

Statistics 5401

32. Hierarchical Clustering

Gary W. Oehlert
School of Statistics
313B Ford Hall
612-625-1557
gary@stat.umn.edu

We will discuss agglomerative or joining hierarchical methods. These methods are hierarchical, because every cluster is formed by the merger of two other clusters. These methods agglomerate, because we start with n clusters of one data point each, and end with one cluster of n data points.

All agglomerative linkage methods follow the same basic algorithm.

1. Start with n clusters of one point each, and an $n \times n$ matrix of distances (or similarities).
2. Merge the two closest (or most similar) clusters.
3. Update the distance (similarity) matrix by removing rows and columns for the two clusters that got merged, and adding a row and column for the new cluster that was the product of the merger.
4. Repeat 2 and 3 until there is only one cluster.

The methods differ only in how they update the distance (similarity) matrix.

```
Cmd> x1 <- rnorm(8)
```

NOTE: random number seeds set to
1341343222 and 1032315086

```
Cmd> x2 <- rnorm(8)
```

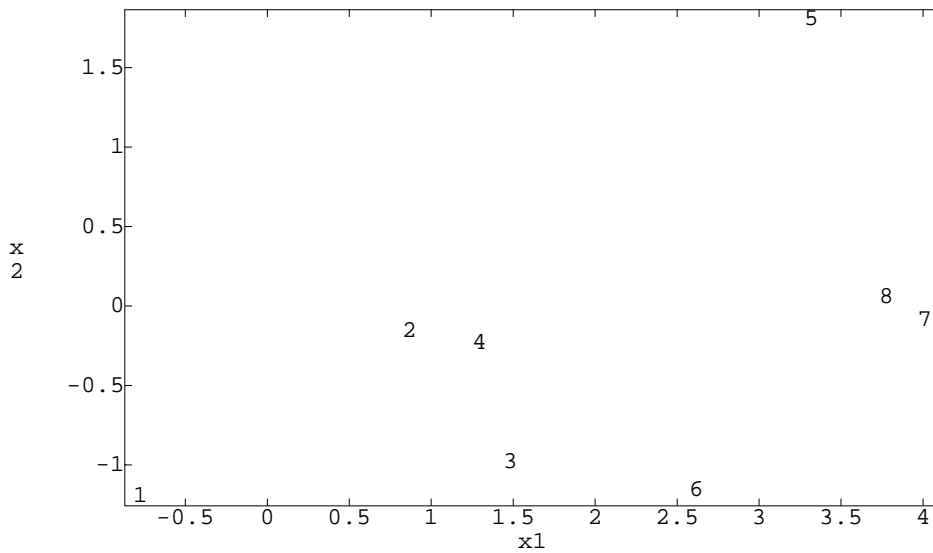
```
Cmd> x1[run(5,8)] <- x1[run(5,8)]+3
```

```
Cmd> D <- sqrt((x1-x1')^2+(x2-x2')^2)
```

```
Cmd> print(D,format:"f4.2",labels:F)
```

```
0.00 1.94 2.27 2.28 5.08 3.39 4.91 4.72
1.94 0.00 1.03 0.43 3.14 2.02 3.15 2.92
2.27 1.03 0.00 0.77 3.33 1.15 2.68 2.52
2.28 0.43 0.77 0.00 2.87 1.61 2.72 2.50
5.08 3.14 3.33 2.87 0.00 3.04 2.02 1.80
3.39 2.02 1.15 1.61 3.04 0.00 1.75 1.68
4.91 3.15 2.68 2.72 2.02 1.75 0.00 0.28
4.72 2.92 2.52 2.50 1.80 1.68 0.28 0.00
```

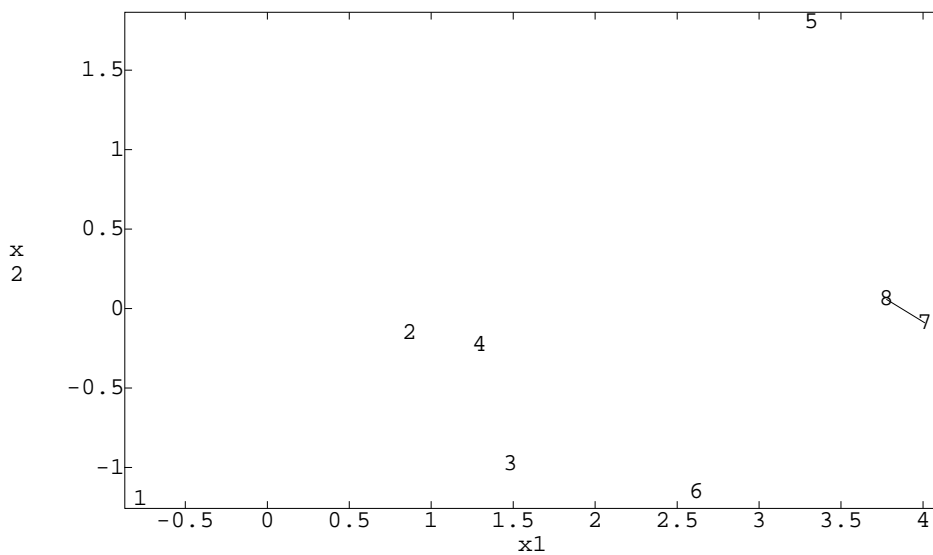
```
Cmd> chplot(x1,x2,xaxis:F,yaxis:F)
```



Single Linkage defines the distance between two clusters as the minimum distance between any two points in the clusters.

The minimum distance is .28, between points 7 and 8, so we join them into a cluster.

We say that 7 and 8 join at distance .28.



```
Cmd> D1 <- D[-vector(7,8),-vector(7,8)]
```

```
Cmd> d <- min(D[run(7,8),-vector(7,8)])
```

```
Cmd> d
(1,1)  4.7196  2.9177  2.5166  2.4992
(1,5)  1.8038  1.6769
```

```
Cmd> D1 <- vconcat(D1,d)
```

```

Cmd> D1 <- hconcat(D1,vector(d,0))

Cmd> print(D1,format:"f4.2",labels:F)
D1:
 0.00  1.94  2.27  2.28  5.08  3.39  4.72
 1.94  0.00  1.03  0.43  3.14  2.02  2.92
 2.27  1.03  0.00  0.77  3.33  1.15  2.52
 2.28  0.43  0.77  0.00  2.87  1.61  2.50
 5.08  3.14  3.33  2.87  0.00  3.04  1.80
 3.39  2.02  1.15  1.61  3.04  0.00  1.68
 4.72  2.92  2.52  2.50  1.80  1.68  0.00

```

Now the smallest distance is .43, between points 2 and 4. Thus 2 and 4 joint at height .43. The clusters are now 1, 3, 5, 6, (7,8), (2,4).

```

Cmd> join <- vector(2,4)

Cmd> D2 <- D1[-join,-join]

Cmd> d <- min(D1[join,-join])

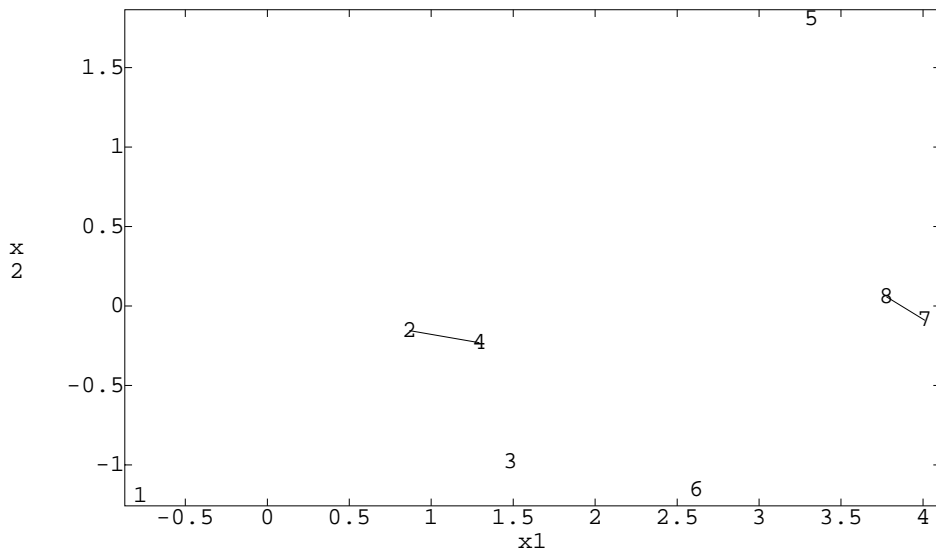
Cmd> d
(1,1)  1.9433  0.77151  2.871  1.6144  2.4992

Cmd> D2 <- vconcat(D2,d)

Cmd> D2 <- hconcat(D2,vector(d,0))

Cmd> print(D2,format:"f4.2",labels:F)
 0.00  2.27  5.08  3.39  4.72  1.94
 2.27  0.00  3.33  1.15  2.52  0.77
 5.08  3.33  0.00  3.04  1.80  2.87
 3.39  1.15  3.04  0.00  1.68  1.61
 4.72  2.52  1.80  1.68  0.00  2.50
 1.94  0.77  2.87  1.61  2.50  0.00

```



Next we join clusters 2 (point 3) and 6 (points 2 and 4), at distance .77. The closest pair is points 3 and 4. The clusters are now 1, 5, 6, (7,8), (2,3,4).

```
Cmd> join <- vector(2,6)
```

```
Cmd> D3 <- D2[-join,-join]
```

```
Cmd> d <- min(D2[join,-join]);d
```

```
(1,1)      1.9433      2.871      1.1483      2.4992
```

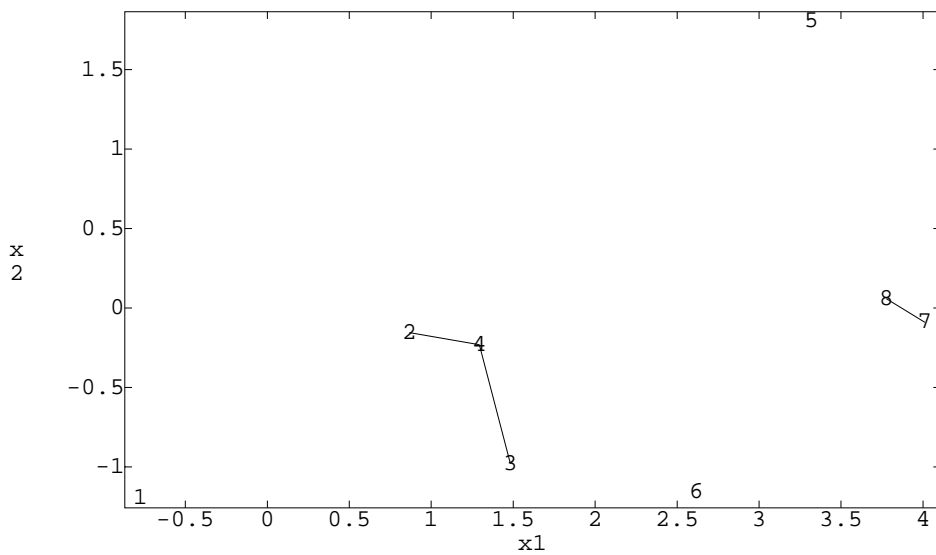
```
Cmd> D3 <- vconcat(D3,d)
```

```
Cmd> D3 <- hconcat(D3,vector(d,0))
```

```
Cmd> print(D3,format:"f5.2",labels:F)
```

```
D3:
```

```
0.00  5.08  3.39  4.72  1.94
5.08  0.00  3.04  1.80  2.87
3.39  3.04  0.00  1.68  1.15
4.72  1.80  1.68  0.00  2.50
1.94  2.87  1.15  2.50  0.00
```



Next we join clusters 3 (point 6) and 5 (points 2,3,4) at distance 1.15. The closest pair is points 3 and 6. The new clusters are 1, 5, (7,8), (2,3,4,6).

```
Cmd> join <- vector(3,5)
```

```
Cmd> D4 <- D3[-join,-join]
```

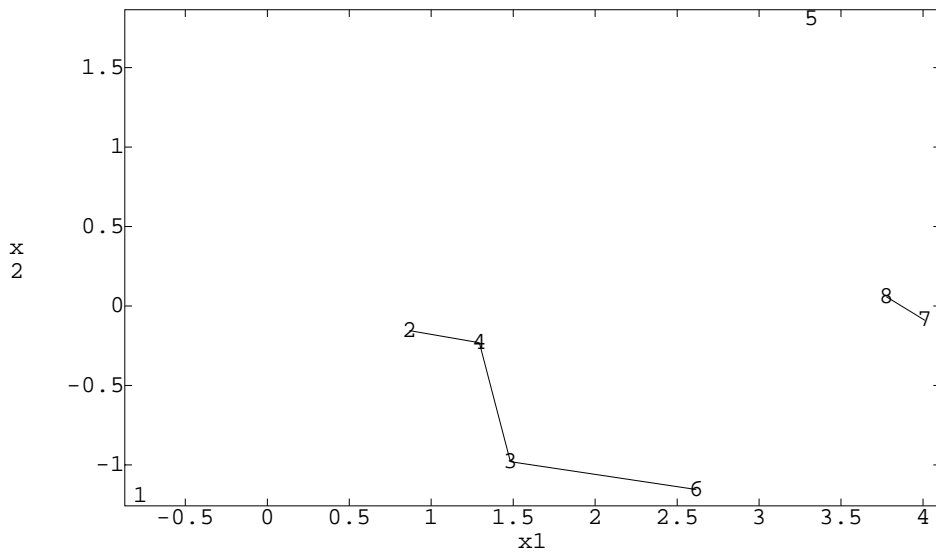
```
Cmd> d <- min(D3[join,-join]);d
(1,1)      1.9433      2.871      1.6769
```

```
Cmd> D4 <- vconcat(D4,d)
```

```
Cmd> D4 <- hconcat(D4,vector(d,0))
```

```
Cmd> print(D4,format:"f5.2",labels:F)
```

```
D4:
 0.00  5.08  4.72  1.94
 5.08  0.00  1.80  2.87
 4.72  1.80  0.00  1.68
 1.94  2.87  1.68  0.00
```



New we join clusters 3 (points 7,8) and 4 (points 2,3,4,6) at height 1.68. The closest pair is points 6 and 8. The new clusters are 1, 5, (2,3,4,6,7,8).

```
Cmd> join <- vector(3,4)
```

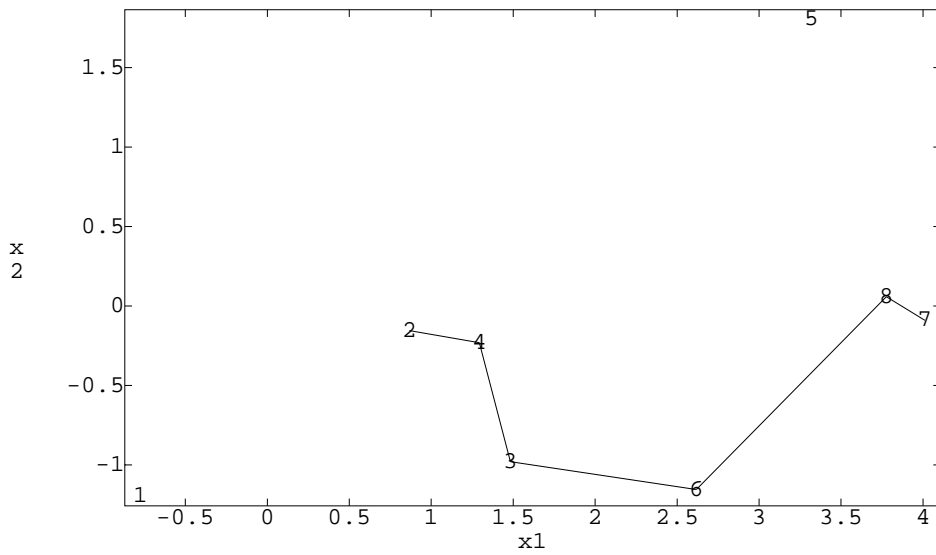
```
Cmd> D5 <- D4[-join,-join]
```

```
Cmd> d <- min(D4[join,-join]);d
(1,1)      1.9433      1.8038
```

```
Cmd> D5 <- vconcat(D5,d)
```

```
Cmd> D5 <- hconcat(D5,vector(d,0))
```

```
Cmd> D5
(1,1)      0      5.0752      1.9433
(2,1)     5.0752      0      1.8038
(3,1)     1.9433     1.8038      0
```



Don't despair; just a couple more to go. Now join clusters 2 (point 5) and 3 (points 2,3,4,6,7,8). The closest pair is points 5 and 8 at a distance of 1.8. The clusters are now 1, (2,3,4,5,6,7,8).

```
Cmd> join <- vector(2,3)
```

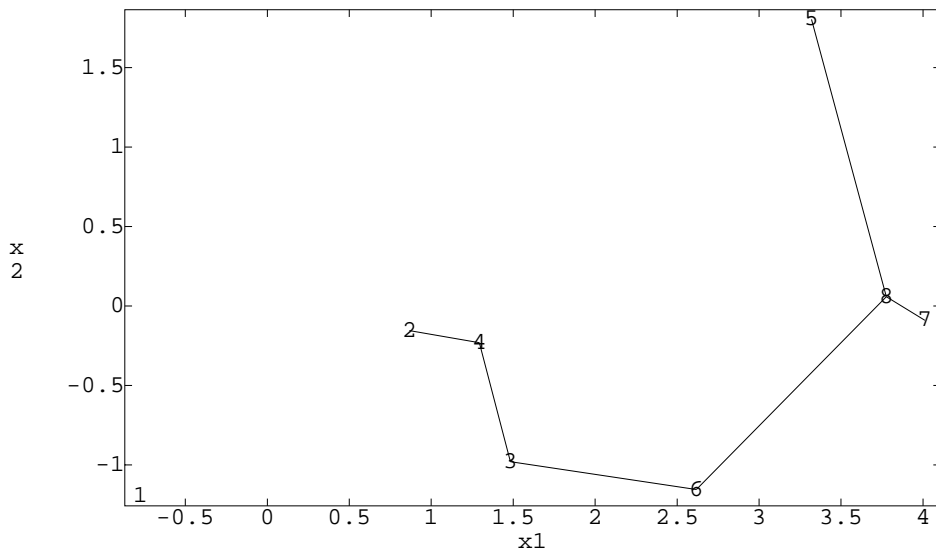
```
Cmd> D6 <- D5[-join,-join]
```

```
Cmd> d <- min(D5[join,-join]);d
(1,1)      1.9433
```

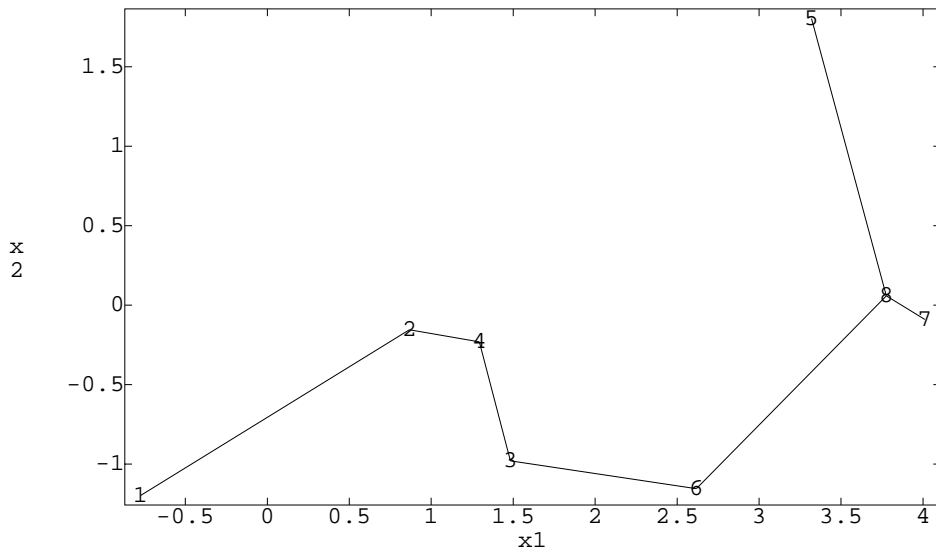
```
Cmd> D6 <- vconcat(D6,d)
```

```
Cmd> D6 <- hconcat(D6,vector(d,0))
```

```
Cmd> D6
(1,1)      0      1.9433
(2,1)     1.9433     0
```



Now we join the last ones (points 1 and 2 are the closest pair) at a height of 1.94. We are done. The graph at the end of single linkage is called a *Minimum spanning tree*. It is the shortest graph that connects all the points.



This is all automated in MacAnova.

```
Cmd> cluster(dissim:D,method:"single")
```

Case No.	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	2	2	2	2	6	6	6
4	2	2	2	2	2	7	7
5	2	3	3	3	3	3	3


```

6  2  2  2  5  5  5  5
7  2  2  4  4  4  4  4
8  2  2  4  4  4  4  8

```

Criterion

```

+
1.9433 +---+
1.8038 | +-----+-----+
1.6769 | +-----+-----+
1.1483 | +-----+-----+
0.77151 | +-----+-----+
0.43435 | +---+ | | | |
0.27813 | | | | | +---+ |
Cluster No. 1 2 7 6 5 4 8 3

```

Clusters 1 to 8 (Top 7 levels of hierarchy).

Clustering method: Single linkage

Distance: Input dissimilarity matrix

Note that the cluster numbers really have nothing to do with the case numbers. You have to go through the table and find which cases go in which clusters.

You can also cluster the data matrix directly. By default, `cluster()` will standardize the variables of the data, which we don't want here, so we'll have to tell it not to standardize.

```
Cmd> X <- hconcat(x1,x2)
```

```
Cmd> cluster(X,standard:F,method:"single")
```

(Output the same as before.)

You can also say how many clusters to produce. Then MacAnova will only show the "upper" part of the dendrogram with that many clusters.

```
Cmd> cluster(dissim:D,nclust:3,method:"single")
```

Case Number of Clusters

```
No. 2 3
```

```
-----
```

```

1  1  1
2  2  2
3  2  2
4  2  2
5  2  3
6  2  2
7  2  2
8  2  2

```

Criterion

+

```

      1.9433  +--+
      1.8038  |  +--+
Cluster No.  1  2  3
      Clusters 1 to 3 (Top 2 levels of hierarchy).
      Clustering method: Single linkage
      Distance: Input dissimilarity matrix

```

You can also ask `cluster()` to return some values.

```

Cmd> cluster(dissim:D,nclust:3,\
method:"single",keep:"criterion")
(1)          1.9433          1.8038

```

```

Cmd> cluster(X,standard:F,nclust:3,\
method:"single",keep:"distances")

```

The latter just returns D.

```

Cmd> cluster(dissim:D,nclust:3,\
method:"single",keep:"classes")
(1,1)          1          1
(2,1)          2          2
(3,1)          2          2
(4,1)          2          2
(5,1)          2          3
(6,1)          2          2
(7,1)          2          2
(8,1)          2          2

```

Or you can use `all:T` to get all three.

```

Cmd> X <- matrix(rnorm(200),100)

Cmd> X[run(51,100),1] <- X[run(51,100),1]+3

Cmd> cluster(X,standard:F,nclust:10,\
method:"single")
(memberships not printed)

```

Criterion

```

      +
1.0994 +-----+
0.89843 +-----+ |
0.81718 +-----+ | |
0.69717 +-----+ | | |
0.69502 +-----+ | | | |

```

```

0.6519 +-----+ | | | | |
0.65104 +---+ | | | | |
0.64323 | | | +-----+ | | | |
0.62301 | | | +---+ | | | | |
Cluster No. 1 8 7 6 10 9 5 4 3 2
Clusters 1 to 10 (Top 9 levels of hierarchy).
Clustering method: Single linkage
Distance: Euclidean

```

```

Cmd> out <- cluster(X,standard:F,\
nclust:10,method:"single",\
keep:"classes")

```

```

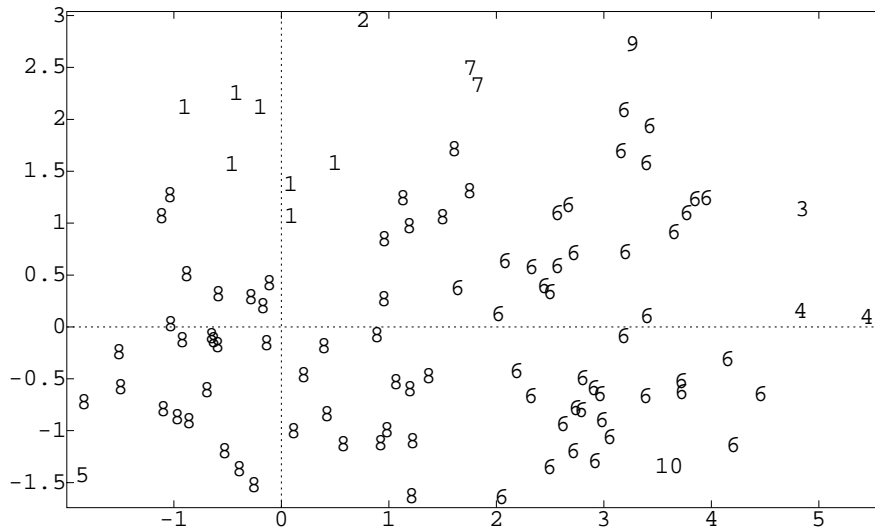
Cmd> tabs(,out[,9])
(1) 7 1 1 2 1
(6) 41 2 43 1 1

```

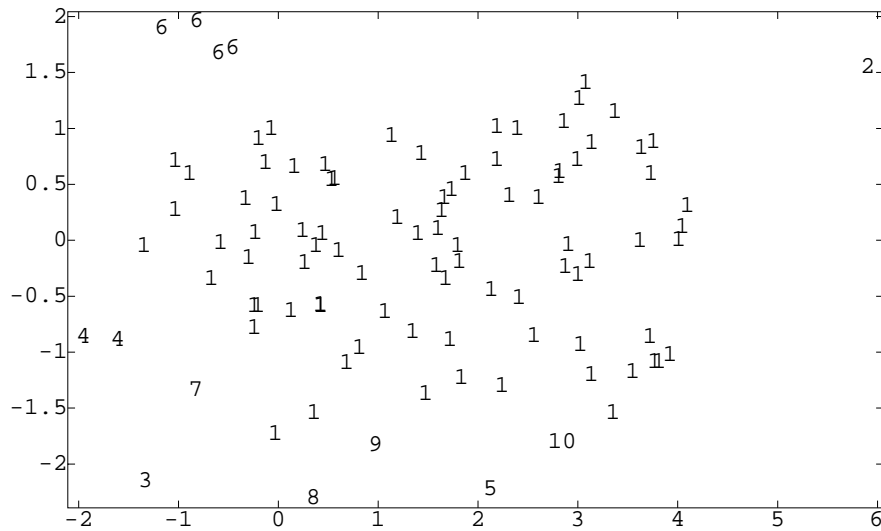
```

Cmd> chplot(X[,1],X[,2],out[,9])

```



But other data sets look like



```
Cmd> X <- matrix(rnorm(200),100)
```

```
Cmd> X[run(51,100),1] <- \
X[run(51,100),1]+3
```

```
Cmd> out <- cluster(X,standard:F,\
nclust:30,method:"single",\
keep:"classes")
```

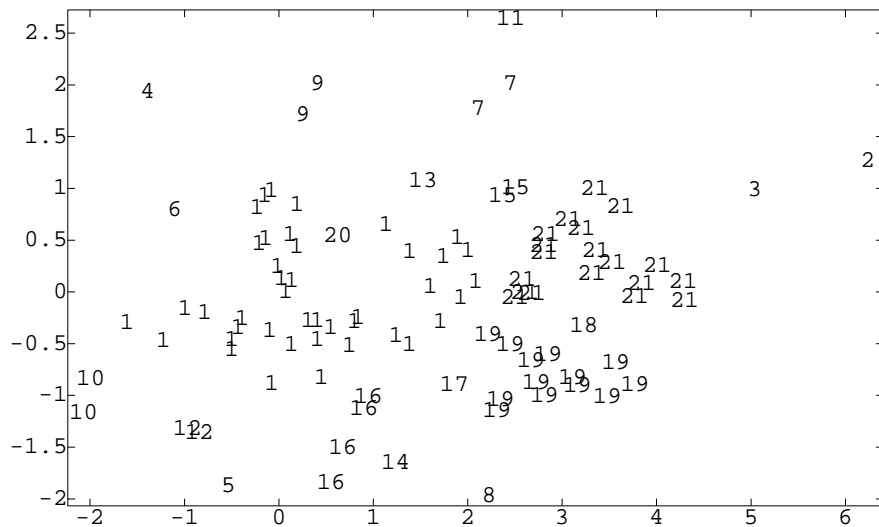
```
Cmd> tabs(,out[,29])
```

```
(1) 3 1 1 1 1
(6) 1 1 1 2 2
(11) 1 2 1 1 2
(16) 2 1 1 2 1
(21) 17 9 2 1 1
(26) 1 2 11 2 26
```

```
Cmd> tabs(,out[,20])
```

```
(1) 42 1 1 1 1
(6) 1 2 1 2 2
(11) 1 2 1 1 2
(16) 4 1 1 13 1
(21) 19
```

```
Cmd> chplot(X[,1],X[,2],out[,20],\
xaxis:F,yaxis:F)
```



Single linkage is subject to chaining, long strands of cluster wandering around feature space. It is also subject to having many of the last joins be “outliers”, or, at least, single points.

On the other hand, single linkage doesn't make clusters where there aren't any.

The other standard linkage methods are average linkage, complete linkage, and Ward's.

The complete linkage distance between two clusters is the maximum distance between any two points in the clusters.

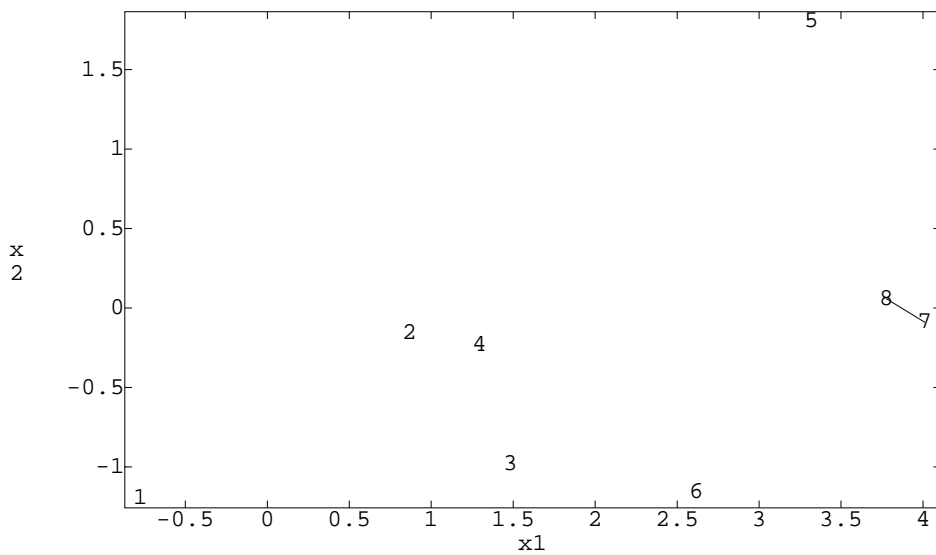
The average linkage distance between two clusters is the average distance between any pair of points, one from each cluster.

Ward's uses a dissimilarity function, not a distance function. At any stage of the clustering, we can do a one-way Manova on the the data using the clusters as groups. Take the trace of the error sums of squares and crossproducts matrix as a measure of goodness for that clustering.

Ward's dissimilarity between two clusters is the increase in the trace of the error SSCP matrix when those clusters are merged.

All of these can be done in MacAnova. First compare single with complete linkage on the 8 point data set.

The minimum distance is .28, between points 7 and 8, so we join them into a cluster. This is the same as single linkage.



```
Cmd> D1 <- D[-vector(7,8),-vector(7,8)]
```

```
Cmd> d <- max(D[run(7,8),-vector(7,8)])
```

```
Cmd> d
```

```
(1,1)    4.91    3.15    2.68    2.72
(1,5)    2.02    1.75
```

```
Cmd> D1 <- vconcat(D1,d)
```

```
Cmd> D1 <- hconcat(D1,vector(d,0))
```

```
Cmd> print(D1,format:"f4.2",labels:F)
```

```
D1:
```

```
0.00 1.94 2.27 2.28 5.08 3.39 4.91
1.94 0.00 1.03 0.43 3.14 2.02 3.15
2.27 1.03 0.00 0.77 3.33 1.15 2.68
2.28 0.43 0.77 0.00 2.87 1.61 2.72
5.08 3.14 3.33 2.87 0.00 3.04 2.02
3.39 2.02 1.15 1.61 3.04 0.00 1.75
4.91 3.15 2.68 2.72 2.02 1.75 0.00
```

Now the smallest distance is .43, between points 2 and 4. Thus 2 and 4 joint at height .43. The clusters are now 1, 3, 5, 6, (7,8), (2,4).

```
Cmd> join <- vector(2,4)
```

```
Cmd> D2 <- D1[-join,-join]
```

```

Cmd> d <- max(D1[join,-join])

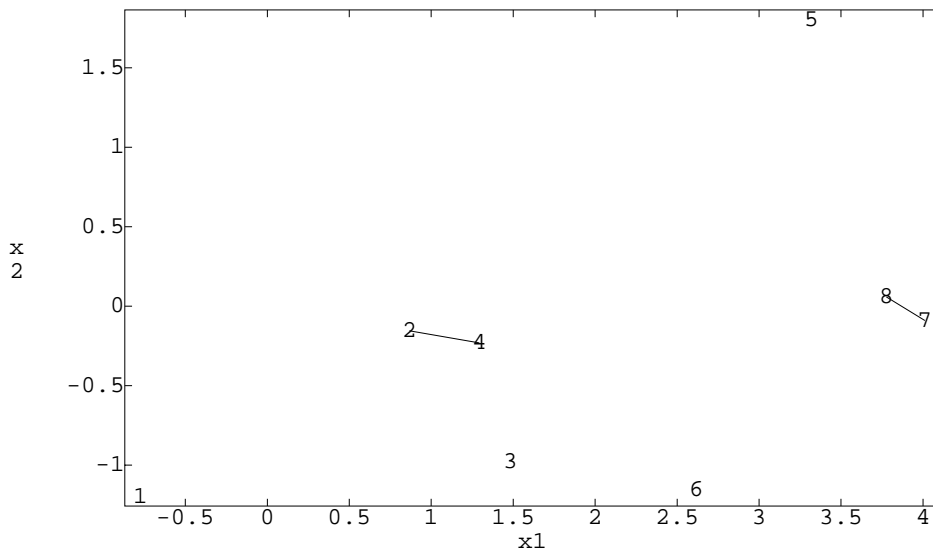
Cmd> d
(1,1)  2.28  1.03  3.14  2.02  3.15

Cmd> D2 <- vconcat(D2,d)

Cmd> D2 <- hconcat(D2,vector(d,0))

Cmd> print(D2,format:"f4.2",labels:F)
0.00 2.27 5.08 3.39 4.91 2.28
2.27 0.00 3.33 1.15 2.68 1.03
5.08 3.33 0.00 3.04 2.02 3.14
3.39 1.15 3.04 0.00 1.75 2.02
4.91 2.68 2.02 1.75 0.00 3.15
2.28 1.03 3.14 2.02 3.15 0.00

```



Next we join clusters 2 (point 3) and 6 (points 2 and 4), at distance .77. The closest pair is points 3 and 4. The clusters are now 1, 5, 6, (7,8), (2,3,4).

```

Cmd> join <- vector(2,6)

Cmd> D3 <- D2[-join,-join]

Cmd> d <- max(D2[join,-join]);d
(1,1)  2.28  3.33  2.02  3.15

Cmd> D3 <- vconcat(D3,d)

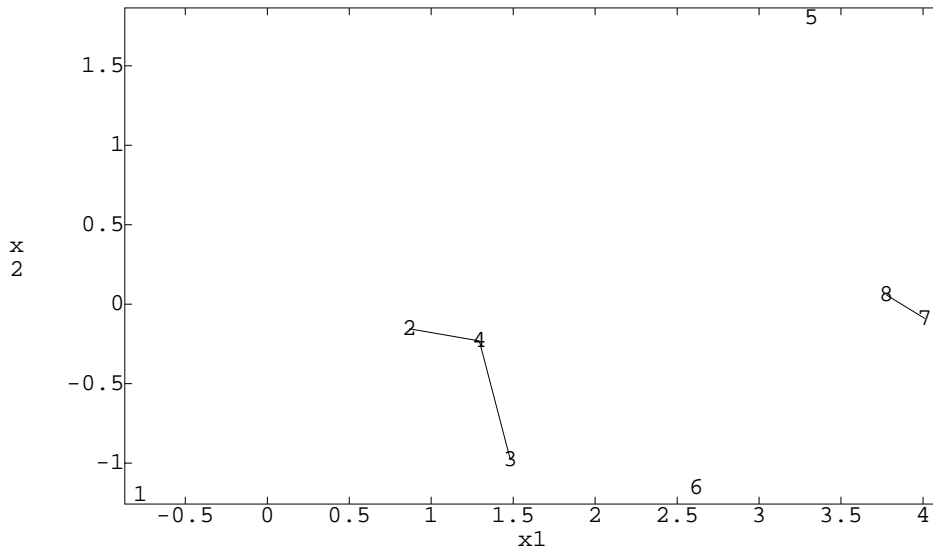
Cmd> D3 <- hconcat(D3,vector(d,0))

```

```
Cmd> print(D3,format:"f5.2",labels:F)
```

```
D3:
```

```
0.00 5.08 3.39 4.91 2.28
5.08 0.00 3.04 2.02 3.33
3.39 3.04 0.00 1.75 2.02
4.91 2.02 1.75 0.00 3.15
2.28 3.33 2.02 3.15 0.00
```



Next we join clusters 3 (point 6) and 4 (points 7,8) at distance 1.75. The closest pair is points 8 and 6. The new clusters are 1, 5, (2,3,4), (6,7,8). We now differ with single linkage.

```
Cmd> join <- vector(3,4)
```

```
Cmd> D4 <- D3[-join,-join]
```

```
Cmd> d <- max(D3[join,-join]);d
(1,1)      4.91    3.04    3.15
```

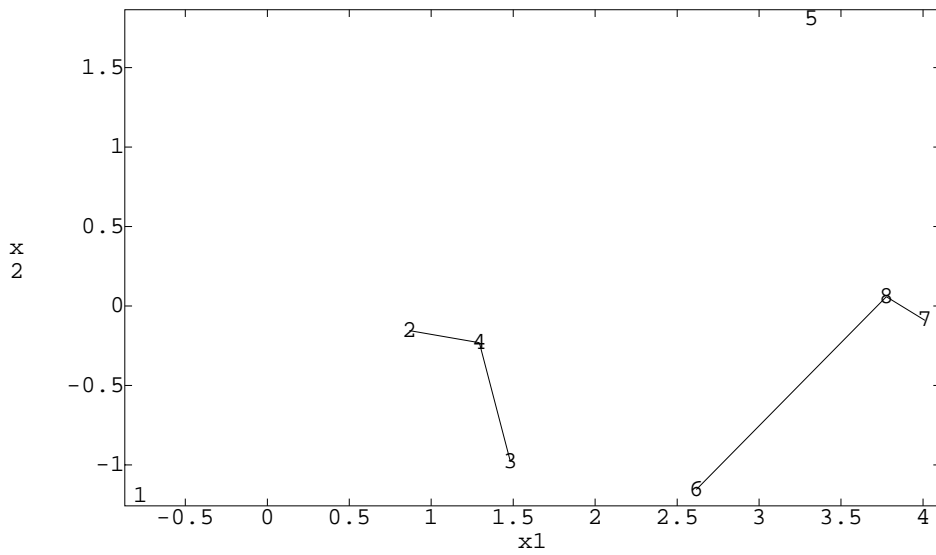
```
Cmd> D4 <- vconcat(D4,d)
```

```
Cmd> D4 <- hconcat(D4,vector(d,0))
```

```
Cmd> print(D4,format:"f5.2",labels:F)
```

```
D4:
```

```
0.00 5.08 2.28 4.91
5.08 0.00 3.33 3.04
2.28 3.33 0.00 3.15
4.91 3.04 3.15 0.00
```

New we join clusters 1 (point 1) and 3 (points 2,3,4) at height 2.28. The closest pair is points 1 and 2. The new clusters are 5, (6,7,8), (1,2,3,4).

```
Cmd> join <- vector(1,3)
```

```
Cmd> D5 <- D4[-join,-join]
```

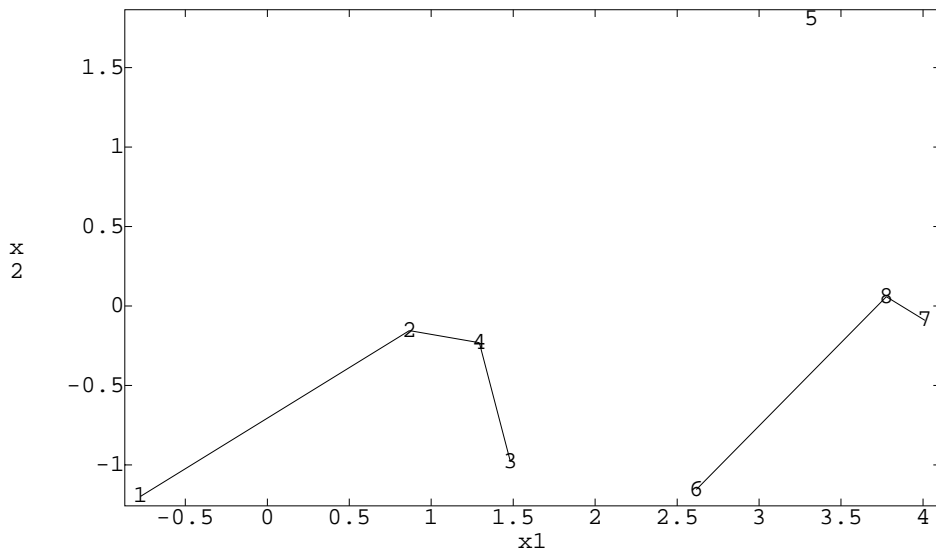
```
Cmd> d <- min(D4[join,-join]);d
(1,1)      5.08      4.91
```

```
Cmd> D5 <- vconcat(D5,d)
```

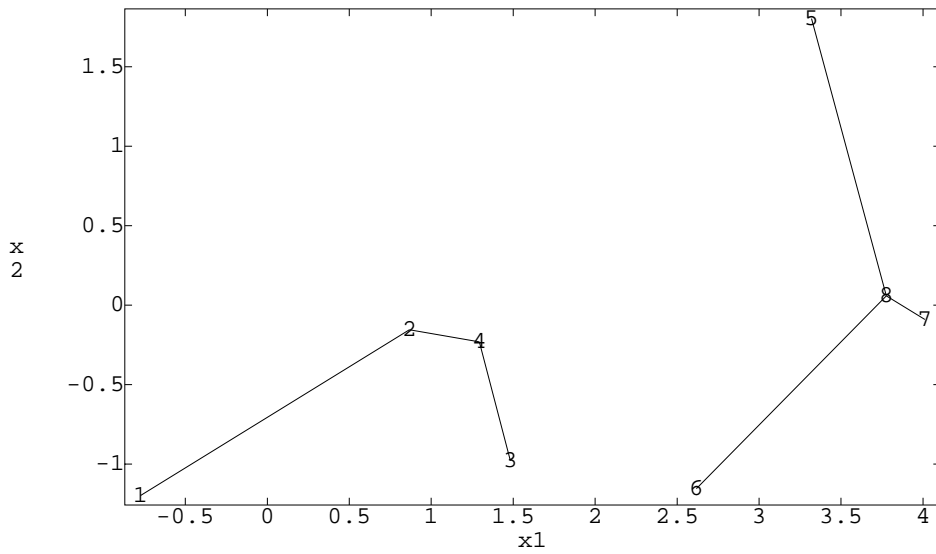
```
Cmd> D5 <- hconcat(D5,vector(d,0))
```

```
Cmd> D5
```

```
D5:
 0.00  3.04  5.08
 3.04  0.00  4.91
 5.08  4.91  0.00
```



Now join clusters 1 (point 5) and 2 (points 6,7,8). The closest pair is points 5 and 8. The clusters are now (1,2,3,4), (5,6,7,8).



```
Cmd> cluster(dissim:D,method:"complete")
```

Case No.	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1
2	1	1	4	4	4	4	4
3	1	1	4	4	6	6	6
4	1	1	4	4	4	7	7
5	2	2	2	2	2	2	2
6	2	3	3	3	3	3	3
7	2	3	3	5	5	5	5

8 2 3 3 5 5 5 8

Criterion

```

+
5.08 +-----+
3.04 |               +---+
2.28 +---+         |   |
1.75 |   |         |   +---+
1.03 |   +-----+ |   |
0.43 |   +---+   |   |   |
0.28 |   |   |   |   |   +---+

```

Cluster No. 1 4 7 6 2 3 5 8

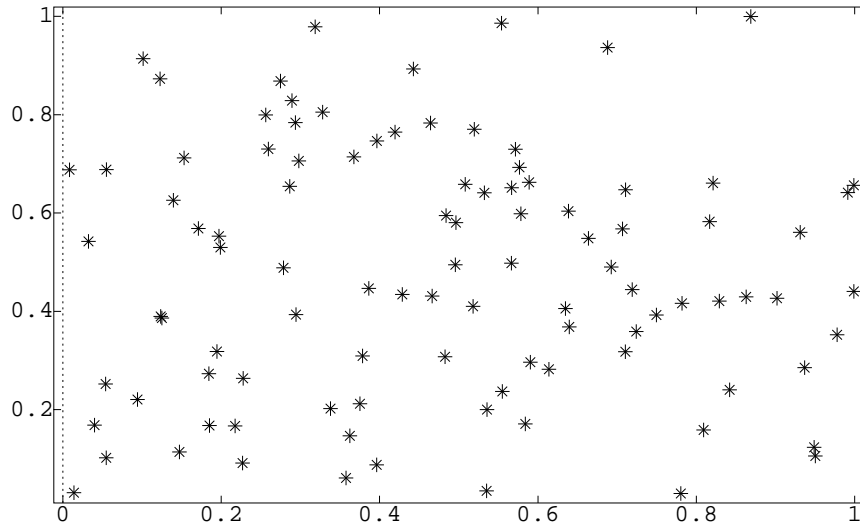
Clusters 1 to 8 (Top 7 levels of hierarchy).

Clustering method: Complete linkage

Distance: Input dissimilarity matrix

Cmd> X <- matrix(runi(200),100)

Cmd> plot(X[,1],X[,2])



```

Cmd> out <- cluster(X,standard:F,\
method:"single",keep:"classes",\
nclust:10,print:T)

```

```

0.19151 +-----+
0.15221 +-----+ |
0.14529 +-----+ |
0.14498 +-----+ |
0.144   +-----+ |
0.14291 |           | +---+ |

```

```

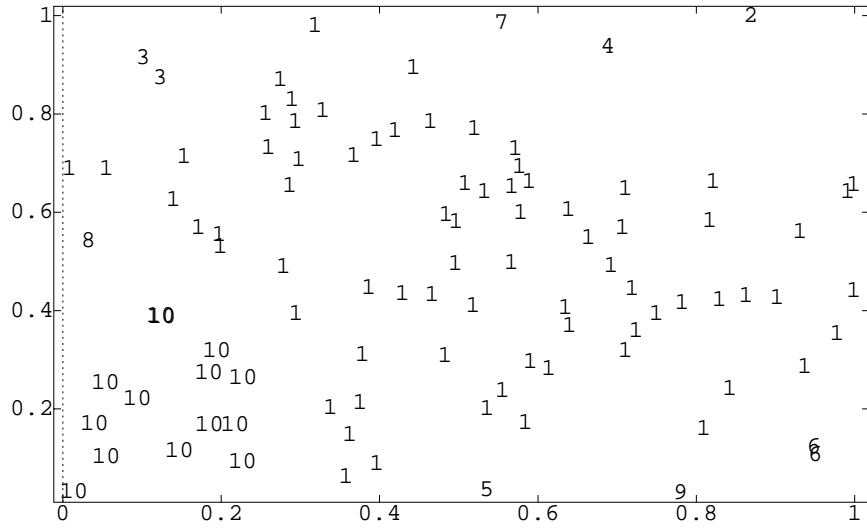
0.13598 +-----+ | | | | | |
0.13218 +-----+ | | | | | |
0.12506 +--+ | | | | | |
Clus No. 1 10 9 8 6 5 4 7 3 2

```

```

Cmd> chplot(X[,1],X[,2],out[,9])

```



```

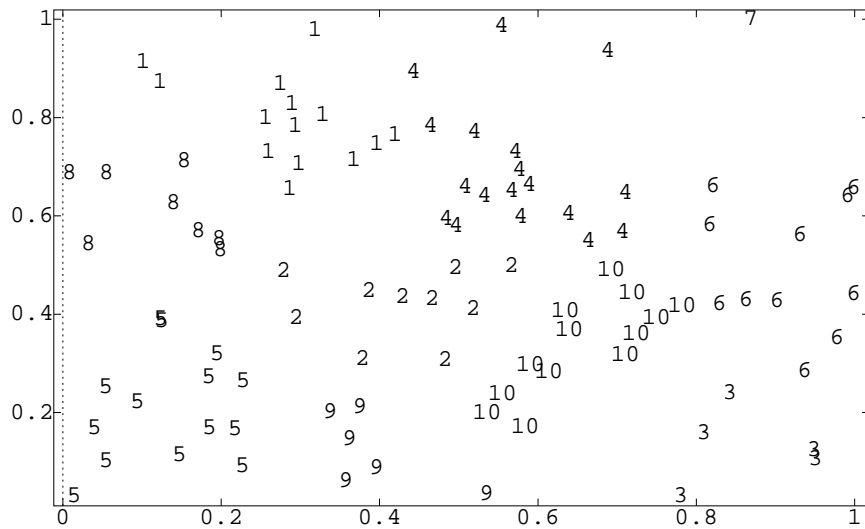
Cmd> out <- cluster(X,standard:F,\
method:"complete",keep:"classes",\
nclust:10,print:T)
  1.292 +-----+
  1.1667 | +-----+
  0.9535 +-----+ |
  0.72338 | +-----+ |
  0.66379 | | +-----+
  0.56086 | +--+ |
  0.52201 +--+ | |
  0.52126 | | +--+ |
  0.46894 | | | +--+ |
Clus No. 1 8 4 7 2 9 5 3 10 6

```

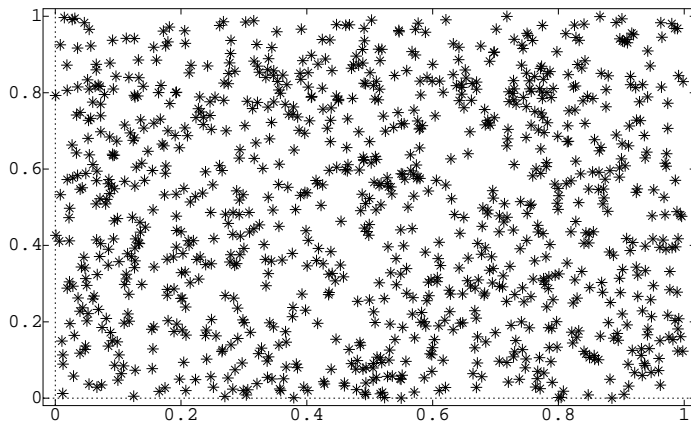
```

Cmd> chplot(X[,1],X[,2],out[,9])

```

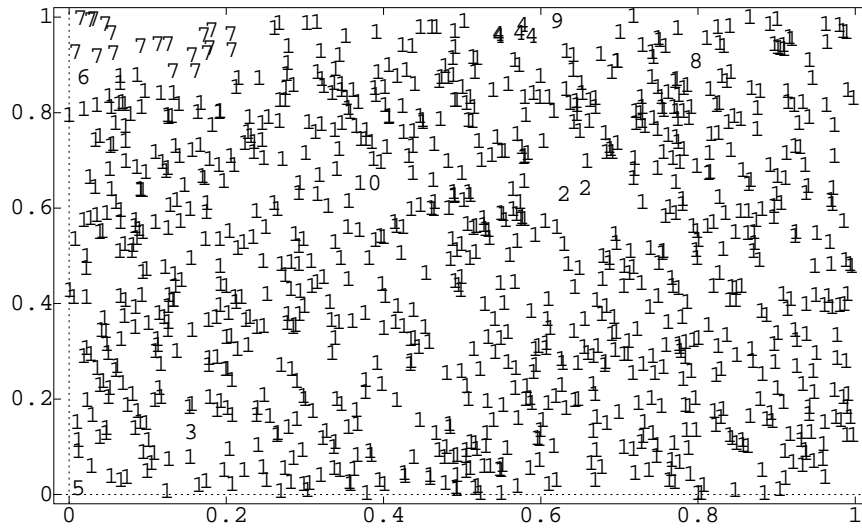


```
Cmd> X <- matrix(runi(2000),1000)
Cmd> plot(X[,1],X[,2])
```



```
Cmd> out <- cluster(X,standard:F,\
method:"single",keep:"classes",\
nclust:10,print:T)
0.057202 +-----+
0.051705 +-----+
0.051192 +-----+
0.048396 +-----+
0.046627 +-----+
0.045836 +-----+
0.045173 +-----+
0.045036 | | | | | +--+
0.044072 +--+ | | | | |
Clus No. 1 10 8 7 6 5 4 9 3 2
```

```
Cmd> chplot(X[,1],X[,2],out[,9])
```



```
Cmd> out <- cluster(X,standard:F,\
method:"complete",keep:"classes",\
nclust:10,print:T)
```

```
  1.3695 +-----+
  1.1528 +-----+ |
  1.0933 |         | +-----+
0.86438 |         +-----+ |
0.76113 +--+ |         |         |
0.66051 | | |         |         +--+
0.63723 | | +--+ |         |         |
0.62297 | | |         +--+ |         |
0.58435 | | |         | +--+ |         |
Clu No.  1  6  3  8  5  2  9 10  4  7
```

```
Cmd> chplot(X[,1],X[,2],out[,9])
```

