(Unsupervised Learning)

Hierarchical Clustering

Examples

STAT 32950-24620

Spring 2023 (5/2-4)

Unsupervised Learning — Cluster Analysis

Cluster analysis:

- Partition items into groups according to certain similarity measures
- Often hierarchical structures are of interests

Definitions of distance between two clusters:

- Single linkage
  — the minimum distance among all pairs of individuals from the two clusters
- Complete linkage
  — the maximum distance among all pairs of individuals from the two clusters
- Average linkage
  — the average distance among all pairs of individuals from the two clusters

Example Comparison of letters for numbers in 11 languages

Data of interests:

Concordance counts of the first letters for the first ten numbers

For example,

English: one, two, three, four, five, six, seven, eight, nine, ten

Norwegian: en, to, the, fire, fem, seeks, sju, atte, ni, ti

French: un, deux, trois, quatre, cinq, six, sept, huit, neuf, dix

A similarity matrix

```
data=read.table("t12-3a.dat.txt")
colnames(data)=c("E","N","Da","Du","G","Fr","Sp","I","P","H","Fi")
data
```

	E	N	Da	Du	G	Fr	Sp	I	P	H	Fi
1	10	8	8	3	4	4	4	4	3	1	1
2	8	10	9	5	6	4	4	4	3	2	1
3	8	9	10	4	5	4	5	5	4	2	1
4	3	5	4	10	5	1	1	1	0	2	1
5	4	6	5	5	10	3	3	3	2	1	1
6	4	4	4	1	3	10	8	9	5	0	1
7	4	4	5	1	3	8	10	9	7	0	1
8	4	4	5	1	3	9	9	10	6	0	1
9	3	3	4	0	2	5	7	6	10	0	1
10	1	2	2	2	1	0	0	0	0	10	2
11	1	1	1	1	1	1	1	1	1	2	10

From similarity to distance

Create a distance measure from the similarity measure

Generate a **distance matrix**

```
distdata = 10 - data
dmat =as.dist(distdata)
dmat

    E  N Da Du  G Fr Sp  I  P  H
N   2
Da  2  1
Du  7  5  6
G   6  4  5  5
Fr  6  6  6  9  7
Sp  6  6  5  9  7  2
I   6  6  5  9  7  1  1
P   7  7  6 10  8  5  3  4
H   9  8  8  8  9 10 10 10 10
Fi  9  9  9  9  9  9  9  9  9  8
```


Hierarchical clustering using single linkage

Start: Every element is a cluster.

First step: Find the closest clusters to merge.

Distance	*E*	*N*	*Da*	*Du*	*G*	*Fr*	*Sp*	*I*	*P*	*H*	*Fi*
E	0										
N	2	0									
Da	2	**1**	0								
Du	7	5	6	0							
G	6	4	5	5	0						
Fr	6	6	6	9	7	0					
Sp	6	6	5	9	7	2	0				
I	6	6	5	9	7	**1**	**1**	0			
P	7	7	6	10	8	5	3	4	0		
H	9	8	8	8	9	10	10	10	10	0	
Fi	9	9	9	9	9	9	9	9	9	8	0

Single linkage after one step: 11 clusters merge into 8 clusters

	E	N-Da	*Du*	*G*	Fr-Sp-I	*P*	*H*	*Fi*
E	0							
N-Da	?	0						
Du	7	?	0					
G	6	?	5	0				
Fr-Sp-I	?	?	?	?	0			
P	7	?	10	8	?	0		
H	9	?	8	9	?	10	0	
Fi	9	?	9	9	?	9	8	0

The distances between the new clusters and others need to be calculated.

Check element level distances between all pairs of clusters (including single member clusters)

element dist.	*E*	*N*	*Da*	*Du*	*G*	*Fr*	*Sp*	*I*	*P*	*H*
N	2									
Da	2	1								
Du	7	5	6							
G	6	4	5	5						
Fr	6	6	6	9	7					
Sp	6	6	5	9	7	2				
I	6	6	5	9	7	1	1			
P	7	7	6	10	8	5	3	4		
H	9	8	8	8	9	10	10	10	10	
Fi	9	9	9	9	9	9	9	9	9	8

Single linkage defines the distance between clusters as the minimum distance among all pairs of individuals from the two clusters.

New assignments of cluster distance values after one merge:

cluster dist.	*E*	N-Da	*Du*	*G*	Fr-Sp-I	*P*	*H*	*Fi*
E	0							
N-Da	2	0						
Du	7	5	0					
G	6	4	5	0				
Fr-Sp-I	6	5	9	7	0			
P	7	6	10	8	3	0		
H	9	8	8	9	10	10	0	
Fi	9	9	9	9	9	9	8	0

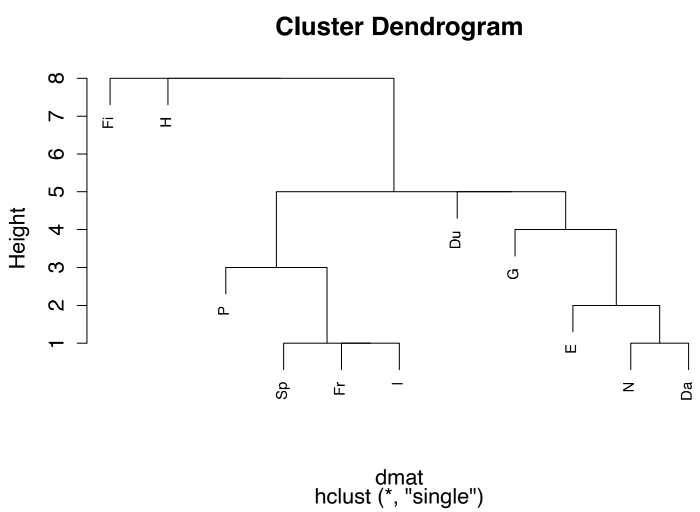
Continue the clustering process from 8 clusters: Find the closest clusters.

	E	N-Da	*Du*	*G*	Fr-Sp-I	*P*	*H*	*Fi*
E	0							
N-Da	2	0						
Du	7	5	0					
G	6	4	5	0				
Fr-Sp-I	6	5	9	7	0			
P	7	6	10	8	3	0		
H	9	8	8	9	10	10	0	
Fi	9	9	9	9	9	9	8	0

The next step is to combine *E* and N-Da.

After the second merge, 8 clusters become 7 clusters.


```
Msingle = hclust(dmat, method="single")
plot(Msingle,cex=.7)
```



Note: "Height" indicates the method-defined "distance" between the merging clusters.

Hierarchical clustering using complete linkage

Start: Every member forms a cluster.
First step: find the nearest clusters to merge.

	E	*N*	*Da*	*Du*	*G*	*Fr*	*Sp*	*I*	*P*	*H*	*Fi*
E	0										
N	2	0									
Da	2	1	0								
Du	7	5	6	0							
G	6	4	5	5	0						
Fr	6	6	6	9	7	0					
Sp	6	6	5	9	7	2	0				
I	6	6	5	9	7	1	1	0			
P	7	7	6	10	8	5	3	4	0		
H	9	8	8	8	9	10	10	10	10	0	
Fi	9	9	9	9	9	9	9	9	9	8	0

Notice that the definition of "nearest" has changed.

Complete linkage after one step: 11 clusters merge into 9 clusters.
Question: Is this step of clustering unique under this method?

	E	N-Da	Du	G	Fr	Sp-I	P	H	Fi
E	0								
N-Da	?	0							
Du	7	?	0						
G	6	?	5	0					
Fr	6	?	9	7	0				
Sp-I	?	?	?	?	?	0			
P	7	?	10	8	5	?	0		
H	9	?	8	9	10	?	10	0	
Fi	9	?	9	9	9	?	9	8	0

Complete linkage clustering after one step: check pairwise distances:

element dist.	E	N	Da	Du	G	Fr	Sp	I	P	H
N	2									
Da	2	1								
Du	7	5	6							
G	6	4	5	5						
Fr	6	6	6	9	7					
Sp	6	6	5	9	7	2				
I	6	6	5	9	7	1	1			
P	7	7	6	10	8	5	3	4		
H	9	8	8	8	9	10	10	10	10	
Fi	9	9	9	9	9	9	9	9	9	8

Complete linkage defines the distance between clusters as the maximum distance among all pairs of individuals from the two clusters.

Complete linkage assignments of distances after one step:

cluster dist.	E	N-Da	Du	G	Fr	Sp-I	P	H	Fi
E	0								
N-Da	2	0							
Du	7	6	0						
G	6	5	5	0					
Fr	6	6	9	7	0				
Sp-I	6	6	9	7	2	0			
P	7	7	10	8	5	4	0		
H	9	8	8	9	10	10	10	0	
Fi	9	9	9	9	9	9	9	8	0

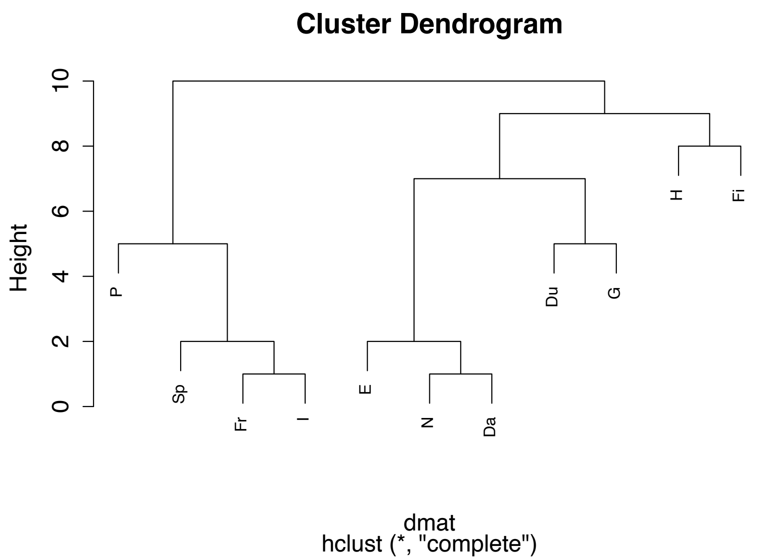
Continue complete linkage clustering, now with 9 clusters.
Find nearest clusters.

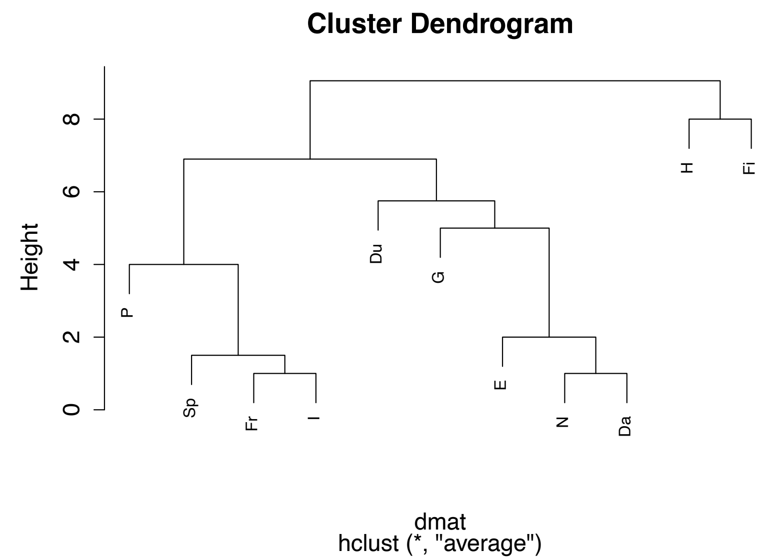
	E	N-Da	Du	G	F	Sp-I	P	H	Fi
E	0								
N-Da	2	0							
Du	7	6	0						
G	6	5	5	0					
Fr	6	6	9	7	0				
Sp-I	6	6	9	7	2	0			
P	7	7	10	8	5	4	0		
H	9	8	8	9	10	10	10	0	
Fi	9	9	9	9	9	9	9	8	0

The next step is to combine *E* and N-Da, and to combine *F* and Sp-I.

After the second merge, 9 clusters become 7 clusters.

```
Mcomplete = hclust(dmat, method="complete")
plot(Mcomplete,cex=.7)
```





Notice the difference in the distance scales.

Cluster Analysis Example: Old public utility data of 22 utility firms

Variables:

- V1 Fixed-charge coverage ratio (income/debt)
- V2 Rate of return on capital
- V3 Cost per kilowatt capacity in place
- V4 Annual load factor
- V5 Peak kilowatt-hour demand growth last year
- V6 Sales (kilowatt-hour used per year)
- V7 Percent nuclear
- V8 Total fuel costs (cents per kilowatt-hour)

Each item (firm) has 8 variables (in $\mathbb{R}^8$ space)

Each variable has 22 observations (in $\mathbb{R}^{22}$ space)


```
data = read.table("T12-4.dat")
data
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	1.06	9.2	151	54.4	1.6	9077	0.0	0.628	Arizona
2	0.89	10.3	202	57.9	2.2	5088	25.3	1.555	Boston
3	1.43	15.4	113	53.0	3.4	9212	0.0	1.058	Central
4	1.02	11.2	168	56.0	0.3	6423	34.3	0.700	Common
5	1.49	8.8	192	51.2	1.0	3300	15.6	2.044	Consolid
6	1.32	13.5	111	60.0	-2.2	11127	22.5	1.241	Florida
7	1.22	12.2	175	67.6	2.2	7642	0.0	1.652	Hawaiian
8	1.10	9.2	245	57.0	3.3	13082	0.0	0.309	Idaho
9	1.34	13.0	168	60.4	7.2	8406	0.0	0.862	Kentucky
10	1.12	12.4	197	53.0	2.7	6455	39.2	0.623	Madison
11	0.75	7.5	173	51.5	6.5	17441	0.0	0.768	Nevada
12	1.13	10.9	178	62.0	3.7	6154	0.0	1.897	NewEngla
13	1.15	12.7	199	53.7	6.4	7179	50.2	0.527	Northern
14	1.09	12.0	96	49.8	1.4	9673	0.0	0.588	Oklahoma
15	0.96	7.6	164	62.2	-0.1	6468	0.9	1.400	Pacific
16	1.16	9.9	252	56.0	9.2	15991	0.0	0.620	Puget
17	0.76	6.4	136	61.9	9.0	5714	8.3	1.920	SanDiego
18	1.05	12.6	150	56.7	2.7	10140	0.0	1.108	Southern
19	1.16	11.7	104	54.0	-2.1	13507	0.0	0.636	Texas
20	1.20	11.8	148	59.9	3.5	7287	41.1	0.702	Wisconsi
21	1.04	8.6	204	61.0	3.5	6650	0.0	2.116	United
22	1.07	9.3	174	54.3	5.9	10093	26.6	1.306	Virginia

Clustering **variables**

To cluster variables, we need **distances between variables**.

Distances can be derived from similarity matrix.

Correlation is one of the most common similarity measures.

```
X = data[,1:8]
print(cor(X),digits=1)
```

```
      V1    V2     V3    V4     V5    V6    V7     V8
V1  1.00  0.64 -0.103 -0.08 -0.259 -0.15  0.04 -0.013
V2  0.64  1.00 -0.348 -0.09 -0.260 -0.01  0.21 -0.328
V3 -0.10 -0.35  1.000  0.10  0.435  0.03  0.11  0.005
V4 -0.08 -0.09  0.100  1.00  0.033 -0.29 -0.16  0.486
V5 -0.26 -0.26  0.435  0.03  1.000  0.18 -0.02 -0.007
V6 -0.15 -0.01  0.028 -0.29  0.176  1.00 -0.37 -0.561
V7  0.04  0.21  0.115 -0.16 -0.019 -0.37  1.00 -0.185
V8 -0.01 -0.33  0.005  0.49 -0.007 -0.56 -0.19  1.000
```


Convert similarity matrix of correlation to distance matrix

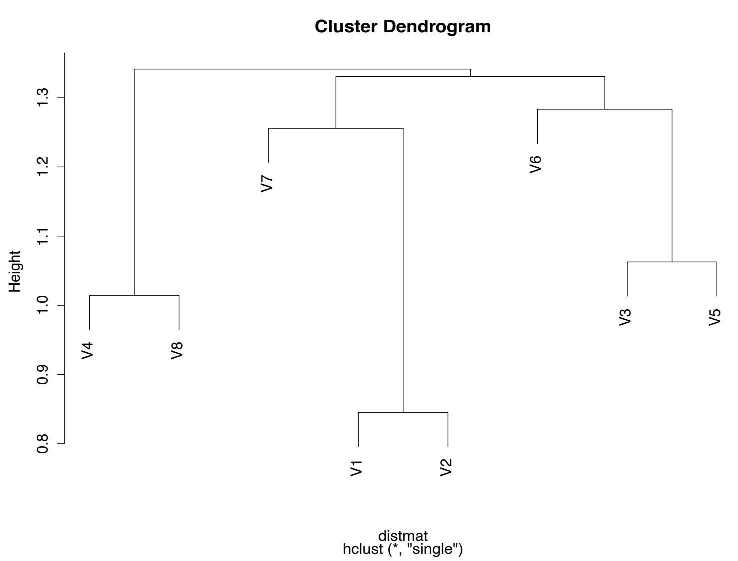
One method: $d = \sqrt{2(1-s)}$

```
     V1   V2   V3   V4   V5   V6   V7
V2 0.85
V3 1.49 1.64
V4 1.47 1.47 1.34
V5 1.59 1.59 1.06 1.39
V6 1.52 1.42 1.39 1.60 1.28
V7 1.38 1.26 1.33 1.53 1.43 1.66
V8 1.42 1.63 1.41 1.01 1.42 1.77 1.54
```

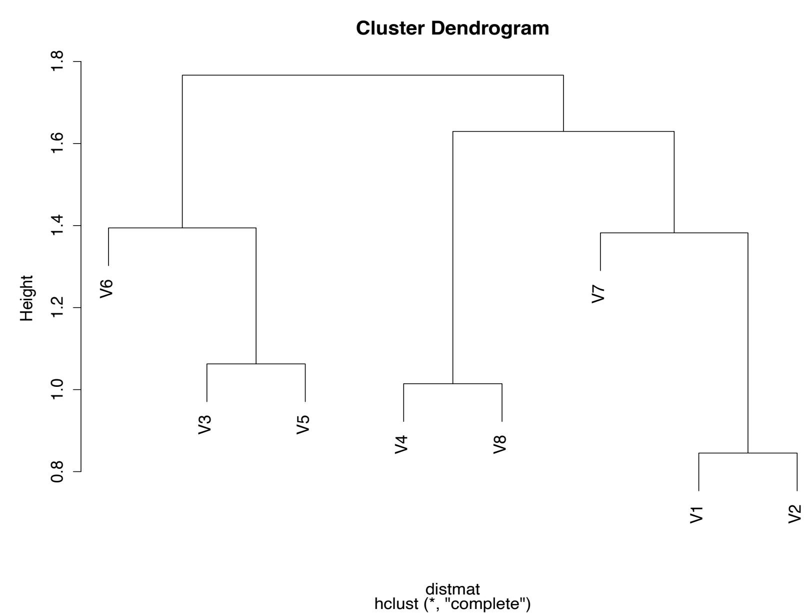
```
## create distance measure from similarity measure - correlations
distmat = sqrt(2*(1- as.dist(cor(X))))
print(distmat, digits=2)
```

Discussion: Is the distance definition reasonable? Other options, and implications?

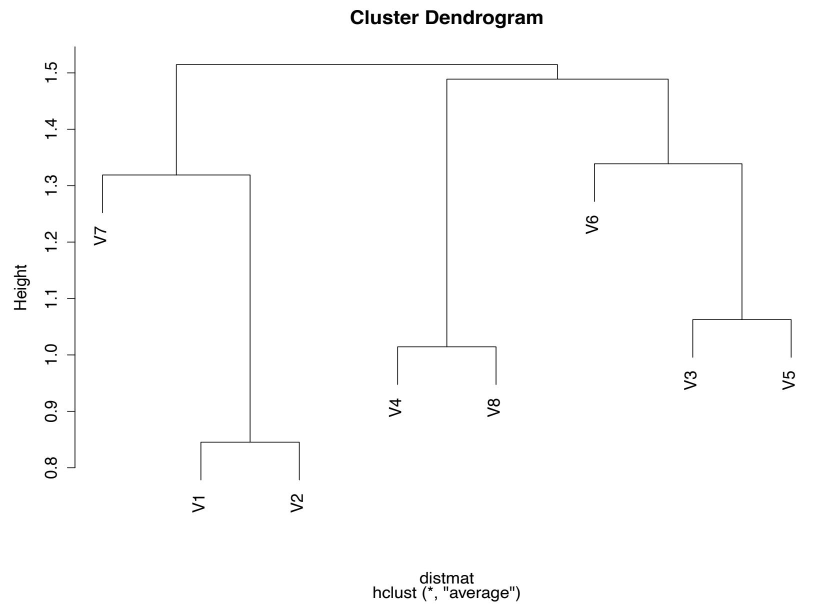
Clustering **variables**: Single linkage



Clustering variables: complete linkage



Clustering variables: average linkage



Are the clusters produced by the three methods different?

Are they reasonable?

The variables are:

- V1 Fixed-charge coverage ratio (income/debt)
- V2 Rate of return on capital
- V3 Cost per kilowatt capacity in place
- V4 Annual load factor
- V5 Peak kilowatt-hour demand growth last year
- V6 Sales (kilowatt-hour used per year)
- V7 Percent nuclear
- V8 Total fuel costs (cents per kilowatt-hour)

Clustering **observations**

Ww need to create a distance matrix **between observations**.

We may use variable values as coordinate and use Euclidean distance.

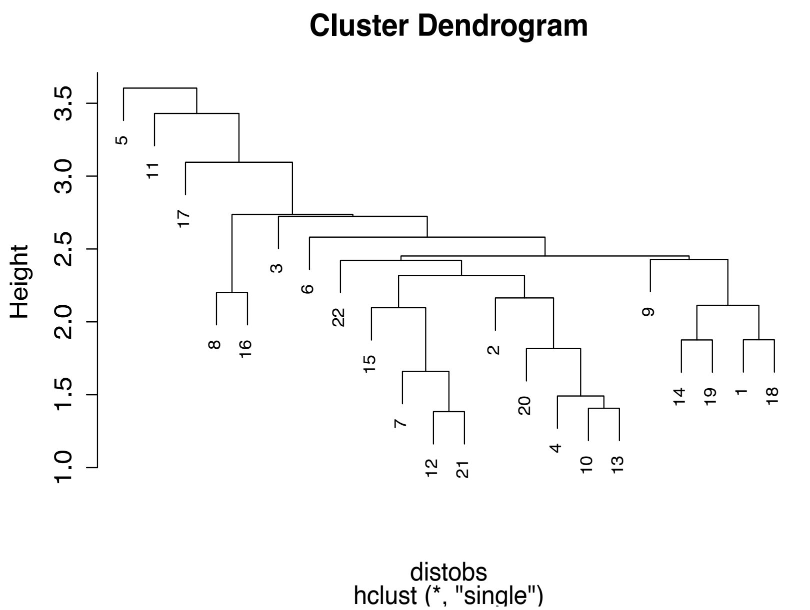
It is often reasonable to use normalized values to create distance measures.

```
NormX = as.matrix(X)%*%solve(diag(sqrt(diag(var(X)))))
distobs=dist(NormX,method="euclidean")
print(distobs,digits =2)
```

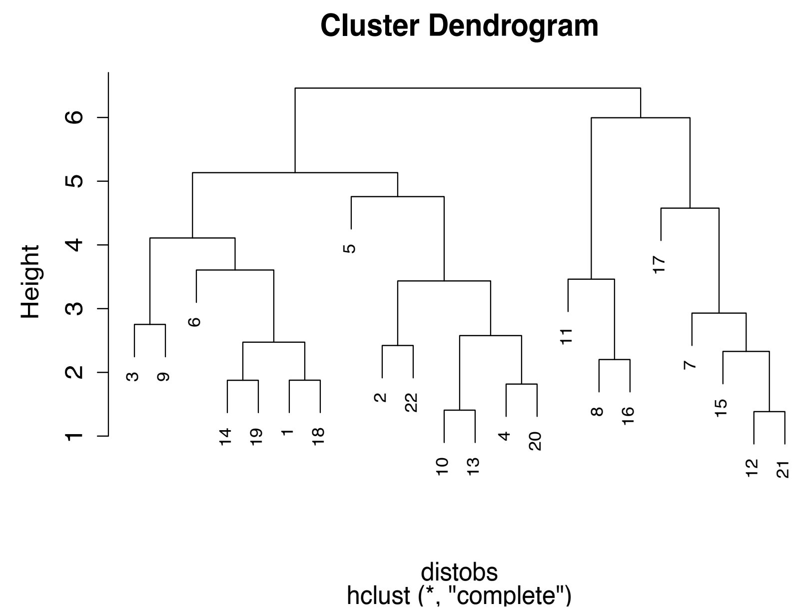

Distance matrix between 22 utility companies

```
     1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
2  3.1
3  3.7 4.9
4  2.5 2.2 4.1
5  4.1 3.9 4.5 4.1
6  3.6 4.2 3.0 3.2 4.6
7  3.9 3.4 4.2 4.0 4.6 3.4
8  2.7 3.9 5.0 3.7 5.2 4.9 4.4
9  3.3 4.0 2.8 3.8 4.5 3.7 2.8 3.6
10 3.1 2.7 3.9 1.5 4.0 3.8 4.5 3.7 3.6
11 3.5 4.8 5.9 4.9 6.5 6.0 6.0 3.5 5.2 5.1
12 3.2 2.4 4.0 3.5 3.6 3.7 1.7 4.1 2.7 3.9 5.2
13 4.0 3.4 4.4 2.6 4.8 4.6 5.0 4.1 3.7 1.4 5.3 4.5
14 2.1 4.3 2.7 3.2 4.8 3.5 4.9 4.3 3.8 3.6 4.3 4.3 4.4
15 2.6 2.5 5.2 3.2 4.3 4.1 2.9 3.8 4.1 4.3 4.7 2.3 5.1 4.2
16 4.0 4.8 5.3 5.0 5.8 5.8 5.0 2.2 3.6 4.5 3.4 4.6 4.4 5.2 5.2
17 4.4 3.6 6.4 4.9 5.6 6.1 4.6 5.4 4.9 5.5 4.8 3.5 5.6 5.6 3.4 5.6
18 1.9 2.9 2.7 2.7 4.3 2.9 2.9 3.2 2.4 3.1 3.9 2.5 3.8 2.3 3.0 4.0 4.4
19 2.4 4.6 3.2 3.5 5.1 2.6 4.5 4.1 4.1 4.1 4.5 4.4 5.0 1.9 4.0 5.2 6.1 2.5
20 3.2 3.0 3.7 1.8 4.4 2.9 3.5 4.1 2.9 2.1 5.4 3.4 2.2 3.7 3.8 4.8 4.9 2.9 3.9
21 3.5 2.3 5.1 3.9 3.6 4.6 2.7 4.0 3.7 4.4 4.9 1.4 4.9 4.9 2.1 4.6 3.1 3.2 5.0 4.1
22 2.5 2.4 4.1 2.6 3.8 4.0 4.0 3.2 3.2 2.6 3.4 3.0 2.7 3.5 3.4 3.5 3.6 2.5 4.0 2.6 3.0
```

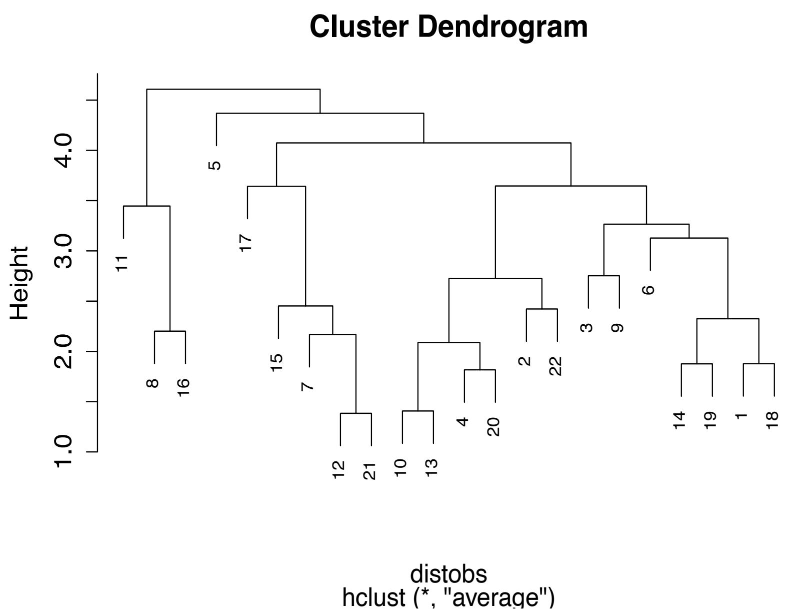
Clustering companies: single linkage



Clustering companies: complete linkage



Clustering companies: average linkage



Firms

```
1    Arizona
2     Boston
3    Central
4     Common
5   Consolid
6    Florida
7   Hawaiian
8      Idaho
9   Kentucky
10   Madison
11    Nevada
12  NewEngla
13  Northern
14  Oklahoma
15   Pacific
16     Puget
17  SanDiego
18  Southern
19     Texas
20  Wisconsi
21    United
22  Virginia
```

Any noticeable patterns in the clusters by firms?