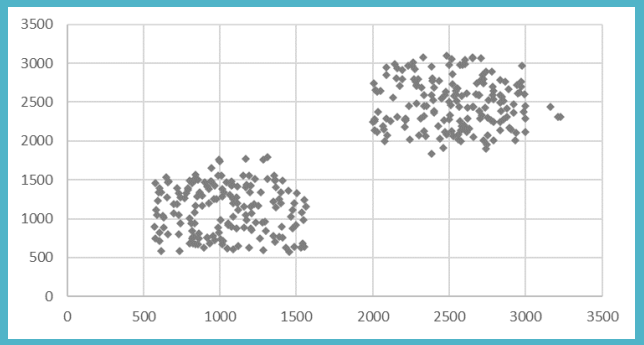
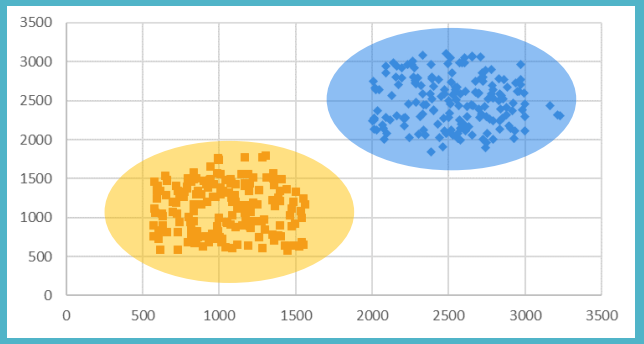
Clustering

(Seminar Solutions)





Dr. Jason Lines
j.lines@uea.ac.uk

University of East Anglia

1. *k*-means Clustering

Given the following data:

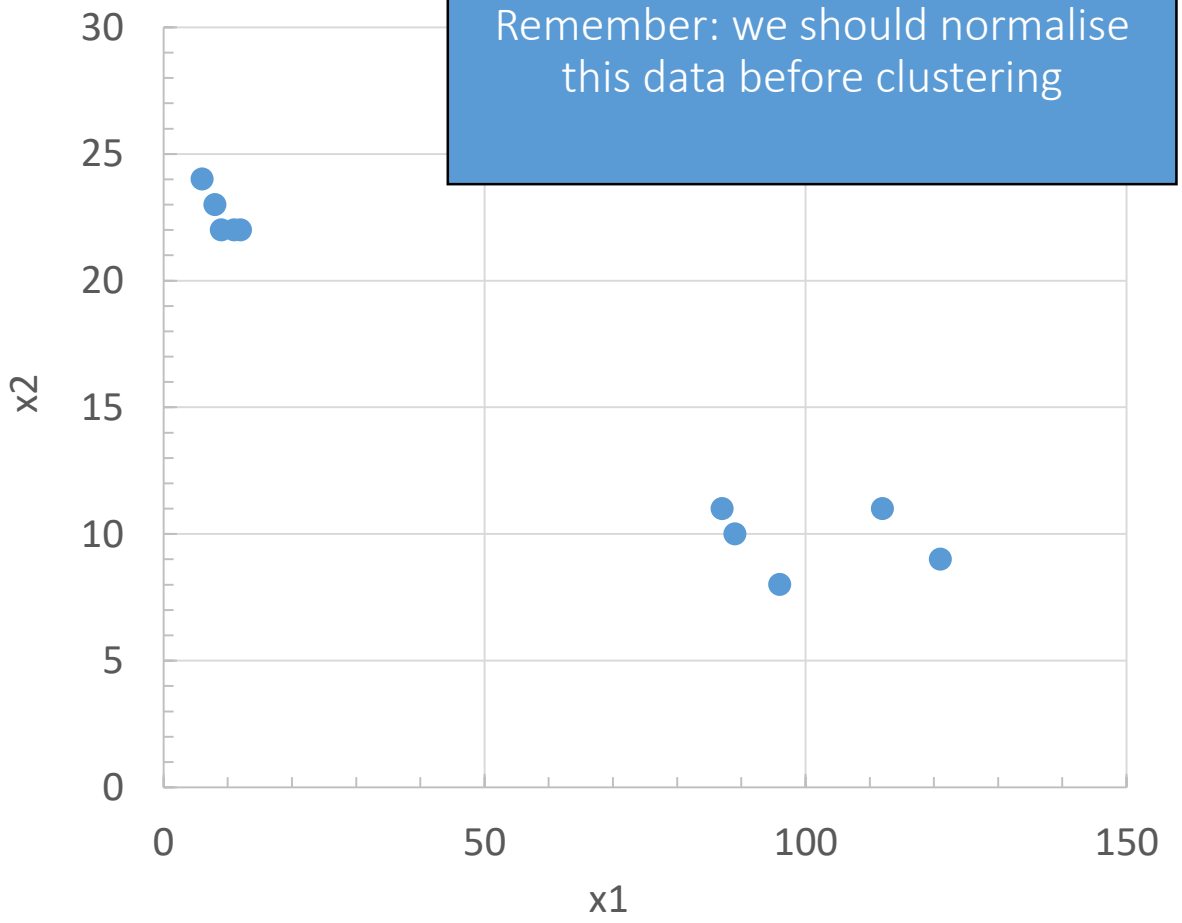
x_1	x_2
96	8
121	9
89	10
87	11
112	11
9	22
6	24
8	23
11	22
12	22

Cluster the data using *k*-means clustering (k=2). You should normalise the data, and then initialise the two centroids to the positions of the first two data (remember: k-means doesn't use medoids – we are just starting the centroids in the same place).

Complete three iterations of the k-means algorithm (or stop early if membership doesn't change for an iteration)

1. Our Problem –Raw Data

x1	x2
96	8
121	9
89	10
87	11
112	11
9	22
6	24
8	23
11	22
12	22

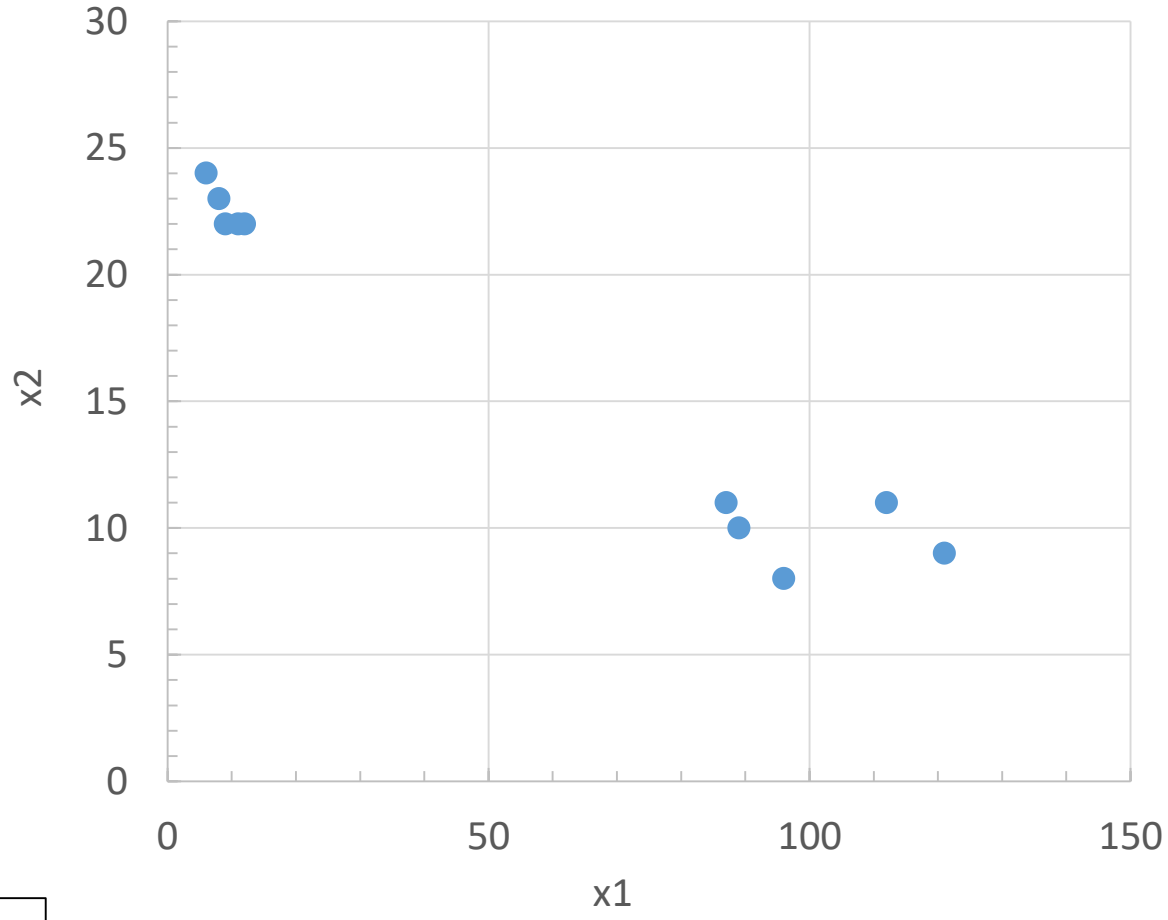


Remember: we should normalise this data before clustering

1. Our Problem –Raw Data

x1	x2
96	8
121	9
89	10
87	11
112	11
9	22
6	24
8	23
11	22
12	22

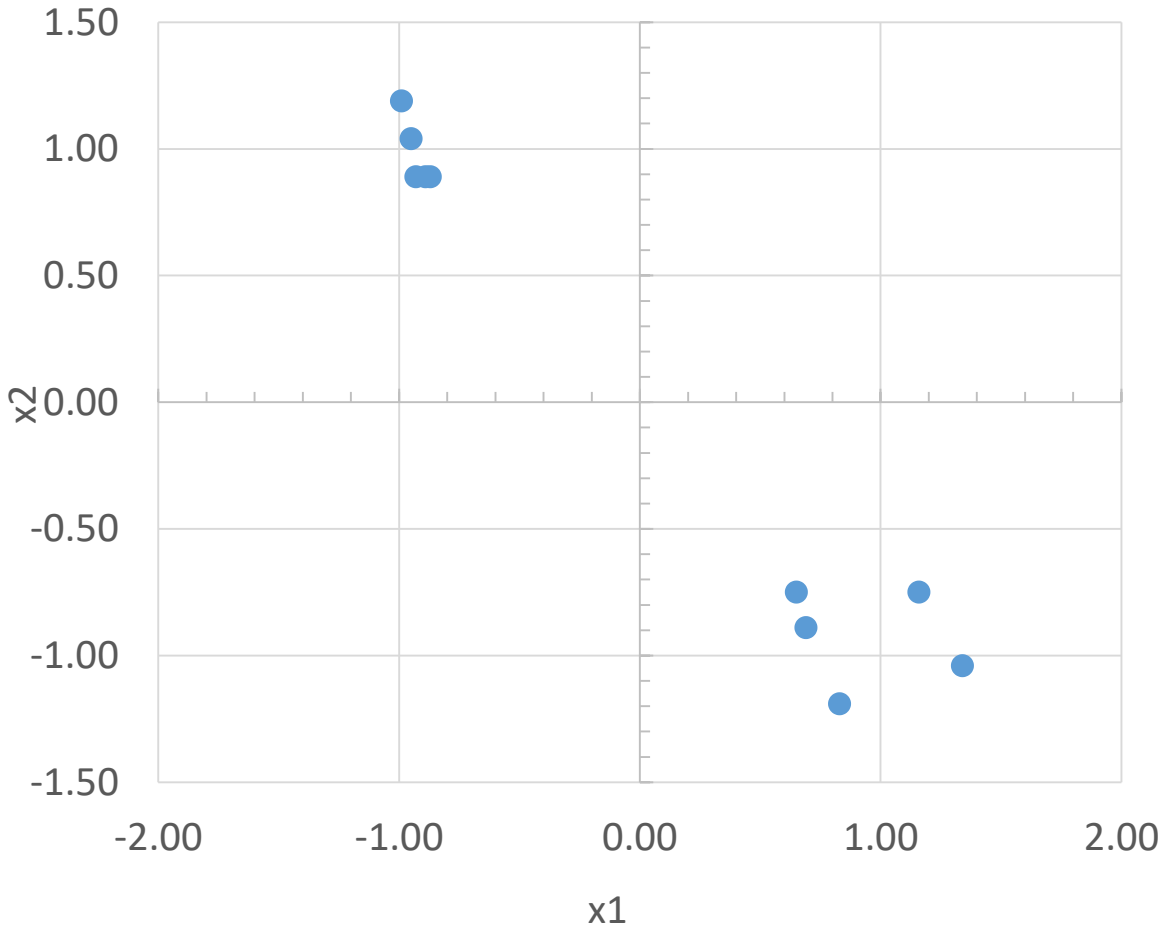
Mean	55	16
Stdev	49.33	6.71



Reminder $a_i' = \frac{a_i - \bar{a}}{stdev(a)}$

1. Our Problem – Standard Normalisation

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89



Question 1

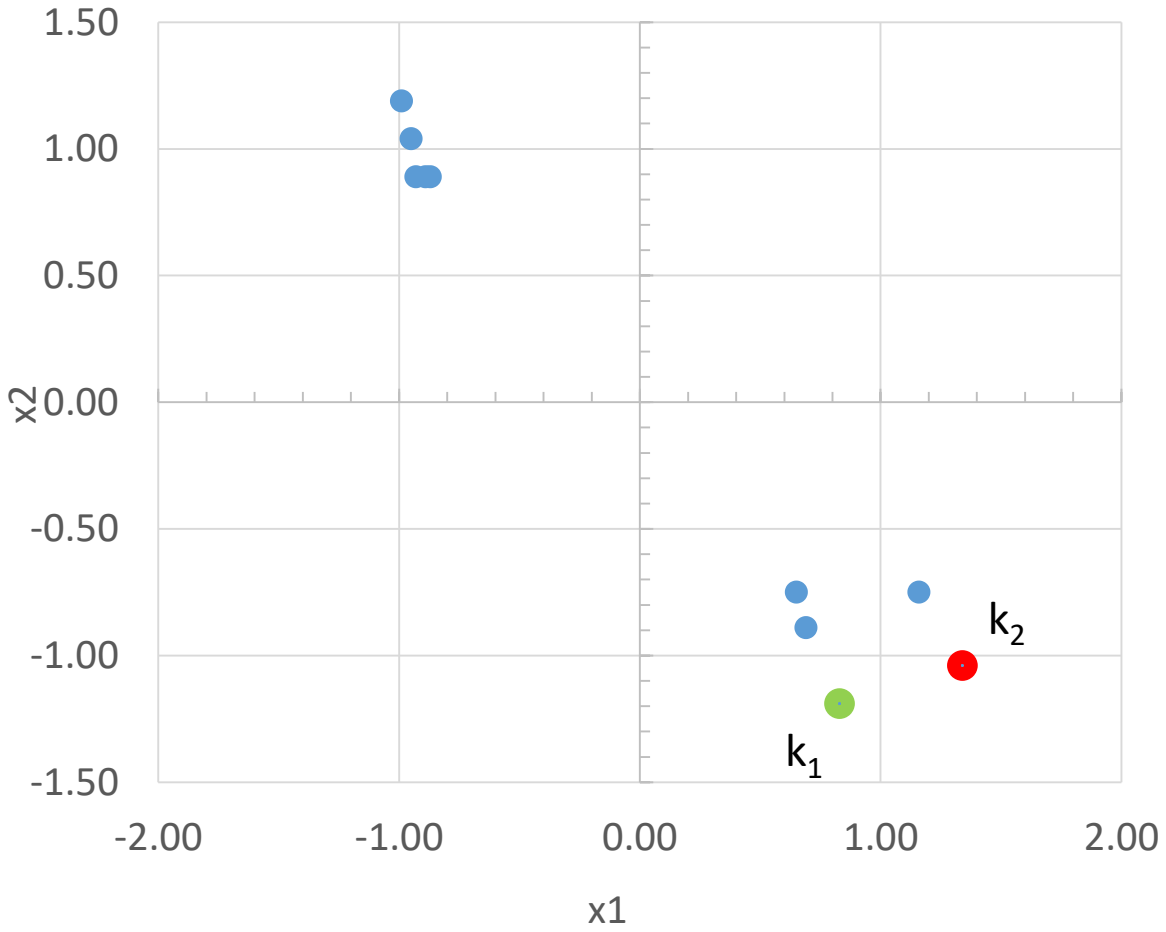
1. With k=2, start with the first two data as the initial **centroids**
   - Note: we're doing *k*-means, so we are using **centroids**, not medoids (the starting values just happen to be the same as the first two instances, but we're not constrained to using instances as the final cluster centers.)

2. Stop after either:
   a) Membership doesn't change
   b) 3 iterations have been performed

(As an extra exercise, cluster with k=3 and k=4 and decide which is better)

1. Our Problem – with Standard Normalisation

	x1	x2
k_1	0.83	-1.19
k_2	1.34	-1.04
	0.69	-0.89
	0.65	-0.75
	1.16	-0.75
	-0.93	0.89
	-0.99	1.19
	-0.95	1.04
	-0.89	0.89
	-0.87	0.89



Iteration 1: Calculate Distances to Centroids

k_1 =

x1	x2
0.83	-1.19

k_2 =

x1	x2
1.34	-1.04

x1	x2	Dist K_1	Dist K_2
0.83	-1.19		
1.34	-1.04		
0.69	-0.89		
0.65	-0.75		
1.16	-0.75		
-0.93	0.89		
-0.99	1.19		
-0.95	1.04		
-0.89	0.89		
-0.87	0.89		

$$\text{Remember:} \quad \text{EuclideanDistance}(x, y) \equiv \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Iteration 1: Calculate Distances to Centroids

k_1 =

x1	x2
0.83	-1.19

k_2 =

x1	x2
1.34	-1.04

Results after square root

x1	x2	Dist K_1	Dist K_2
0.83	-1.19	0	0.53
1.34	-1.04	0.53	0
0.69	-0.89	0.33	0.67
0.65	-0.75	0.48	0.75
1.16	-0.75	0.55	0.34
-0.93	0.89	2.72	2.98
-0.99	1.19	3.00	3.23
-0.95	1.04	2.85	3.09
-0.89	0.89	2.70	2.95
-0.87	0.89	2.69	2.93

Iteration 1: Assign Cluster Membership

k_1 =

x1	x2
0.83	-1.19

k_2 =

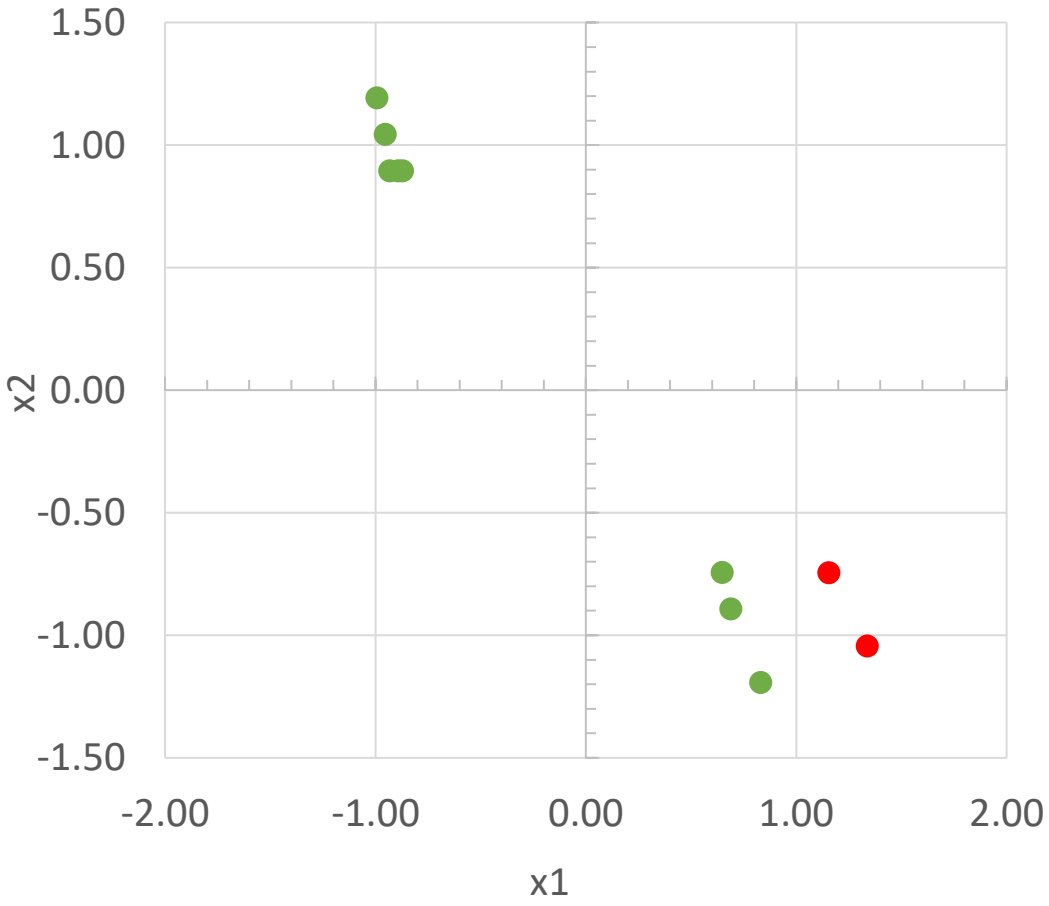
x1	x2
1.34	-1.04

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89

Dist K_1	Dist K_2
0	0.53
0.53	0
0.33	0.67
0.48	0.75
0.55	0.34
2.72	2.98
3.00	3.23
2.85	3.09
2.70	2.95
2.69	2.93

Iteration 1: New Clusters

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89



Recalculate centroids

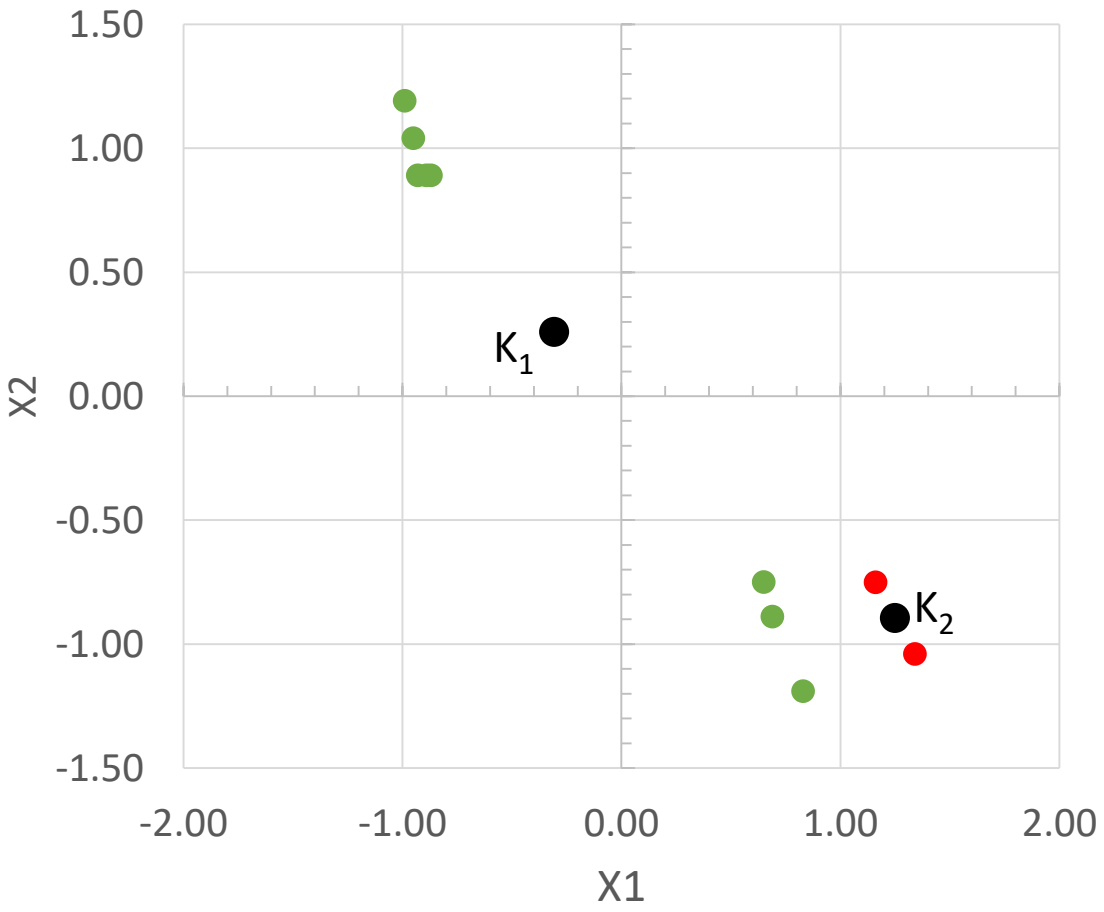
Iteration 2: New Centroids

K_1 =

x1	x2
0.83	-1.19
0.69	-0.89
0.65	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89
Avg	Avg
-0.31	0.26

K_2 =

x1	x2
1.34	-1.04
1.16	-0.75
Avg	Avg
1.25	-0.90



Iteration 2: Recalculate Membership

k_1 =

x1	x2
-0.31	0.26

k_2 =

x1	x2
1.25	-0.90

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89

Iteration 2: Recalculate Membership

k_1 =

x1	x2
-0.31	0.26

k_2 =

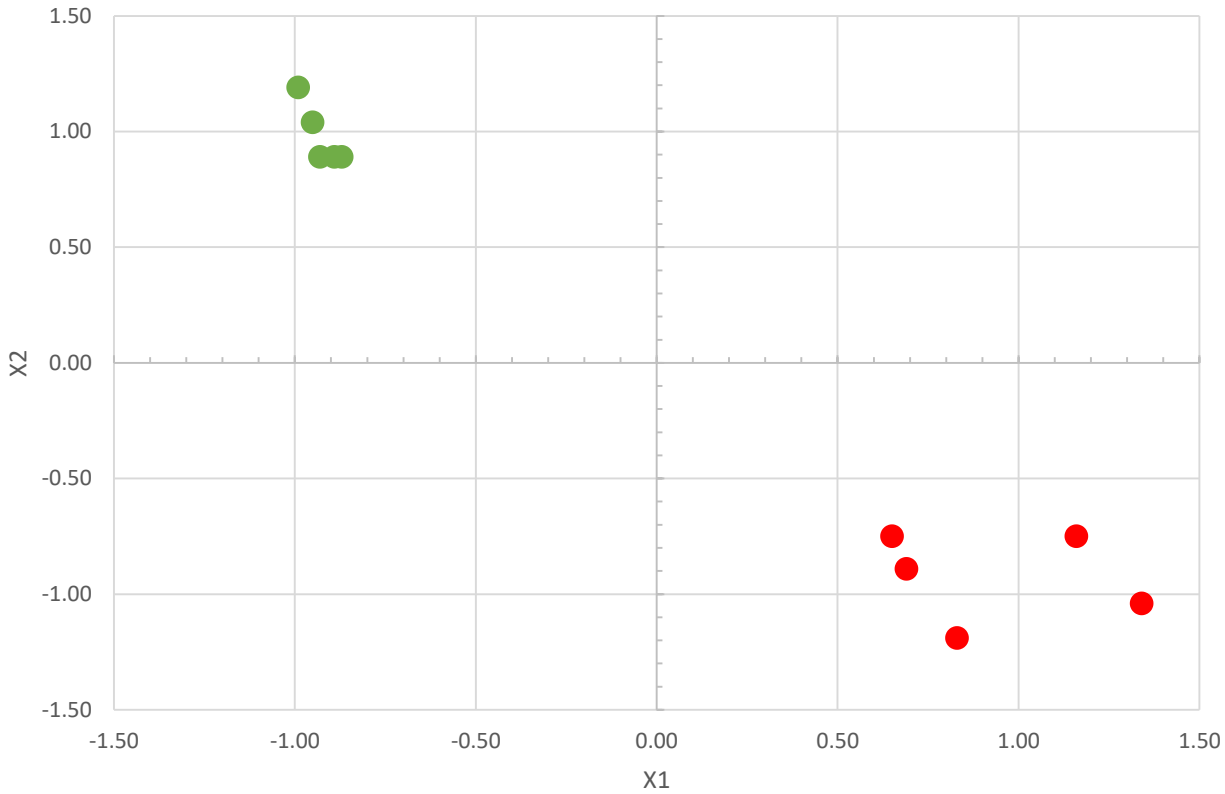
x1	x2
1.25	-0.90

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89

Dist K_1	Dist K_2
1.84	0.51
2.10	0.17
1.52	0.56
1.39	0.62
1.78	0.17
0.89	2.82
1.15	3.06
1.01	2.93
0.86	2.79
0.85	2.77

Iteration 2: New Clusters

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89



Recalculate centroids

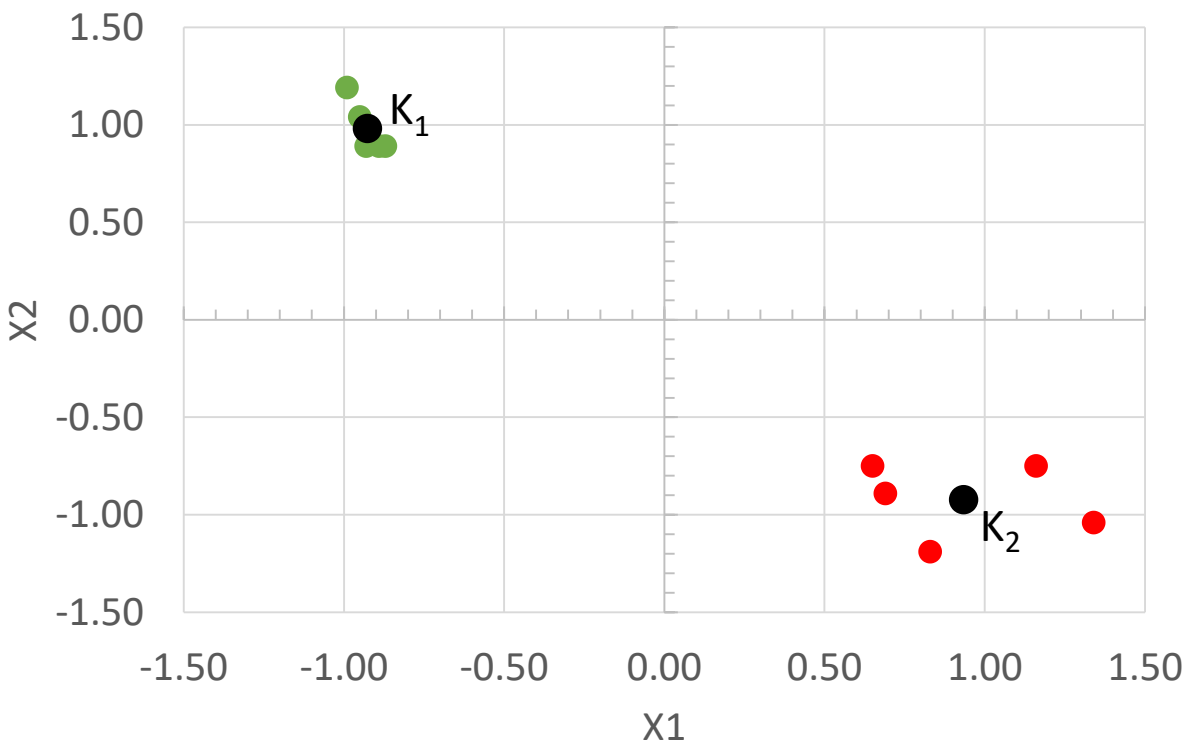
Iteration 3: New Centroids

K_1 =

x1	x2
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89
Avg	Avg
-0.93	0.98

K_2 =

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
Avg	Avg
0.93	-0.92



Iteration 3: Recalculate Membership

k_1 =

x1	x2
-0.93	0.98

k_2 =

x1	x2
0.93	-0.92

x1	x2
0.83	-1.19
1.34	-1.04
0.69	-0.89
0.65	-0.75
1.16	-0.75
-0.93	0.89
-0.99	1.19
-0.95	1.04
-0.89	0.89
-0.87	0.89

Iteration 3: Recalculate Membership

k_1 =

x1	x2
-0.93	0.98

k_2 =

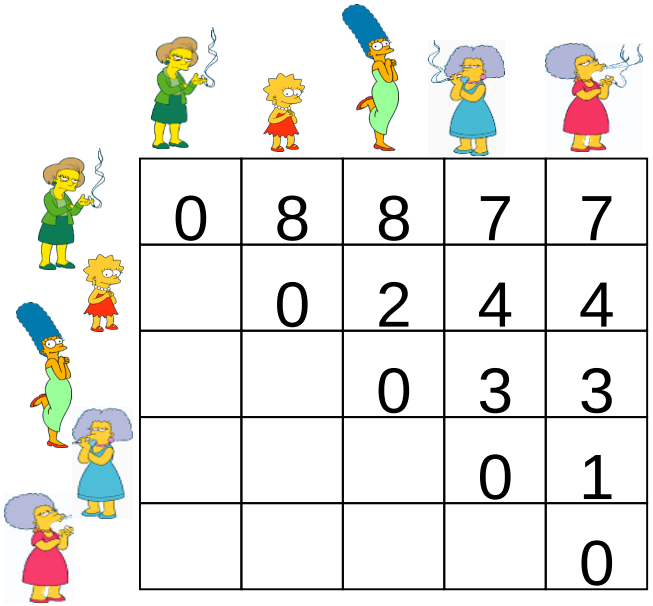
x1	x2
0.93	-0.92

x1	x2	Dist K_1	Dist K_2
0.83	-1.19	2.79	0.29
1.34	-1.04	3.04	0.43
0.69	-0.89	2.47	0.24
0.65	-0.75	2.34	0.33
1.16	-0.75	2.71	0.29
-0.93	0.89	0.09	2.60
-0.99	1.19	0.22	2.85
-0.95	1.04	0.06	2.72
-0.89	0.89	0.10	2.57
-0.87	0.89	0.11	2.55

No Change - Stop

Question 2

- Cluster the characters using k=2 and PAM as the clustering algorithm.
  - Use Edna and Lisa as the initial **medoids**

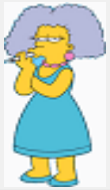
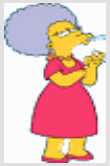


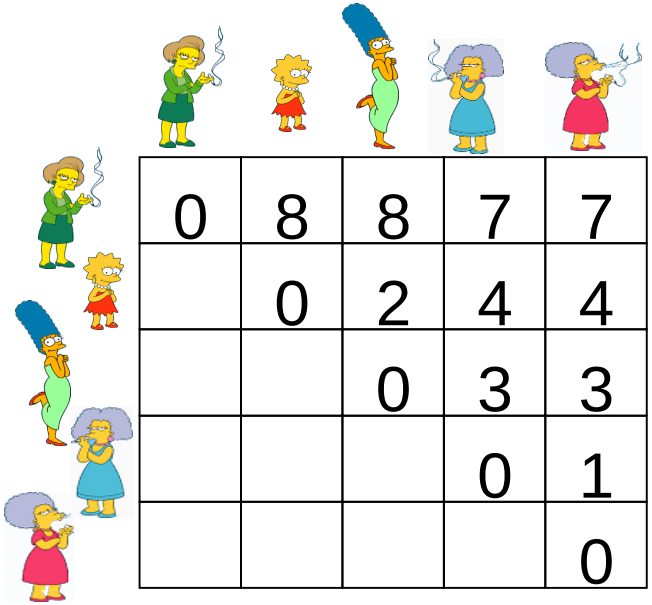
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Iteration 1: Work out cluster membership

- Current medoids: 

- Distances from other data to the medoids:

		
	8	2
	7	4
	7	4



0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Iteration 1: Work out medoids

Edna's Cluster (we'll call it K1)

- Just her, so we don't need to do anything as we can't switch medoids

Lisa's cluster (K2)

- Try each member as the medoid
- Calculate the within-cluster sum of distances for each medoid
- Choose the smallest sum as the new medoid

If any medoids change, iterate through again to recalculate memberships. Else, stop.

Iteration 1: Within-cluster sum for K2

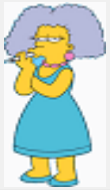
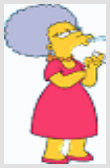
	0	2	4	4	= 10
	2	0	3	3	= 8
	4	3	0	1	= 8
	4	3	1	0	= 8

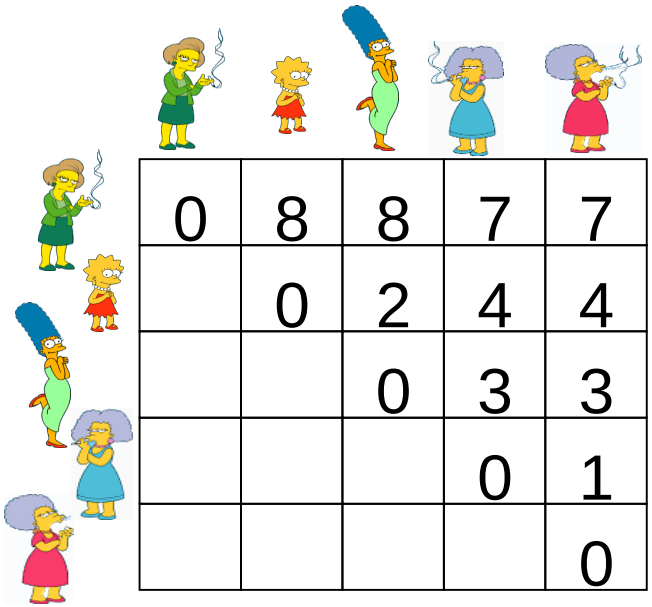
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Medoid changed – Recalculate membership

Iteration 2: Work out cluster membership

- Current medoids: 
- Distances from other data to the medoids:

		
	8	2
	7	3
	7	3

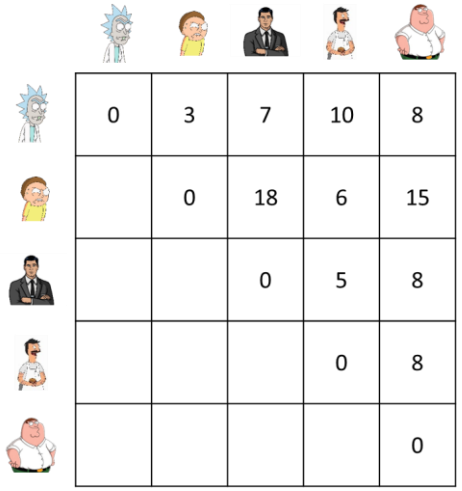


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

No change - Stop

Question 3: Hierarchical

Cluster the following data using bottom-up average-linkage clustering



0	3	7	10	8
	0	18	6	15
		0	5	8
			0	8
				0

To reiterate:

- Start with each data in it's own cluster (unlike top-down, which has all data in a single cluster)
- Find best two clusters to merge (the two with the smallest distance/dissimilarity)
- Repeat for all clusters until we have a single cluster
- For k=2, split at the highest level in the dendrogram where there are two clusters

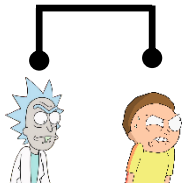
Question 3:
Bottom Up with Average Linkage

					
	0	3	7	10	8
		0	18	6	15
			0	5	8
				0	8
					0

Step 1:
Enumerate all possible combinations and pick the one with the smallest distance

(the smallest distance in the matrix is 3 so it must be that to begin with)

D( , ) = 3



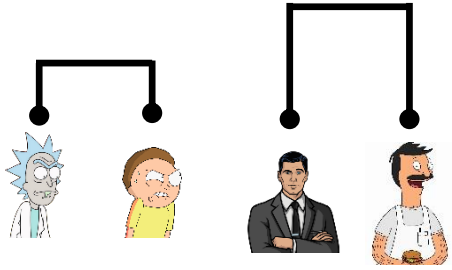
Question 3:
Bottom Up with Average Linkage

					
	0	3	7	10	8
		0	18	6	15
			0	5	8
				0	8
					0

Step 2:
Enumerate all possible combinations and pick the one with the smallest distance

(the next smallest is 5, which isn't to the existing cluster, so the next move must be this)

D( , ) = 5



Question 3:
Bottom Up with Average Linkage

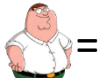
					
	0	3	7	10	8
		0	18	6	15
			0	5	8
				0	8
					0

Step 3:

We can add Peter to either cluster, or instead combine those two clusters

REMEMBER: group average linkage

D(( , ) +  = (8+15)/2 = 11.5

D(( , ) +  = (8+8)/2 = 8

D(( , ) + D( , ) = (7+10+18+6)/4
= 10.25

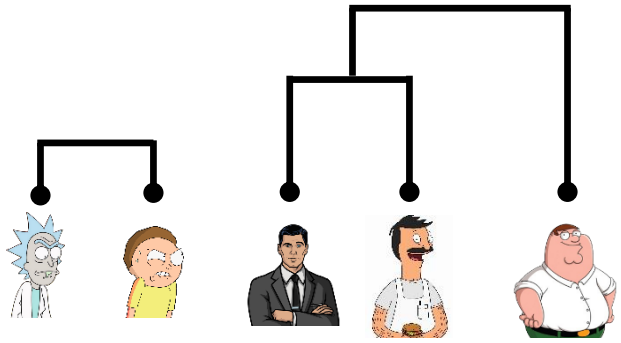
Question 3:
Bottom Up with Average Linkage

	0	3	7	10	8
		0	18	6	15
			0	5	8
				0	8
					0

Step 3:

D(( , ) +  = (8+8)/2 = 8

Step 4:
only 1 option left, join (Rick, Morty) with
(Archer, Bob, Peter)



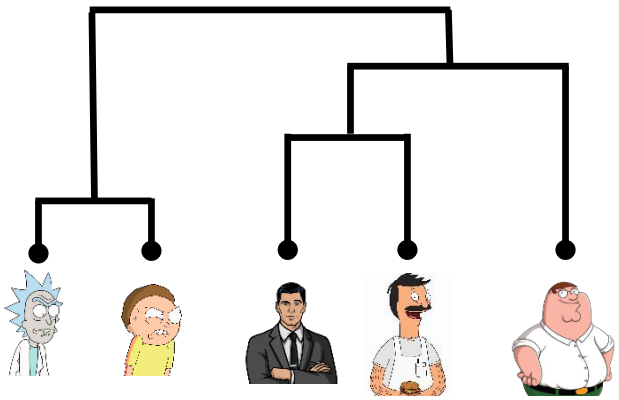
Question 3:
Bottom Up with Average Linkage

0	3	7	10	8
	0	18	6	15
		0	5	8
			0	8
				0

Step 3:

D((,) + = (8+8)/2 = 8

Step 4:
only 1 option left, join (Rick, Morty) with
(Archer, Bob, Peter)



Question 4

Given the following data:

id	x	y
1	5	20
2	19	7
3	16	9
4	17	5
5	7	25
6	6	22
7	4	24

Informal Algorithm: Density Peaks

Five steps:

1. Calculate the distance matrix for the input data
2. Find the local density value for each data object (ρ). This quantifies how many other objects are "close" to each object.
3. For each object, find the distance to its nearest neighbour with a higher local density (δ)
4. Select cluster centres (medoid) using ρ and δ
5. Assign cluster labels to objects based on closest centre

Use the density peaks algorithm, with a cut-off kernel and $d_c=4$, to cluster the data with k=2. You do **not** need to normalise this data

Step 1: Calculate the distance Matrix

Distance measure isn't specified, so default to Euclidean Distance

id	x	y
1	5	20
2	19	7
3	16	9
4	17	5
5	7	25
6	6	22
7	4	24

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

e.g.

$$D(id = 1, id = 5) = \sqrt{(5-7)^2+(20-25)^2} = \sqrt{4+25} = 5.39$$

$$D(id = 6, id = 4) = \sqrt{(6-17)^2+(22-5)^2} = \sqrt{121+289} = 20.25$$

Step 3: For each object, find the distance to its nearest neighbour with a higher local density (δ)

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)
6	3	
2	2	
5	2	
7	2	
1	1	
3	1	
4	1	

Step 3: For each object, find the distance to its nearest neighbour with a higher local density (δ)

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)
6	3	23.02
2	2	
5	2	
7	2	
1	1	
3	1	
4	1	

Special case for the first value – set to max of the matrix

Step 3: For each object, find the distance to its nearest neighbour with a higher local density (δ)

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)
6	3	23.02
2	2	19.85
5	2	3.16
7	2	2.83
1	1	
3	1	
4	1	

Special case for the first value – set to max of the matrix

For 2, 5 and 7 – this can only be the distance to id=6, as its ρ is 3 and these have ρ=2

Step 3: For each object, find the distance to its nearest neighbour with a higher local density (δ)

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)
6	3	23.02
2	2	19.85
5	2	3.16
7	2	2.83
1	1	min(2.24, 19.10, 5.39, 4.12)
3	1	min(16.40, 3.61, 18.36, 19.21)
4	1	min(20.25, 2.83, 22.36,23.02)

Special case for the first value – set to max of the matrix

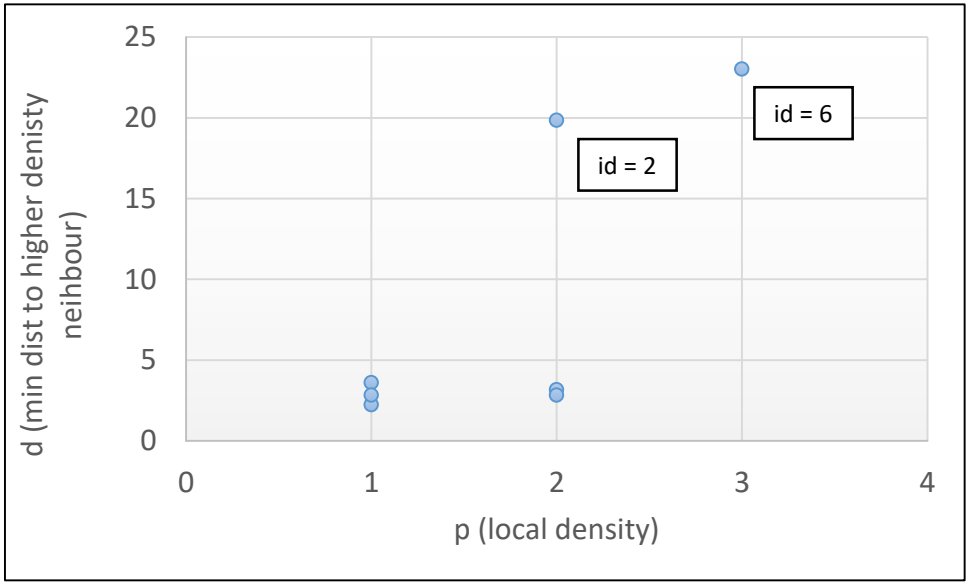
For 2, 5 and 7 – this can only be the distance to id=6, as its ρ is 3 and these have ρ=2

For 1, 3 and 4 – this will be min of the distances to id=6, 2, 5 or 7

Step 4: Select cluster centres

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

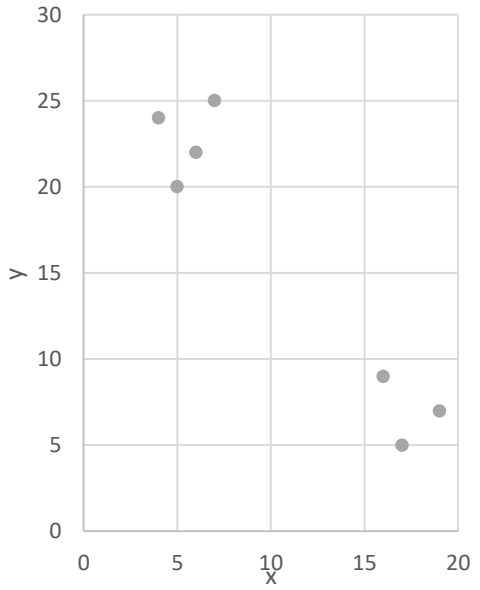
id	ρ	δ (min dist to neighbour with higher ρ)	ρ × δ
6	3	23.02	69.07
2	2	19.85	39.70
5	2	3.16	6.32
7	2	2.83	5.66
1	1	2.24	2.24
3	1	3.61	3.61
4	1	2.83	2.83



Step 5: Assign data to closest medoid

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

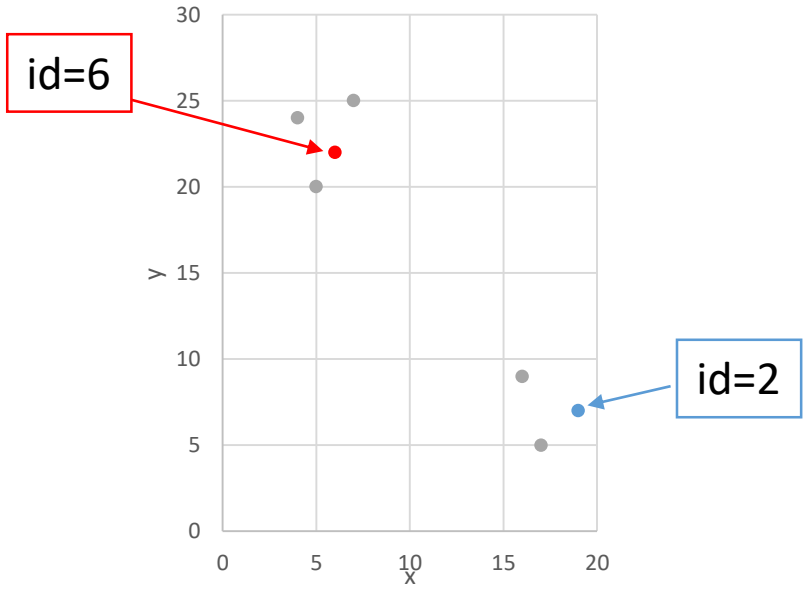
id	ρ	δ (min dist to neighbour with higher ρ)	Membership
6	3	23.02	C1
2	2	19.85	C2
5	2	3.16	
7	2	2.83	
1	1	2.24	
3	1	3.61	
4	1	2.83	



Step 5: Assign data to closest medoid

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)	Membership
6	3	23.02	C1
2	2	19.85	C2
5	2	3.16	
7	2	2.83	
1	1	2.24	
3	1	3.61	
4	1	2.83	



Step 5: Assign data to closest medoid

	1	2	3	4	5	6	7
1	0	19.10	15.56	19.21	5.39	2.24	4.12
2	19.10	0	3.61	2.83	21.63	19.85	22.67
3	15.56	3.61	0	4.12	18.36	16.40	19.21
4	19.21	2.83	4.12	0	22.36	20.25	23.02
5	5.39	21.63	18.36	22.36	0	3.16	3.16
6	2.24	19.85	16.40	20.25	3.16	0	2.83
7	4.12	22.67	19.21	23.02	3.16	2.83	0

id	ρ	δ (min dist to neighbour with higher ρ)	Membership
6	3	23.02	C1
2	2	19.85	C2
5	2	3.16	C1
7	2	2.83	C1
1	1	2.24	C1
3	1	3.61	C2
4	1	2.83	C2