Cluster Analysis

Data Sets: Wholesale customers data and USArrests

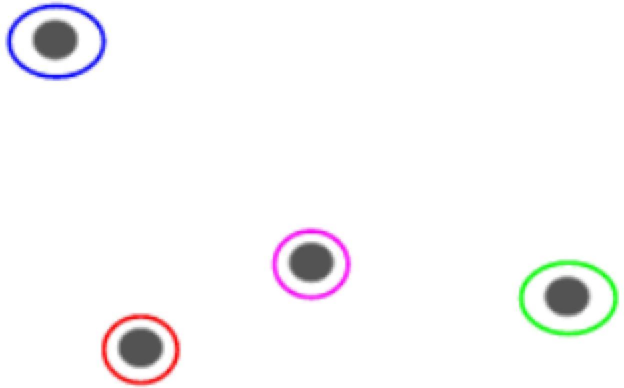
Hierarchical Clustering

What is Hierarchical Clustering?

Let's say we have the below points and we want to cluster them into groups:

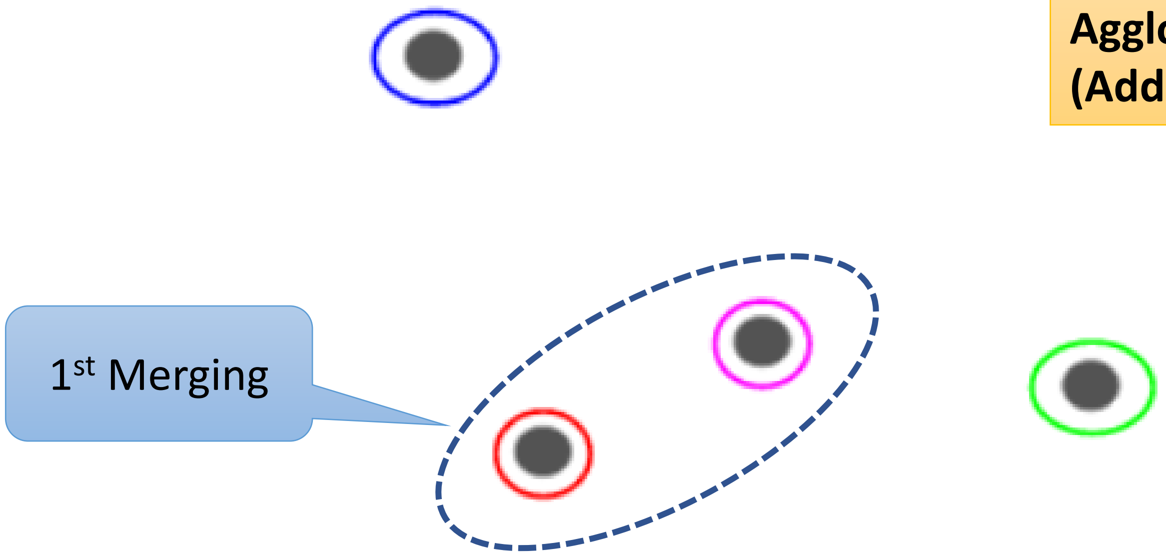
Each data point as one cluster

We can assign each of these points to a separate cluster:



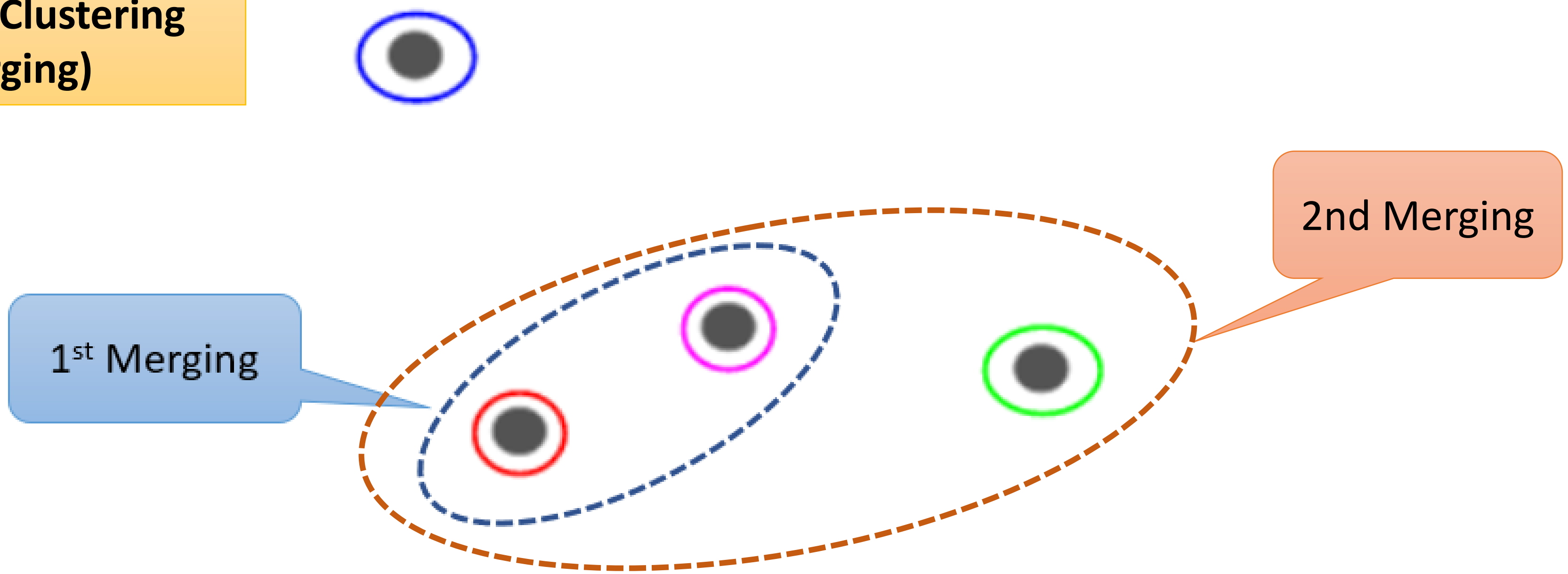
1^{st} merging of two nearest clusters/(data points)

Agglomerative Clustering (Adding or Merging)



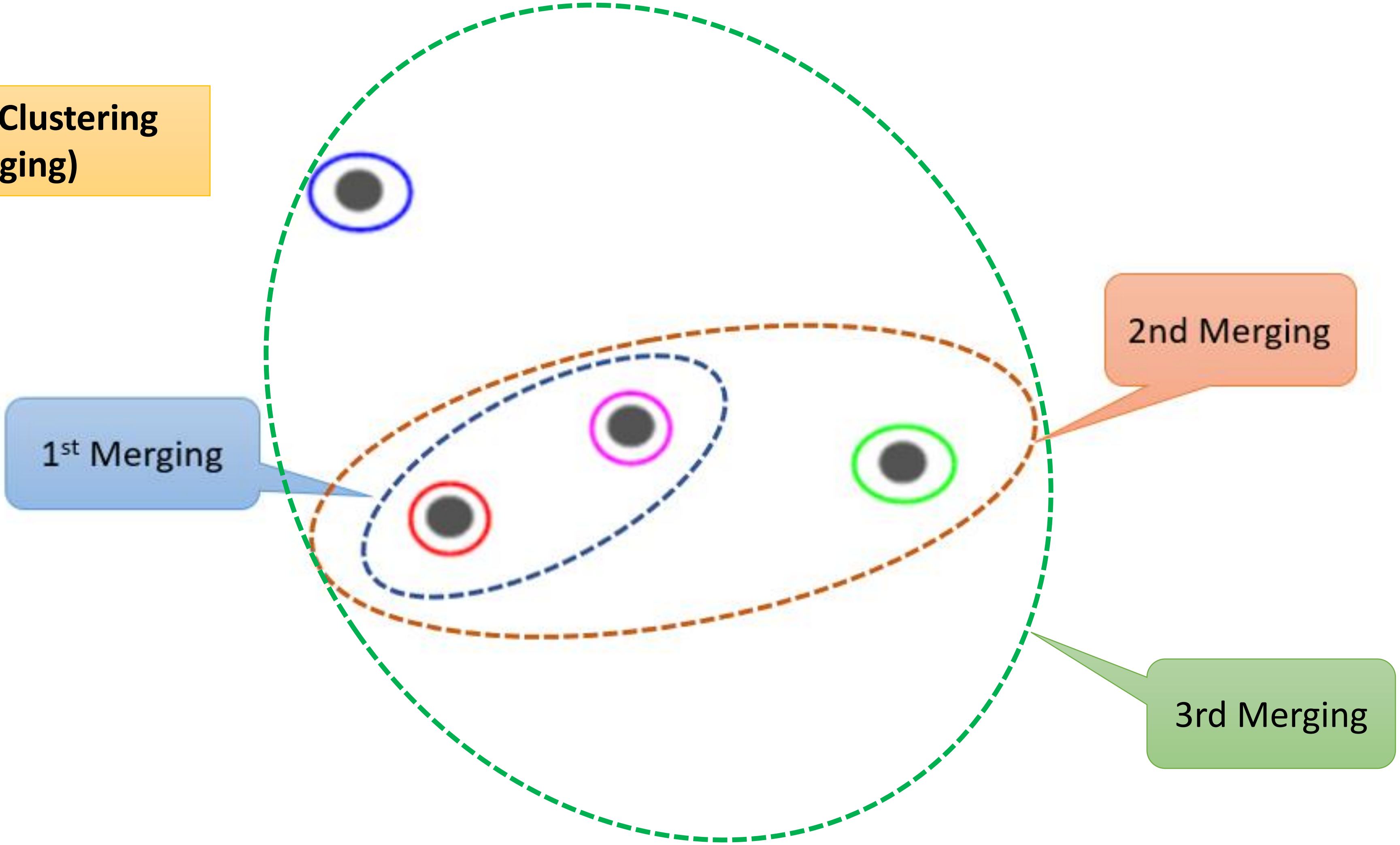
2nd merging of two nearest clusters/(data points)

Agglomerative Clustering
(Adding or Merging)



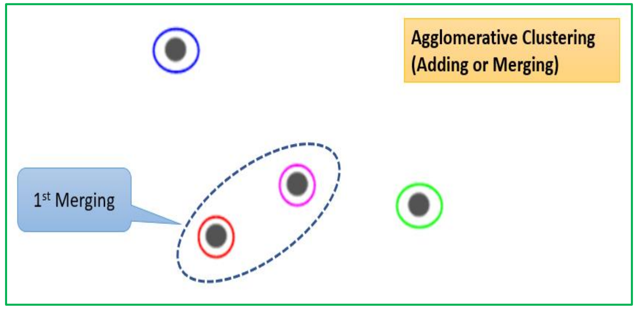
3rd merging of two nearest clusters/(data points)

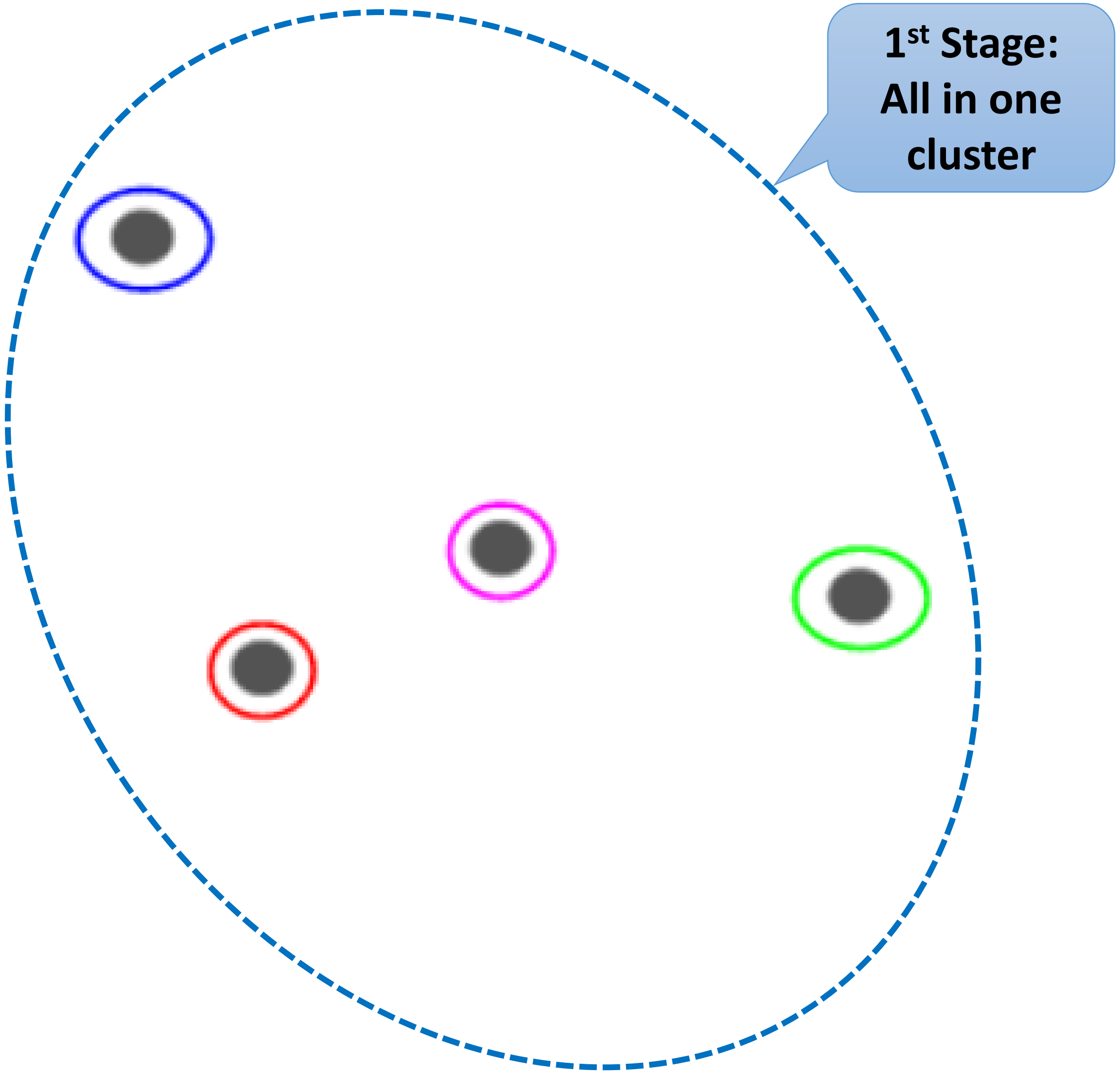
Agglomerative Clustering (Adding or Merging)



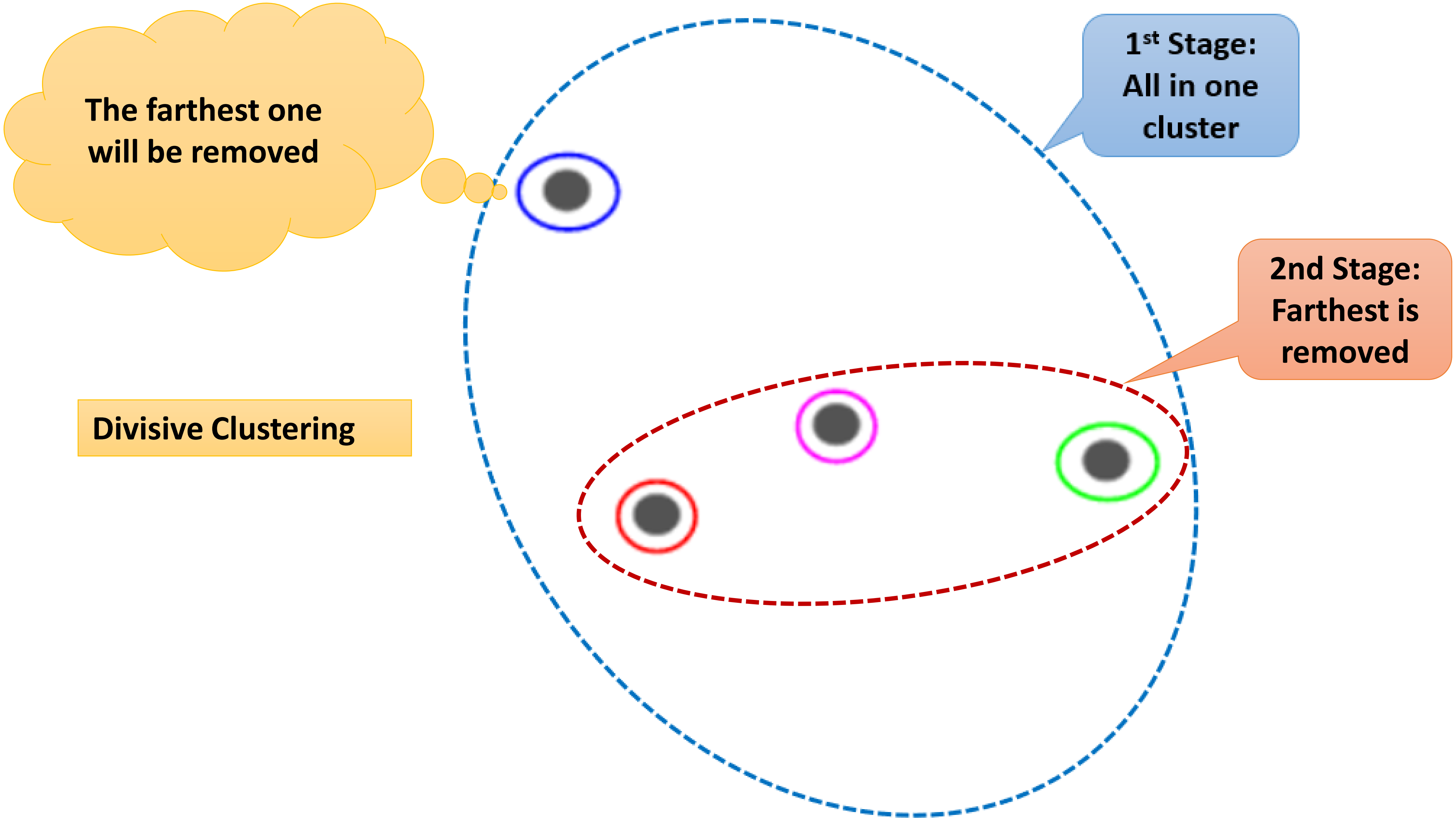
Divisive Hierarchical Clustering

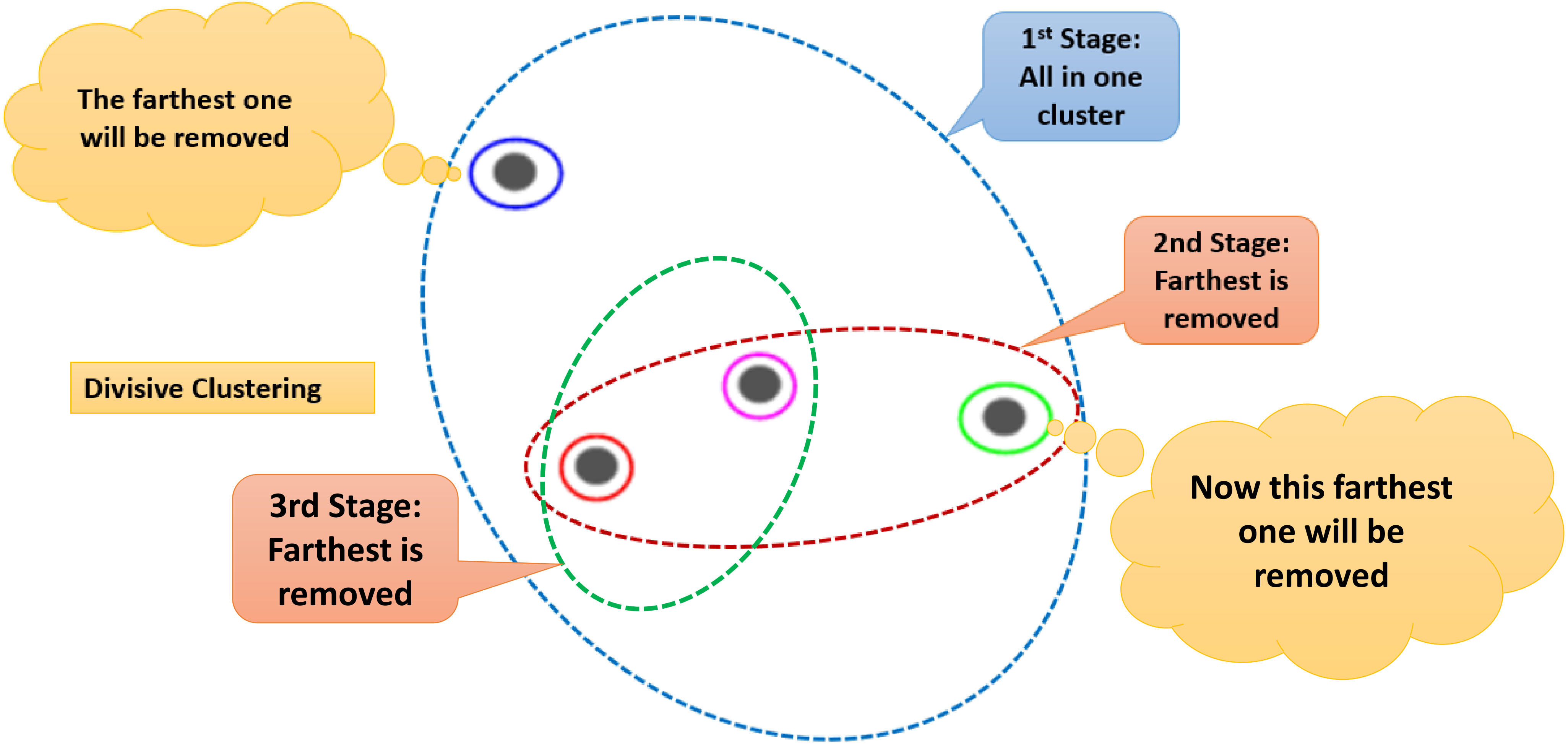
Divisive Clustering
(All in one cluster)

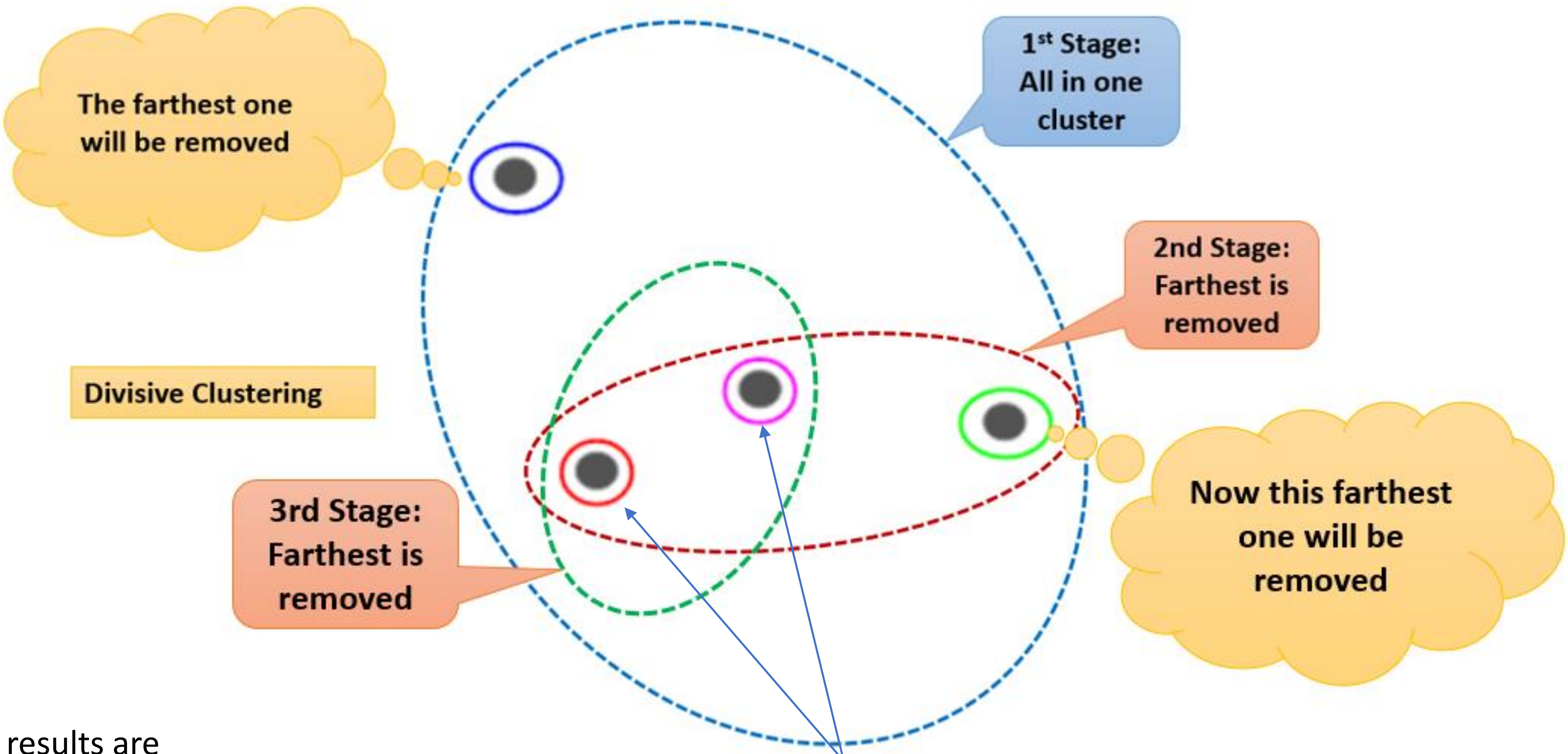




Divisive Clustering




The farthest one will be removed
1st Stage: All in one cluster
2nd Stage: Farthest is removed
Divisive Clustering
Now this farthest one will be removed
3rd Stage: Farthest is removed



Finally, these 2 will be identified as separate clusters

80% results are produced by 10% persons [90 % are game watchers]

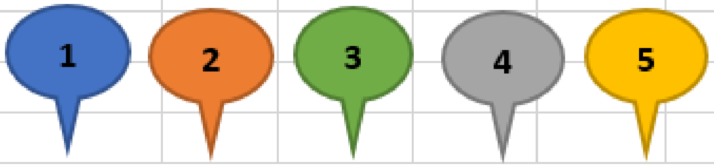
Agglomerative Clustering (Adding or Merging)

ID	Marks
1	10
2	7
3	28
4	20
5	35

Stage 1: Five clusters

Distance Matrix

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

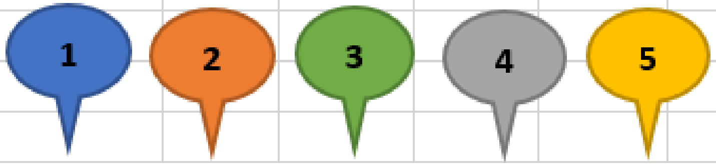


As 1 & 2 have smallest distance, so merging them

ID	Marks
1	10
2	7
3	28
4	20
5	35

Stage 1: Five clusters

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0



As 1 & 2 have smallest distance, so merging them

ID	Marks	
1-2	10	max of 10 & 7
3	28	
4	20	
5	35	

Stage 2: Four clusters

ID	1-2	3	4	5
1-2	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

As 3 & 5 have smallest distance, so merging them

ID	Marks	
1-2	10	max of 10 & 7
3	28	
4	20	
5	35	

Stage 2: Four clusters

ID	1-2	3	4	5
1-2	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	**7**	15	0

As 3 & 5 have smallest distance, so merging them

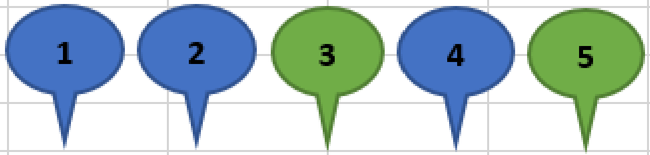
ID	Marks	
1-2	10	
3-5	35	max of 28 & 35
4	20	

25

Stage 3: Three clusters

ID	1-2	3-5	4
1-2	0	15	10
3-5	15	0	15
4	**10**	15	0

As 1-2 & 4 have smallest distance, so merging them

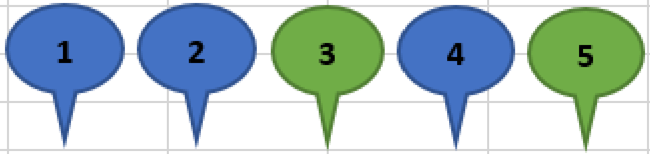


ID	Marks	
1-2	10	
3-5	35	max of 28 & 35
4	20	

Stage 3: Three clusters

ID	1-2	3-5	4
1-2	0	15	10
3-5	15	0	15
4	**10**	15	0

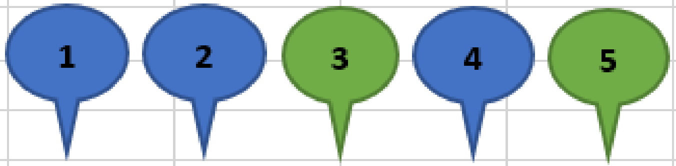
As 1-2 & 4 have smallest distance, so merging them



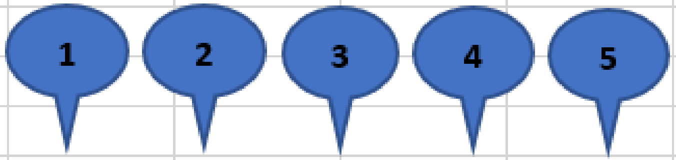
ID	Marks	
1-2-4	20	max of 10 & 20
3-5	35	max of 28 & 35

Stage 4: Two clusters

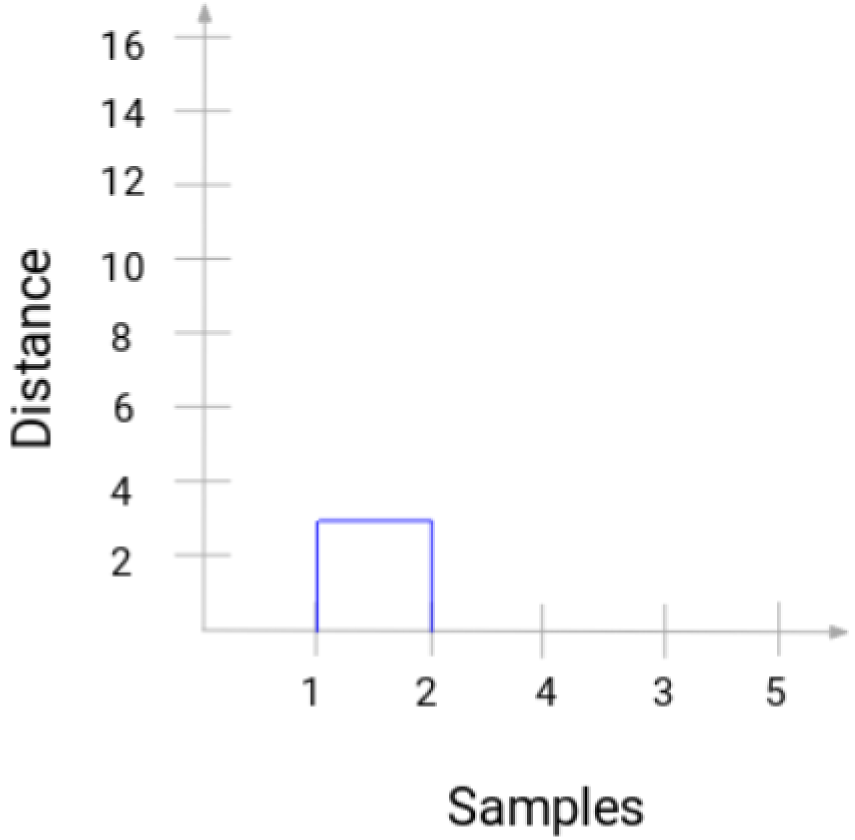
ID	1-2-4	3-5
1-2-4	0	15
3-5	**15**	0



As only 2 clusters remained, we merge them



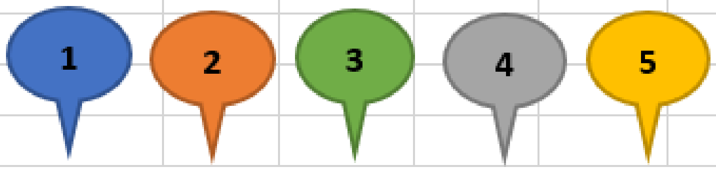
Dendrogram



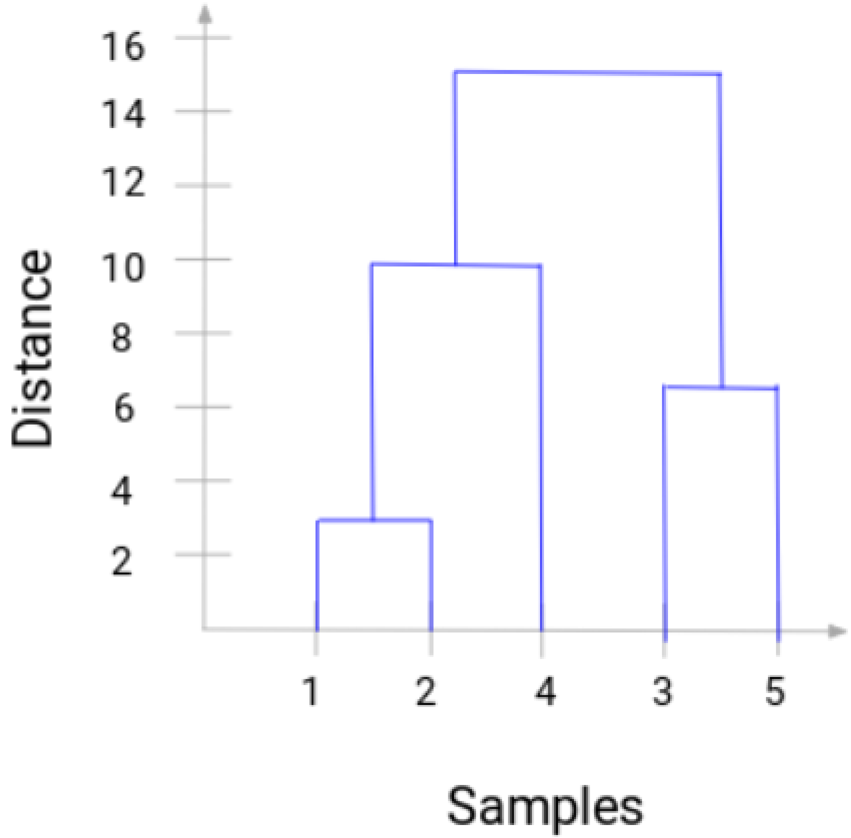
ID	Marks
1	10
2	7
3	28
4	20
5	35

Stage 1: Five clusters

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0



As 1 & 2 have smallest distance, so merging them

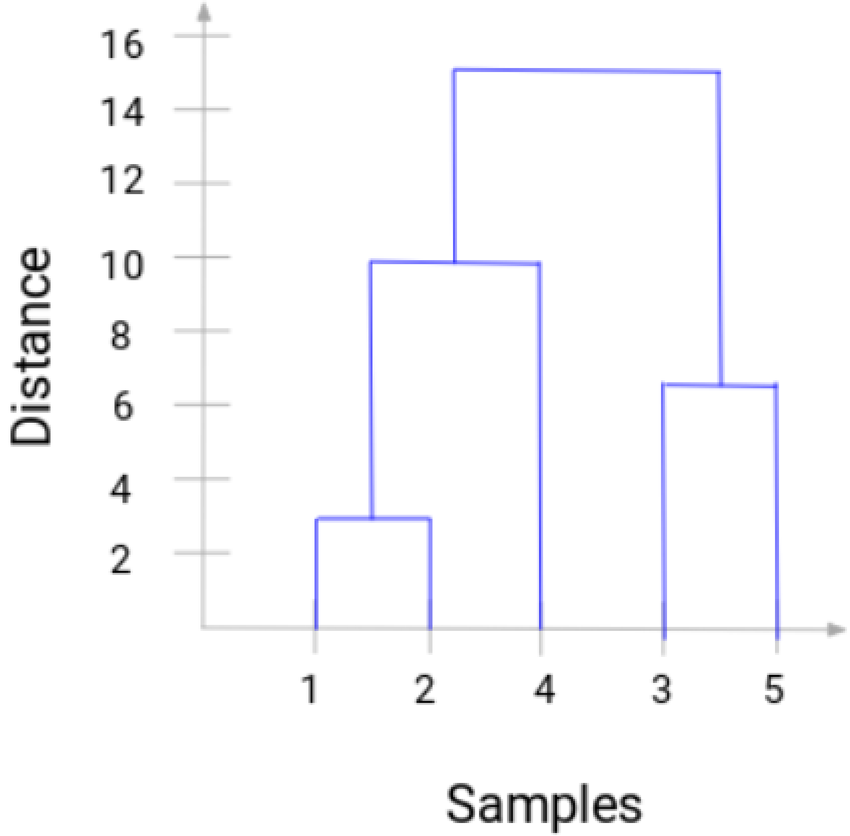


ID	Marks
1-2	10
3	28
4	20
5	35

Stage 2: Four clusters

ID	1-2	3	4	5
1-2	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

As 3 & 5 have smallest distance, so merging them

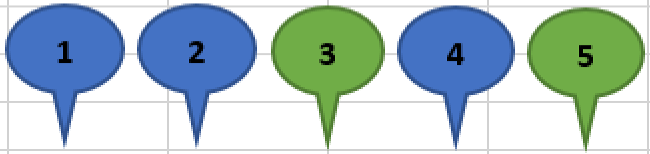


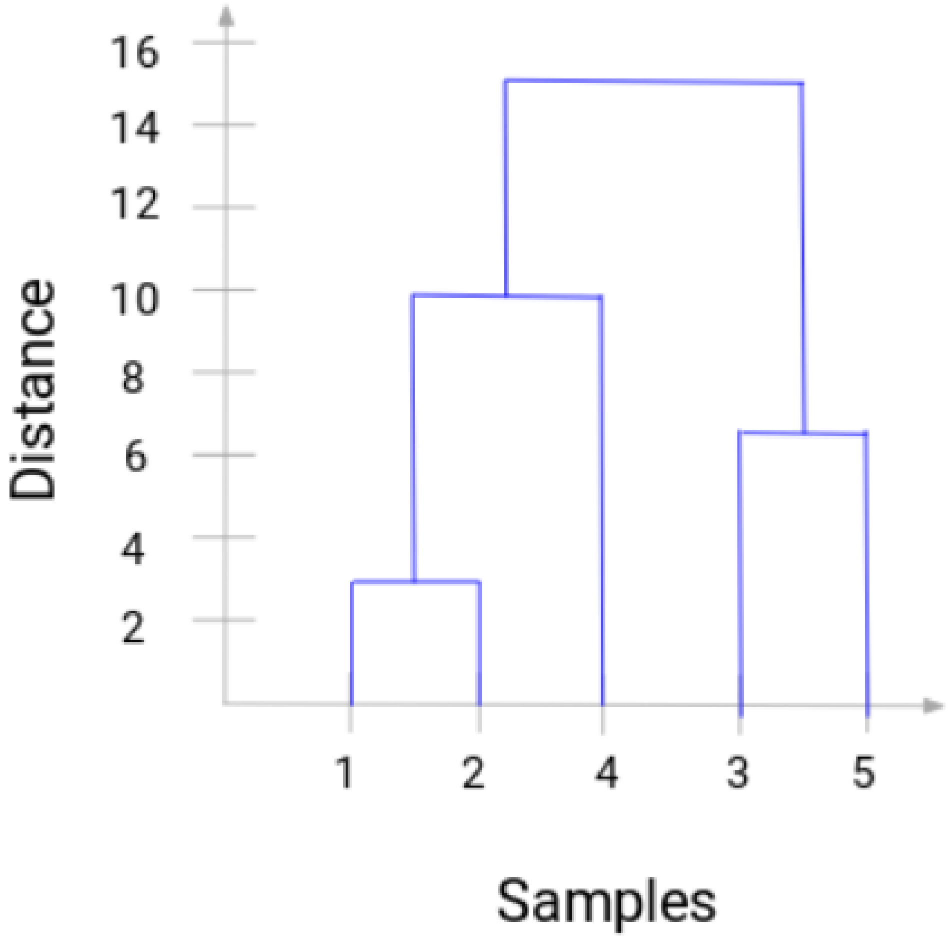
ID	Marks	
1-2	10	
3-5	35	max of 28 & 35
4	20	

Stage 3: Three clusters

ID	1-2	3-5	4
1-2	0	15	10
3-5	15	0	15
4	10	15	0

As 1-2 & 4 have smallest distance, so merging them

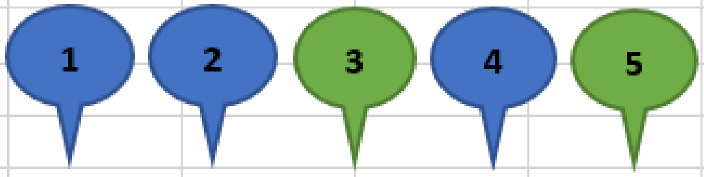




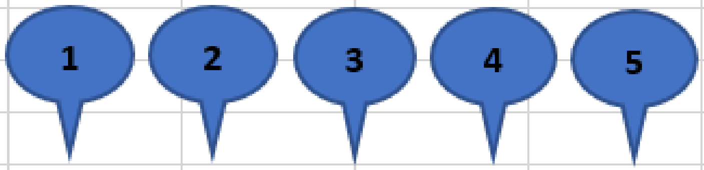
ID	Marks	
1-2-4	20	max of 10 & 20
3-5	35	max of 28 & 35

Stage 4: Two clusters

ID	1-2-4	3-5
1-2-4	0	15
3-5	15	0

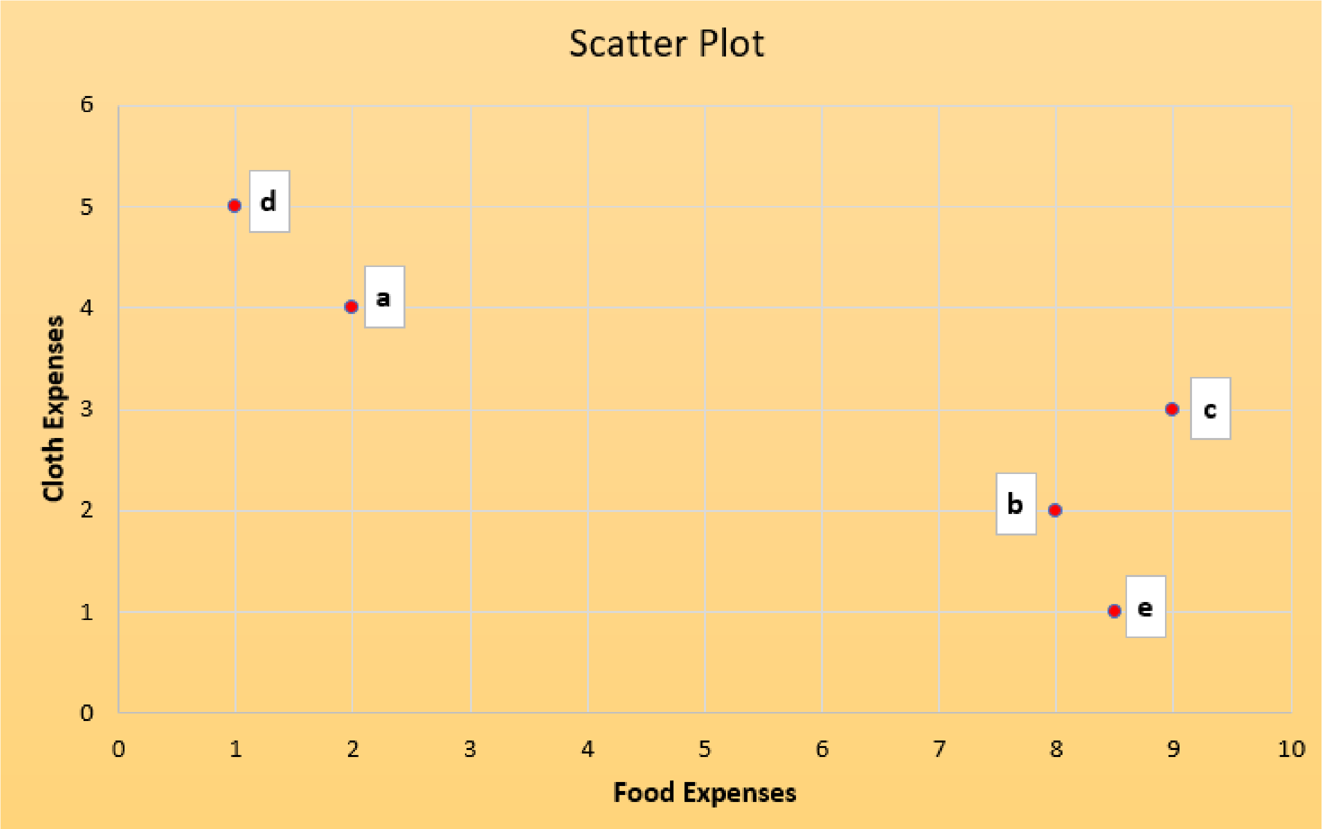


As only 2 clusters remained, we merge them

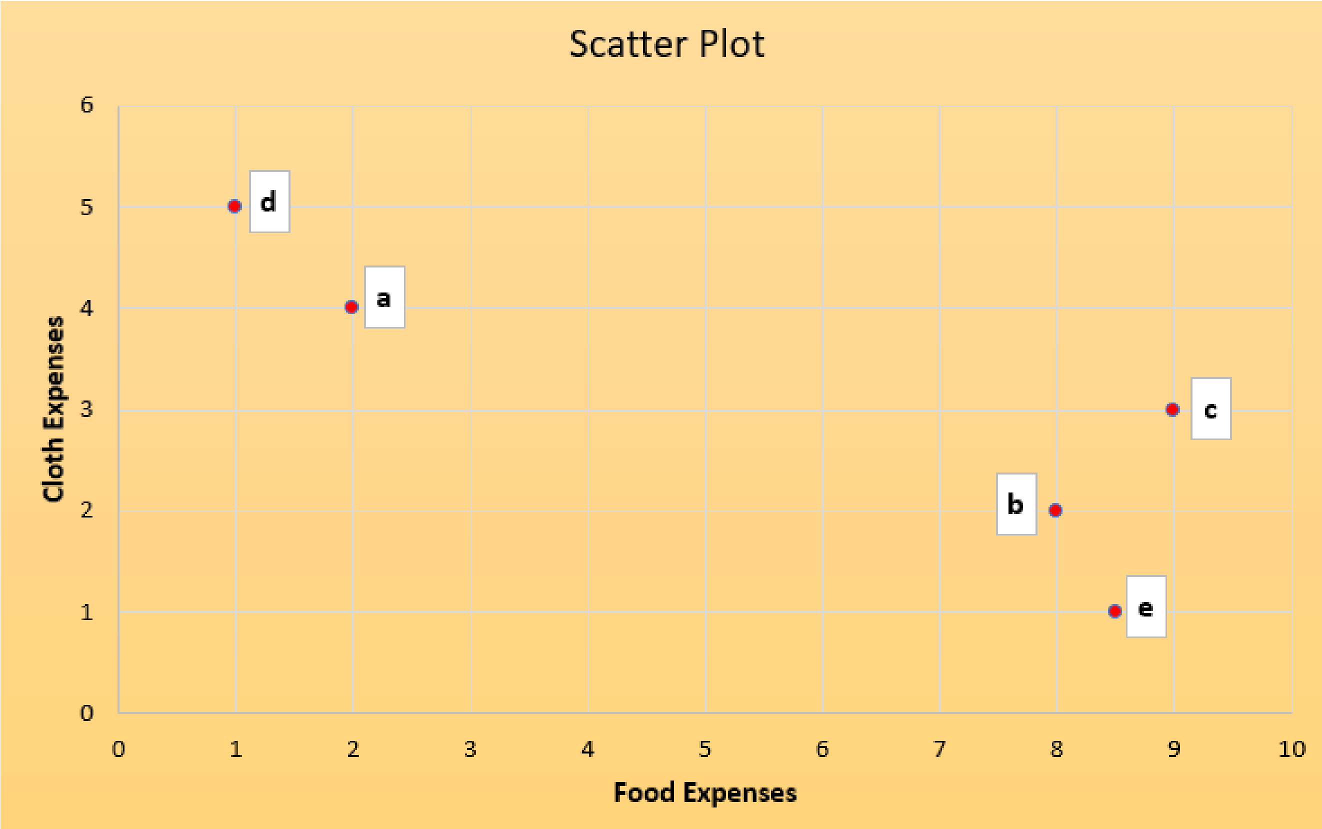


Data with X and Y: Single Linkage (Min)

	Food Exp	Cloth Exp
	X	Y
a	2	4
b	8	2
c	9	3
d	1	5
e	8.5	1

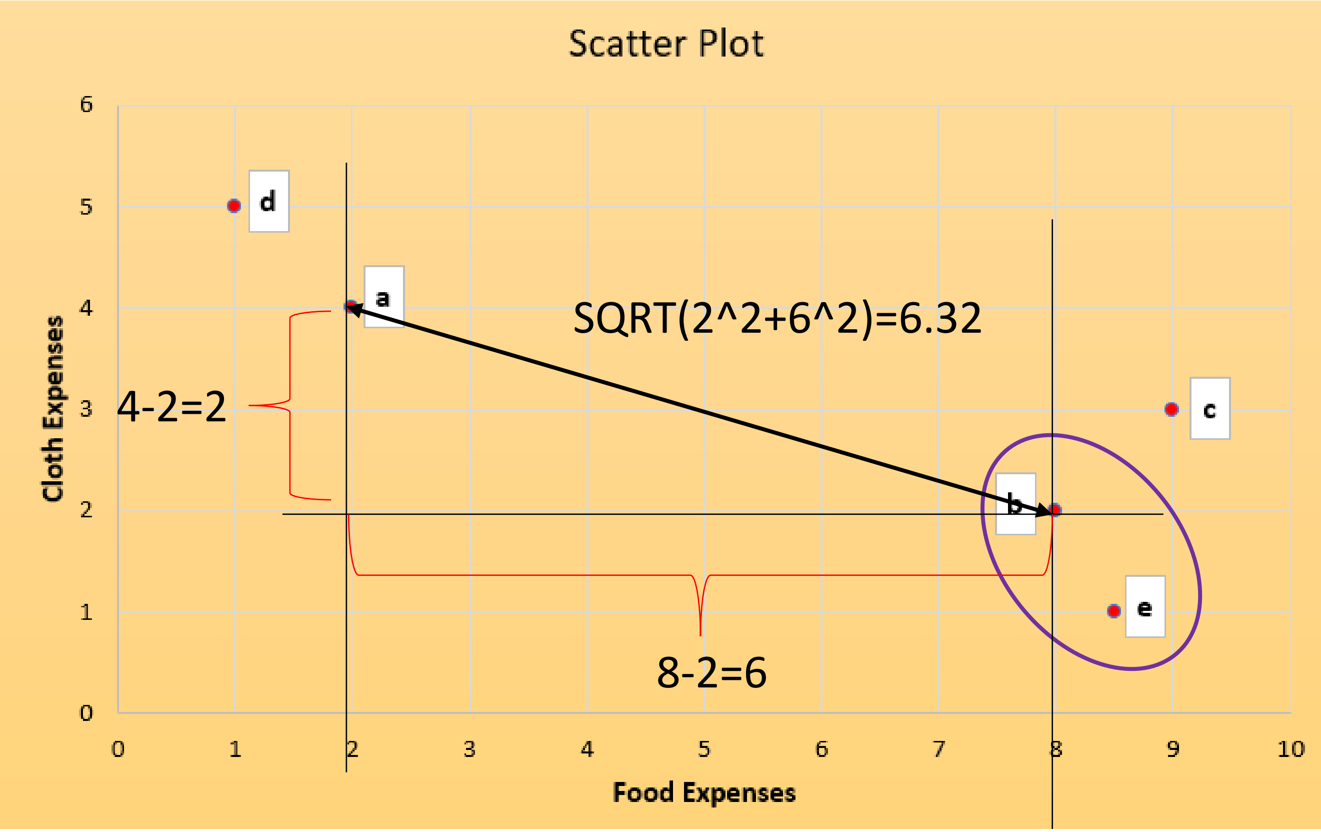


Give a try….make agglomerative process intuitively!



Stage 1: b and e can be clubbed

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00



Now *b* and *e* are in one cluster. How their values will be treated for finding DIST vis-à-vis another point/s?

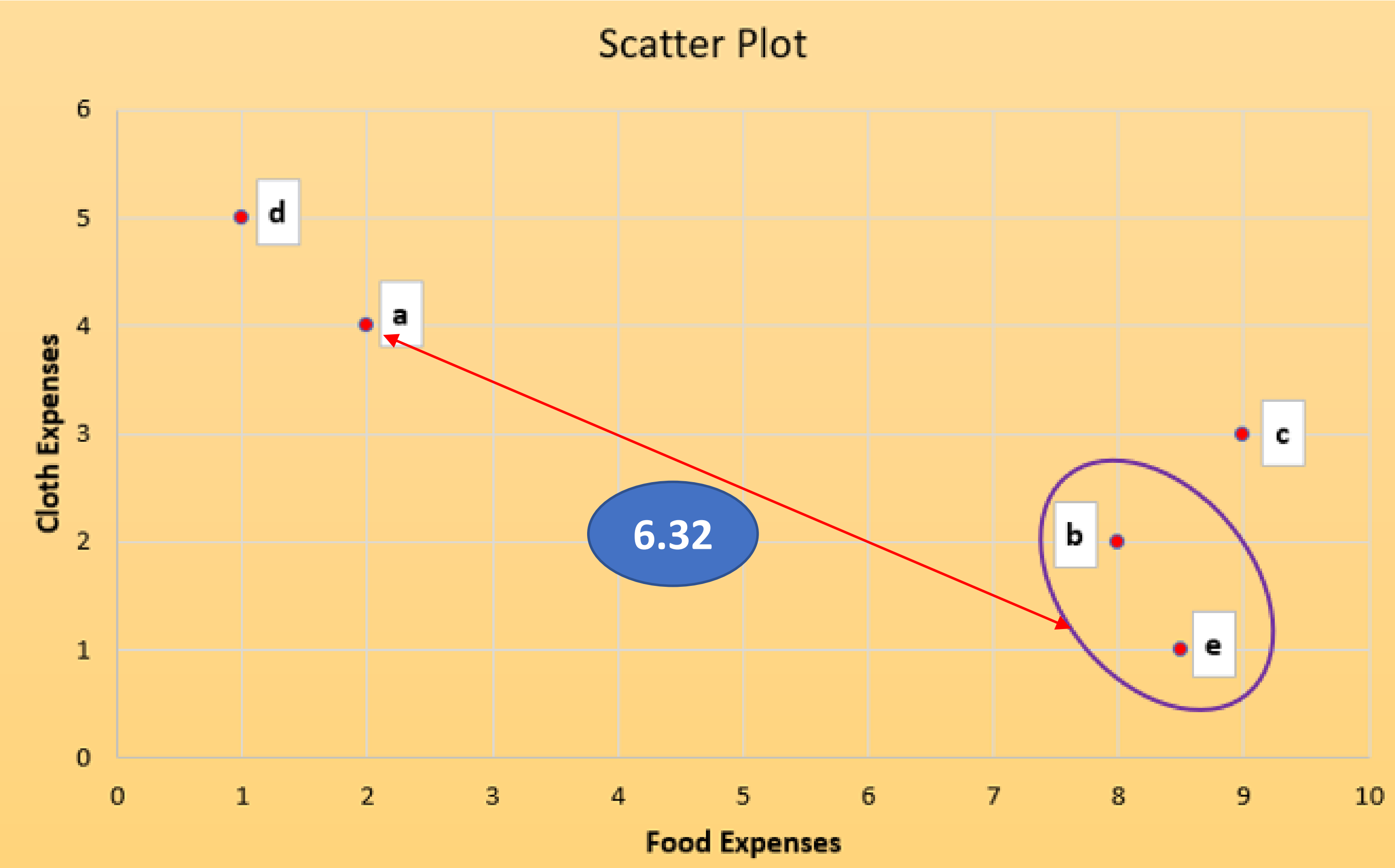
Distance between b, e and a

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

$$D(be, a) = min\{D(b, a), D(e, a)\}$$

$$D(be, a) = min\{6.32, 7.16\}$$

$$D(be, a) = \mathbf{6.32}$$



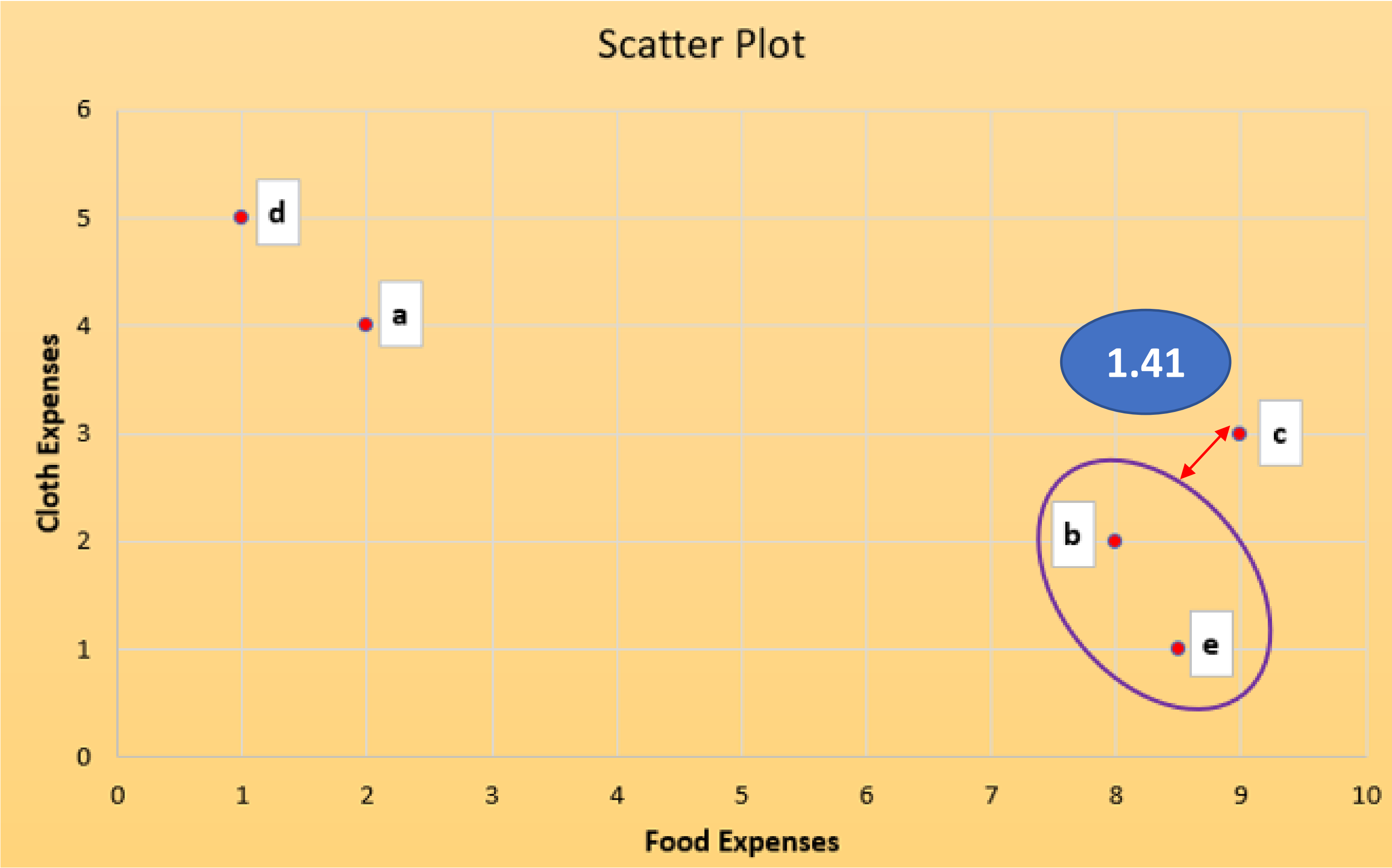
Distance between b, e and c

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

$$D(be,c) = min\{D(b,c), D(e,c)\}$$

$$D(be,c) = min\{1.41, 2.06\}$$

$$D(be,c) = 1.41$$



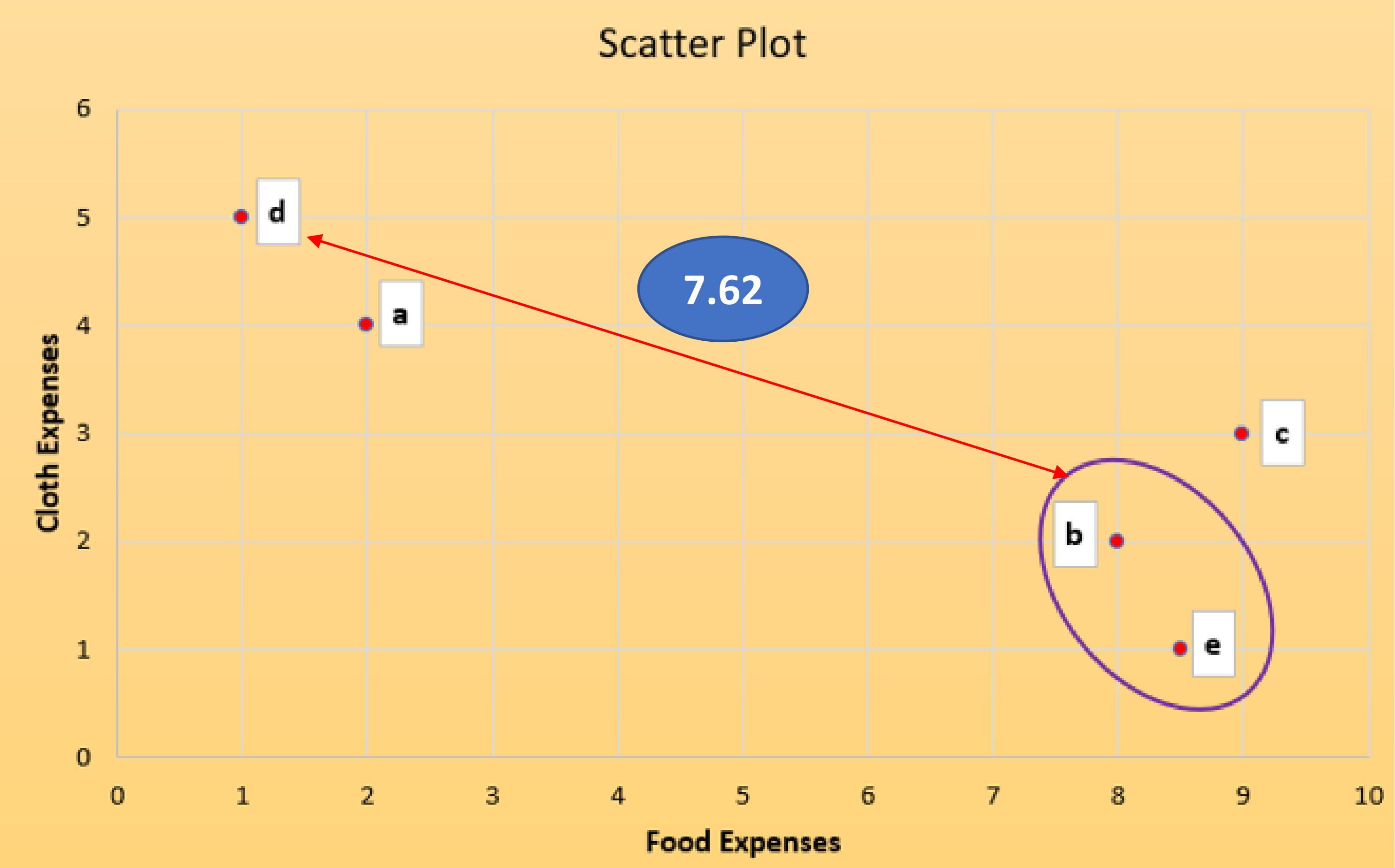
Distance between b, e and d

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

$$D(be, d) = min\{D(b, d), D(e, d)\}$$

$$D(be, d) = min\{7.62, 8.50\}$$

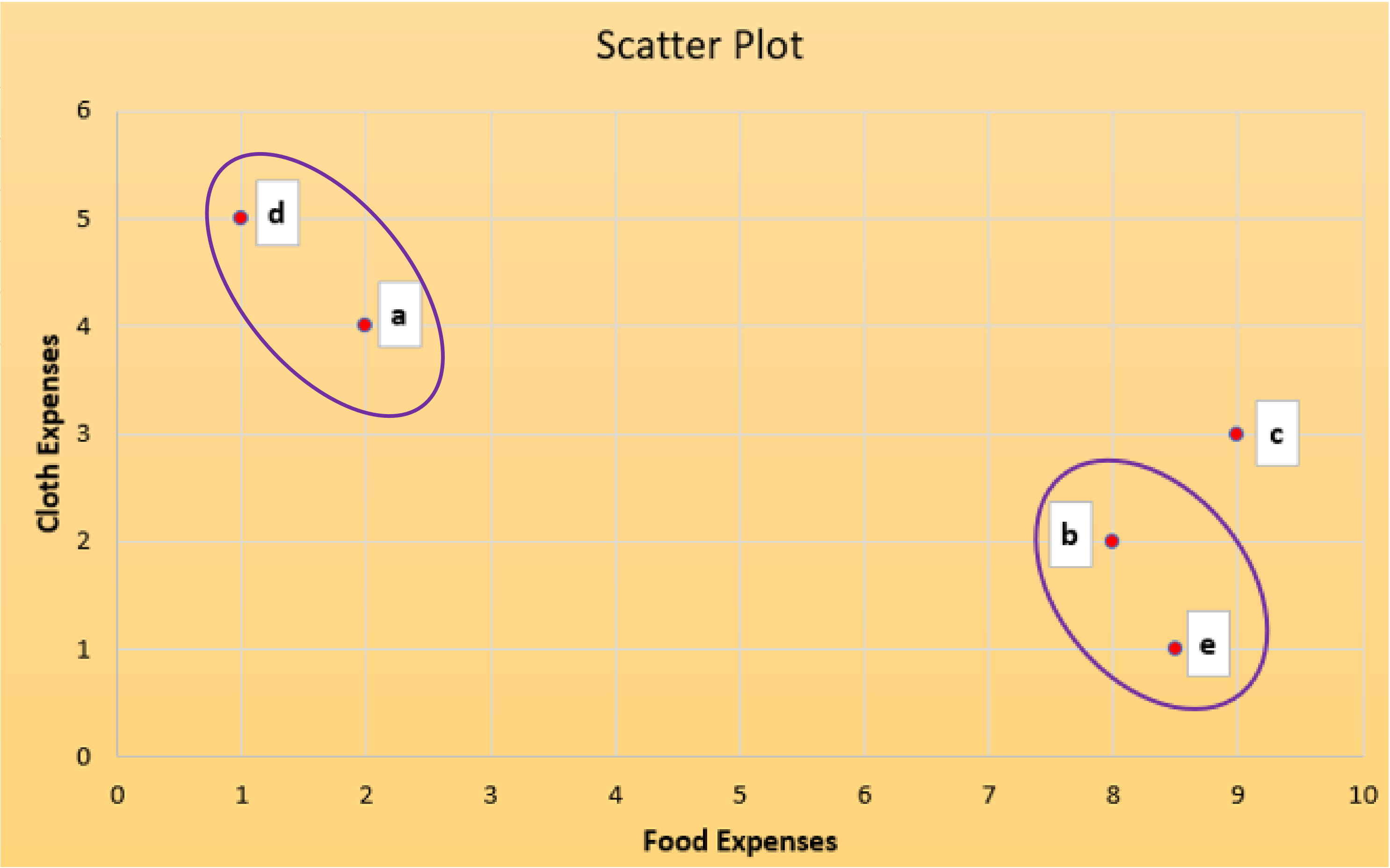
$$D(be, d) = 7.62$$



Stage 2: d and a can be clubbed

	b e	a	c	d
b e	0	6.32	1.41	7.62
a		0	7.07	1.41
c			0	8.25
d				0

Distance between clusters (b, e) and (a, d)?



Distance between clusters (b, e) and (a, d)?

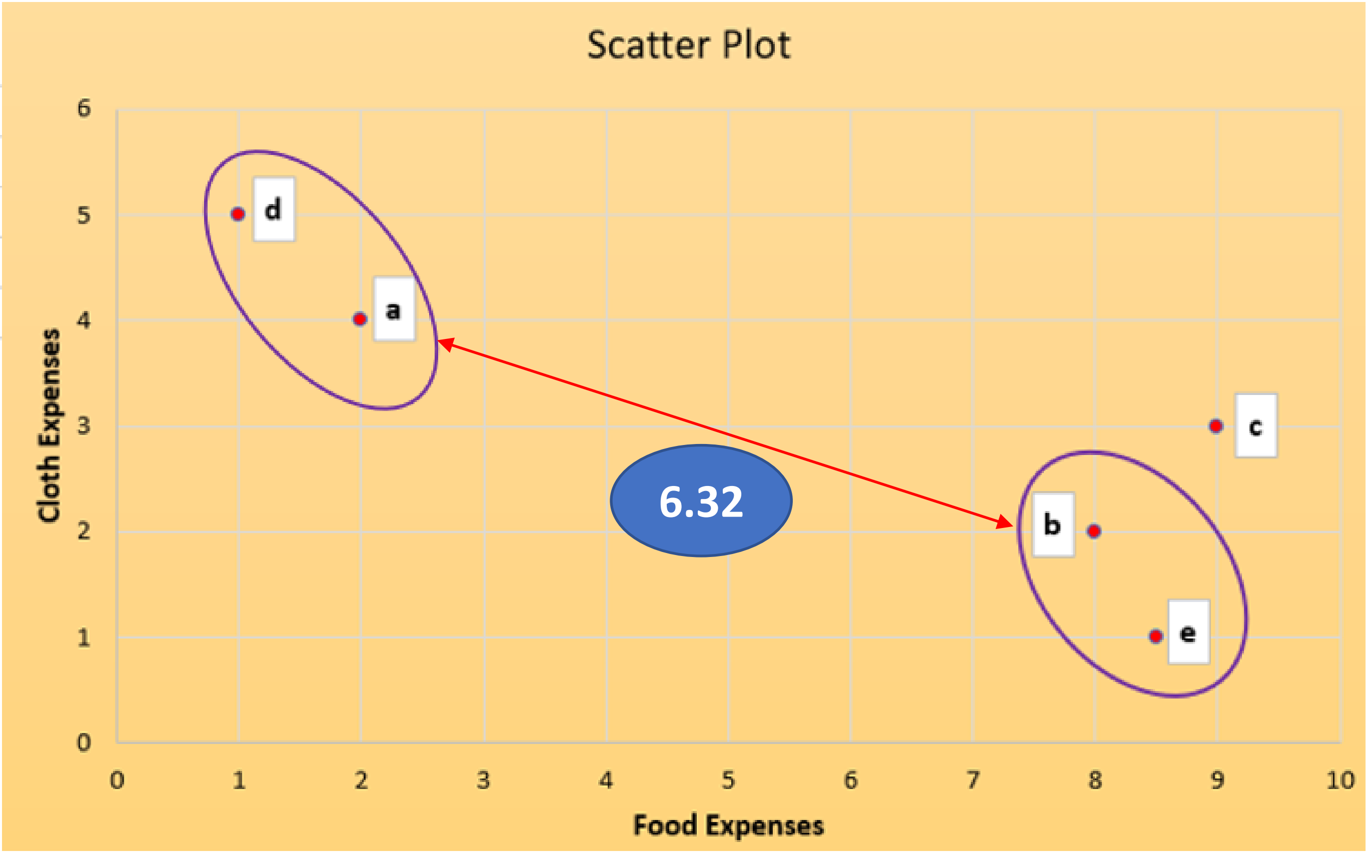
	b e	a	c	d
b e	0	6.32	1.41	7.62
a		0	7.07	1.41
c			0	8.25
d				0

$$D(be, ad) = min\{D(be, a), D(be, d)\}$$

$$D(be, ad) = min\{6.32, 7.62\}$$

$$D(be, ad) = 6.32$$

	b e	a d	c
b e	0	6.32	1.41
a d		0	7.07
c			0



Distance between clusters (a, d) and (c)?

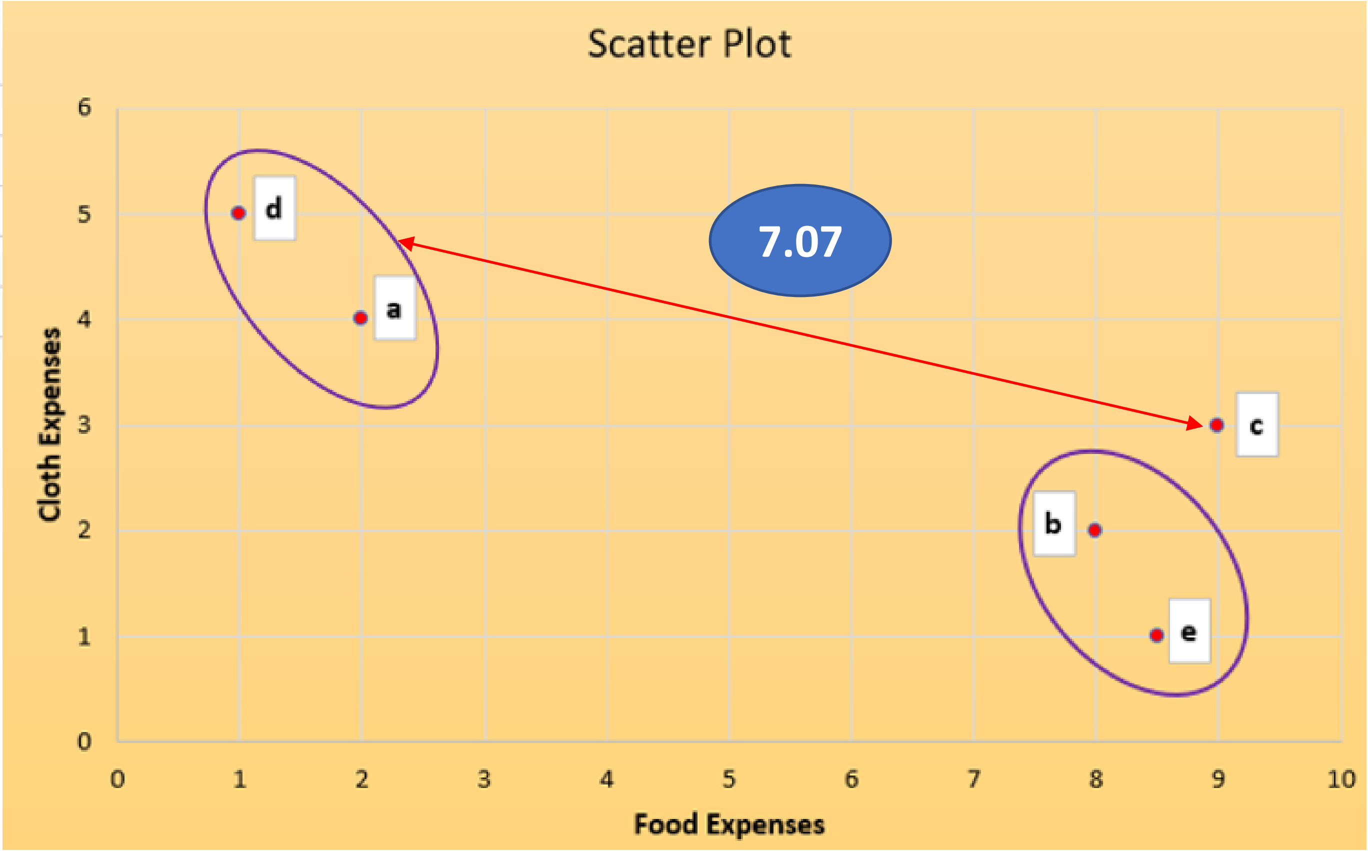
	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

	b e	a d	c
b e	0	6.32	1.41
a d		0	7.07
c			0

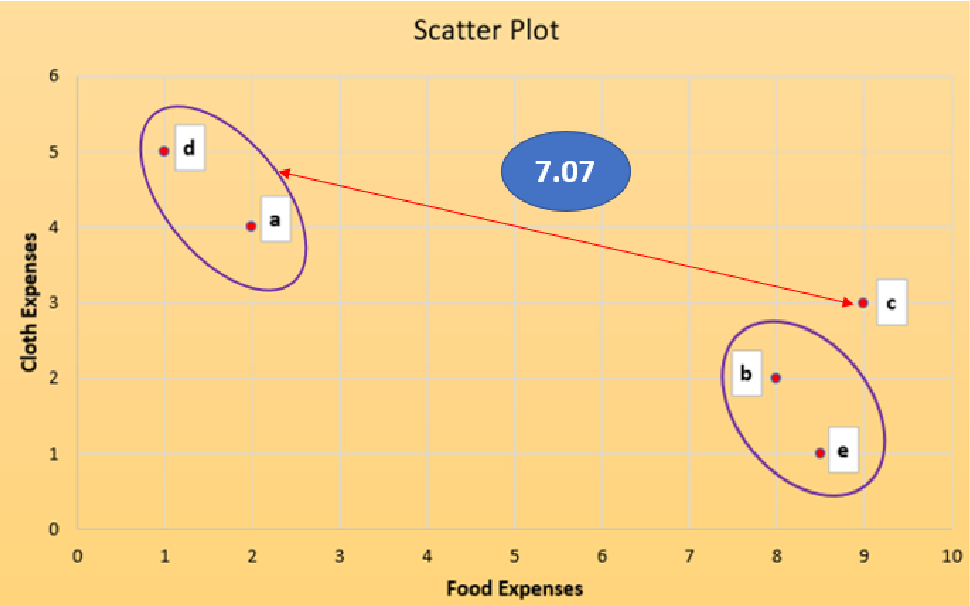
$$D(ad,c) = min\{D(a,c), D(d,c)\}$$

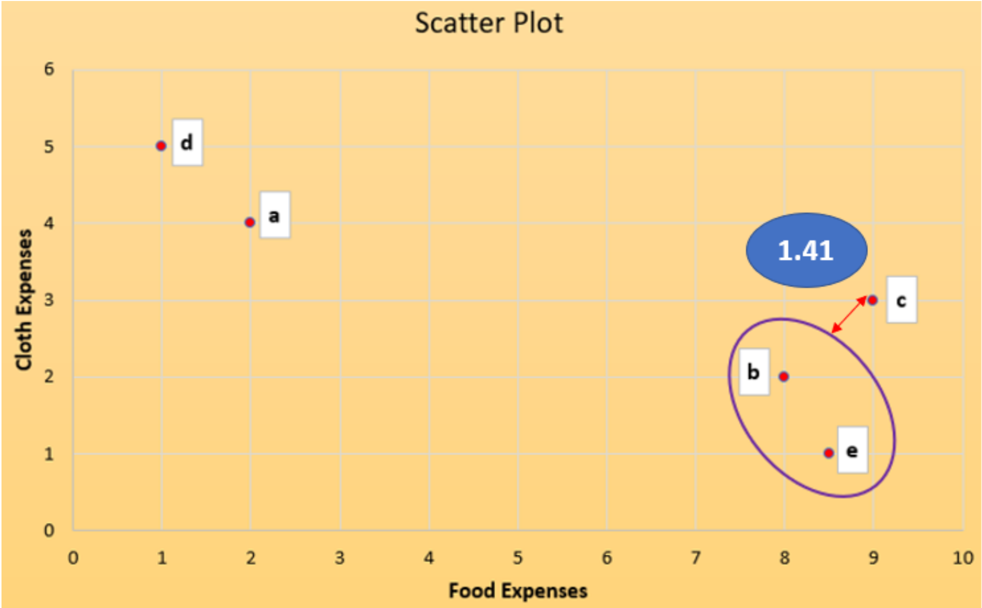
$$D(ad,c) = min\{7.07, 8.25\}$$

$$D(ad,c) = 7.07$$



Where c should go?





Stage 3: Cluster b e must envelop c

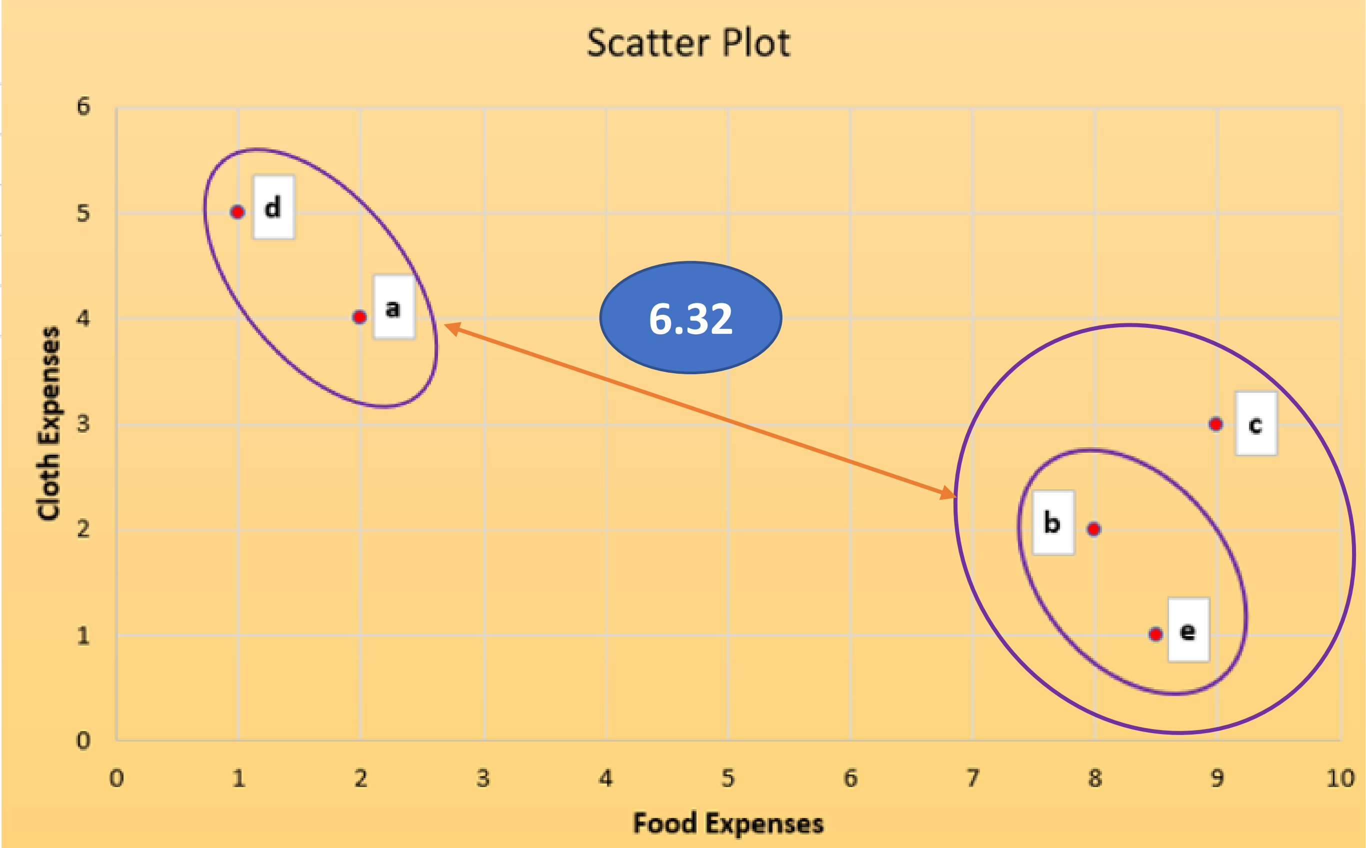
Stage 4

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

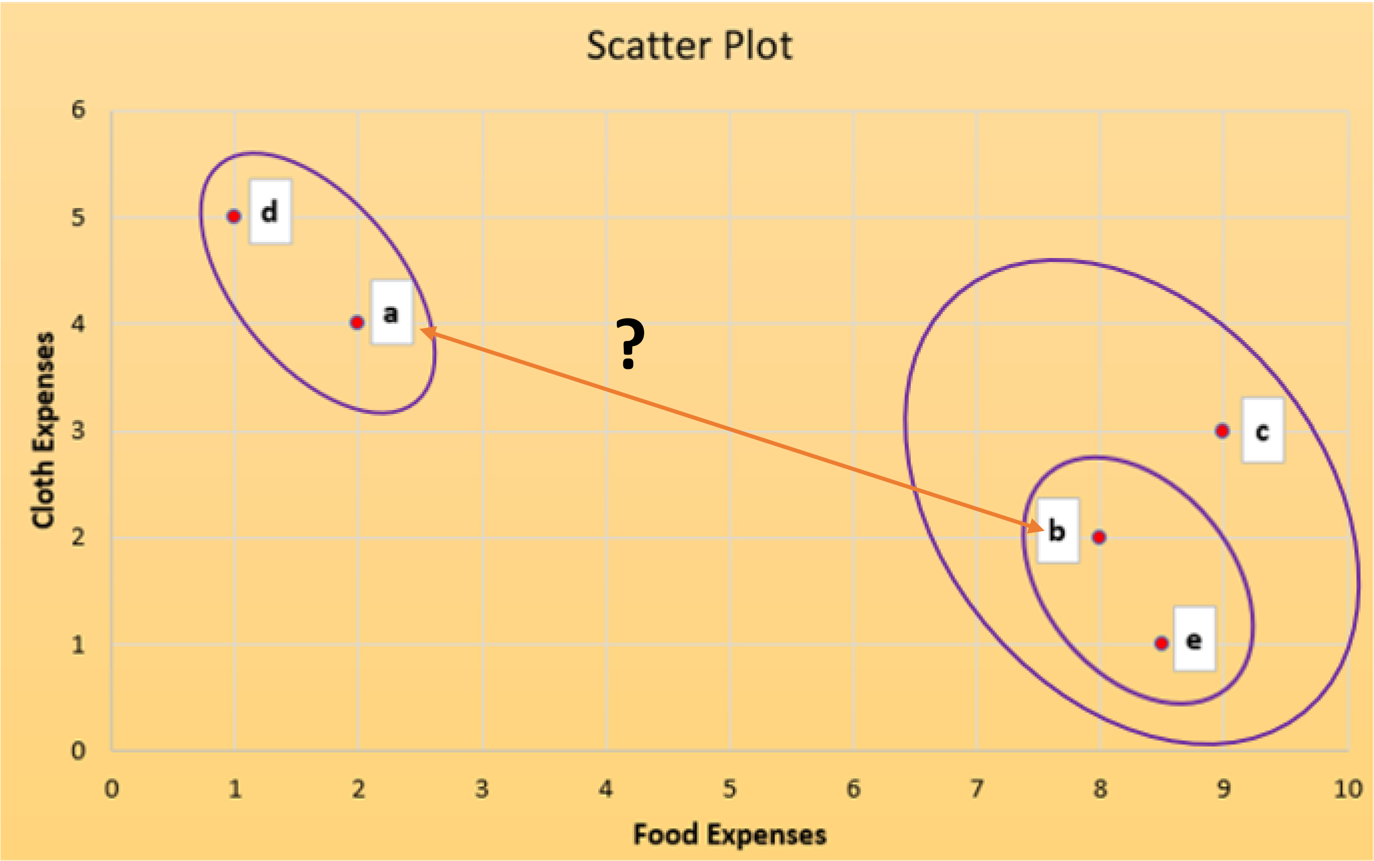
	b e c	a d
b e c	0	6.32
a d		0

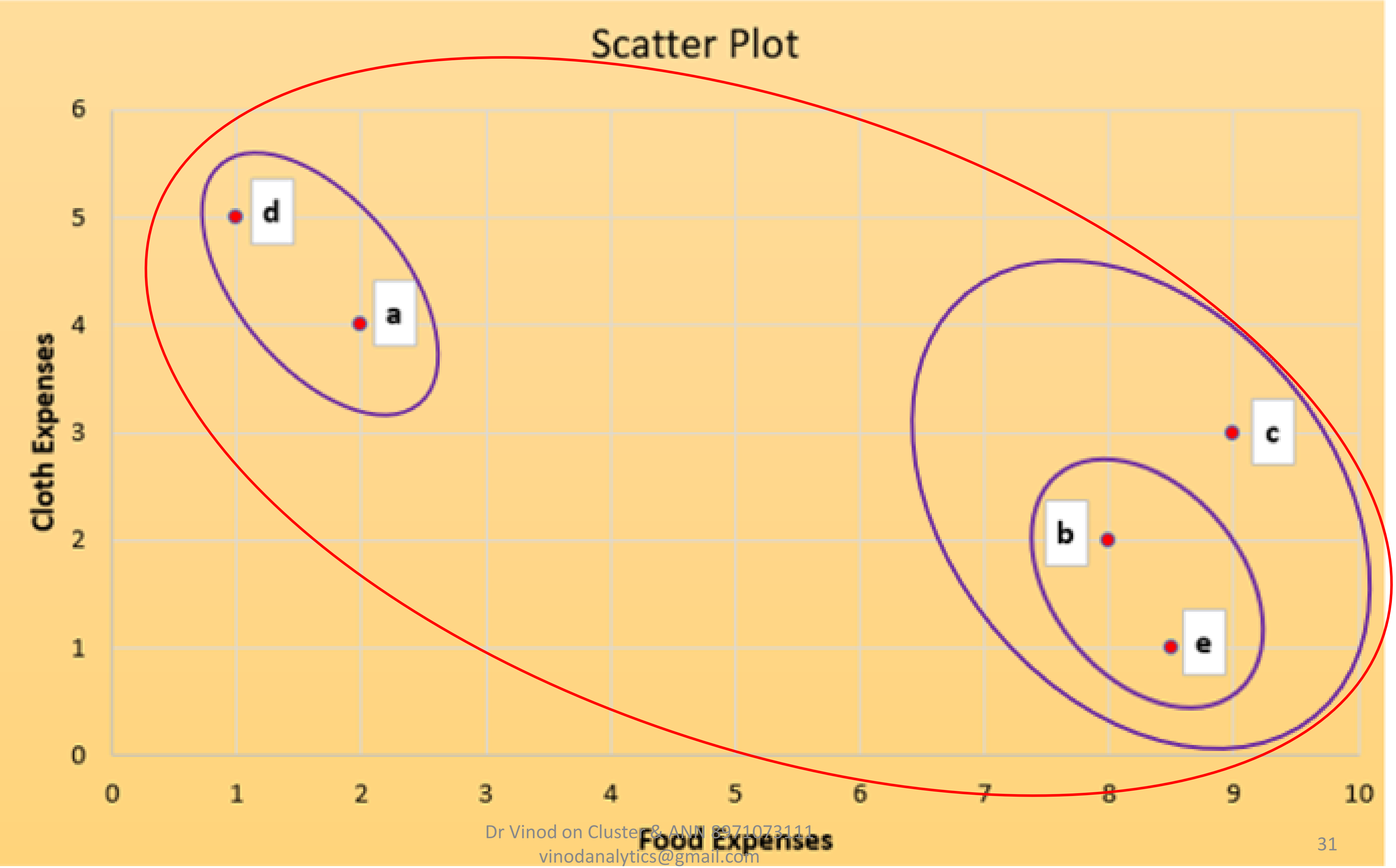
$$D(bec, ad) = min\{D(b,a), D(b,d), D(e,a), D(e,d)\ \ D(c,a), D(c,d)\}$$

$$D(bec, ad) = min\{\mathbf{6.32}, 7.62, 7.16, 8.50, 7.07, 8.25\ \}$$



Final round: Distance between a d and b e c




Scatter Plot
Cloth Expenses
Food Expenses
0
1
2
3
4
5
6
7
8
9
10
d
a
c
b
e

Single Linkage: Nearest Neighbor

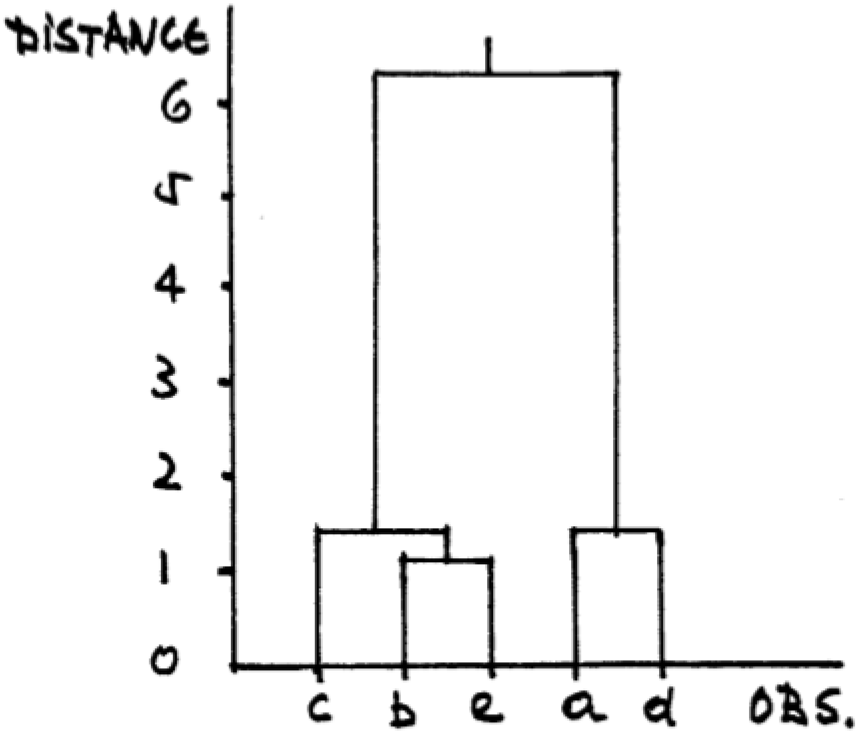


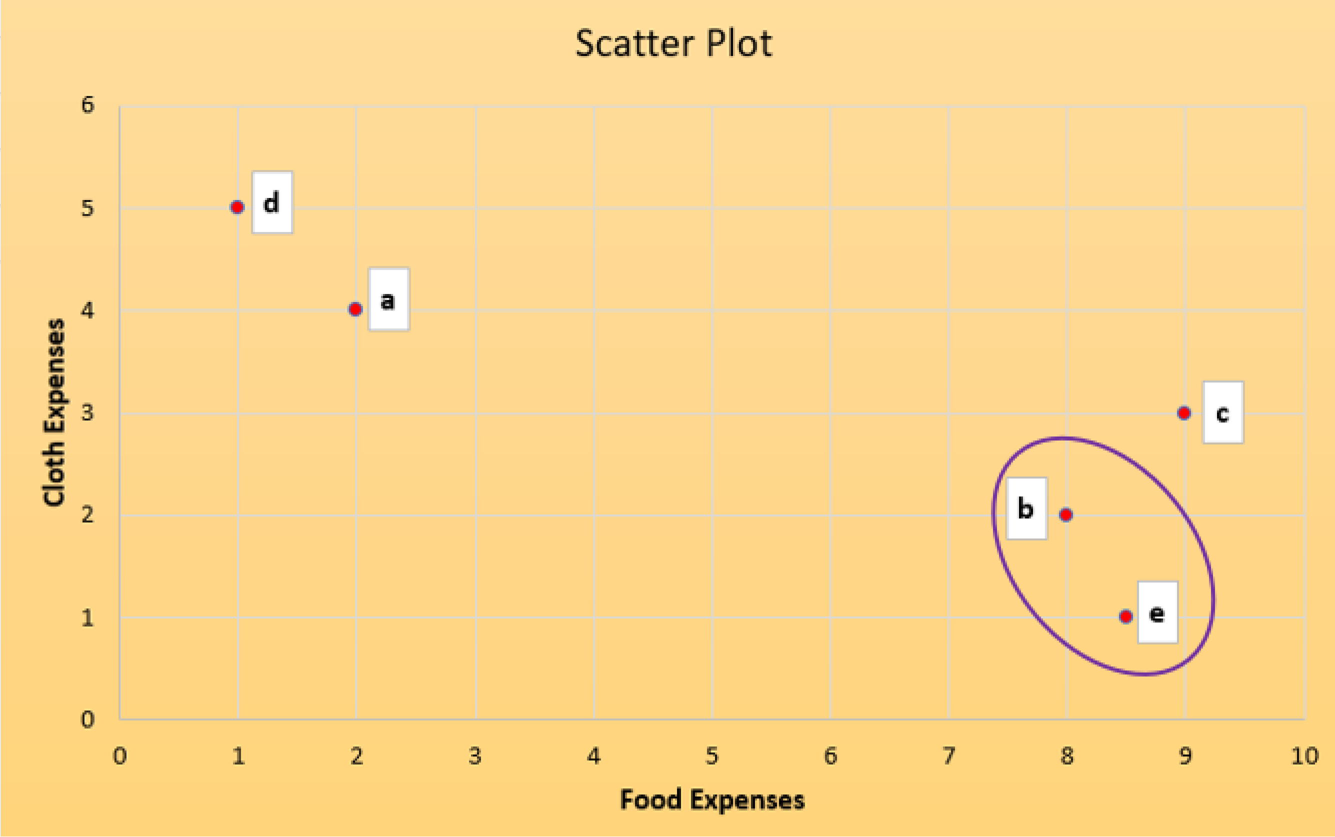
Figure 15.8
Nearest neighbor method, dendrogram

Complete Linkage: Farthest Neighbor

Stage 1: b and e can be clubbed

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00

Now *b* and *e* are in one cluster. How their values will be treated for finding DIST vis-à-vis another point/s?



Stage 2: Complete Linkage
Points a and d can be clubbed (same as Single Linkage)

$$D(be,a) = max\{D(b,a), D(e,a)\}$$

$$D(be,a) = max\{6.32, 7.16\}$$

$$D(be,a) = \mathbf{7.16}$$

$$D(be,c) = max\{D(b,c), D(e,c)\}$$

$$D(be,c) = max\{1.41, 2.06\}$$

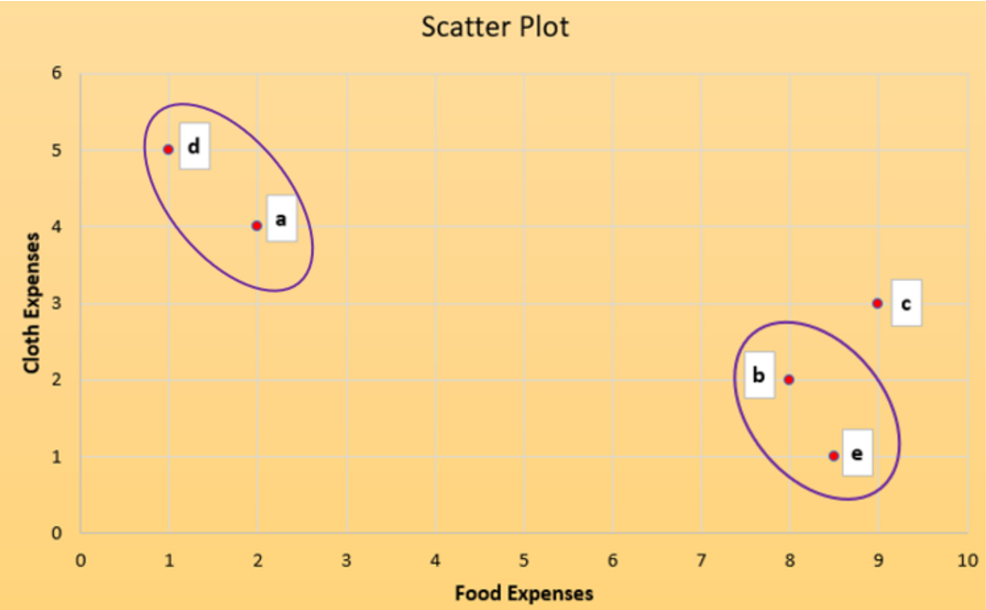
$$D(be,c) = \mathbf{2.06}$$

$$D(be,d) = max\{D(b,d), D(e,d)\}$$

$$D(be,d) = max\{7.62, 8.50\}$$

$$D(be,d) = \mathbf{8.50}$$

	b e	a	c	d
b e	**0**	***7.16***	***2.06***	***8.5***
a		**0**	**7.07**	**1.41**
c			**0**	**8.25**
d				**0**



Stage 3: Complete Linkage
Point c can be clubbed with b e (same as Single Linkage)

	b e	a	c	d
b e	0	*7.16*	*2.06*	*8.5*
a		0	7.07	1.41
c			0	8.25
d				0

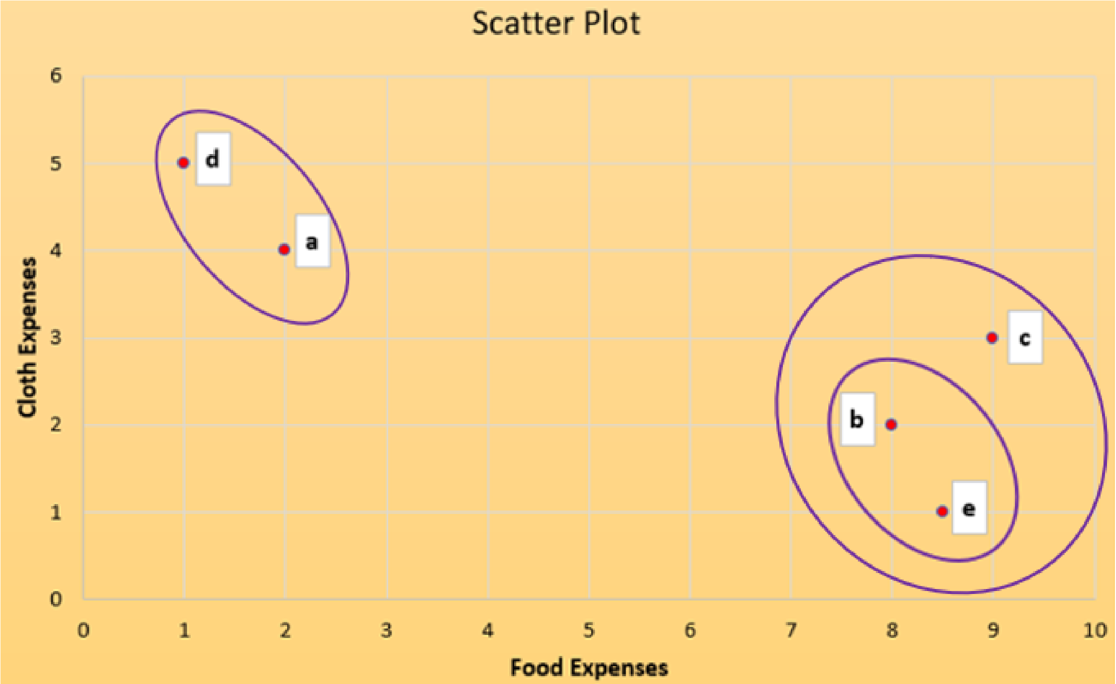
$$D(be, ad) = max\{D(be, a), D(be, d)\}$$

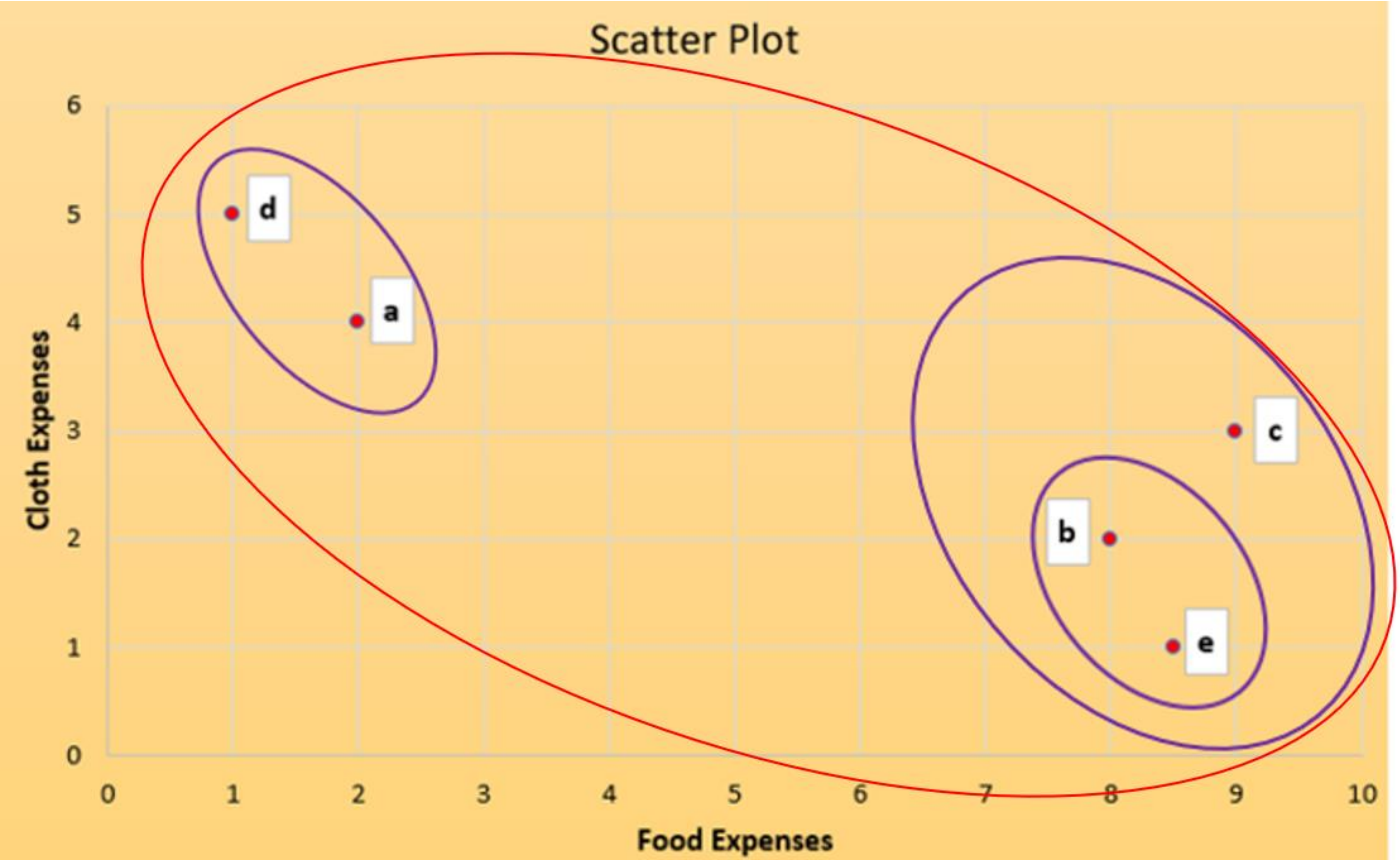
$$D(be, ad) = max\{7.16, 8.50\}$$

$$D(be, ad) = \mathbf{8.50}$$

Only **D**(**be, ad**) need to be calculated
Remaining are available in above table

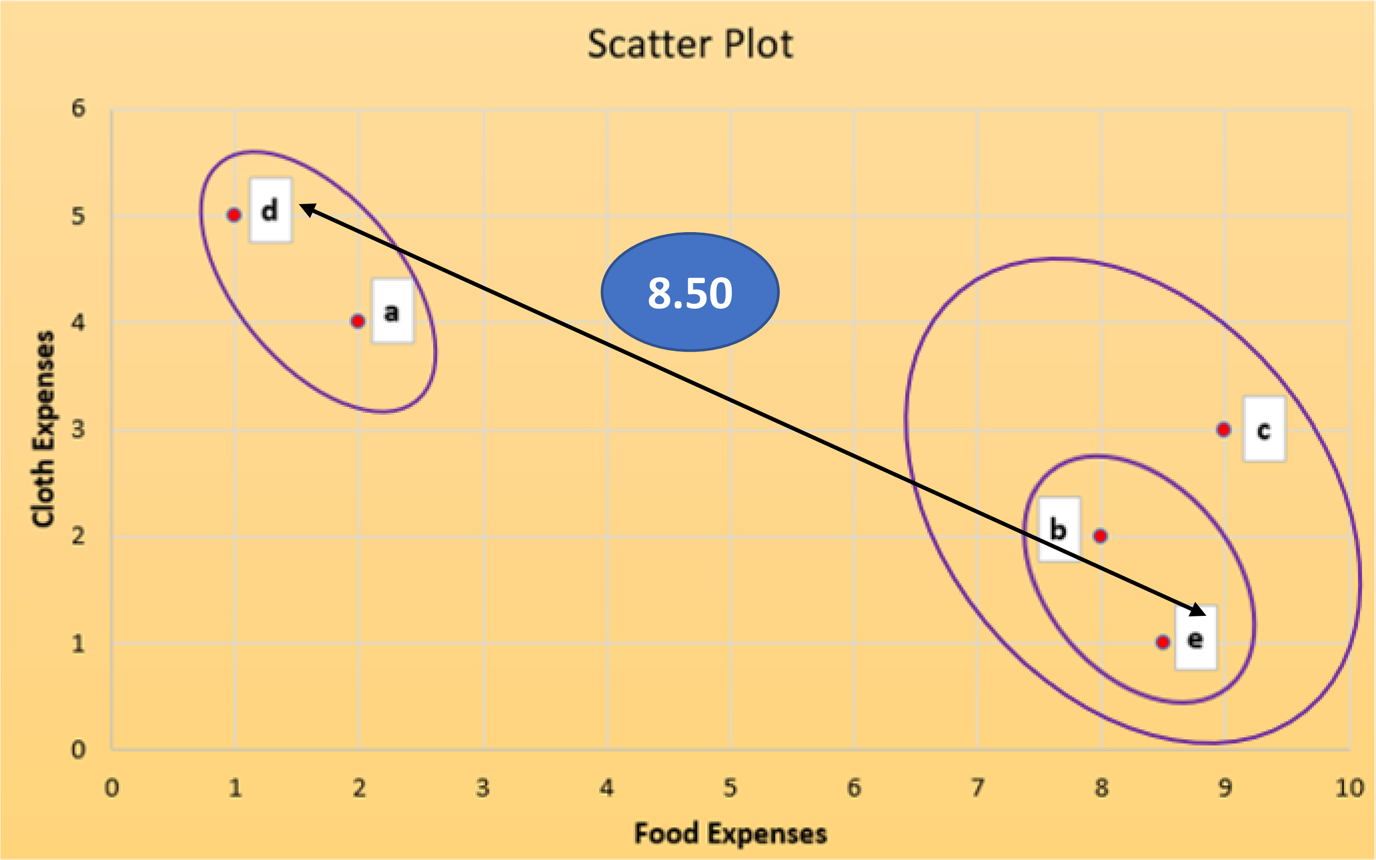
	b e	a d	c
b e	0	8.5	2.06
a d		0	8.25
c			0





vinodanalytics@gmail.com

Final round: Distance between a d and b e c



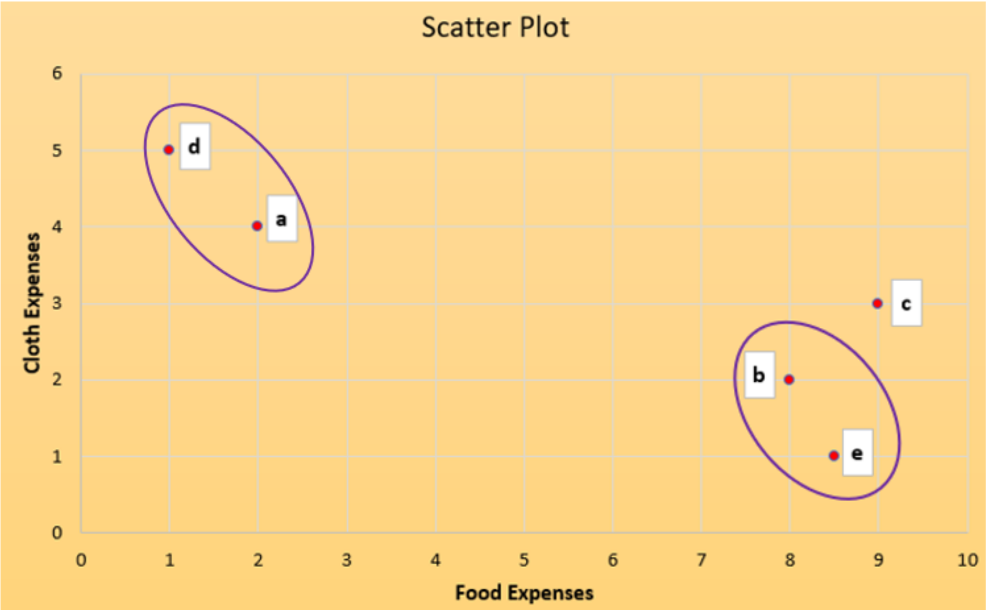
$$D(bec, ad) = max\{D(b,a), D(b,d), D(e,a), D(e,d) \;\; D(c,a), D(c,d)\}$$

$$D(bec, ad) = max\{6.32, 7.62, 7.16, 8.50, 7.07, 8.25\}$$

Average Linkage: Compromise (Min Max)

Stage 1 & 2

	a	b	c	d	e
a	0.00	6.32	7.07	1.41	7.16
b		0.00	1.41	7.62	1.12
c			0.00	8.25	2.06
d				0.00	8.50
e					0.00



	b e	a	c	d
b e	0	6.74	1.74	8.06
a		0	7.07	1.41
c			0	8.25
d				0

First line only to be calculated (as averages), Rest from upper table

$$D(be,a) = \{D(b,a) + D(e,a)\}/2$$

$$D(be,a) = \frac{\{6.32 + 7.16\}}{2} = \mathbf{6.74}$$

$$D(be,d) = \{D(b,d) + D(e,d)\}/2$$

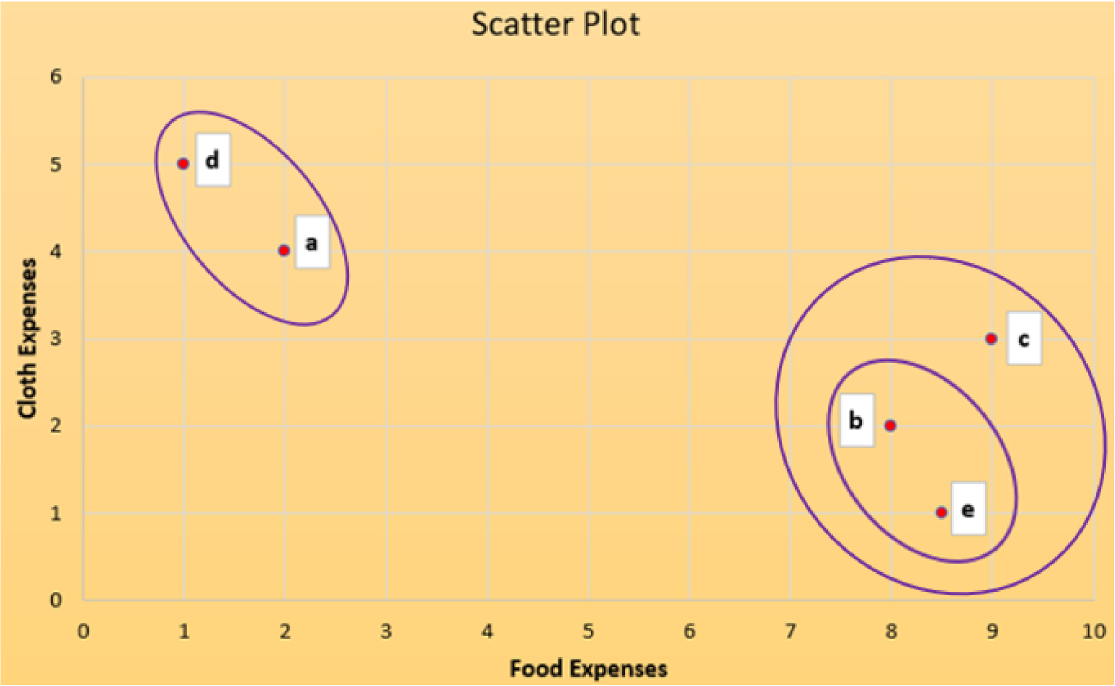
$$D(be,d) = \frac{\{7.62 + 8.50\}}{2} = \mathbf{8.06}$$

$$D(be,c) = \{D(b,c) + D(e,c)\}/2$$

$$D(be,c) = \frac{\{1.41 + 2.06\}}{2} = \mathbf{1.74}$$

Stage 3

	b e	a	c	d
b e	0	6.74	1.74	8.06
a		0	7.07	1.41
c			0	8.25
d				0



	b e	a d	c
b e	0	7.4	1.74
a d		0	7.66
c			0

1.74 is from upper table, rest two were calculated

$$D(be, ad) = \{D(be, a) + D(be, d)\}/2$$

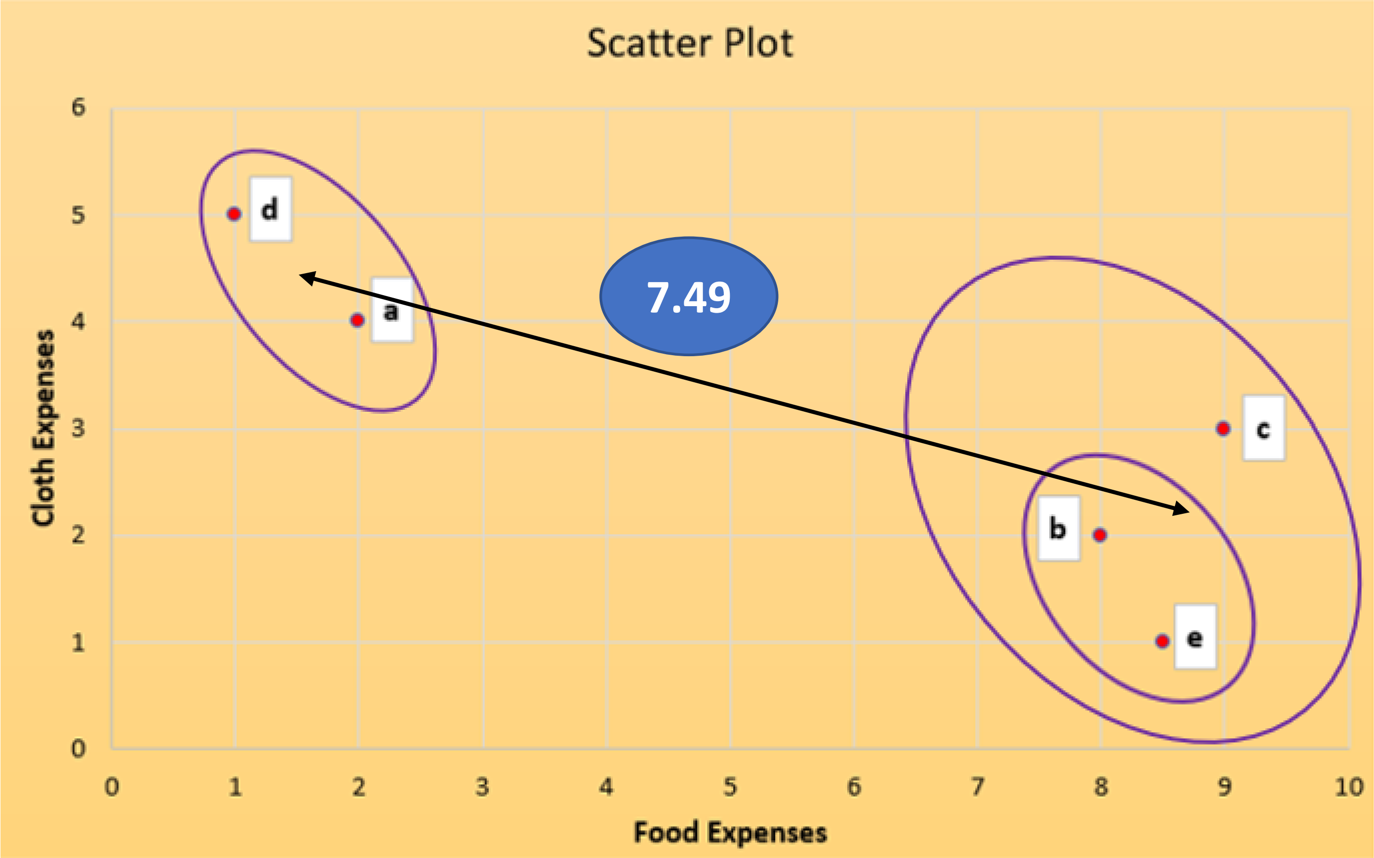
$$D(be, ad) = \frac{\{6.74 + 8.06\}}{2} = \mathbf{7.4}$$

$$D(ad, c) = \{D(a, c) + D(d, a)\}/2$$

Actually c

$$D(ad, c) = \frac{\{7.07 + 8.25\}}{2} = \mathbf{7.66}$$

Final round: Distance between a d and b e c



$$D(bec, ad) = average\{D(b,a), D(b,d), D(e,a), D(e,d)\ \ D(c,a), D(c,d)\}$$

$$D(bec, ad) = average\{6.32, 7.62, 7.16, 8.50, 7.07, 8.25\} = \mathbf{7.49}$$

Ward Linkage: Finds centroids and then SSE

Distance Matrix

	Food Exp	Cloth Exp
	X	Y
a	2	4
b	8	2
c	9	3
d	1	5
e	8.5	1

Ward's Method

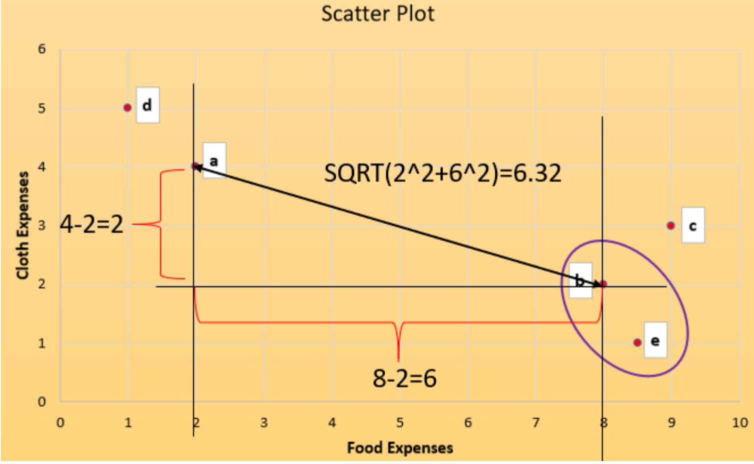
Ward's method means calculating the incremental sum of squares. *Half square Euclidean distance* is the only distance measure that can be used with this clustering method. Therefore, the distance measure is automatically set to *Half square Euclidean distance* when Ward's method is selected.

	a	b	c	d	e	stage 1
a	0.00	20.00	25.00	1.00	25.63	
b		0.00	1.00	29.00	0.63	
c			0.00	34.00	2.13	
d				0.00	30.13	
e					0.00	

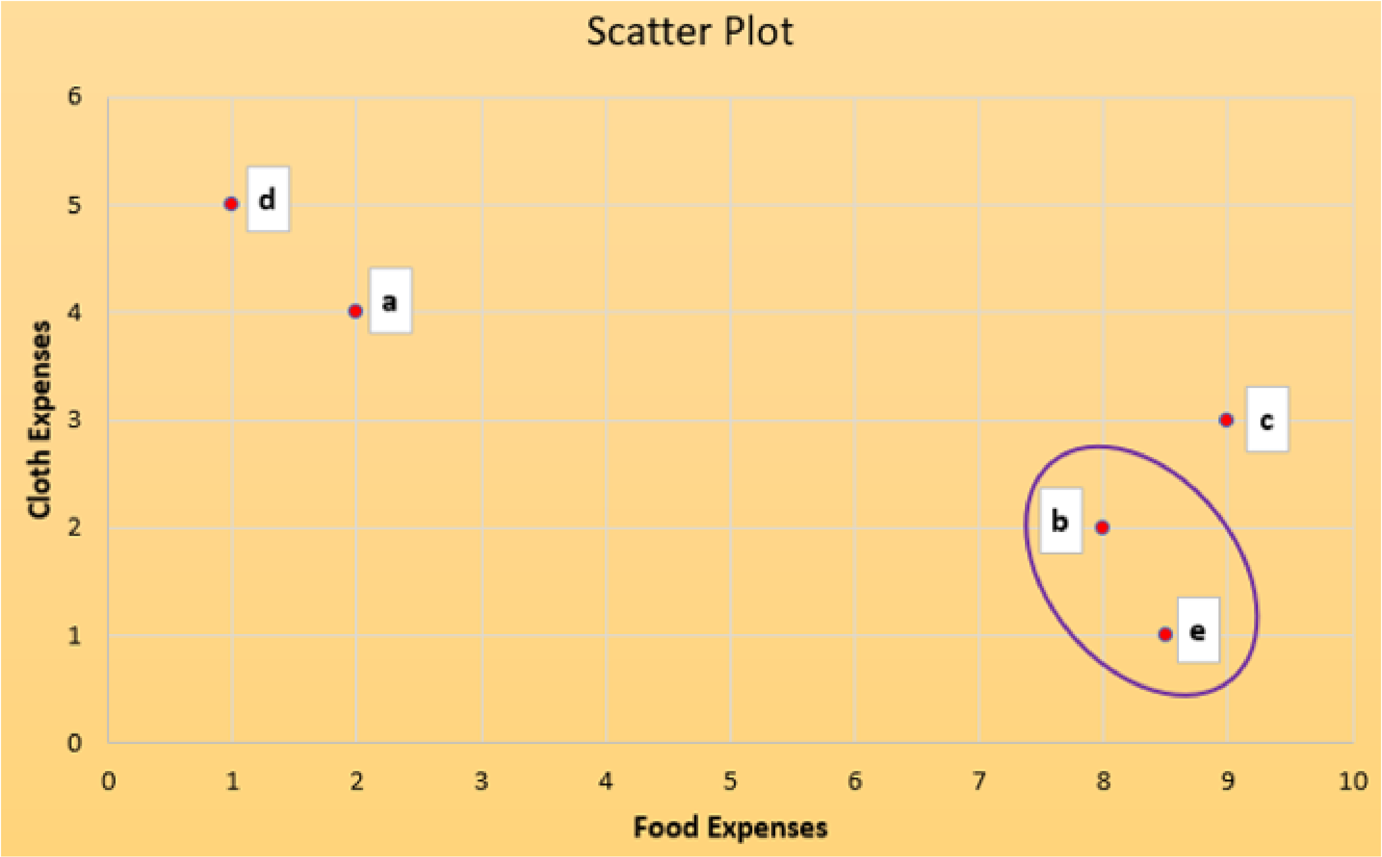
36.125

Euclidean Distances

	b e	a	c	d
b e	0	6.74	1.74	8.06
a		0	7.07	1.41
c			0	8.25
d				0

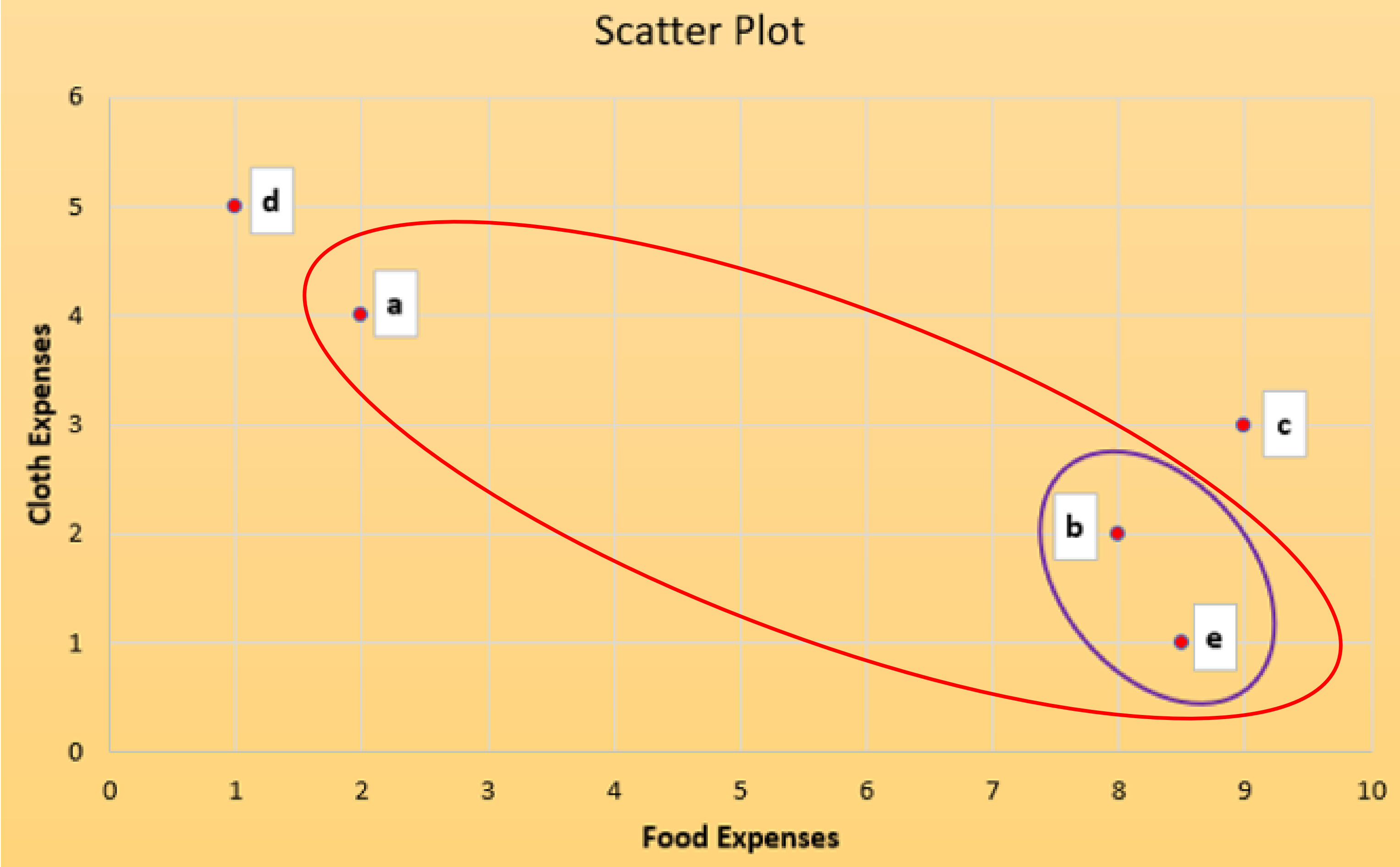


Stage 1



Scatter Plot
Cloth Expenses
Food Expenses
0
1
2
3
4
5
6
7
8
9
10
d
a
c
b
e

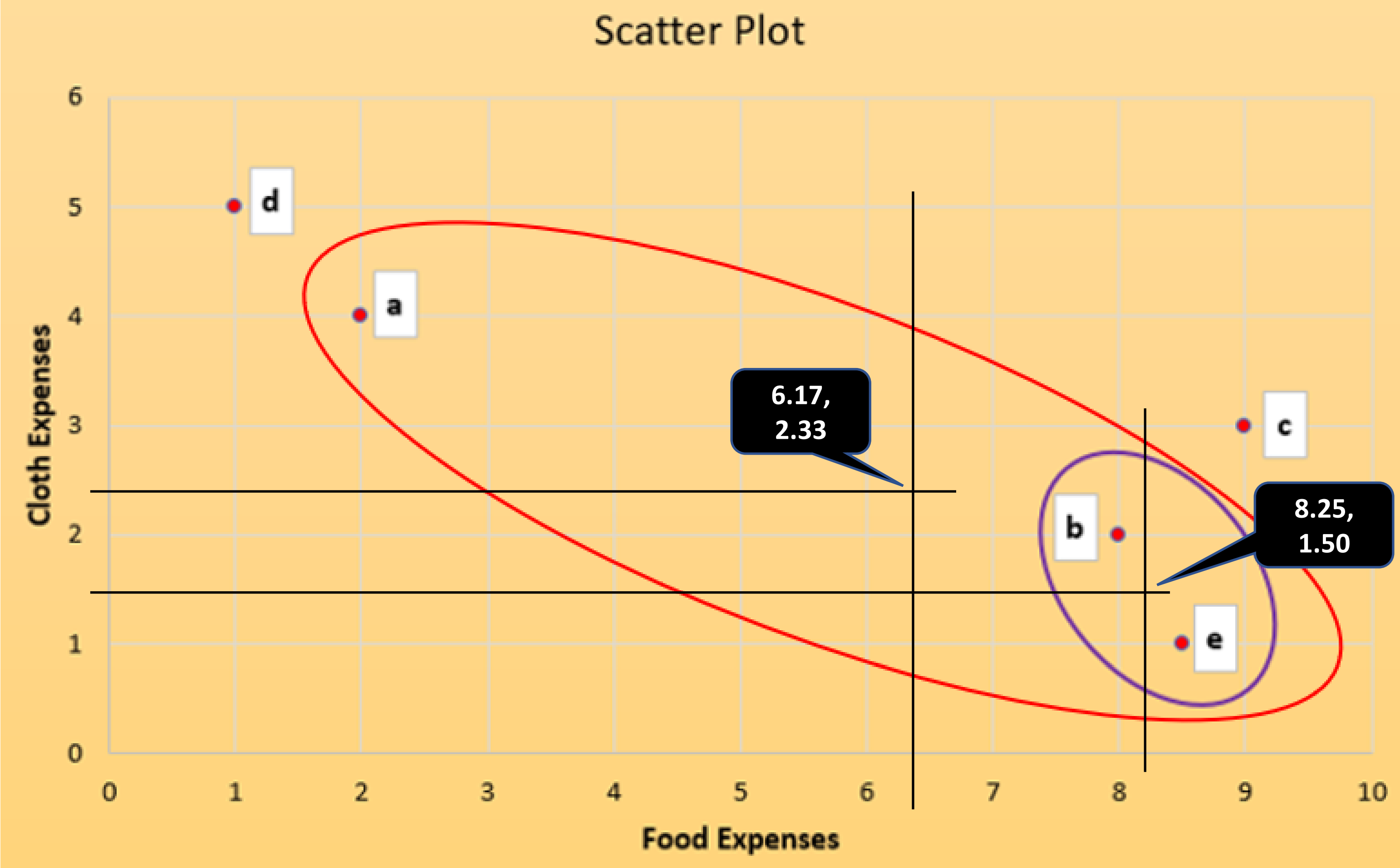
Distance Between (be) and a



Distance Between (be) and a
First find centroids

Centroid for cluster (bea)		
	X	Y
b	8	2
e	8.5	1
a	2	4
	6.17	**2.33**

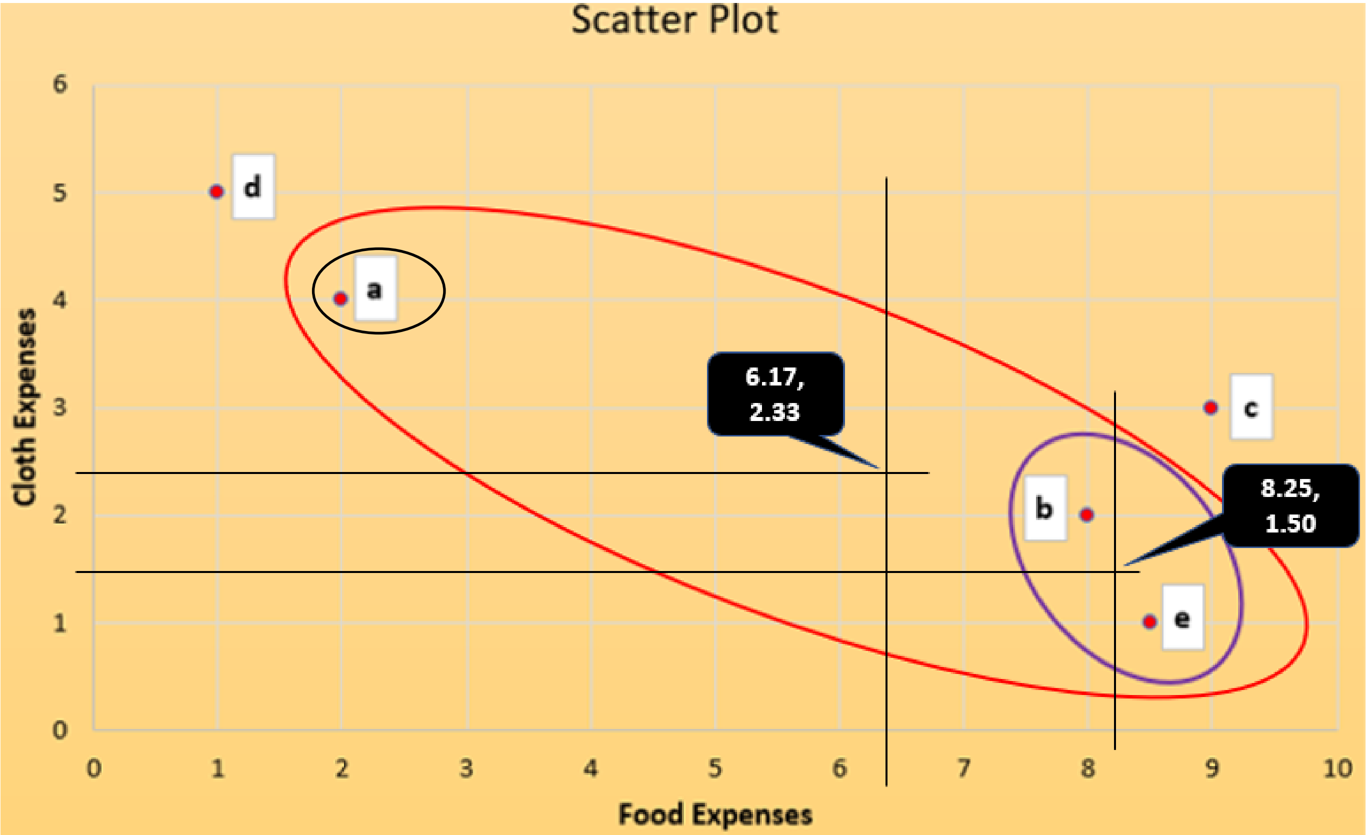
Centroid for cluster (be)		
	X	Y
b	8	2
e	8.5	1
	8.25	**1.50**



Distance Between (be) and a

Centroid for cluster (bea)		
	X	Y
b	8	2
e	8.5	1
a	2	4
	6.17	2.33
SSE(bea)	30.83	

Centroid for cluster (be)		
	X	Y
b	8	2
e	8.5	1
	8.25	1.50
SSE(be)	0.63	

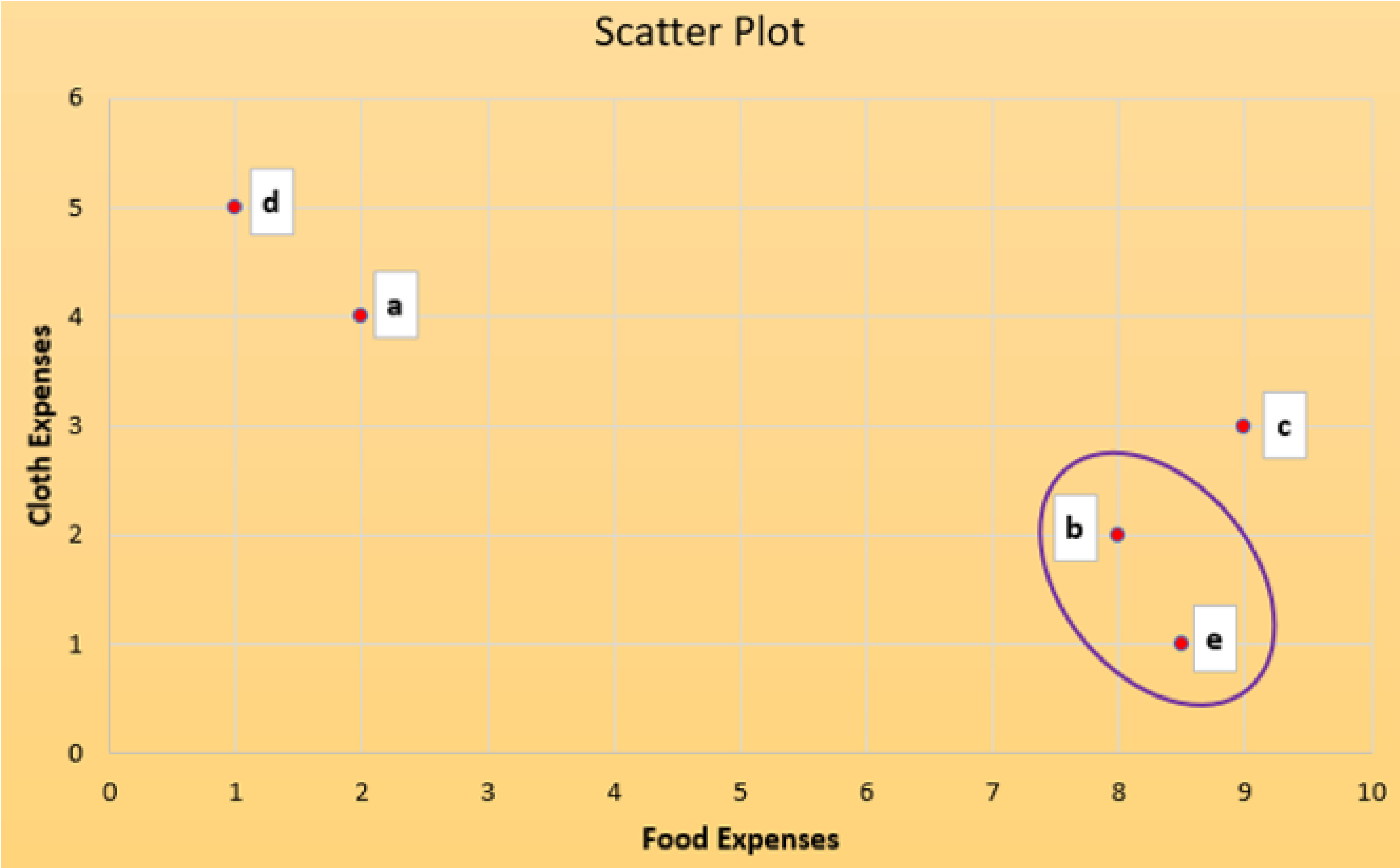


$$dist_{(be,a)} = SSE_{(bea)} - SSE_{be} - SSE_{a}$$

$$dist_{(be,a)} = 30.83 - 0.63 - 0 = 30.20$$

Distance Between (be) and d
First find centroids then find SSEs

Centroid for cluster (bed)		
	X	Y
b	8	2
e	8.5	1
d	1	5
	5.83	**2.67**
SSE(bed)	**43.83**	
Centroid for cluster (be)		
	X	Y
b	8	2
e	8.5	1
	8.25	**1.50**
SSE(be)	**0.63**	
dist(be,d)	**43.21**	



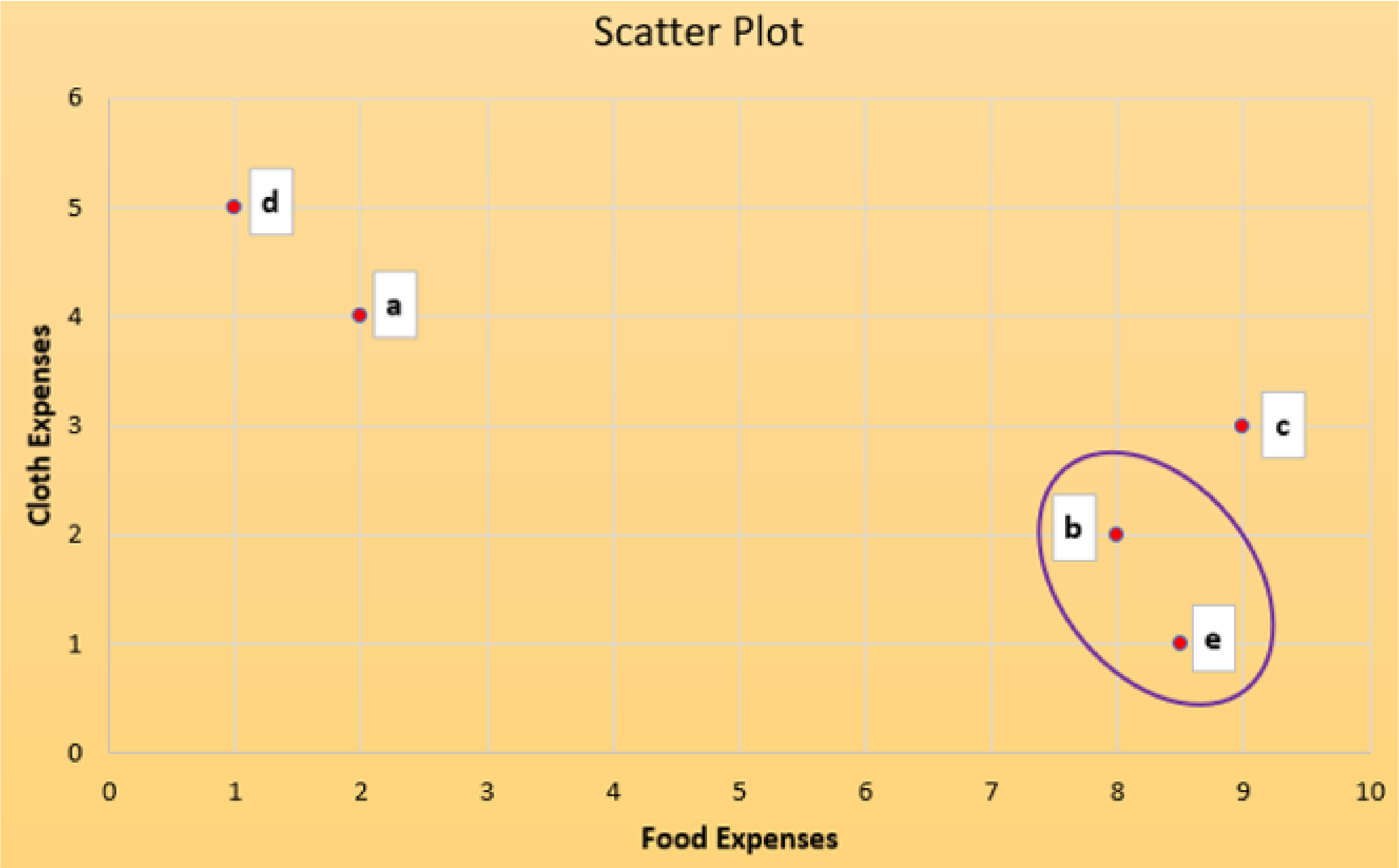
Dr Vinod on Cluster & ANN 8971073111
vinodanalytics@gmail.com

Distance Between (be) and c
First find centroids then find SSEs

Centroid for cluster (bec)		
	X	Y
b	8	2
e	8.5	1
c	9	3
	8.50	**2.00**
SSE(bec)	**2.50**	

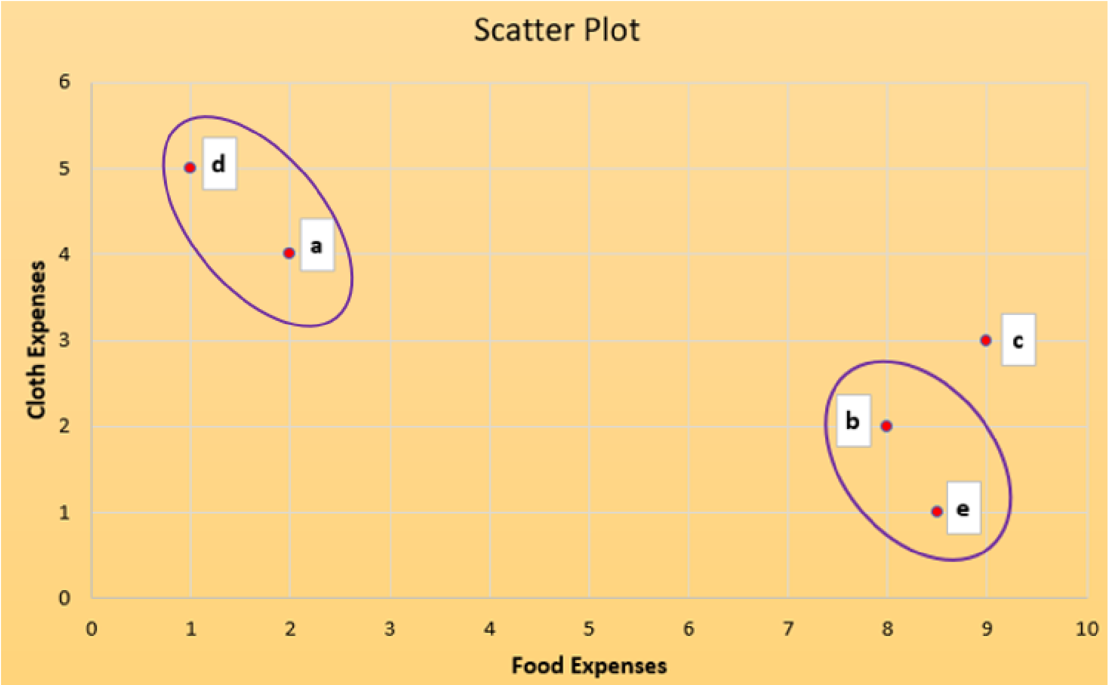
Centroid for cluster (be)		
	X	Y
b	8	2
e	8.5	1
	8.25	**1.50**
SSE(be)	**0.63**	

dist(be,c)	**1.88**



Stage 2

	b e	a	c	d	stage 2
b e	0	30.21	1.88	43.21	
a		0	25.00	1.00	
c			0	34.00	
d				0	



Lets make it

```
In [1]: # Jesus is my Saviour!

In [2]: import os

In [3]: os.chdir('C:\\Users\\Dr Vinod\\Desktop\\WD_python')

In [4]: # our exported file will appear here

In [5]: import pandas as pd

In [6]: import numpy as np

In [7]: import matplotlib.pyplot as plt

In [8]: from sklearn.preprocessing import normalize
```

Data: Wholesale customers data

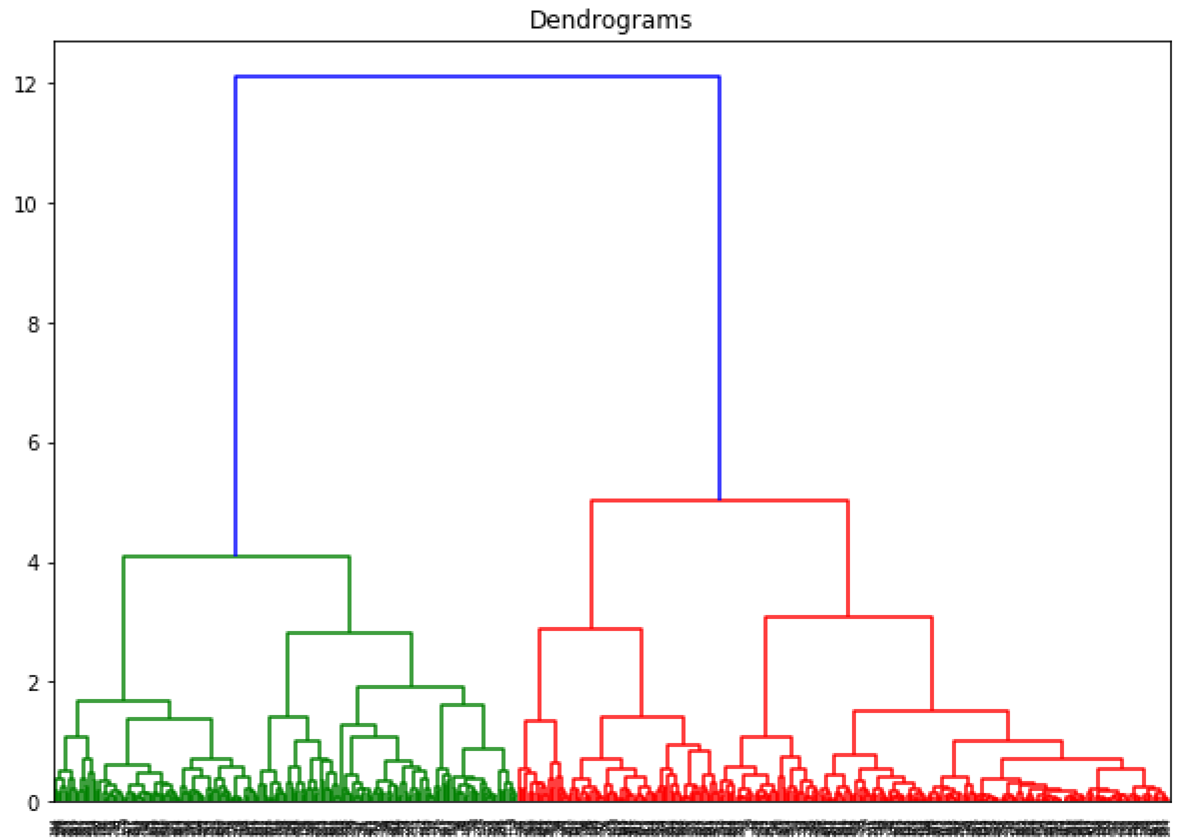
```
In [9]: data = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/Wholesale customers
data.csv")

In [10]: data = pd.DataFrame(data)

In [11]: data.head(3)
Out[11]:
   Channel  Region  Fresh  Milk  Grocery  Frozen  Detergents_Paper  Delicassen
0        2       3  12669  9656     7561     214              2674        1338
1        2       3   7057  9810     9568    1762              3293        1776
2        2       3   6353  8808     7684    2405              3516        7844
```

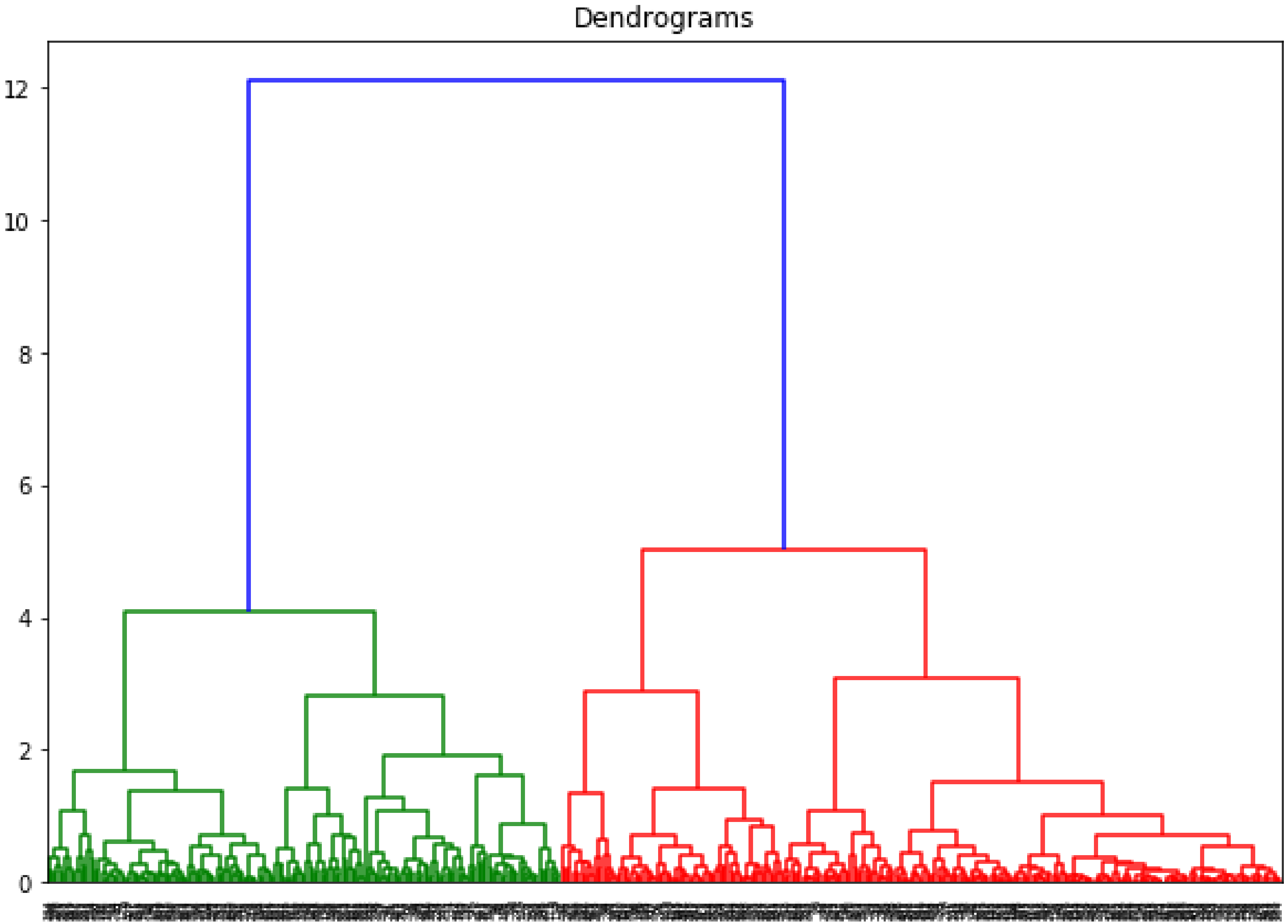
Scale the data

```
In [12]: data_scaled = normalize(data)

In [13]: data_scaled = pd.DataFrame(data_scaled, columns = data.columns)

In [14]: data.head(data_scaled)
Traceback (most recent call last):
```

Dendrogram

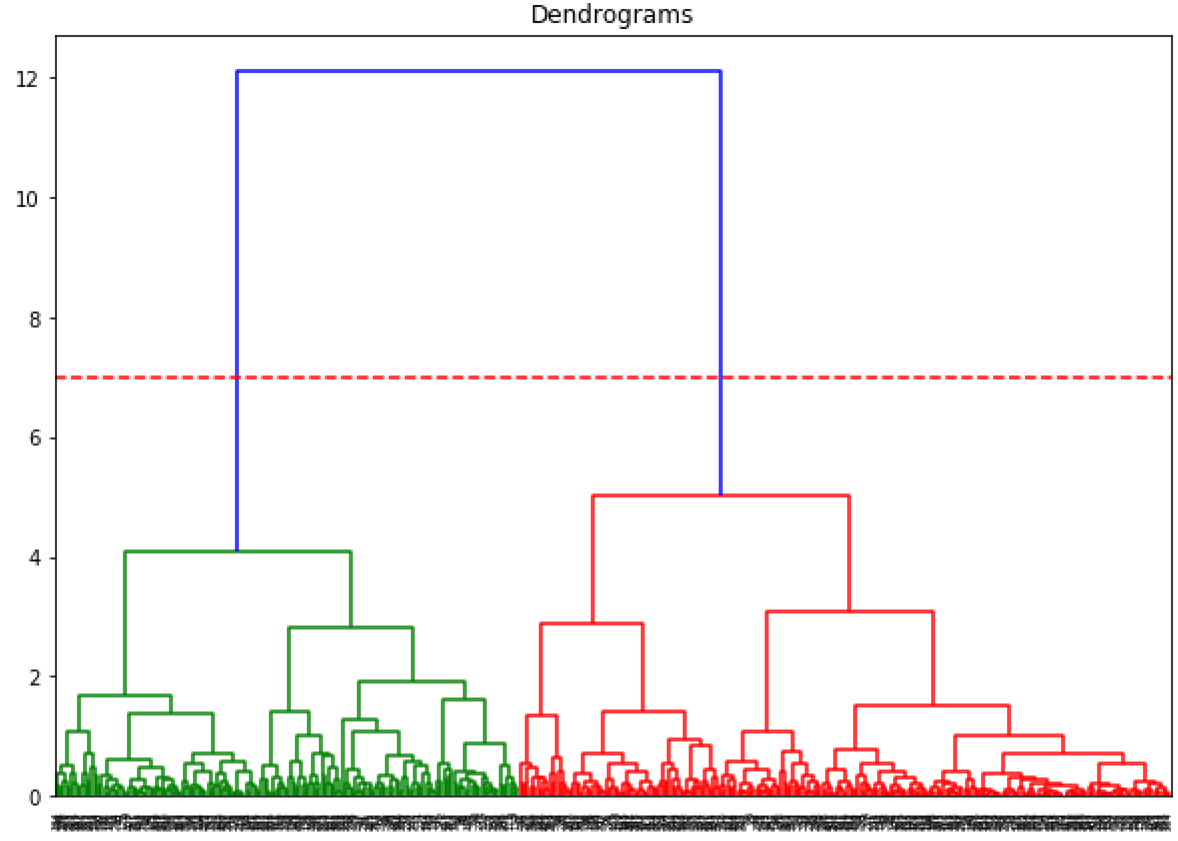
```
In [15]: import scipy.cluster.hierarchy as shc

In [16]: plt.figure(figsize = (10,7))
    ...: plt.title("Dendrograms")
    ...: dend = shc.dendrogram(shc.linkage(data_scaled, method = 'ward'))
```

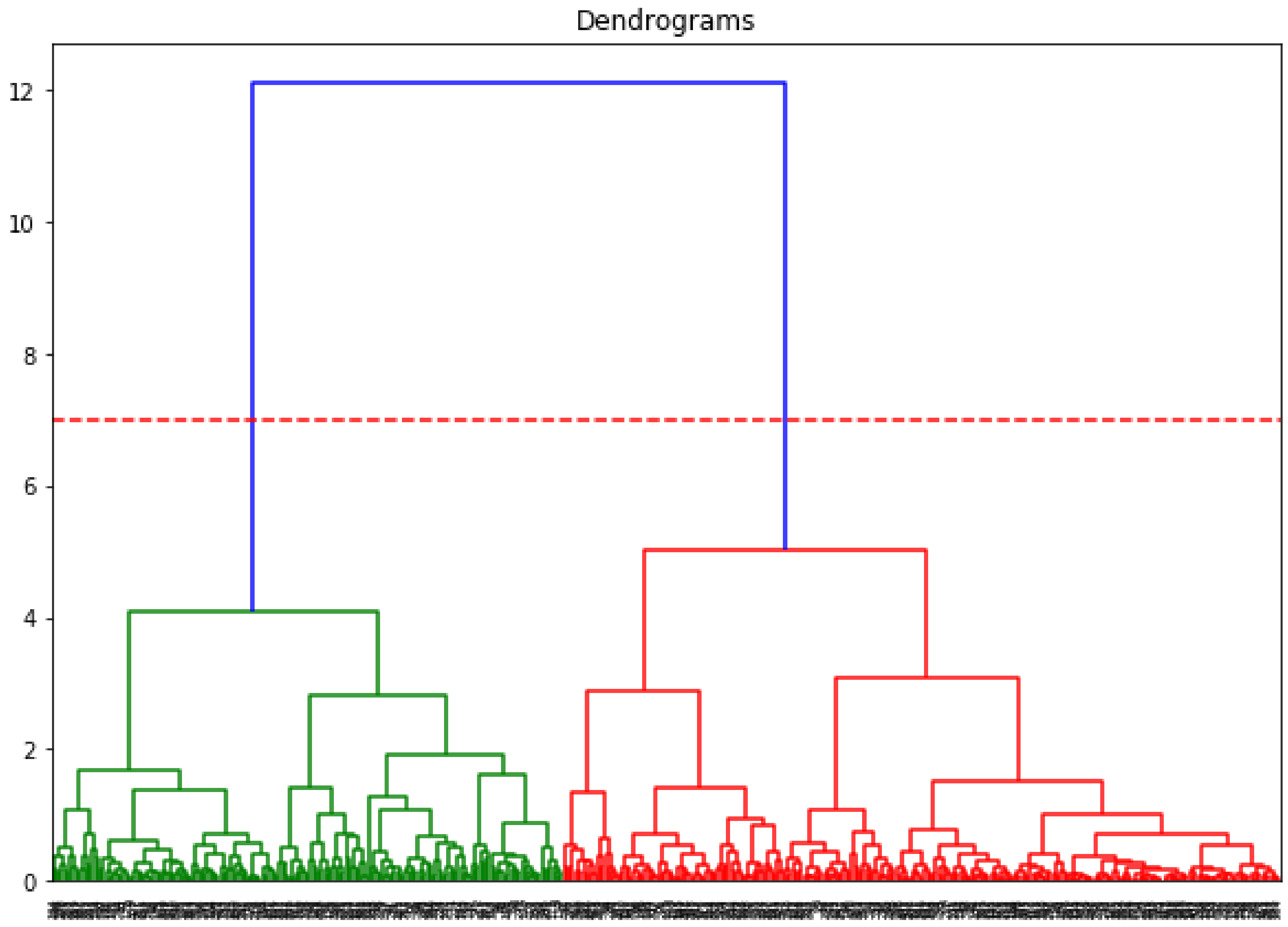



Dendrograms
12
10
8
6
4
2
0

2 clusters

```
In [17]: plt.figure(figsize = (10,7))
    ...: plt.title("Dendrograms")
    ...: dend = shc.dendrogram(shc.linkage(data_scaled, method = 'ward'))
    ...: plt.axhline(y=7, color = "r", linestyle = '--')
Out[17]: <matplotlib.lines.Line2D at 0x2242398ec50>
```



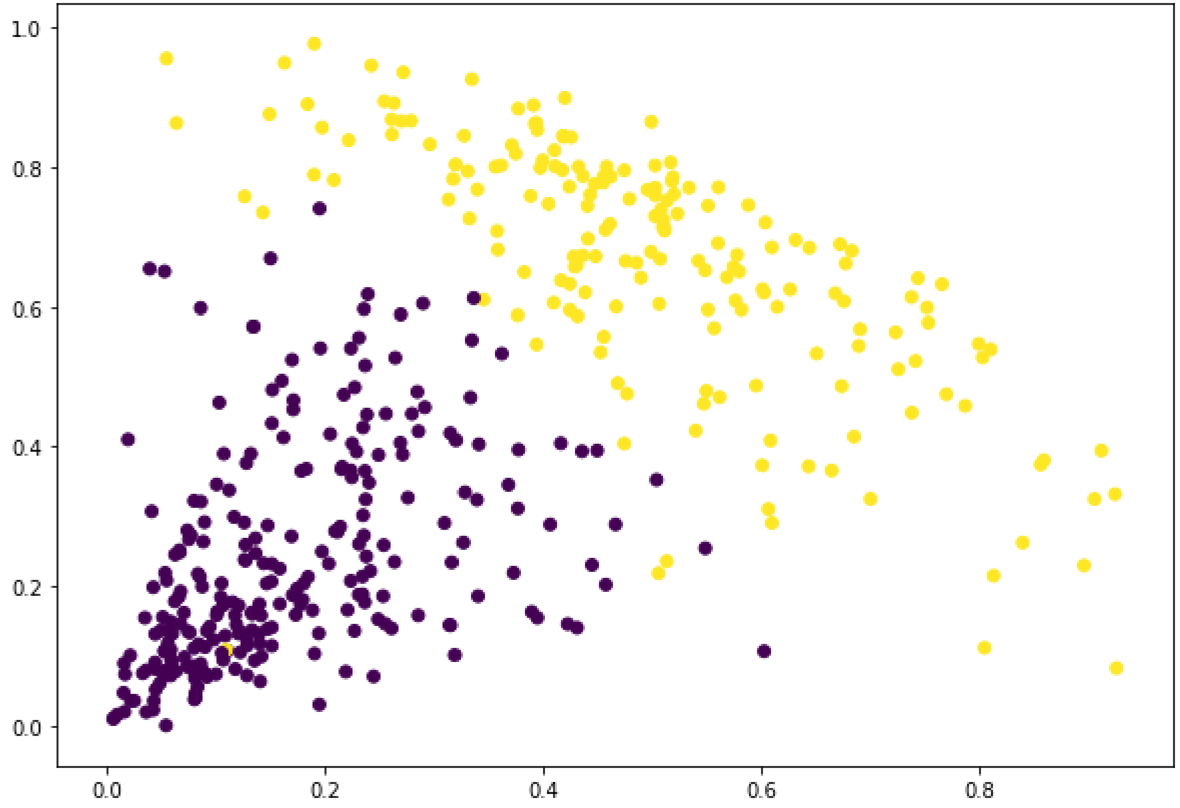

Dendrograms
12
10
8
6
4
2
0

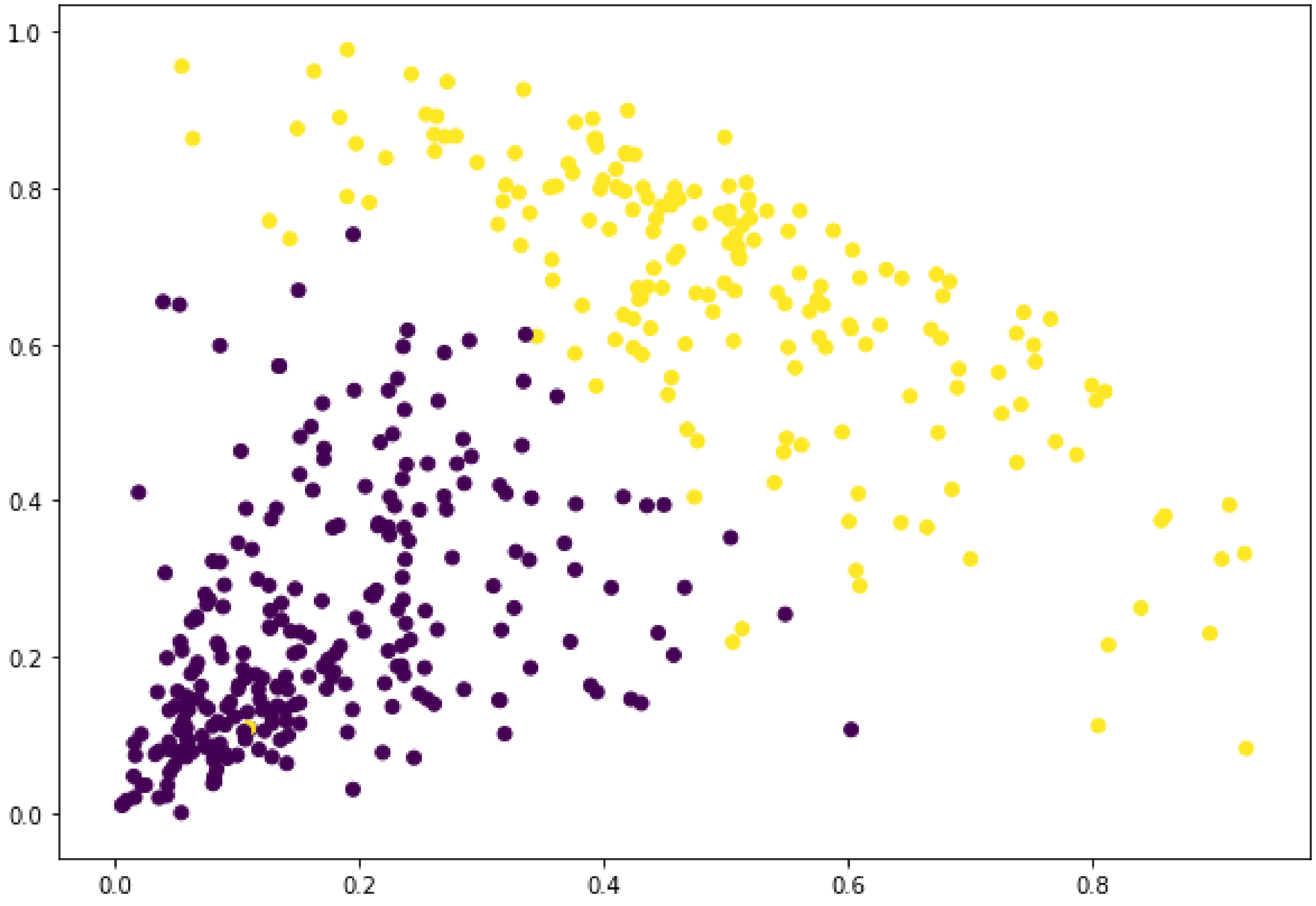
Lets make clusters

```
In [18]: #_______lets apply Hierarchical clustering for 2 clusters

In [19]: from sklearn.cluster import AgglomerativeClustering

In [20]: cluster = AgglomerativeClustering(n_clusters = 2, affinity = 'euclidean',
linkage = 'ward')

In [21]: cluster.fit_predict(data_scaled)
Out[21]:
array([1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
```

Plot

```
In [22]: plt.figure(figsize = (10,7))
    ...: plt.scatter(data_scaled['Milk'], data_scaled['Grocery'], c=cluster.labels_)
    ...:
Out[22]: <matplotlib.collections.PathCollection at 0x22423e25898>
```



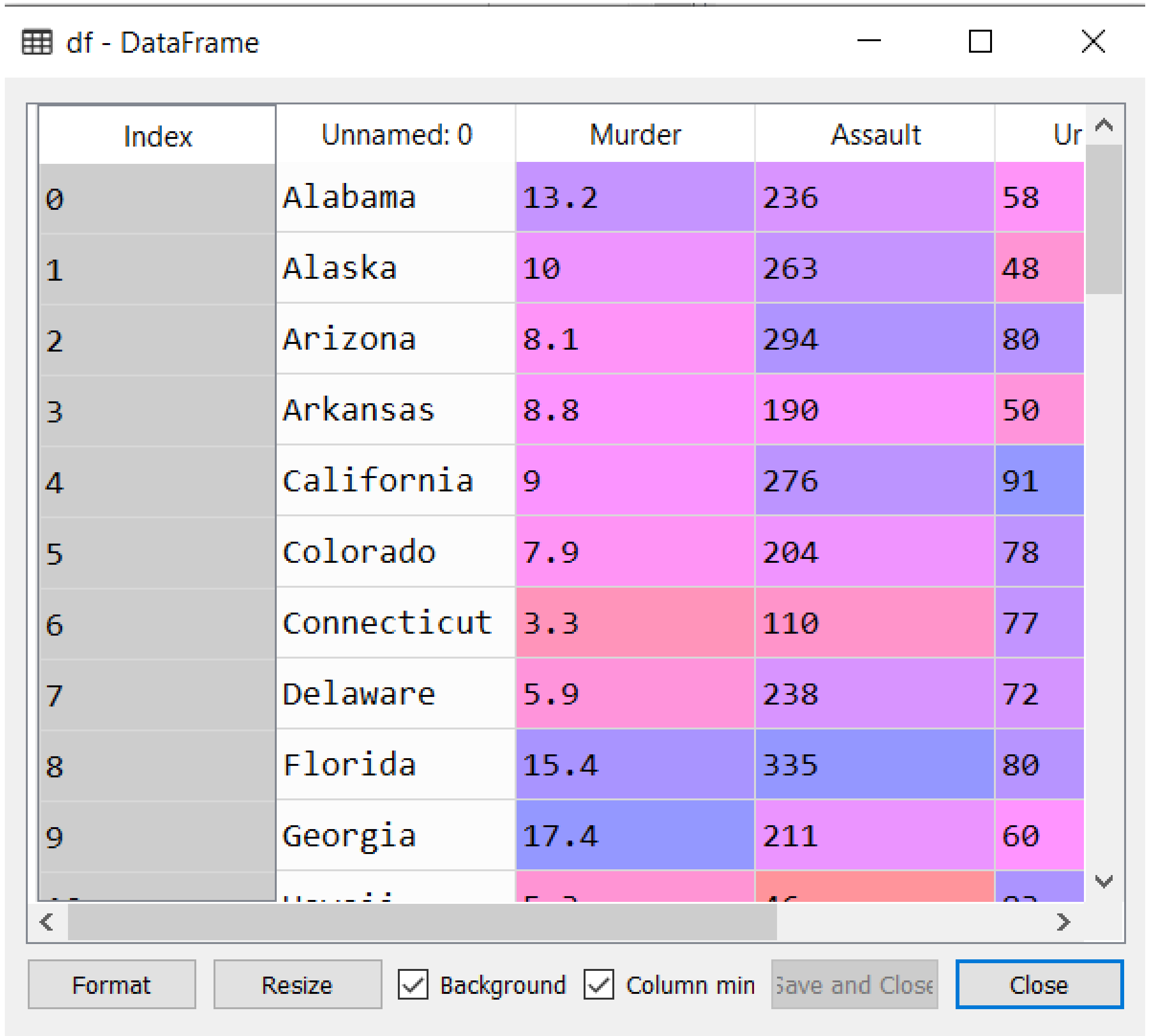

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8

K-means

Libraries

```
In [1]: # importing needed libraries

In [2]: import pandas as pd

In [3]: import matplotlib.pyplot as plt

In [4]: from sklearn import preprocessing

In [5]: from sklearn.preprocessing import scale

In [6]: from sklearn.preprocessing import StandardScaler

In [7]: from sklearn.cluster import KMeans

In [8]: import seaborn as sns
```

```python
In [9]: df = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/USArrests.csv")

In [10]: df = pd.DataFrame(df)
```



Index	Unnamed: 0	Murder	Assault	Ur
0	Alabama	13.2	236	58
1	Alaska	10	263	48
2	Arizona	8.1	294	80
3	Arkansas	8.8	190	50
4	California	9	276	91
5	Colorado	7.9	204	78
6	Connecticut	3.3	110	77
7	Delaware	5.9	238	72
8	Florida	15.4	335	80
9	Georgia	17.4	211	60

USArrests Data

df - DataFrame

Index	Unnamed: 0	Murder	Assault	Ur
0	Alabama	13.2	236	58
1	Alaska	10	263	48
2	Arizona	8.1	294	80
3	Arkansas	8.8	190	50
4	California	9	276	91
5	Colorado	7.9	204	78
6	Connecticut	3.3	110	77
7	Delaware	5.9	238	72
8	Florida	15.4	335	80
9	Georgia	17.4	211	60

Format | Resize | Background | Column min | Save and Close | Close

Bring state's name as row names

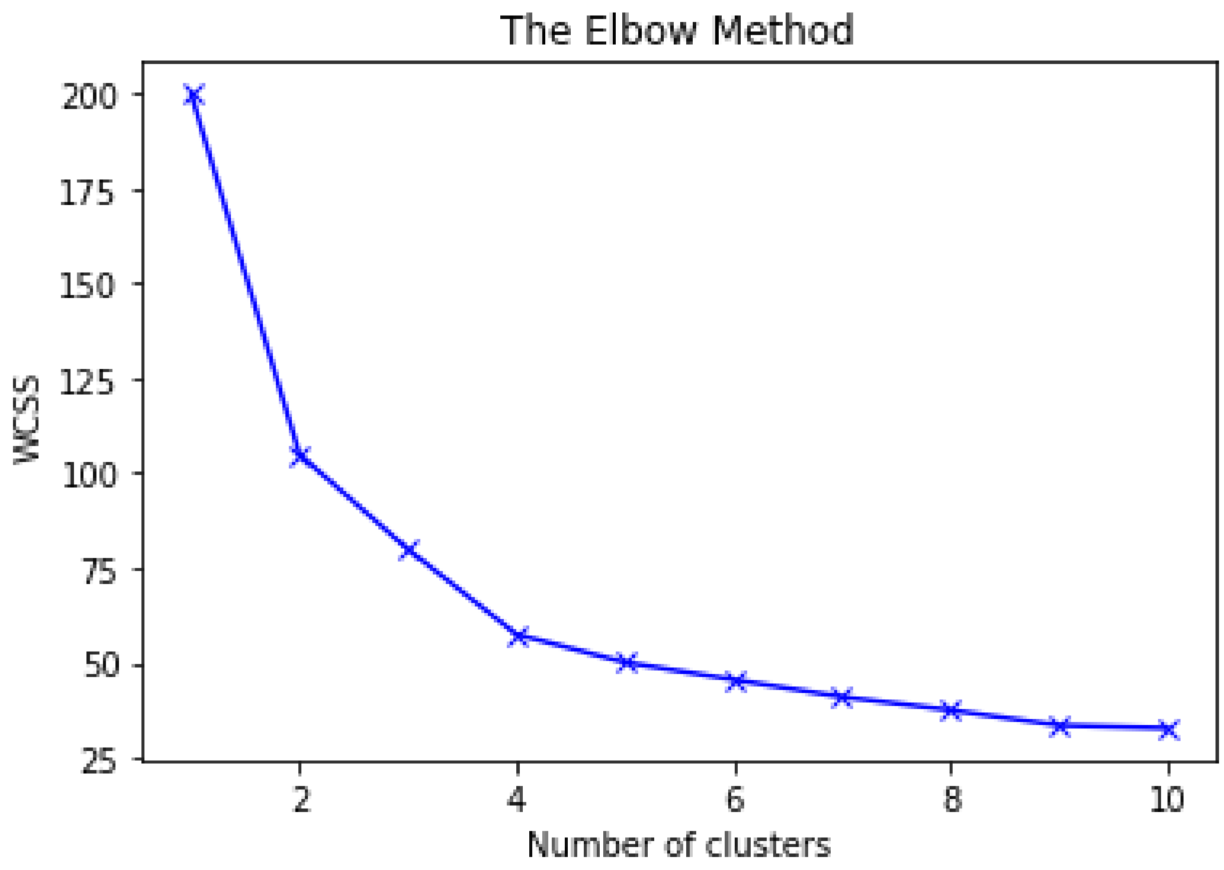
```
In [11]: # Naming the 1st Unnamed column as States

In [12]: df.rename(columns={'Unnamed: 0':'States'}, inplace=True)

In [13]: # Setting States column as index

In [14]: df.set_index('States', inplace=True)

In [15]: df
Out[15]:
            Murder  Assault  UrbanPop  Rape
States
Alabama       13.2      236        58  21.2
Alaska        10.0      263        48  44.5
Arizona        8.1      294        80  31.0
Arkansas       8.8      190        50  19.5
California     9.0      276        91  40.6
```

Scaling of variables

```
In [16]: # Creating X variable from df without States

In [17]: X = df[['Murder', 'Assault', 'Rape', 'UrbanPop']]

In [18]: # Scaling X variable

In [19]: scaler = StandardScaler()

In [20]: X_scaled = scaler.fit_transform( X )
```

Elbow Plot

```
In [21]: # Plotting for optimum nos of clusters

In [22]: plt.figure(figsize=(10, 8))
Out[22]: <Figure size 720x576 with 0 Axes><Figure size 720x576 with 0 Axes>

In [23]: wcss = []

In [24]: #____below in one block

In [25]: for i in range(1, 11):
    ...:     kmeans = KMeans(n_clusters = i, init = 'random', random_state = 42)
    ...:     kmeans.fit(X_scaled)
    ...:     wcss.append(kmeans.inertia_)
    ...:
    ...:

In [26]: plt.plot(range(1, 11), wcss, 'bx-')
    ...: plt.title('The Elbow Method')
    ...: plt.xlabel('Number of clusters')
    ...: plt.ylabel('WCSS')
    ...: plt.show()
```


The Elbow Method
200
175
150
125
100
75
50
25
WCSS
2
4
6
8
10
Number of clusters

Build clusters

```
In [27]: # Running kmeans to our optimal number of clusters

In [28]: kmeans = KMeans(n_clusters= 4)

In [29]: clusters = kmeans.fit_predict(X_scaled)

In [30]: clusters
Out[30]:
array([2, 1, 1, 2, 1, 1, 0, 0, 1, 2, 0, 3, 1, 0, 3, 0, 3, 2, 3, 1, 0, 1,
       3, 2, 1, 3, 3, 1, 3, 0, 1, 1, 2, 3, 0, 0, 0, 0, 0, 2, 3, 2, 1, 0,
       3, 0, 0, 3, 3, 0])
```

Cluster Membership

```
In [31]: # Naming Clusters 1-4 instead of 0-3 and adding to dataframe

In [32]: y_kmeans1 = clusters + 1

In [33]: cluster = list(y_kmeans1)

In [34]: df['cluster'] = cluster

In [35]: df.head()
Out[35]:
            Murder  Assault  UrbanPop  Rape  cluster
States
Alabama       13.2      236        58  21.2        3
Alaska        10.0      263        48  44.5        2
Arizona        8.1      294        80  31.0        2
Arkansas       8.8      190        50  19.5        3
California     9.0      276        91  40.6        2
```

Cluster-1

```
In [36]: # States and counts in different clusters

In [37]: df[df['cluster']==1]
Out[37]:
               Murder  Assault  UrbanPop  Rape  cluster
States
Connecticut       3.3      110        77  11.1        1
Delaware          5.9      238        72  15.8        1
Hawaii            5.3       46        83  20.2        1
Indiana           7.2      113        65  21.0        1
Kansas            6.0      115        66  18.0        1
Massachusetts     4.4      149        85  16.3        1
New Jersey        7.4      159        89  18.8        1
Ohio              7.3      120        75  21.4        1
Oklahoma          6.6      151        68  20.0        1
Oregon            4.9      159        67  29.3        1
Pennsylvania      6.3      106        72  14.9        1
Rhode Island      3.4      174        87   8.3        1
Utah              3.2      120        80  22.9        1
Virginia          8.5      156        63  20.7        1
Washington        4.0      145        73  26.2        1
Wyoming           6.8      161        60  15.6        1

In [38]: len(df[df['cluster']==1])
Out[38]: 16
```

Cluster-2

```
In [39]: df[df['cluster']==2]
Out[39]:
            Murder  Assault  UrbanPop  Rape  cluster
States
Alaska        10.0      263        48  44.5        2
Arizona        8.1      294        80  31.0        2
California     9.0      276        91  40.6        2
Colorado       7.9      204        78  38.7        2
Florida       15.4      335        80  31.9        2
Illinois      10.4      249        83  24.0        2
Maryland      11.3      300        67  27.8        2
Michigan      12.1      255        74  35.1        2
Missouri       9.0      178        70  28.2        2
Nevada        12.2      252        81  46.0        2
New Mexico    11.4      285        70  32.1        2
New York      11.1      254        86  26.1        2
Texas         12.7      201        80  25.5        2

In [40]: len(df[df['cluster']==2])
Out[40]: 13
```

Cluster-3

```
In [41]: df[df['cluster']==3]
Out[41]:
                Murder  Assault  UrbanPop  Rape  cluster
States
Alabama           13.2      236        58  21.2        3
Arkansas           8.8      190        50  19.5        3
Georgia           17.4      211        60  25.8        3
Louisiana         15.4      249        66  22.2        3
Mississippi       16.1      259        44  17.1        3
North Carolina    13.0      337        45  16.1        3
South Carolina    14.4      279        48  22.5        3
Tennessee         13.2      188        59  26.9        3

In [42]: len(df[df['cluster']==3])
Out[42]: 8
```

Cluster-4

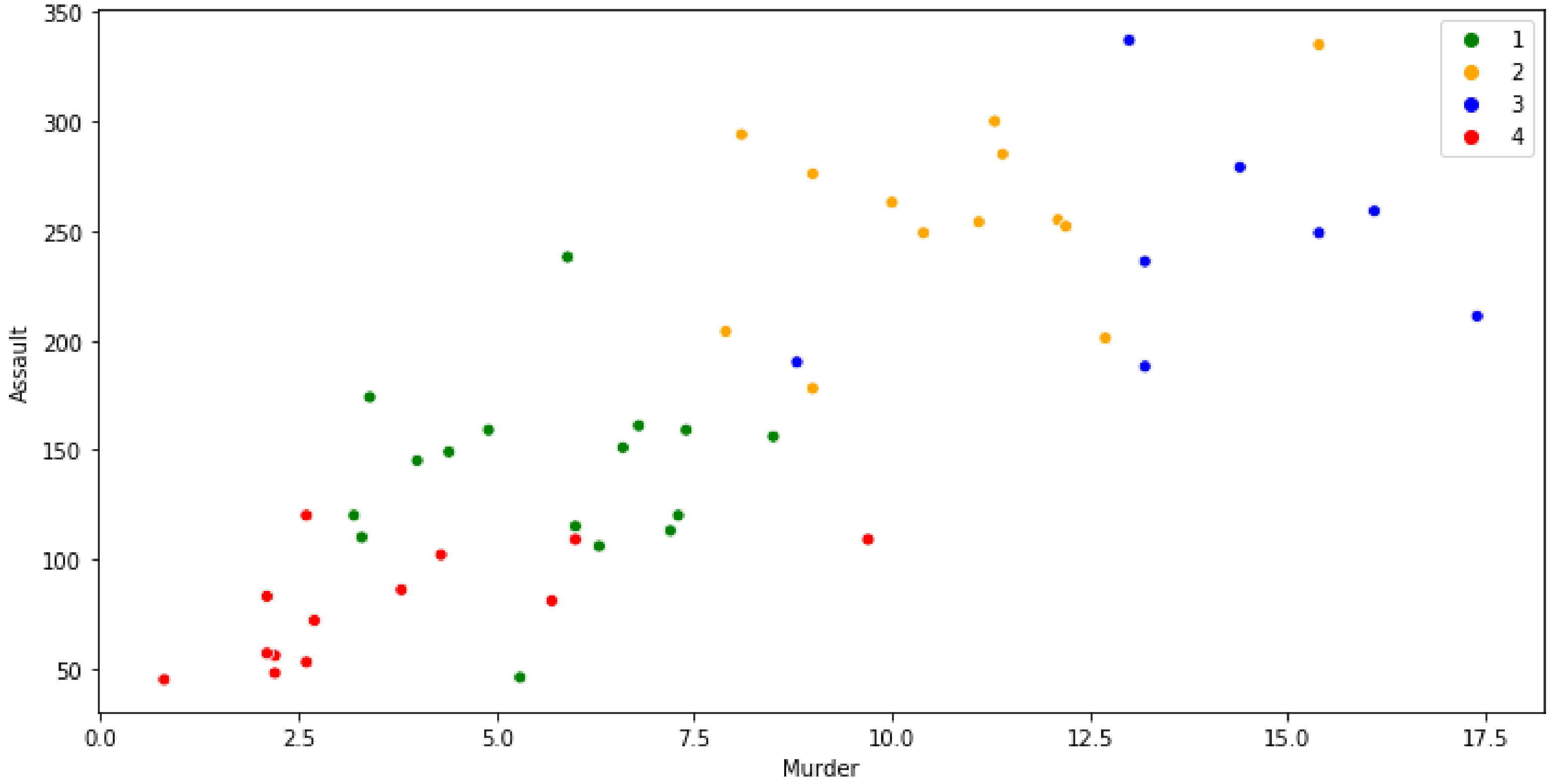
```
In [43]: df[df['cluster']==4]
Out[43]:
               Murder  Assault  UrbanPop  Rape  cluster
States
Idaho             2.6      120        54  14.2        4
Iowa              2.2       56        57  11.3        4
Kentucky          9.7      109        52  16.3        4
Maine             2.1       83        51   7.8        4
Minnesota         2.7       72        66  14.9        4
Montana           6.0      109        53  16.4        4
Nebraska          4.3      102        62  16.5        4
New Hampshire     2.1       57        56   9.5        4
North Dakota      0.8       45        44   7.3        4
South Dakota      3.8       86        45  12.8        4
Vermont           2.2       48        32  11.2        4
West Virginia     5.7       81        39   9.3        4
Wisconsin         2.6       53        66  10.8        4

In [44]: len(df[df['cluster']==4])
Out[44]: 13
```

Cluster Profiling

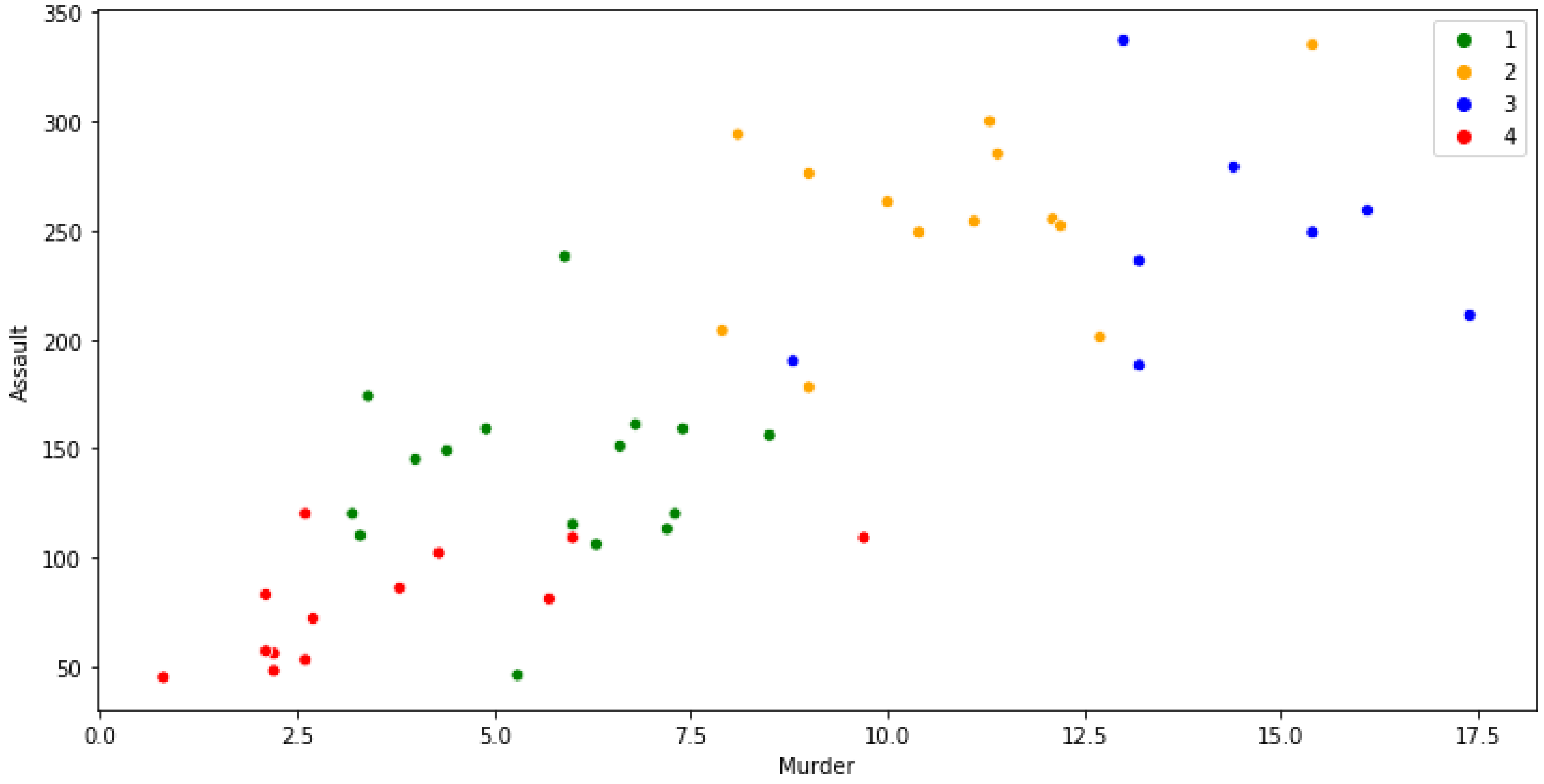
```
In [45]: # Mean of clusters 1 to 4

In [46]: kmeans_mean_cluster = pd.DataFrame(round(df.groupby('cluster').mean(),1))

In [47]: kmeans_mean_cluster
Out[47]:
         Murder  Assault  UrbanPop  Rape
cluster
1           5.7    138.9      73.9  18.8
2          10.8    257.4      76.0  33.2
3          13.9    243.6      53.8  21.4
4           3.6     78.5      52.1  12.2
```

Plot

```
In [48]: x = df['Murder']
    ...: y = df['Assault']
    ...: plt.figure(figsize=(12,6))
    ...: sns.scatterplot(x, y, hue=y_kmeans1,
    ...:     palette=['green','orange','blue','red'], legend='full')
    ...:
Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x23393fb3710>
```



Plot



350
300
250
200
150
100
50
Assault
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Murder
1
2
3
4

Cophenetic Correlation

Average Linkage

	A	B	C	D	E
A	*	6.32 [7.49]	7.07 [7.49]	1.41 [1.41	7.16 [7.49]
B		*	1.41 [1.74]	7.62 [7.49]	1.12 [1.12]
C			*	8.25 [7.49]	2.06 [1.74]
D				*	8.50 [7.49]
E					*

$$We\ merged\ b\ \&\ e\ with\ (b, e)\ at\ a\ distance\ 1.12$$

$$Then\ we\ merged\ a\ \&\ d\ with\ (a, d)\ at\ a\ distance\ 1.41$$

$$We\ merged\ c\ with\ ((b, e), c)\ at\ a\ distance\ 1.74$$

$$Then\ we\ merged\ (a, d)\ with\ ((b, e), c)\ at\ a\ distance\ 7.49$$

Average Linkage	
Dist	**Merged at dist**
6.32	7.49
7.07	7.49
1.41	1.41
7.16	7.49
1.41	1.74
7.62	7.49
1.12	1.12
8.25	7.49
2.06	1.74
8.5	7.49
Correl =	**0.980423166**

Silhouette Analysis

Silhouette analysis can be used to determine the degree of separation between clusters. For each sample:

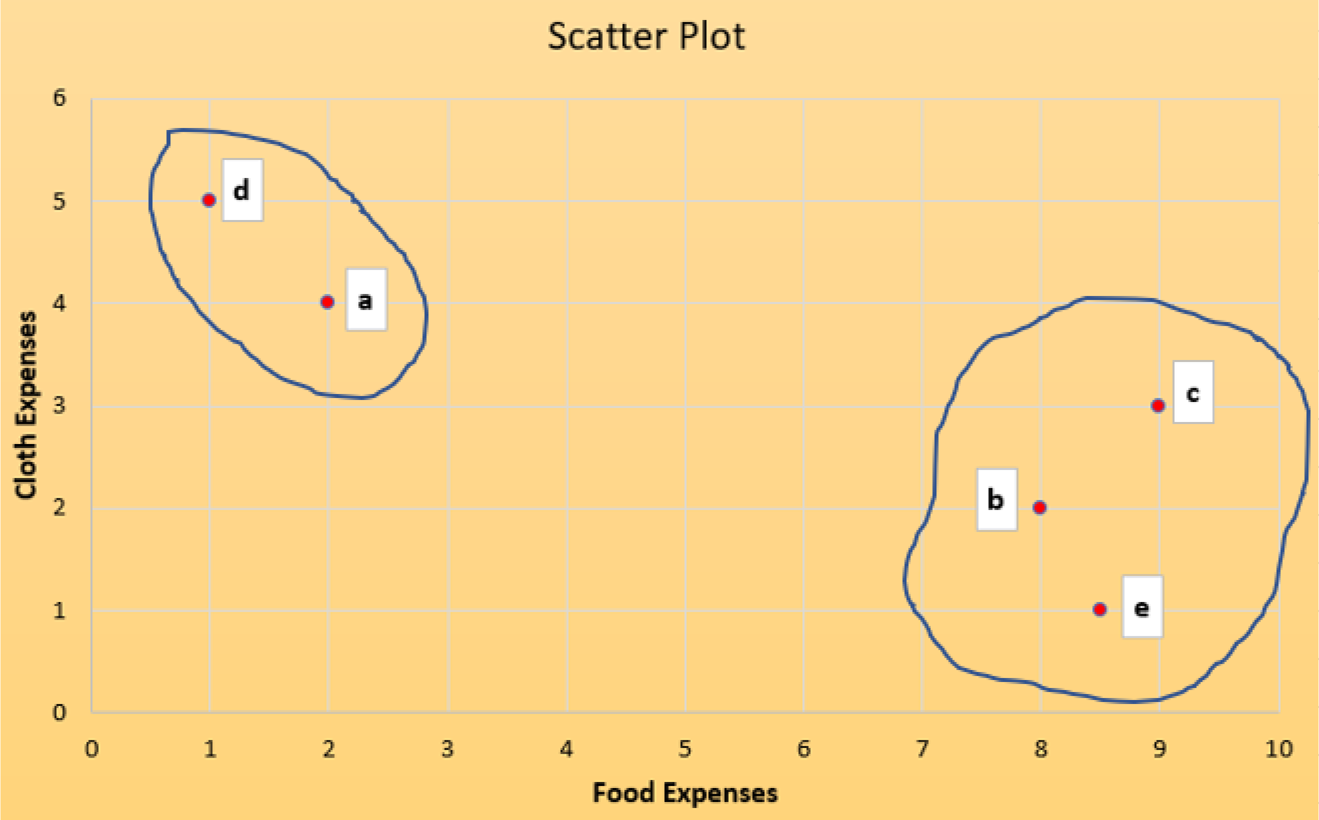
- Compute the average distance from all data points in the same cluster (ai).
- Compute the average distance from all data points in the closest cluster (bi).
- Compute the coefficient:

$$\frac{b^i - a^i}{max(a^i, b^i)}$$

The coefficient can take values in the interval [-1, 1].

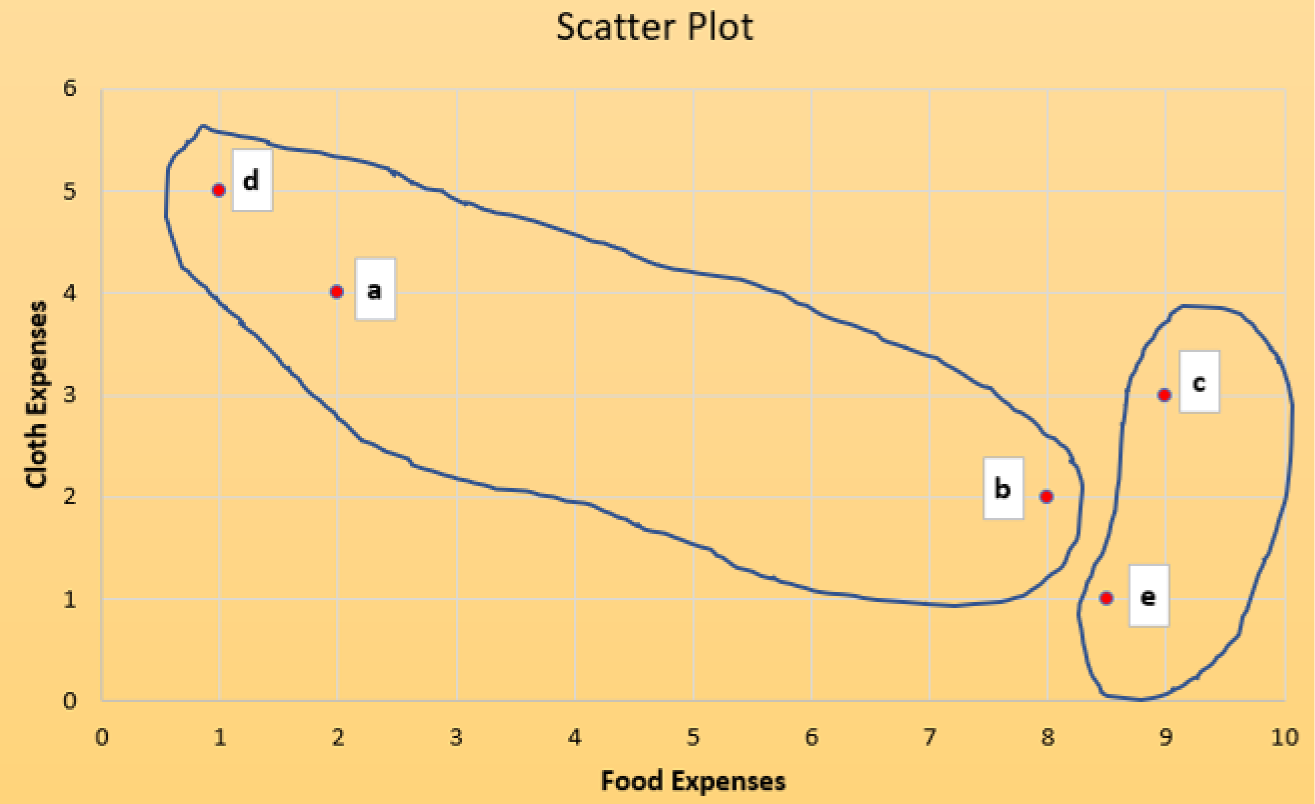
- If it is 0 –> the sample is very close to the neighbouring clusters.
- If it is 1 –> the sample is far away from the neighbouring clusters.
- If it is -1 –> the sample is assigned to the wrong clusters.

Silhoutte_score (K Means)



	Silhoutte Score for a		
		d	
C1	ad	1.41	A
C2	ab	6.32	
C2	ac	7.07	
C2	ae	7.16	
	AVG =	6.85	B
	Sil_Sc(a)=	0.794161	
	As it is close to 1,		
	righty allotted C1		

Silhoutte_score (K Means)



Silhoutte Score for b			
		d	
C1	ba	6.32	
C1	bd	7.62	
	AVG=	6.97	A
C2	be	1.12	
C2	bc	1.41	
	AVG=	1.265	B
	Sil_Sc(b)=	-0.81851	
	As it is close to -1,		
	wrongly allotted to C1		

Artificial Neural Networks