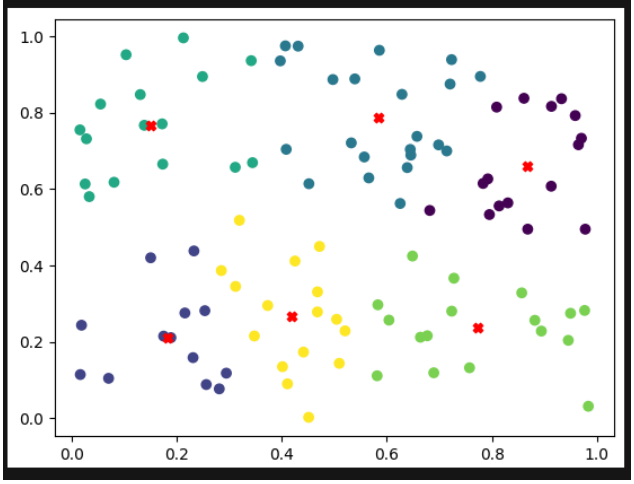


Scatter plot with k=5

b. k=6

```
model_kmeans = cluster.KMeans(n_clusters=6, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[0.8664977  0.66124364 1.        ]
 [0.18363854 0.21083205 3.        ]
 [0.58401396 0.78610906 4.        ]
 [0.15123791 0.76708817 0.        ]
 [0.77315914 0.23617367 2.        ]
 [0.41944145 0.26610349 3.        ]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

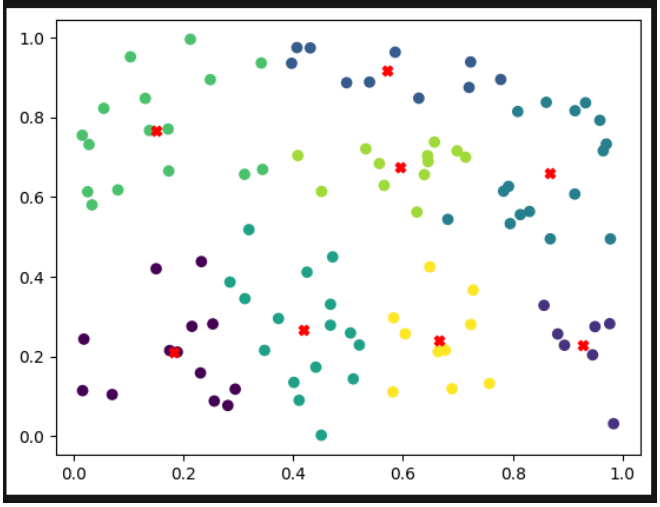


Scatter plot with k=6

c. k=8

```python
model_kmeans = cluster.KMeans(n_clusters=8, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[ 1.83638542e-01  2.10832045e-01  1.00000000e+00]
 [ 9.26765461e-01  2.28958321e-01  4.00000000e+00]
 [ 5.70828739e-01  9.17949947e-01  2.00000000e+00]
 [ 8.66497703e-01  6.61243644e-01 -4.44089210e-16]
 [ 4.19441446e-01  2.66103486e-01  5.00000000e+00]
 [ 1.51237915e-01  7.67088174e-01  3.00000000e+00]
 [ 5.95001651e-01  6.76241645e-01  2.00000000e+00]
 [ 6.65634715e-01  2.41224414e-01  4.00000000e+00]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

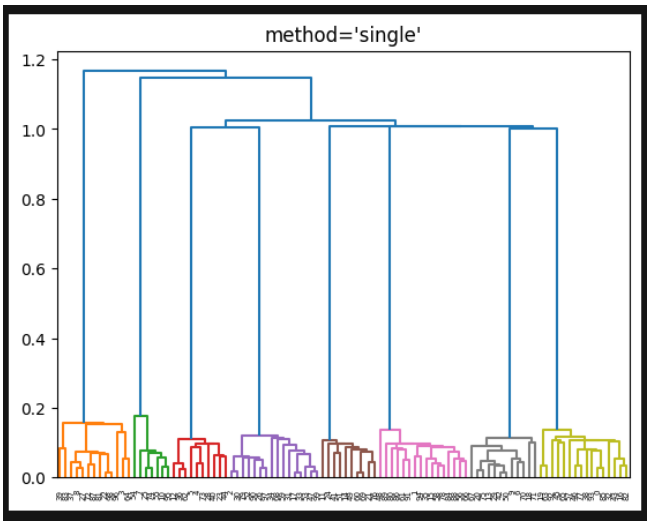


Scatter plot with k=8

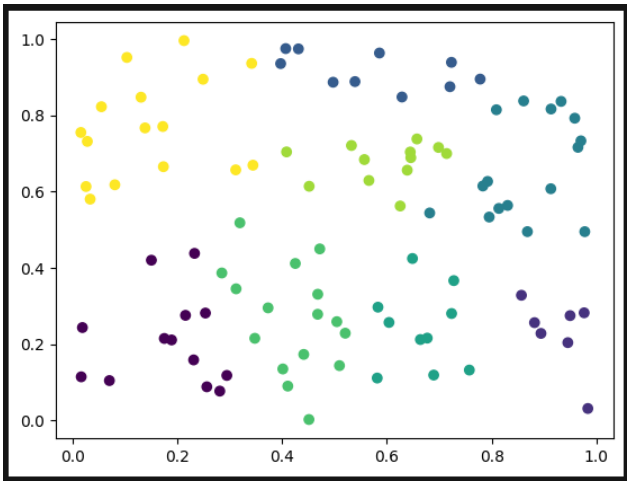
โดยสรุปแล้ว k ที่ดีที่สุดคือ k=6 ถึงแม้ข้อมูลที่มีจะดูกระจัดกระจาย แต่เมื่อลองใช้ k=6 แล้วจะเห็นข้อมูลที่จัดกลุ่มไว้ได้ชัดเจนและไม่กว้างไม่ละเอียดเกินไป หากเลข k มากกว่านี้จะได้ข้อมูลที่อ่านและวิเคราะห์ได้ยาก

4. จัดกลุ่มข้อมูลด้วย Hierarchical Clustering
    a. Method single

```
linkage_data = linkage(data, method='single', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='single'")
plt.show()
```



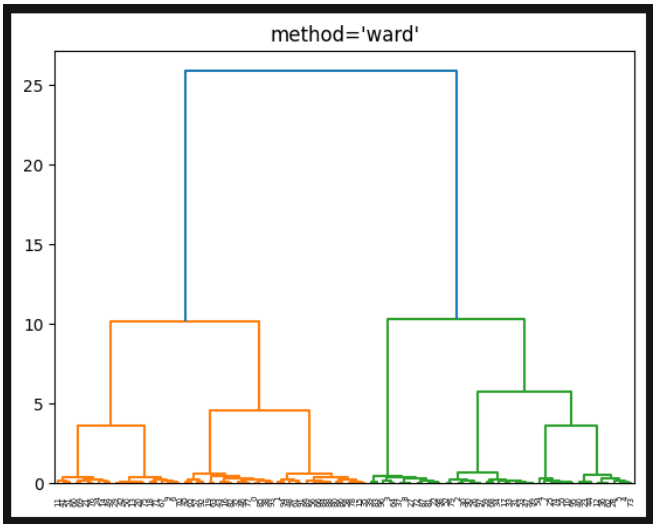

```
cluster_id = fcluster(linkage_data, t=0.2, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



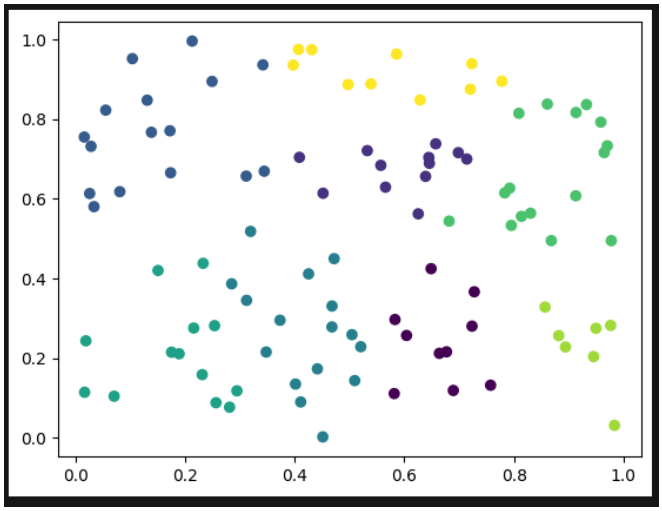
Cutoff ที่ t=0.2

b. Method ward

```
linkage_data = linkage(data, method='ward', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='ward'")
plt.show()
```



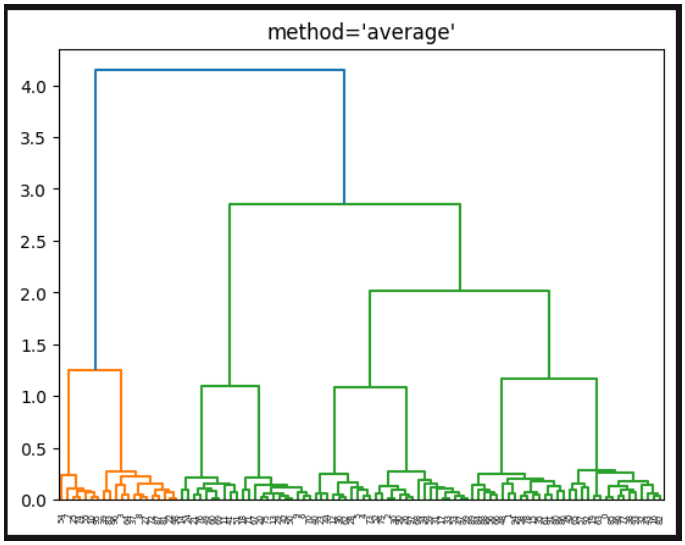

```
cluster_id = fcluster(linkage_data, t=2.5, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



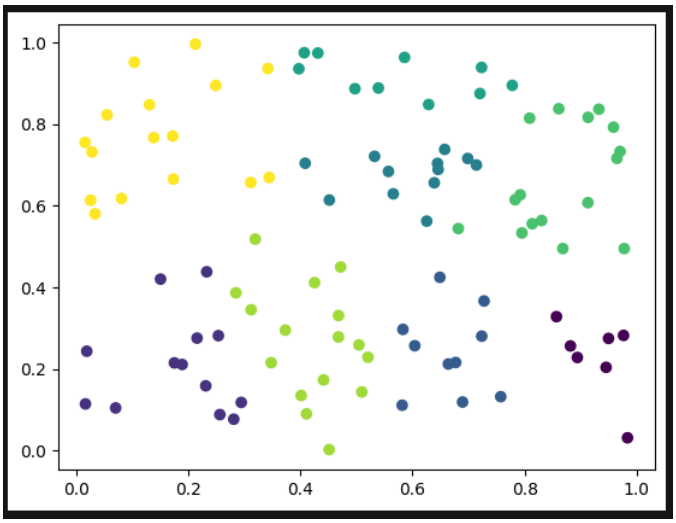
Cutoff ที่ t=2.5

c. Method average

```
linkage_data = linkage(data, method='average', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='average'")
plt.show()
```




```
cluster_id = fcluster(linkage_data, t=0.5, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



Cutoff ที่ t=0.5

62130500048 นางสาวปฏิญญา ทองอ่วม

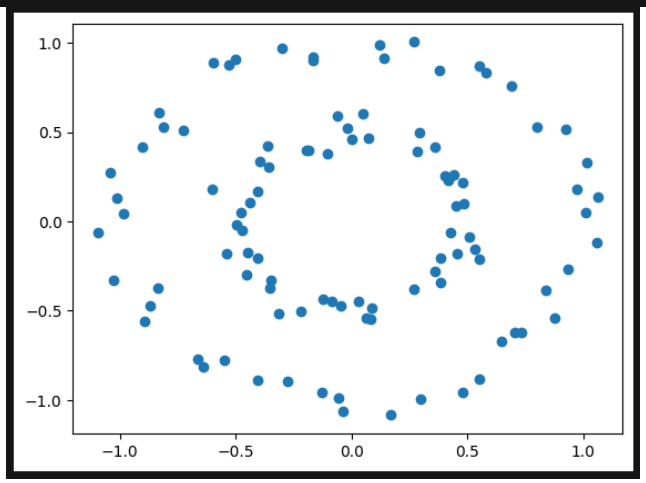
ชุดข้อมูลที่ 4

1. อ่านข้อมูล

```
data = pd.read_csv('data2Dset4.csv', header=None)
```

2. Plot จุดข้อมูล

```
x = data[0]
y = data[1]
plt.scatter(x,y)
plt.show()
```

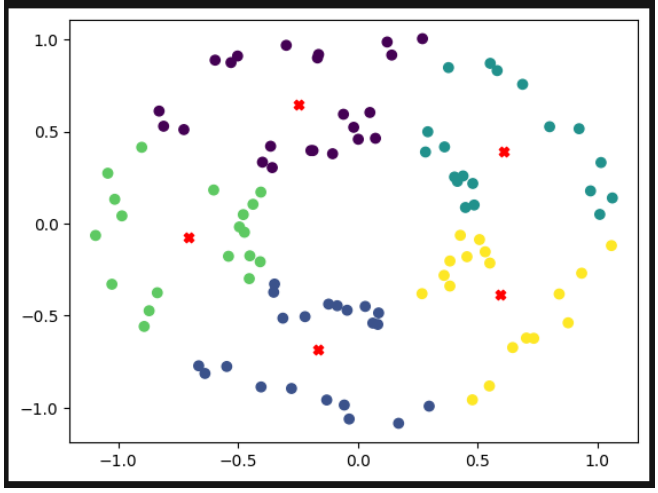


3. จัดกลุ่มข้อมูลด้วย K-Means

    a. k=5

```
model_kmeans = cluster.KMeans(n_clusters=5, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[-0.24486397  0.6478188 ]
 [-0.16621059 -0.68228806]
 [ 0.61069197  0.39494785]
 [-0.70477236 -0.07126065]
 [ 0.59463731 -0.3872353 ]]
```

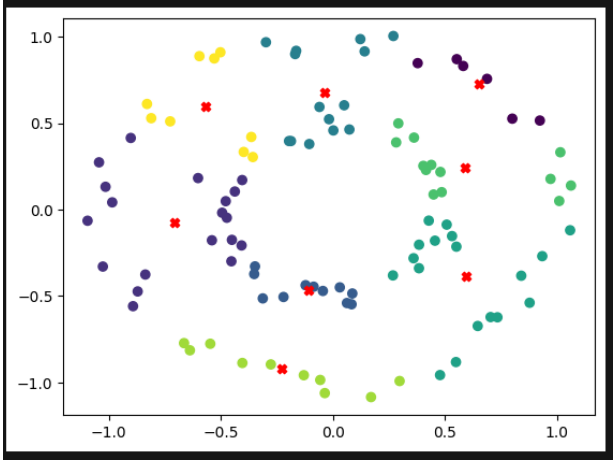


Scatter plot with k=5

b. k=8

```
model_kmeans = cluster.KMeans(n_clusters=8, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[ 6.54656933e-01  7.24993913e-01  2.22044605e-16]
 [-7.04772361e-01 -7.12606479e-02  2.00000000e+00]
 [-1.10284225e-01 -4.63482595e-01  3.00000000e+00]
 [-3.76307517e-02  6.79608970e-01  1.00000000e+00]
 [ 5.94637310e-01 -3.87235298e-01  2.00000000e+00]
 [ 5.90400449e-01  2.42618904e-01  0.00000000e+00]
 [-2.27729599e-01 -9.22974075e-01  3.00000000e+00]
 [-5.67226759e-01  5.98367420e-01  1.00000000e+00]]

c:\Users\patip\AppData\Local\Programs\Python\Python310\lib\site-packages\s
dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

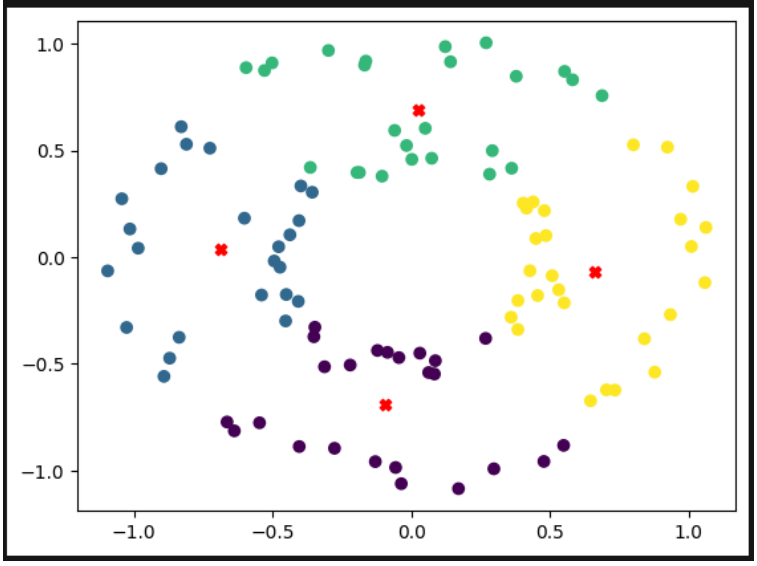


Scatter plot with k=8

c. k=4

```
model_kmeans = cluster.KMeans(n_clusters=4, max_iter=50, random_state=1)
model_kmeans.fit(data)

data['cluster_id'] = model_kmeans.labels_

centroids = model_kmeans.cluster_centers_
print(centroids)

plt.scatter(x,y, c=data['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X', c='r')
plt.show()
```

```
[[-0.09139111 -0.68952081]
 [-0.68783056  0.03900941]
 [ 0.025057    0.68891146]
 [ 0.66179799 -0.06884846]]
```

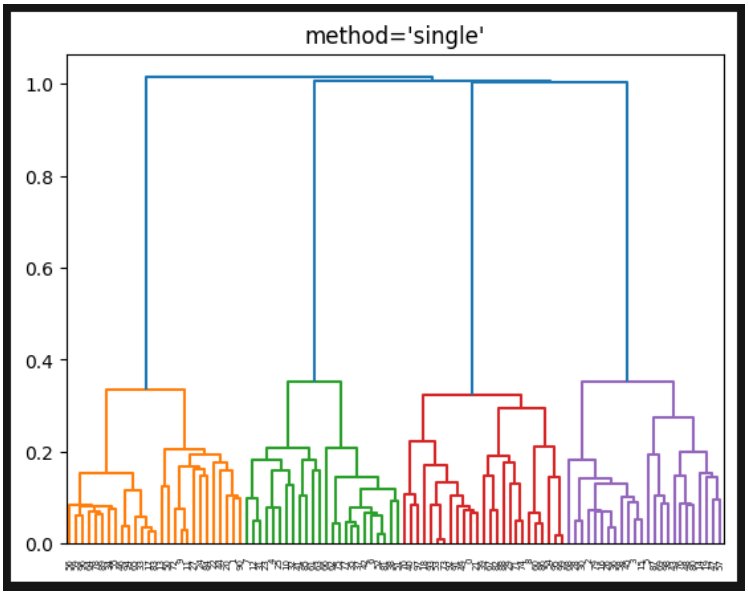


Scatter plot with k=4

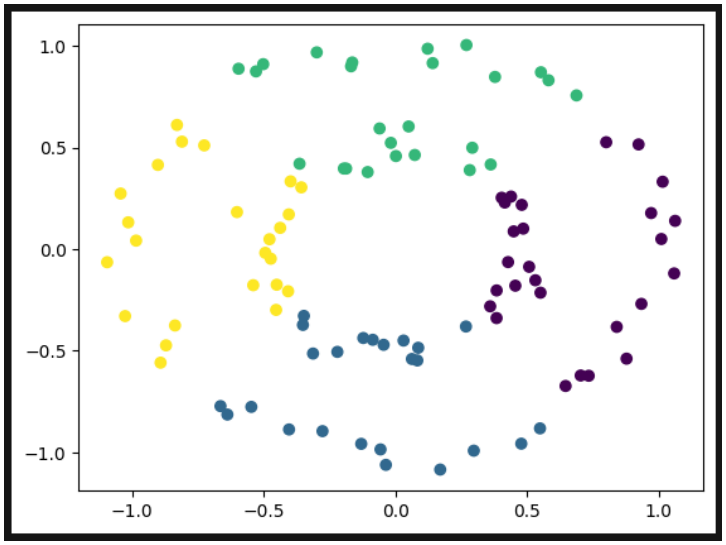
โดยสรุป k ที่ดีที่สุดคือ 4 เพราะข้อมูลแบ่งสัดส่วนกันสวยงามชัดเจน ดูง่าย แต่ถ้าหากมองว่าชุดข้อมูลกว้างเกินไปก็สามารถเลือก k=5 ได้เช่นกัน

4. จัดกลุ่มข้อมูลด้วย Hierarchical Clustering
    a. Method single

```python
linkage_data = linkage(data, method='single', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='single'")
plt.show()
```



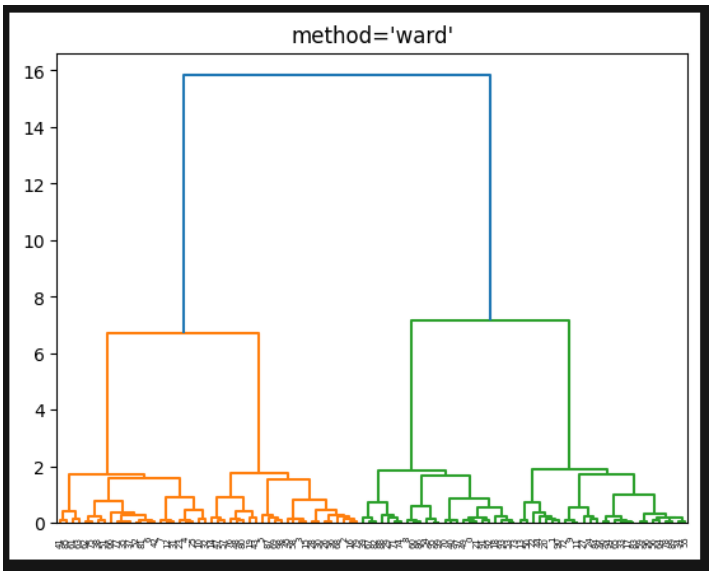

```
cluster_id = fcluster(linkage_data, t=0.4, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



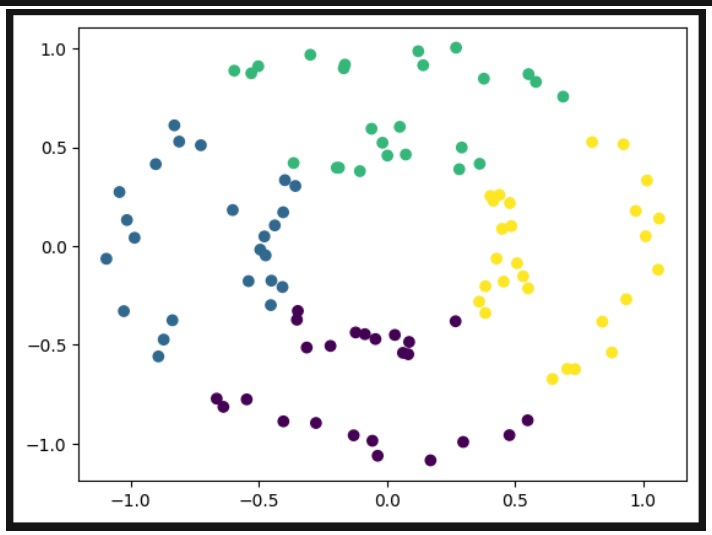
Cutoff ที่ t=0.4

b. Method ward

```
linkage_data = linkage(data, method='ward', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='ward'")
plt.show()
```



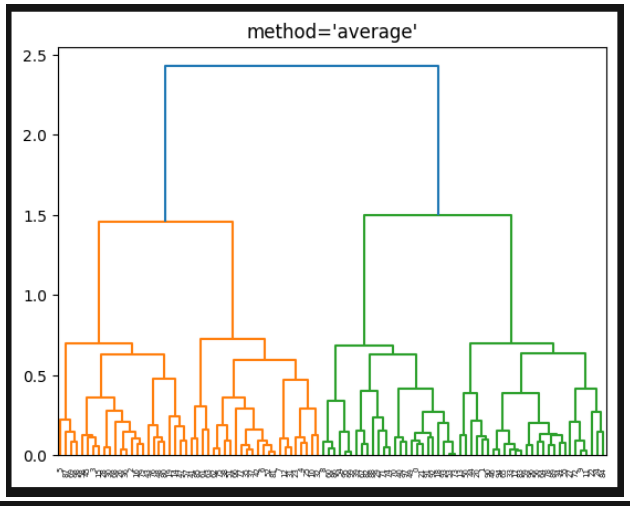

```
cluster_id = fcluster(linkage_data, t=2, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



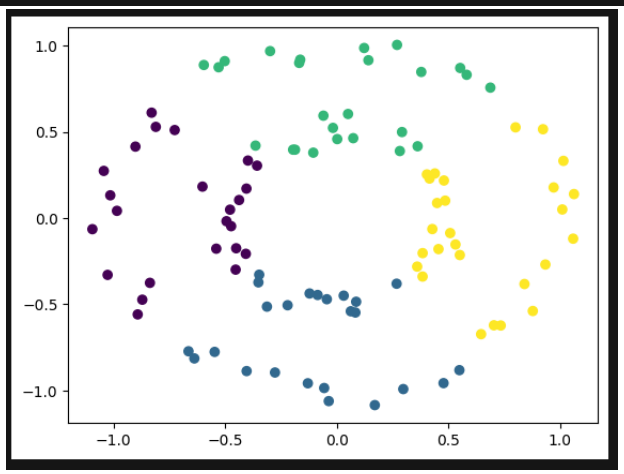
Cutoff ที่ t=2

c. Method average

```
linkage_data = linkage(data, method='average', metric='euclidean')
dendrogram(linkage_data)

plt.title("method='average'")
plt.show()
```




```
cluster_id = fcluster(linkage_data, t=1, criterion='distance')
plt.scatter(x,y, c=cluster_id)
plt.show()
```



Cutoff ที่ t=1

สรุปผลการทดลอง

K-means นั้นเป็นการระบุจำนวน cluster ที่เราต้องการจะแยกหมวดหมู่ ซึ่งวิธีนี้ควรใช้กับชุดข้อมูลที่เราเห็น pattern ชัดเจน สามารถระบุจำนวน cluster หรือ k ได้ง่าย ซึ่งการที่เราสามารถระบุ k ที่เราต้องการลงไปได้เลยนั้นก็คือข้อดีของ K-means แต่ข้อเสียคือค่อนข้างเสีย performance ในการทำงาน ใช้เวลามาก

ส่วน Hierarchical Clustering จากการทดลอง จะเป็นการลองใช้ method ทีละแบบ ซึ่งแต่ละแบบอาจจะได้ผลลัพธ์หรือการทำ dendrogram ได้ผลต่างหรือใกล้เคียงกันขึ้นอยู่กับข้อมูล แต่ทุกแบบจะทำให้เราเห็น hierarchy ที่ชัดเจนเห็น pattern ได้ดีขึ้น และเมื่อเราเห็นข้อมูลที่ถูกจัดอยู่ในรูปแบบของ dendrogram แล้ว เราก็สามารถที่จะวิเคราะห์เพื่อเลือกค่า t ที่จะนำไปใช้ cutoff เพื่อทำให้เกิด cluster ที่เหมาะสมได้ ซึ่งหมายความว่า

62130500048 นางสาวปฏิญญา ทองอ่วม

วิธีนี้จะใช้ได้ดีก็ต่อเมื่อข้อมูลมีจำนวนไม่เยอะมาก หากข้อมูลมีเยอะเกินไป จะทำให้วิเคราะห์ค่า t ที่เหมาะสมได้ยาก