2 Hierarchical Agglomerative Clustering (HAC)

(a)

Hierarchical agglomerative clustering is a type of hierarchical clustering algorithm, meaning that it outputs a set of nested clusters. With HAC, you start out with all of the data points as individual clusters, and then merge the clusters that are closest to each other until there is just one cluster left. You can visualize the clusters in a dendrogram, which is a tree-like structure that shows how all of the clusters are nested within each other. Then, you can decide how many clusters you want by simply cutting the tree where you want.

The biggest decision you have to make with HAC, is choosing how to compute the proximity/distance between two clusters. With MIN-link, you use the distance between the two points in each cluster that are closest to each other, while with MAX-link you use the distance between the two points in each cluster that are the furthest apart.

(b)

MIN-link

Datapoints:

ID	x	y
A	5	7
B	4	3
C	9	8
D	5	6
E	11	3

Proximity matrix:

Using Euclidean distance d(p, q) = $\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$

	A	B	C	D	E
A	0	4.12	4.12	1	7.21
B	4.12	0	7.07	3.16	7

C	4.12	7.07	0	4.47	5.39
D	1	3.16	4.47	0	6.71
E	7.21	7	5.39	6.71	0

First merge

In the proximity matrix, we can see that the lowest value is 1, which is the distance between A and D. Therefore, we merge A and D.

Second proximity matrix

	A,D	B	C	E
A,D	0	3.16	4.12	6.71
B	3.16	0	7.07	7
C	4.12	7.07	0	5.39
E	6.71	7	5.39	0

Second merge

In the proximity matrix, we can see that the lowest value is 3.16, which is the distance between (A,D) and B. Therefore, we merge (A,D) and B.

Third proximity matrix

	A,D,B	C	E
A,D,B	0	4.12	6.71
C	4.12	0	5.39
E	6.71	5.39	0

Third merge

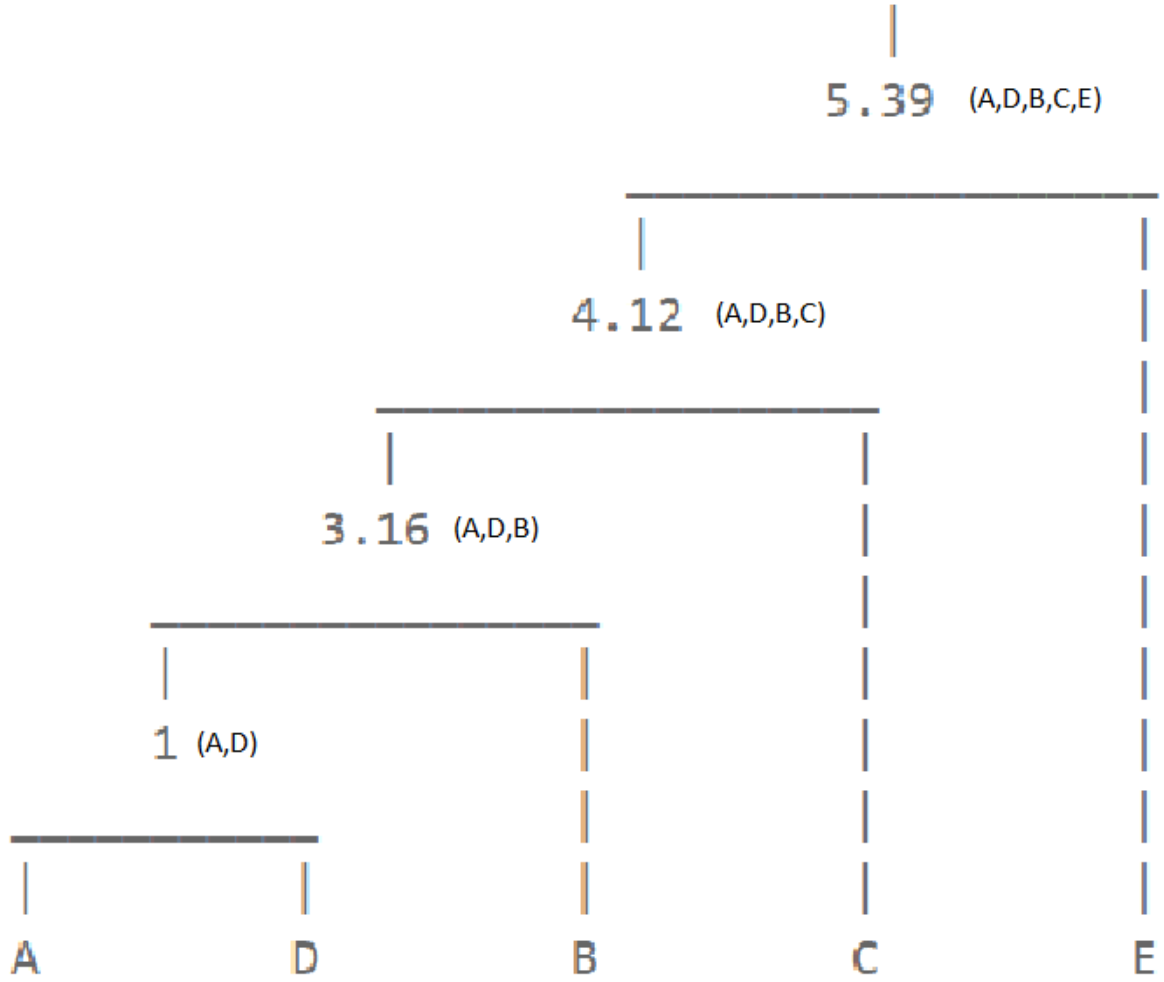
In the proximity matrix, we can see that the lowest value is 4.12, which is the distance between (A,D,B) and C. Therefore, we merge (A,D,B) and C.

Fourth proximity matrix

	A,D,B,C	E
A,D,B,C	0	5.39
E	5.39	0

Fourth merge

In the proximity matrix, we can see that the last distance is 5.39. We can now perform the last merge, and the dendrogram will look like this:



MAX-link

Datapoints:

ID	x	y
A	5	7
B	4	3
C	9	8
D	5	6
E	11	3

Proximity matrix:

Using Euclidean distance d(p, q) = $\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$

	A	B	C	D	E
A	0	4.12	4.12	1	7.21
B	4.12	0	7.07	3.16	7
C	4.12	7.07	0	4.47	5.39
D	1	3.16	4.47	0	6.71
E	7.21	7	5.39	6.71	0

First merge

The first merge is the same as with MIN-link, because each datapoint is a cluster. We therefore merge A and D.

Second proximity matrix (using distance between the points that are furthest apart)

	A,D	B	C	E
A,D	0	4.12	4.47	7.21

B	4.12	0	7.07	7
C	4.47	7.07	0	5.39
E	7.21	7	5.39	0

Second merge

In the proximity matrix, we can see that the lowest value is 4.12, which is the distance between (A,D) and B. Therefore, we merge (A,D) and B.

Third proximity matrix

	A,D,B	C	E
A,D,B	0	7.07	7.21
C	7.07	0	5.39
E	7.21	5.39	0

Third merge

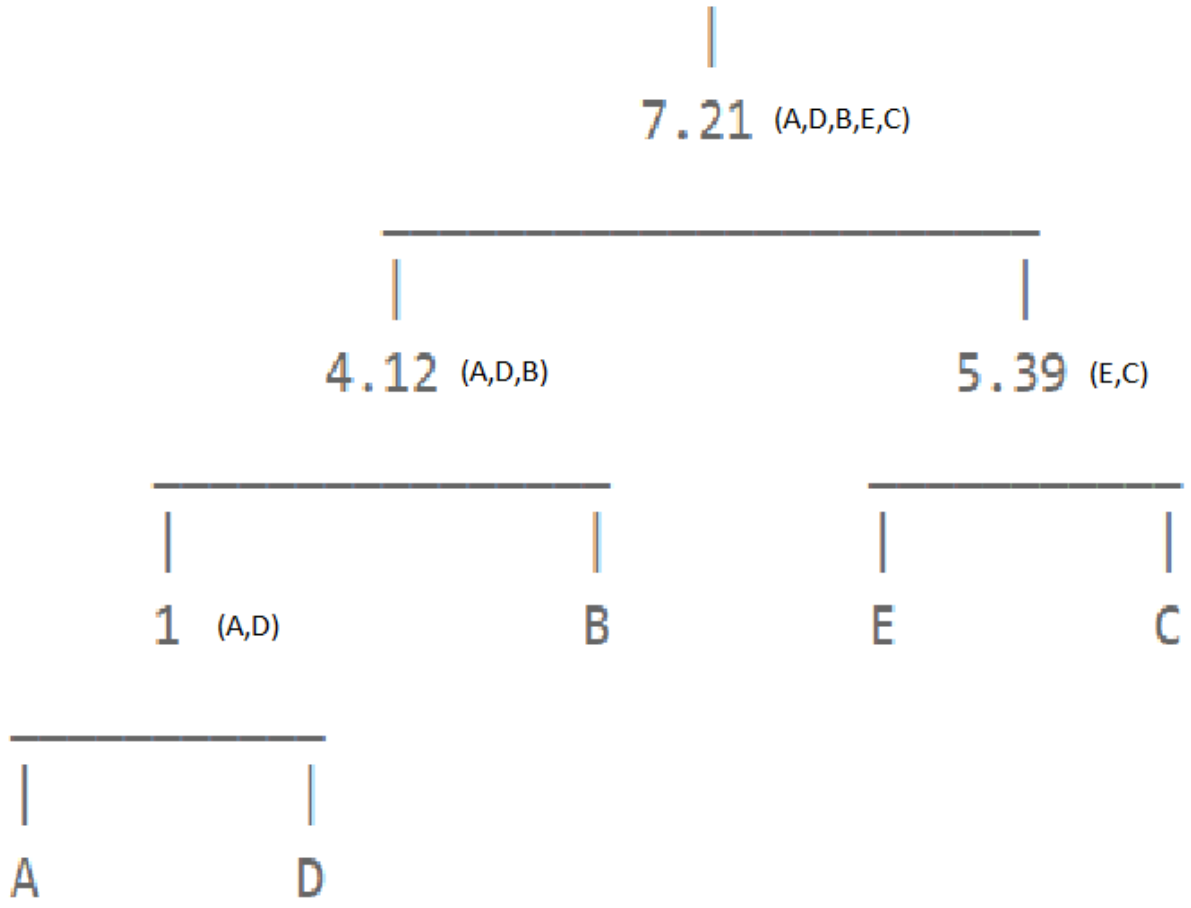
In the proximity matrix, we can see that the lowest value is 5.39, which is the distance between E and C. Therefore, we merge E and C.

Fourth proximity matrix

	A,D,B	E,C
A,D,B	0	7.21
E,C	7.21	0

Fourth merge

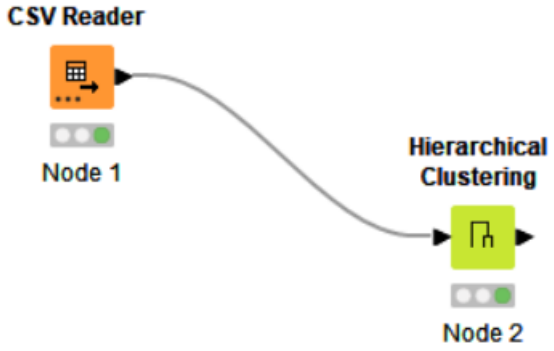
In the proximity matrix, we can see that the last distance is 7.21. We can now perform the last merge, and the dendrogram will look like this:



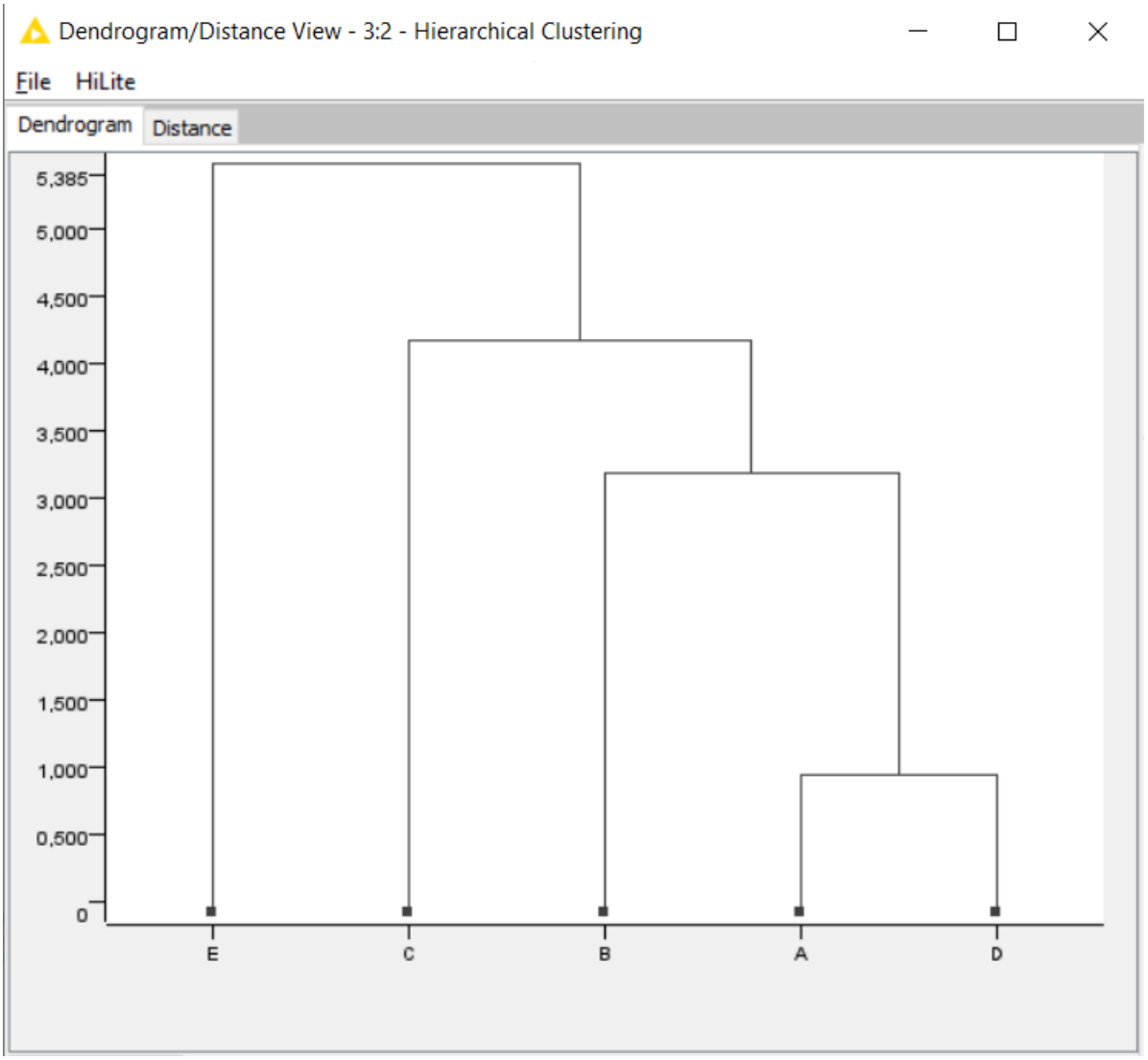
(c)

MIN-link

Workflow



Resulting dendrogram

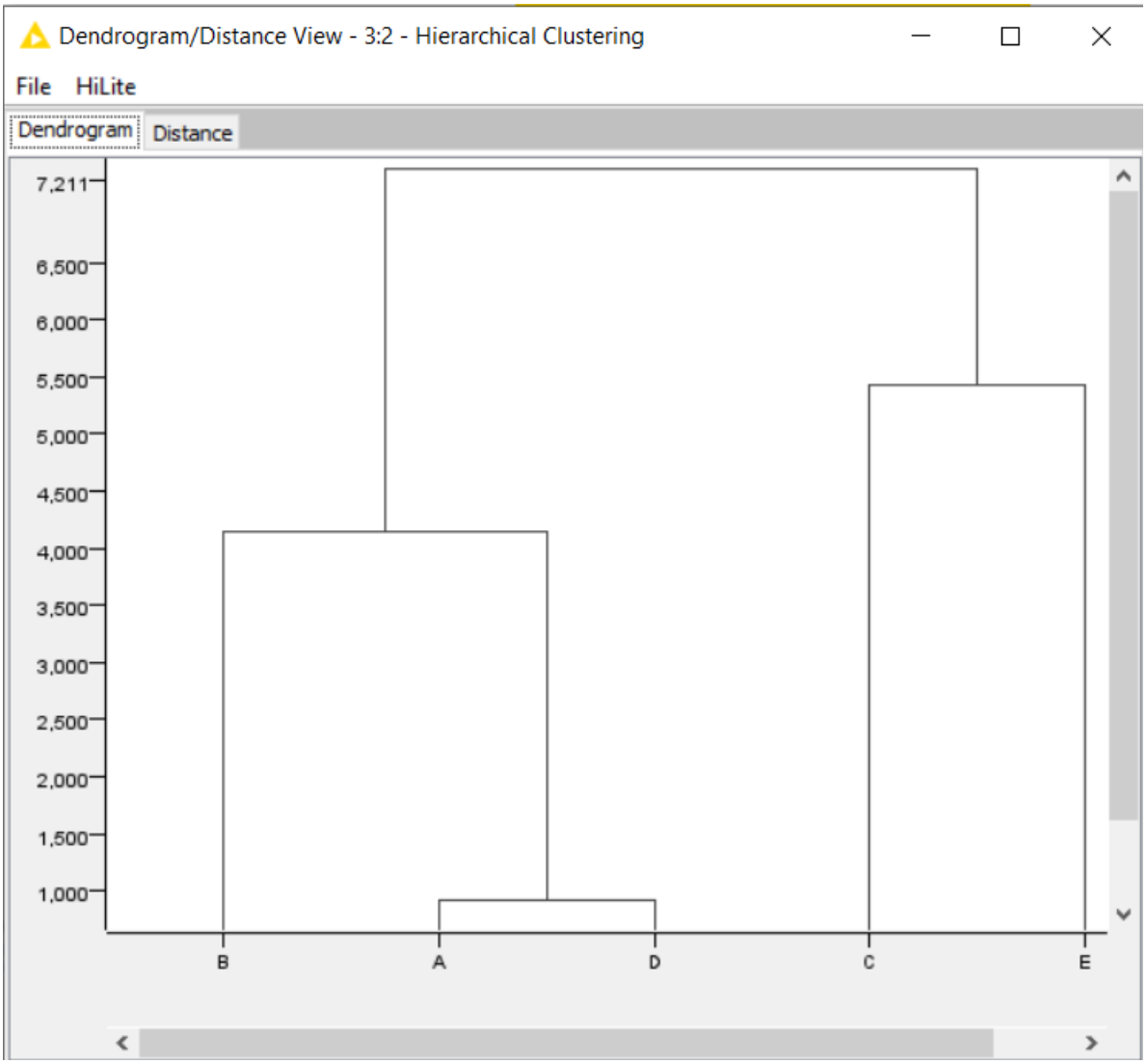


We can see that it matches.

MAX-link

Using the same workflow as above, but with slightly different configuration on the Hierarchical clustering node.

Resulting dendrogram



We can see that it is drawn in a different way, but still shows the same merges and the same distances, so it matches.