เริ่มต้นด้วยการ import

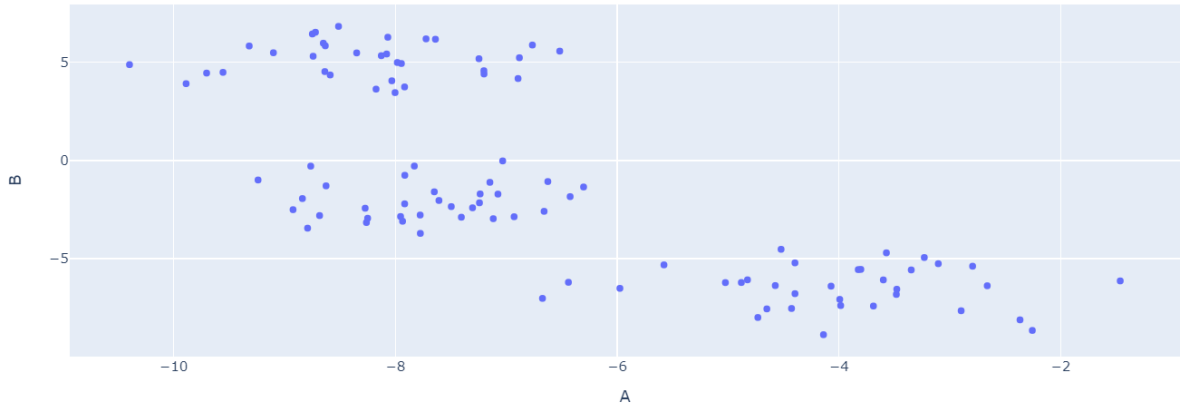
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import cluster
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage
from scipy.spatial import distance_matrix
```

```
data = pd.read_csv("data/data2Dset1.csv",header=None)
data.columns = ["A","B"]
data.head()
```

	A	B
0	-4.575007	-6.364897
1	-7.202692	4.560245
2	-7.148368	-1.115191
3	-7.915773	-0.757674
4	-7.118251	-2.965019

2) Plot จุดข้อมูล data1

```
import plotly.express as px
fig = px.scatter(data1, x="A", y="B")
fig.show()
```



3) Data2Dset1 เริ่มด้วยการเขียนโปรแกรม plot จุดข้อมูลโดยใช้วิธี kmeans

K=1

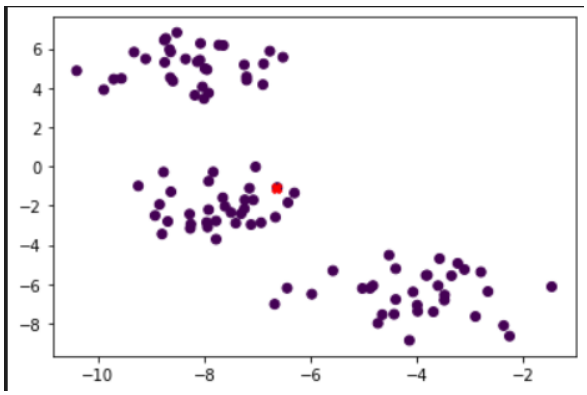
```
model_kmeans = cluster.KMeans(n_clusters=1, max_iter=50, random_state=1)
```

```
model_kmeans.fit(data1)
data1['cluster_id'] = model_kmeans.labels_
centroids = model_kmeans.cluster_centers_
print(centroids)


  [[-6.63872652 -1.18919951]]
```

```
plt.scatter(data1['A'],data1['B'], c=data1['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')
plt.show()
```

ผลลัพธ์ K=1

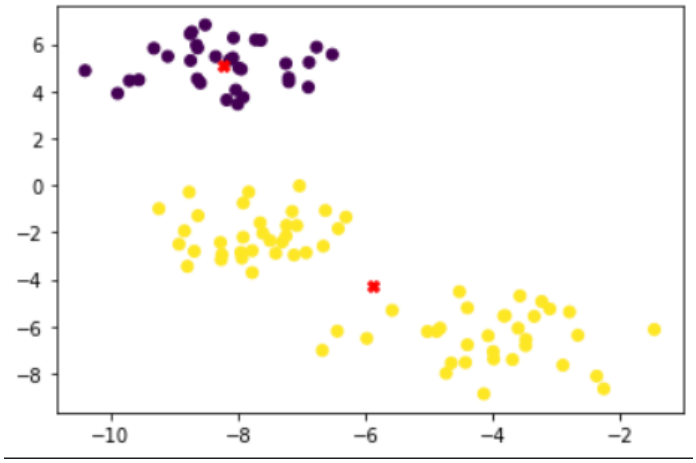


K=2

```
model_kmeans = cluster.KMeans(n_clusters=2, max_iter=50, random_state=1)
model_kmeans.fit(data1)
data1['cluster_id'] = model_kmeans.labels_
centroids = model_kmeans.cluster_centers
print(centroids)
   [[-5.84938602 -4.29998479  1.01492537]
    [-8.24132694  5.12663729  1.        ]]
```

```
plt.scatter(data1['A'],data1['B'], c=data1['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')
plt.show()
```

ผลลัพธ์ K=2



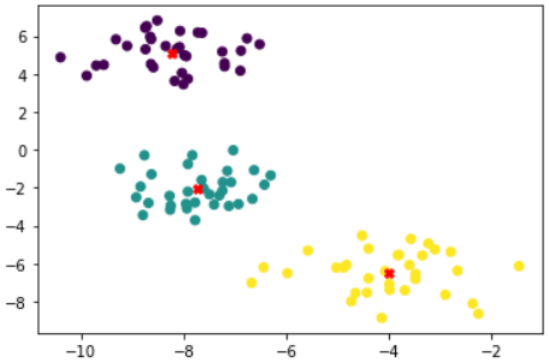
K=3

```
model_kmeans = cluster.KMeans(n_clusters=3, max_iter=50, random_state=1)
model_kmeans.fit(data1)
data1['cluster_id'] = model_kmeans.labels_
centroids = model_kmeans.cluster_centers_
print(centroids)
```

```
[[-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]]
```

```
plt.scatter(data1['A'],data1['B'], c=data1['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')
plt.show()
```

ผลลัพธ์ K=3



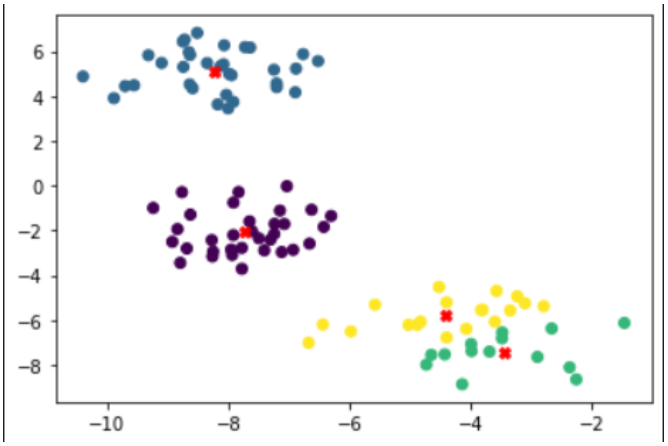
K=4

```
model_kmeans = cluster.KMeans(n_clusters=4, max_iter=50, random_state=1)
model_kmeans.fit(data1)
data1['cluster_id'] = model_kmeans.labels_
centroids = model_kmeans.cluster_centers_
print(centroids)
```

```
[[-7.72806305 -2.06698658  0.        ]
 [-8.24132694  5.12663729  0.        ]
 [-3.44525215 -7.43309475  0.        ]
 [-4.43246263 -5.79125489  0.        ]]
```

```
plt.scatter(data1['A'],data1['B'], c=data1['cluster_id'])
plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')
plt.show()
```

ผลลัพธ์K=4



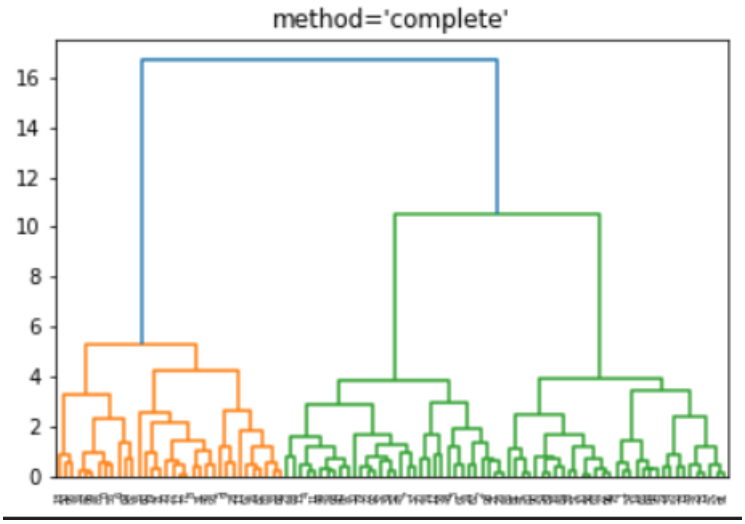
จากผลลัพธ์สรุปได้ว่า K=3 ดีที่สุดสำหรับ Data2Dset1 เพราะจะข้อมูลจะเกาะกลุ่มกันเป็นกลุ่มใหญ่ๆอย่างเห็นได้ชัดสามกลุ่ม ถ้าเป็นK4 จะมีตัวข้อมูลที่เกาะกลุ่มกันมากและถ้าเป็นK1,K2 จะเกิดข้อมูลที่ไม่แน่ชัด

4) เขียนโปแกรมจัดกลุ่มชุดข้อมูลที่อ่านเข้ามา โดยใช้วิธี Hierarchical Clustering

4.1) ให้เลือกใช้method ที่ต่างกัน 3 แบบ แสดง dendrogram ที่ได้แต่ละแบบ

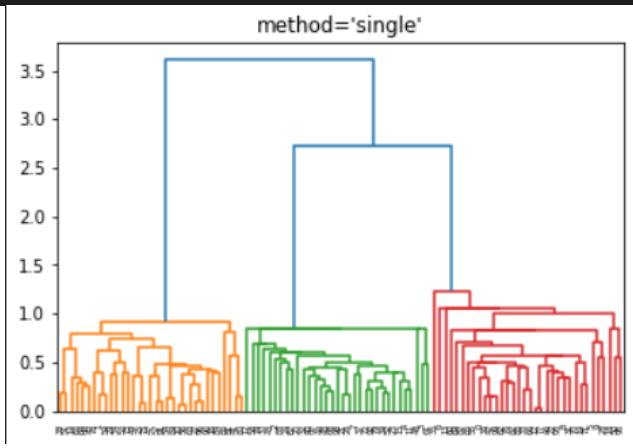
Complete:

```
linkage_data = linkage(data1, method='complete' , metric='euclidean')
dendrogram(linkage_data)
plt.title("method='complete'")
plt.show()
```

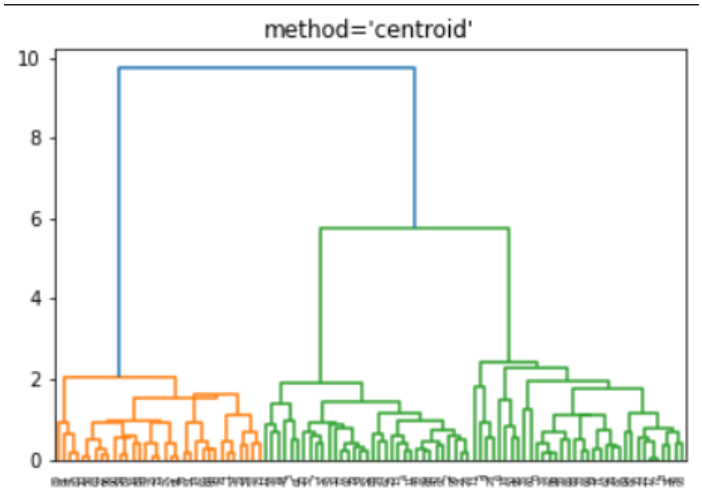


Single:

```
linkage_data = linkage(data1, method='single' , metric='euclidean')
dendrogram(linkage_data)
plt.title("method='single'")
plt.show()
```



Centroid:



4.2) เลือก cut-off โดยกำหนด criterion='distance' และให้นักศึกษาเลือกระบุค่า t ที่คิดว่า เหมาะสม สำหรับแต่ละ dendrogram ที่ได้ในข้อ 4.1)

Complete: ค่า t ที่เหมาะที่สุดคือ 6

```
cluster_id = fcluster(linkage_data,t=6,criterion='distance')
plt.scatter(data1["A"],data1["B"],c=cluster_id)
plt.show()
```

Single: ค่า t ที่เหมาะที่สุดคือ 2

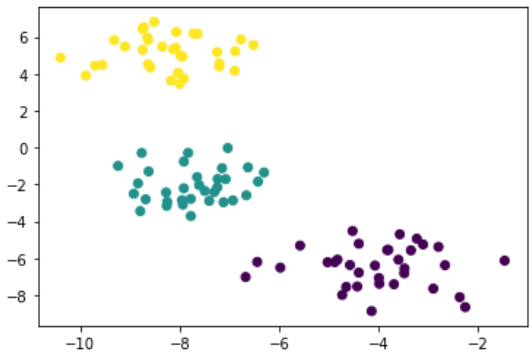
```
cluster_id = fcluster(linkage_data,t=2,criterion='distance')
plt.scatter(data1["A"],data1["B"],c=cluster_id)
plt.show()
```

Centroid: ค่า t ที่เหมาะที่สุดคือ 3 ถึง 5

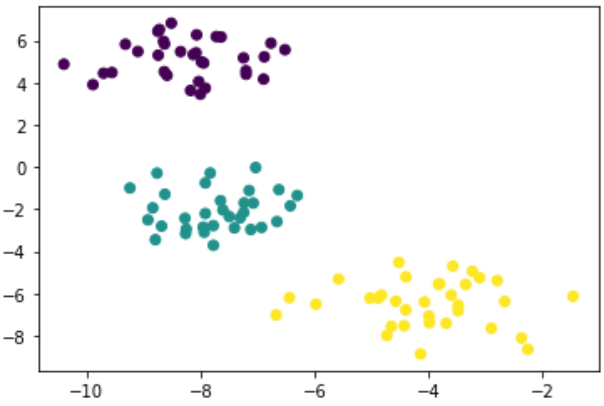
```
cluster_id = fcluster(linkage_data,t=5,criterion='distance')
plt.scatter(data1["A"],data1["B"],c=cluster_id)
plt.show()
```

4.3) Plot ผลการจัดกลุ่ม ที่ได้แต่ละแบบในข้อ 4.2)

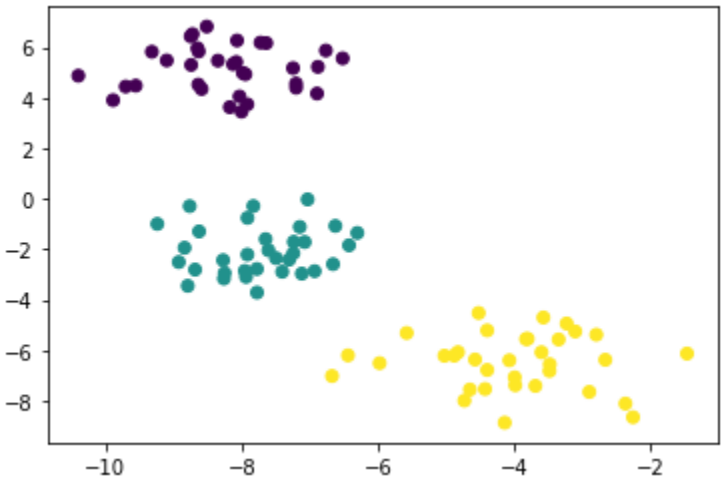
Complete:



Single:



Centroid:



6. เขียนบรรยายสรุปผลการทดลอง แสดงความคิดเห็น วิธีใด เหมาะกับ ชุดข้อมูลแบบไหน แต่ละวิธีมีข้อดี/ ข้อเสีย อย่างไร

- Data2DSet1 จากการทดลองพบว่า วิธี k-Means สามารถทำได้ง่ายกว่า สามารถแบ่งกลุ่มข้อมูลได้ง่ายกว่า และตัวข้อมูลมีดูมีความละเอียดแบ่งเป็นกลุ่มๆอย่างเห็นได้ชัด