Hierarchical method of clustering- II

Dr. A. Ramesh
DEPARTMENT OF MANAGEMENT STUDIES

Agenda

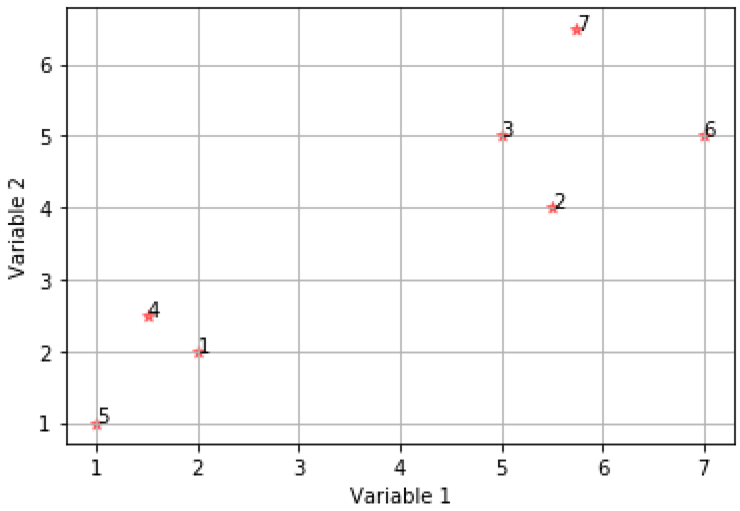
- Agglomerative hierarchical algorithm
- Python demo

Example for Hierarchical Agglomerative Clustering (HAC)

- A data set consisting of seven objects for which two variables were measured.

Object	Variable 1	Variable 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Scatter plot



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

- Calculate Euclidean Distance and create the distance matrix.

$$\text{Distance}[(x_1, y_1), (x_2, y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance (1,2)

$$(2.00, 2.00)\ (5.50, 4.00) = \sqrt{(5.50 - 2.00)^2 + (4.00 - 2.00)^2} = 4.02$$

Distance (1,3)

$$(2.00, 2.00)\ (5.00, 5.00) = \sqrt{(5.00 - 2.00)^2 + (5.00 - 2.00)^2} = 4.24$$

Distance (1,4)

$$(2.00, 2.00)\ (1.50, 2.50) = \sqrt{(1.50 - 2.00)^2 + (2.50 - 2.00)^2} = 0.71$$

Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (1,5)

$$(2.00, 2.00)\ (1.00, 1.00) = \sqrt{(1.00-2.00)^2+(1.00-2.00)^2}$$

$$= 1.41$$

Distance (1,6)

$$(2.00, 2.00)\ (7.00, 5.00) = \sqrt{(7.00-2.00)^2+(5.00-2.00)^2} = 5.83$$

Distance (1,7)

$$(2.00, 2.00)\ (5.75, 6.50) = \sqrt{(5.75-2.00)^2+(6.50-2.00)^2} = 5.86$$

Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (2,3)

$$(5.50, 4.00)\ (5.00, 5.00) = \sqrt{(5.00-5.50)^2+(5.00-4.00)^2}$$

$$= 1.12$$

Distance (2,4)

$$(5.50, 4.00)\ (1.50, 2.50) = \sqrt{(1.50-5.50)^2+(2.50-4.00)^2} = 4.27$$

Distance (2,5)

$$(5.50, 4.00)\ (1.00, 1.00) = \sqrt{(1.00-5.50)^2+(1.00-4.00)^2} = 5.41$$

Distance (2,6)

$$(5.50, 4.00)\ (7.00, 5.00) = \sqrt{(7.00-5.50)^2+(5.00-4.00)^2} = 1.80$$

Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (2,7)

$$(5.50, 4.00)\ (5.75, 6.50) = \sqrt{(5.75 - 5.50)^2 + (6.50 - 4.00)^2}$$

$$= 2.51$$

Distance (3,4)

$$(5.00, 5.00)\ (1.50, 2.50) = \sqrt{(1.50 - 5.00)^2 + (2.50 - 5.00)^2} = 4.30$$

Distance (3,5)

$$(5.00, 5.00)\ (1.00, 1.00) = \sqrt{(1.00 - 5.00)^2 + (1.00 - 5.00)^2} = 5.66$$

Distance (3,6)

$$(5.00, 5.00)\ (7.00, 5.00) = \sqrt{(7.00 - 5.00)^2 + (5.00 - 5.00)^2} = 2.00$$

Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (3,7)

$$(5.00, 5.00)\ (5.75, 6.50) = \sqrt{(5.75 - 5.00)^2 + (6.50 - 5.00)^2}$$

$$= 1.68$$

Distance (4,5)

$$(1.50, 2.50)\ (1.00, 1.00) = \sqrt{(1.00 - 1.50)^2 + (1.00 - 2.50)^2} = 1.58$$

Distance (4,6)

$$(1.50, 2.50)\ (7.00, 5.00) = \sqrt{(7.00 - 1.50)^2 + (5.00 - 2.50)^2} = 6.04$$

Distance (4,7)

$$(1.50, 2.50)\ (5.75, 6.50) = \sqrt{(5.75 - 1.50)^2 + (6.50 - 2.50)^2} = 5.84$$

Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (5,6)

$$(1.00, 1.00)\ (7.00, 5.00) = \sqrt{(7.00 - 1.00)^2 + (5.00 - 1.00)^2}$$

$$= 7.21$$

Distance (5,7)

$$(1.00, 1.00)\ (5.75, 6.50) = \sqrt{(5.75 - 1.00)^2 + (6.50 - 1.00)^2} = 7.27$$

Distance (6,7)

$$(7.00, 5.00)\ (5.75, 6.50) = \sqrt{(5.75 - 7.00)^2 + (6.50 - 5.00)^2} = 1.95$$

Distance Matrix

- The distance matrix is-

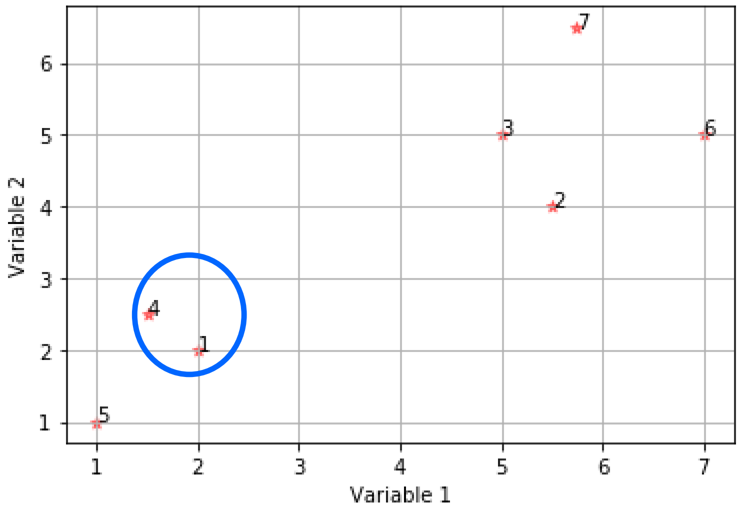
	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

Example for HAC

- Select minimum element to build first cluster formation-

	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

Example for HAC



Example for HAC

- Recalculate distance to update distance matrix

	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

- MIN[dist(1,4), 2] = MIN(dist(1,2), (4,2))
  = MIN(4.0, 4.3) = 4.0
- MIN[dist(1,4), 3] = MIN(dist(1,3), (4,3))
  = MIN(4.2, 4.3) = 4.2
- MIN[dist(1,4), 5] = MIN(dist(1,5), (4,5)) = MIN(1.4, 1.6) = 1.4
- MIN[dist(1,4), 6] = MIN(dist(1,6), (4,6)) = MIN(5.8, 6.0) = 5.8
- MIN[dist(1,4), 7] = MIN(dist(1,7), (4,7)) = MIN(5.9, 5.8) = 5.8

Example for HAC

- Updated distance matrix for the cluster (1, 4)

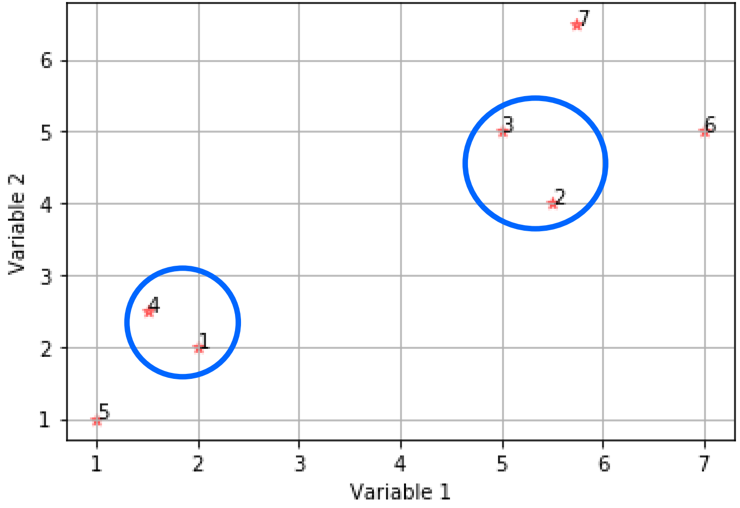
	1,4	**2**	**3**	**5**	**6**	**7**
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

Example for HAC

- Select minimum element to build next cluster formation-

	1,4	2	3	5	6	7
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

Example for HAC



Example for HAC

	1,4	2	3	5	6	7
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

- Recalculate distance to update distance matrix

  - MIN[dist(2,3), (1,4)] = MIN(dist(2,(1,4)), (3,(1,4))
    = MIN(4.0, 4.2) = 4.0
  - MIN[dist(2,3), 5] = MIN(dist(2,5), (3,5)) = MIN(5.4, 5.7) = 5.4
  - MIN[dist(2,3), 6] = MIN(dist(2,6), (3,6)) = MIN(1.8, 2.0) = 1.8
  - MIN[dist(2,3), 7] = MIN(dist(2,7), (3,7)) = MIN(2.5, 1.7) = 1.7

Example for HAC

- Updated distance matrix for the cluster (2, 3)

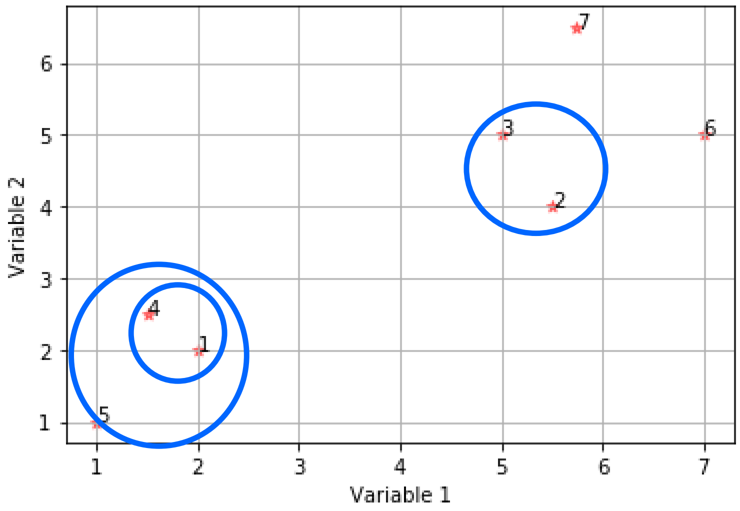
	1,4	2,3	5	6	7
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

Example for HAC

- Select minimum element to build next cluster formation-

	1,4	**2,3**	**5**	**6**	**7**
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

Example for HAC



Example for HAC

- Recalculate distance to update distance matrix

	1,4	2,3	5	6	7
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

- MIN[dist((1,4),5), (2,3)] = MIN(dist((1,4),(2,3)), (5,(2,3))

= MIN(4.0, 5.4) = 4.0

- MIN[dist((1,4),5), 6] = MIN(dist((1,4),6), (5,6)) = MIN(5.8, 7.2) = 5.8

- MIN[dist((1,4),5), 7] = MIN(dist((1,4),7), (5,7)) = MIN(5.8, 7.3) = 5.8

Example for HAC

- Updated distance matrix for the cluster ((1,4), 5)

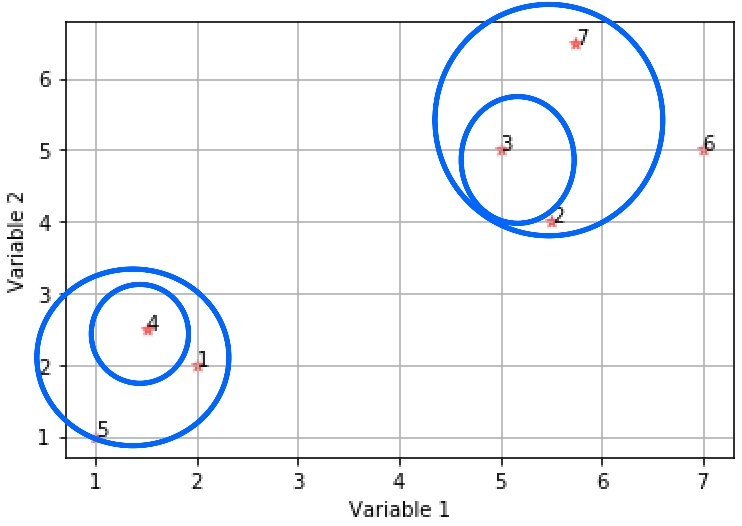
	1,4,5	**2,3**	**6**	**7**
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

Example for HAC

- Select minimum element to build next cluster formation-

	1,4,5	2,3	6	7
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

Example for HAC



Example for HAC

- Recalculate distance to update distance matrix

	1,4,5	2,3	6	7
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

- MIN[dist((2,3),7), (1,4,5)] = MIN(dist((2,3),(1,4,5)), (7,(1,4,5))
= MIN(4.0, 5.8) = 4.0

- MIN[dist((2,3),7), 6] = MIN(dist((2,3),6), (7,6)) = MIN(1.8, 2.0) = 1.8

Example for HAC

- Updated distance matrix for the cluster ((2,3), 7)

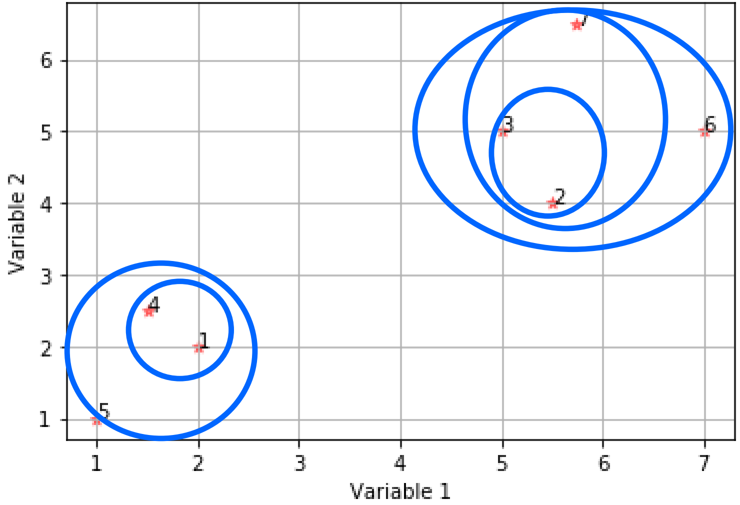
	1,4,5	**2,3,7**	**6**
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

Example for HAC

- Select minimum element to build next cluster formation-

	1,4,5	**2,3,7**	**6**
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

Example for HAC



Example for HAC

- Recalculate distance to update distance matrix

	1,4,5	2,3,7	6
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

- MIN[dist((2,3,7),6), (1,4,5)] = MIN(dist((2,3,7),(1,4,5)), (6,(1,4,5))

= MIN(4.0, 5.8)

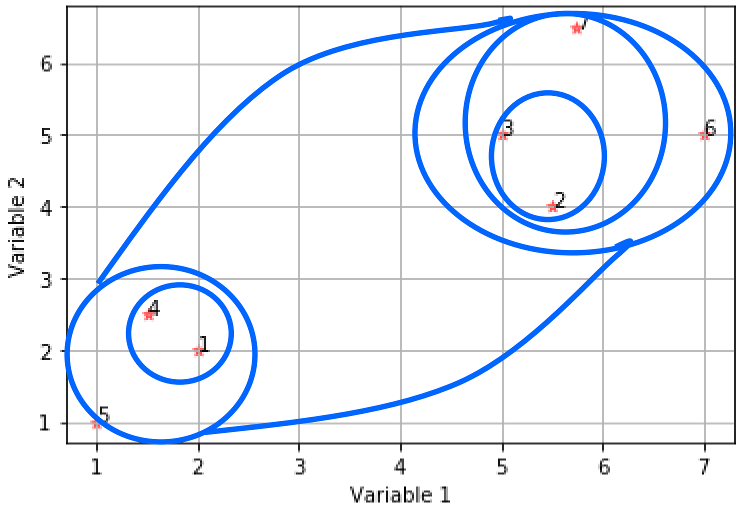
= 4.0

Example for HAC

- Updated distance matrix for the cluster ((2,3,7), 6)

	1,4,5	**2,3,7,6**
1,4,5	0.0	
2,3,7,6	4.0	0.0

Example for HAC



Python demo for HAC

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from scipy.cluster.hierarchy import fcluster
from scipy.cluster.hierarchy import cophenet
from scipy.spatial.distance import pdist
```

In [2]:
```
data = pd.read_excel("hierarchical_clustering.xlsx")
data
```

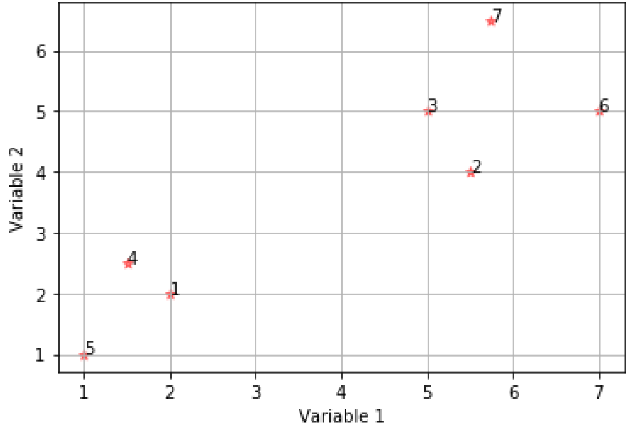
Out[2]:

	Variable 1	Variable 2
0	2.00	2.0
1	5.50	4.0
2	5.00	5.0
3	1.50	2.5
4	1.00	1.0
5	7.00	5.0
6	5.75	6.5

Python demo for HAC

In [3]:
```
x = data['Variable 1']
y = data['Variable 2']
n = range(1,8)

fig, ax = plt.subplots()
ax.scatter(x, y, marker='*', c='red', alpha=0.5)
plt.grid()
plt.xlabel("Variable 1")
plt.ylabel("Variable 2")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
```



Python demo for HAC

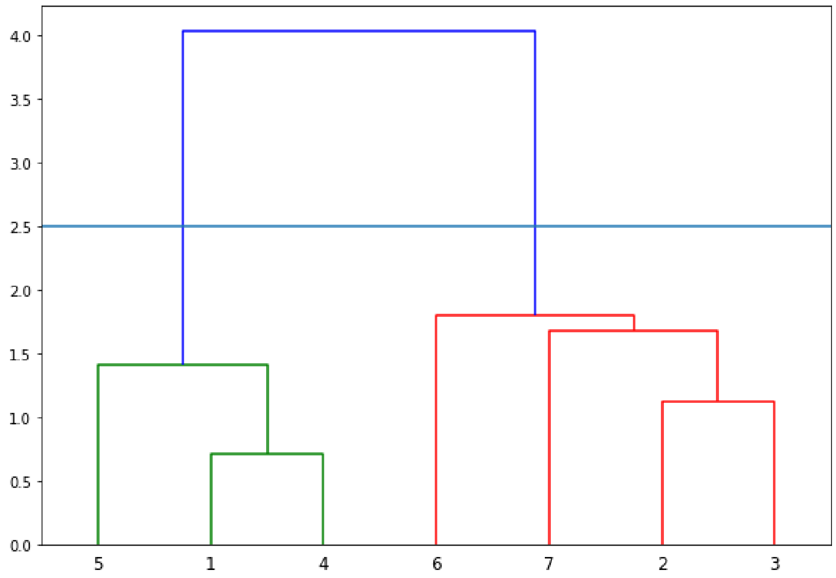
In [4]:

```python
from scipy.cluster.hierarchy import dendrogram, linkage

linked = linkage(data, 'single')

labelList = range(1, 8)

plt.figure(figsize=(10, 7))
dendrogram(linked,
           orientation='top',
           labels=labelList,
           distance_sort='descending',
           show_leaf_counts=True)
plt.axhline(y=2.5)
plt.show()
```



Python demo for HAC

In [5]:
```
import sklearn
from sklearn.cluster import AgglomerativeClustering


k=2
Hclustering = AgglomerativeClustering(n_clusters = k, affinity = 'euclidean', linkage = 'single')
Hclustering.fit(data)
```

K=3

Out[5]:
```
AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                        connectivity=None, distance_threshold=None,
                        linkage='single', memory=None, n_clusters=2,
                        pooling_func='deprecated')
```

In [6]:
```
Hclustering.fit_predict(data)
```

Out[6]:
```
array([1, 0, 0, 1, 1, 0, 0], dtype=int64)
```

In [7]:
```
print(Hclustering.labels_)
```

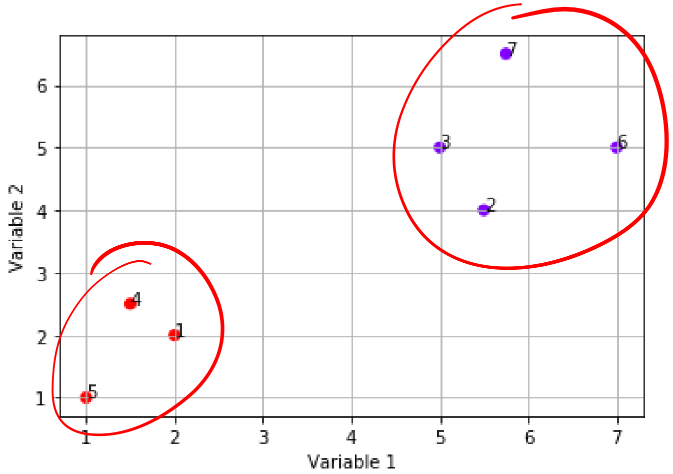
```
[1 0 0 1 1 0 0]
```

Python demo for HAC

In [8]:

```
x = data['Variable 1']
y = data['Variable 2']
n = range(1,8)

fig, ax = plt.subplots()
ax.scatter(x, y, c=Hclustering.labels_, cmap='rainbow')
plt.grid()
plt.xlabel("Variable 1")
plt.ylabel("Variable 2")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
```



THANK YOU