

CLUSTERING: HIERARCHICAL

-APPLIED MULTIVARIATE ANALYSIS-

Lecturer: Darren Homrighausen, PhD

FROM K -MEANS TO HIERARCHICAL CLUSTERING

Recall two properties of K -means clustering

1. It fits exactly K clusters.
2. Final clustering assignments depend on the chosen initial cluster centers.

Alternatively, we can use **hierarchical clustering**. This has the advantage that

1. No need to choose the number of clusters before hand.
2. There is no random component (nor choice of starting point).

FROM K -MEANS TO HIERARCHICAL CLUSTERING

There is a catch: we need to choose a way to measure dissimilarity between clusters, called the **linkage**.

Given the linkage, hierarchical clustering produces a sequence of clustering assignments.

At one end, all points are in their **own** cluster.

At the other, all points are in **one** cluster.

In the middle, there are **nontrivial** solutions.

AGGLOMERATIVE VS. DIVISIVE

Two types of hierarchical clustering algorithms

AGGLOMERATIVE: Start with each point in its own cluster.
Merge until all in same cluster.

(ie: top-down) (think of forward selection)

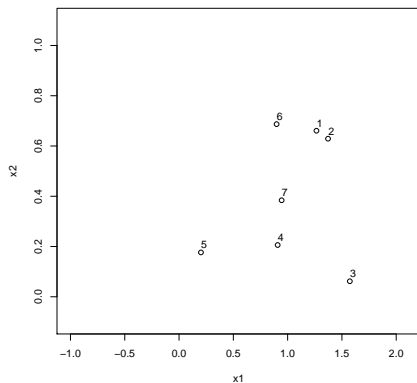
DIVISIVE: Until every point is assigned to its own cluster, repeatedly split the group into two parts that result in the biggest dissimilarity (think of backwards selection).

Agglomerative methods are simpler, so we'll focus on them.

AGGLOMERATIVE EXAMPLE

Given these data points, an agglomerative algorithm **might** decide on the following clustering sequence:

(**IMPORTANT:** Different choices of linkage would result in different solutions)



1. $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2. $\{1, 2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
3. $\{1, 2\}, \{3\}, \{5\}, \{4, 7\}$
4. $\{1, 2, 6\}, \{3\}, \{5\}, \{4, 7\}$
5. $\{1, 2, 4, 6, 7\}, \{3\}, \{5\}$
6. $\{1, 2, 3, 4, 6, 7\}, \{5\}$
7. $\{1, 2, 3, 4, 5, 6, 7\}$

REMINDER: WHAT'S A DENDROGRAM?

We encountered dendrograms when we talked about classification and regression trees.

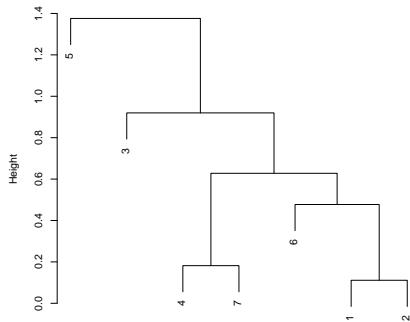
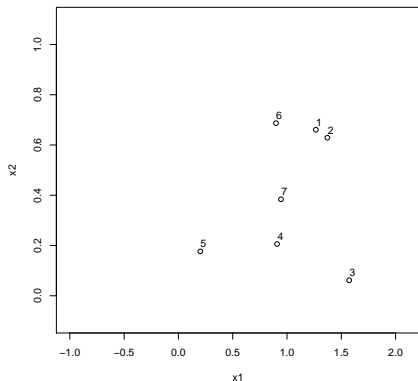
Dendrogram: A convenient graphic to display a hierarchical sequence of clustering assignments. This is simply a tree where:

- Each branch represents a group
- Each leaf (terminal) node is a singleton
(ie: a group containing a single data point)
- The root node is a group containing the whole data set
- Each internal node has two daughter nodes (children), representing the groups that were merged to form it.

Remember: the choice of **linkage** determines how we measure dissimilarity between groups.

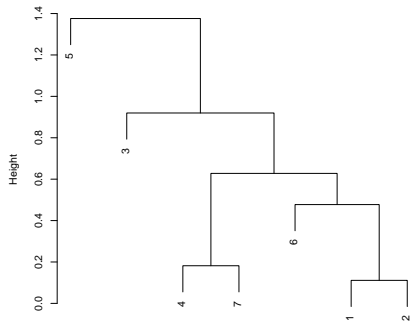
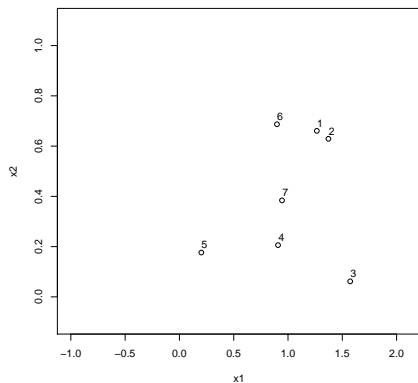
Each internal node is drawn at a height proportional to the **linkage distance** between its two daughter nodes.

We can also represent the sequence of clustering assignments as a **dendrogram**



Note that cutting the dendrogram horizontally partitions the data points into clusters

BACK TO THE EXAMPLE



For instance, the **linkage distance** between the cluster $\{4, 7\}$ and the cluster $\{1, 2, 6\}$ is about .65.

LINKAGES

Notation: Define X_1, \dots, X_n to be the data

Let the **dissimilarities** be d_{ij} between each pair X_i, X_j

At any level, clustering assignments can be expressed by sets $G = \{i_1, i_2, \dots, i_r\}$. given the **indices** of points in this group. Define $|G|$ to be the size of G .

Linkage: The function $d(G, H)$ that takes two groups G, H and returns the **linkage distance** between them.

Agglomerative clustering, given the linkage:

- Start with each point in its own group
- Until there is only one cluster, repeatedly merge the two groups G, H that minimize $d(G, H)$.

SINGLE LINKAGE

In **single linkage** (a.k.a nearest-neighbor linkage), the **linkage distance** between G, H is the smallest dissimilarity between two points in different groups:

$$d_{\text{single}}(G, H) = \min_{i \in G, j \in H} d_{ij}$$

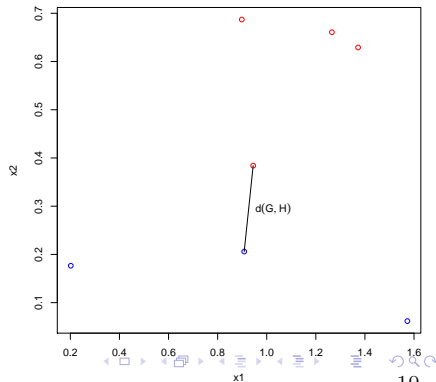
EXAMPLE: There are two clusters G and H (red and blue).

The single linkage distance

(i.e. $d_{\text{single}}(G, H)$)

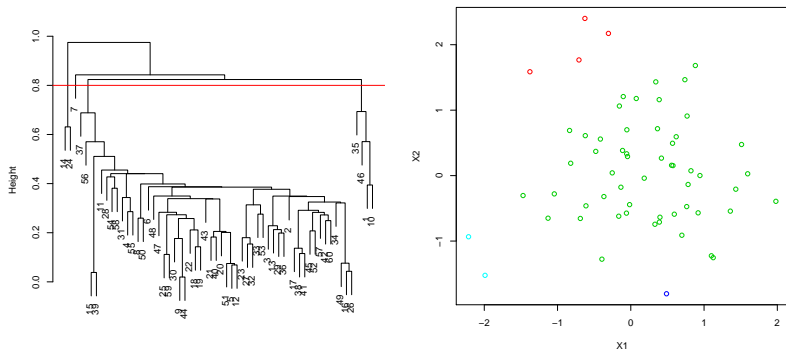
is the dissimilarity between the closest pair

(length of black line segment)



SINGLE LINKAGE EXAMPLE

Here $n = 60$, $X_i \in \mathbb{R}^2$, $d_{ij} = \|X_i - X_j\|_2$. Cutting the tree at $h = 0.8$ gives the cluster assignments marked by colors



Cut interpretation: For each point X_i , there is another point X_j in the same cluster with $d_{ij} \leq 0.8$ (assuming more than 1 point in cluster). Also, no points in different clusters are closer than 0.8.

COMPLETE LINKAGE

In **complete linkage** (i.e. farthest-neighbor linkage), dissimilarity between G, H is the **largest** dissimilarity between two points in different clusters:

$$d_{\text{complete}}(G, H) = \max_{i \in G, j \in H} d_{ij}.$$

EXAMPLE: There are two clusters G and H (red and blue).

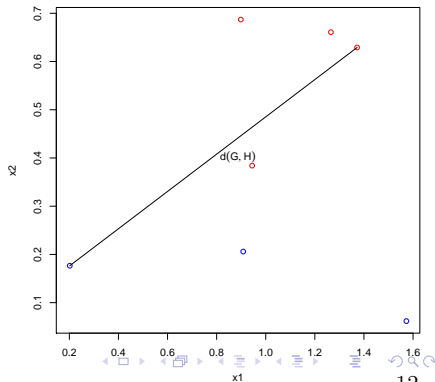
The complete linkage distance

(i.e. $d_{\text{complete}}(G, H)$)

is the dissimilarity between the

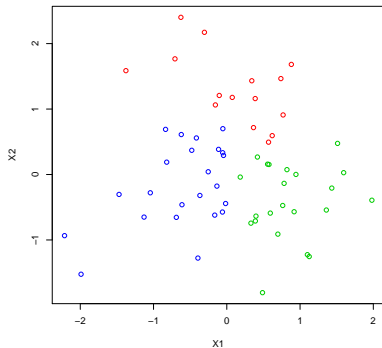
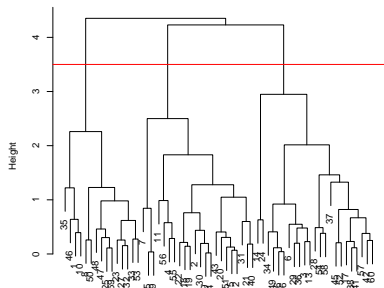
farthest pair

(length of black line segment)



COMPLETE LINKAGE EXAMPLE

Same data as before. Cutting the tree at $h = 3.5$ gives the clustering assignment



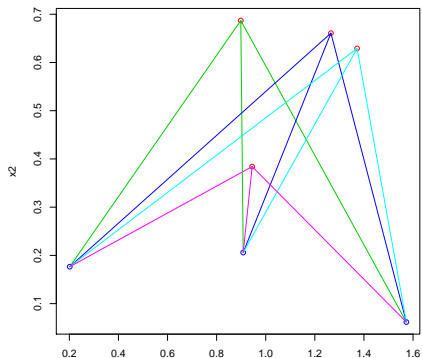
Cut interpretation: For each point X_i , every other point X_j in the same cluster has $d_{ij} \leq 3.5$.

AVERAGE LINKAGE

In **average linkage**, the **linkage distance** between G, H is the **average** dissimilarity over all points in different clusters:

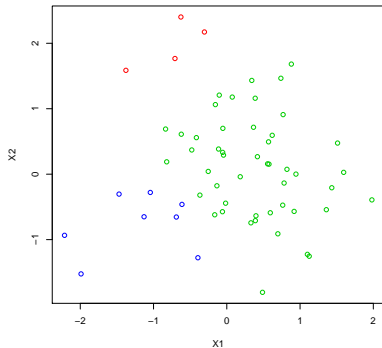
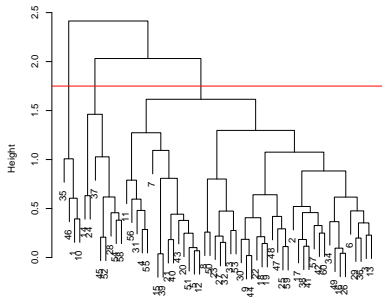
$$d_{\text{average}}(G, H) = \frac{1}{|G| \cdot |H|} \sum_{i \in G, j \in H} d_{ij}.$$

EXAMPLE: There are two clusters G and H (red and blue). The average linkage distance (i.e. $d_{\text{average}}(G, H)$) is the average dissimilarity between **all** points in different clusters (average of lengths of colored line segments)



AVERAGE LINKAGE EXAMPLE

Same data as before. Cutting the tree at $h = 1.75$ gives the clustering assignment



Cut interpretation: ??

COMMON PROPERTIES

Single, complete, and average linkage share the following:

- They all operate on the dissimilarities d_{ij} . This means that the points we are clustering can be quite general (number of mutations on a genome, polygons, faces, whatever).
- Running agglomerative clustering with any of these linkages produces a dendrogram with no **inversions**.

No inversions means that the **linkage distance** between merged clusters only **increases** as we run the algorithm.

In other words, we can draw a proper dendrogram, where the height of a parent is always higher than the height of either daughter.

(We'll return to this again shortly)

SHORTCOMINGS OF SINGLE AND COMPLETE LINKAGE

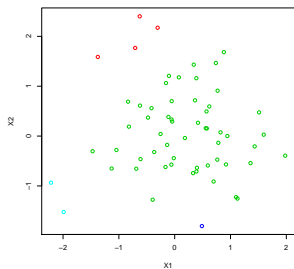
Single and complete linkage have practical problems:

SINGLE LINKAGE: Often suffers from **chaining**, that is, we only need a single pair of points to be close to merge two clusters. Therefore, clusters can be too spread out and not compact enough.

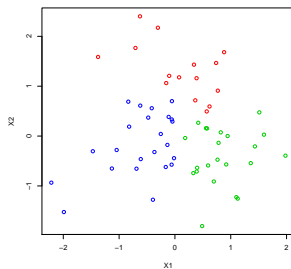
COMPLETE LINKAGE: Often suffers from **crowding**, that is, a point can be closer to points in other clusters than to points in its own cluster. Therefore, the clusters are compact, but not far enough apart.

Average linkage tries to strike a balance between these two.

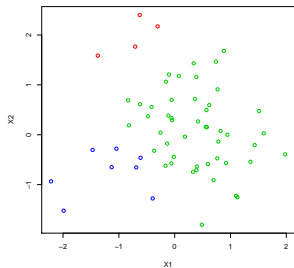
EXAMPLE OF CHAINING AND CROWDING



Single linkage



Complete linkage



Average linkage

SHORTCOMINGS OF AVERAGE LINKAGE

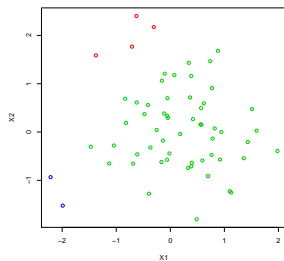
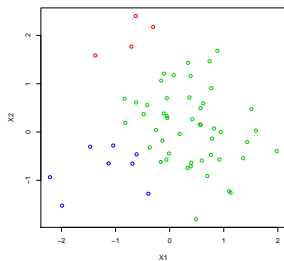
Average linkage isn't perfect.

- It isn't clear what properties the resulting clusters have when we cut an average linkage tree.
- Results of average linkage clustering can change with a monotone increasing transformation of the dissimilarities (that is, if we changed the distance, but maintained the ranking of the distances, the cluster solution could change).

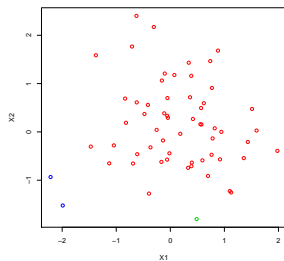
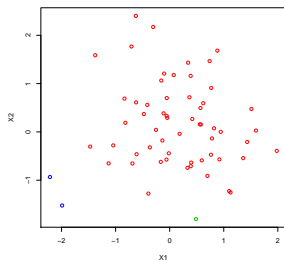
Neither of these problems afflict single or complete linkage.

EXAMPLE OF MONOTONE INCREASING PROBLEM

Average



Single



Left: $d_{ij} = \|X_i - X_j\|_2$

Right: $d_{ij} = \|X_i - X_j\|_2^2$

HIERARCHICAL AGGLOMERATIVE CLUSTERING IN R

There's an easy way to generate **distance-based** dissimilarities in R

```
> x = c(1,2)
> y = c(2,1)
> dist(cbind(x,y),method='euclidean')
      1
2 1.414214
> dist(cbind(x,y),method='euclidean',diag=T,upper=T)
      1      2
1 0.000000 1.414214
2 1.414214 0.000000
```

HIERARCHICAL AGGLOMERATIVE CLUSTERING IN R

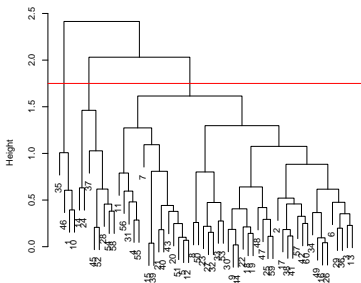
The function `hclust` is in base R

```
Delta = dist(x)
```

```
out.average = hclust(Delta,method='average')
```

```
plot(out.average)
```

```
abline(h = 1.75,col='red')
```



```
cutree(out.tree,k=3)
```

```
cutree(out.tree,h=1.75)
```

RECAP

HIERARCHICAL AGGLOMERATIVE CLUSTERING: Start with all data points in their own groups, and repeatedly merge groups, based on linkage function. Stop when points are in one group (this is agglomerative; there is also divisive)

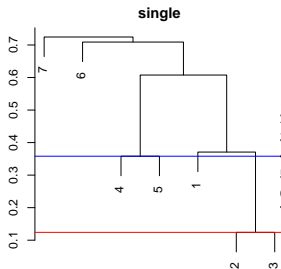
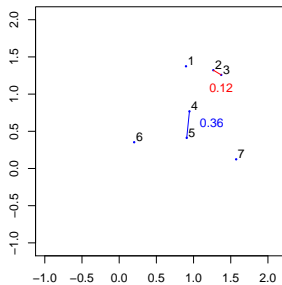
This produces a sequence of clustering assignments, visualized by a dendrogram (i.e., a tree). Each node in the tree represents a group, and its height is proportional to the **linkage distance** of its daughters

Three most common linkage functions: **single**, **complete**, **average** linkage. Single linkage measures the least dissimilar pair between groups, complete linkage measures the most dissimilar pair, average linkage measures the average dissimilarity over all pairs

Each linkage has its strengths and weaknesses

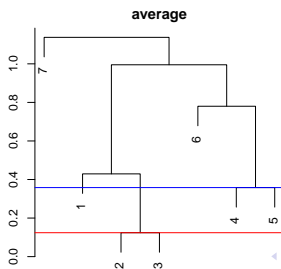
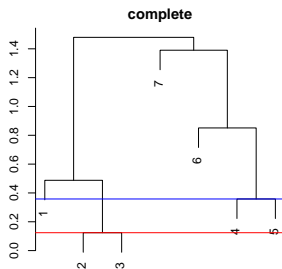
Careful example

CAREFUL EXAMPLE



Distance matrix (Δ)

	1	2	3	4	5	6	7
1	0.00	0.37	0.49	0.61	0.96	1.24	1.42
2	0.37	0.00	0.12	0.64	0.98	1.44	1.24
3	0.49	0.12	0.00	0.65	0.97	1.48	1.15
4	0.61	0.64	0.65	0.00	0.36	0.85	0.90
5	0.96	0.98	0.97	0.36	0.00	0.71	0.72
6	1.24	1.44	1.48	0.85	0.71	0.00	1.39
7	1.42	1.24	1.15	0.90	0.72	1.39	0.00



(All Merging {1} and {2, 3})

SINGLE: 0.37
 COMPLETE: 0.49
 AVERAGE: $(0.37 + 0.49)/2 = 0.43$

(Next Agglomeration)

SINGLE: Merging {4, 5} & {1, 2, 3}
 0.61
 COMPLETE: Merging {4, 5} & {6}
 0.85
 AVERAGE: Merging {4, 5} & {6}
 $(0.85 + 0.71)/2 = 0.78$

ANOTHER LINKAGE

Centroid linkage is a commonly used and relatively new approach.

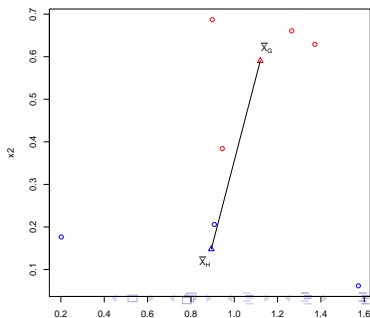
Assume

- $X_i \in \mathbb{R}^p$
- $d_{ij} = \|X_i - X_j\|_2$

Let \bar{X}_G and \bar{X}_H denote group averages for G, H . Then

$$d_{\text{centroid}} = \|\bar{X}_G - \bar{X}_H\|_2$$

Example: There are two clusters (red and blue). The centroid linkage distance ($d_{\text{centroid}}(G, H)$) is the distance between the centroids (black line segment).



CENTROID LINKAGE

Centroid linkage is

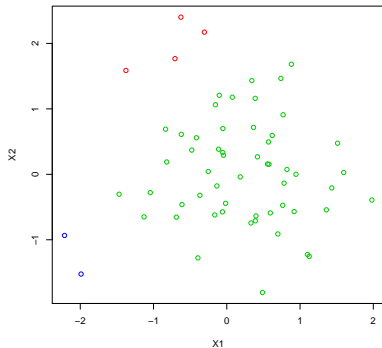
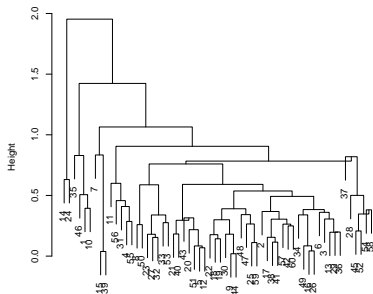
- ... quite intuitive
- ... widely used
- ... nicely analogous to K -means.
- ... very related to average linkage (and much, much faster)

However, it has a very unsavory feature: **inversions**.

An inversion is when an agglomeration doesn't reduce the linkage distance.

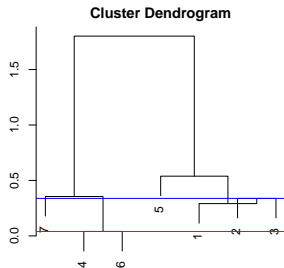
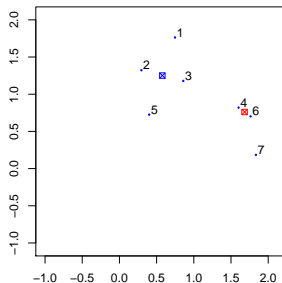
CENTROID LINKAGE EXAMPLE

Same data as before. We can't look at cutting the tree, but we can still look at a 3 cluster solution.



Cut interpretation: Even if there are no inversions, there still is no cut interpretation.

CAREFUL EXAMPLE: STEPS 1,2,3



Distance matrix (Δ)

	1	2	3	4	5	6	7
1	0.00	0.40	0.35	1.62	1.20	2.16	3.67
2	0.40	0.00	0.34	1.96	0.37	2.54	3.66
3	0.35	0.34	0.00	0.68	0.42	1.05	1.94
4	1.62	1.96	0.68	0.00	1.45	0.04	0.46
5	1.20	0.37	0.42	1.45	0.00	1.86	2.35
6	2.16	2.54	1.05	0.04	1.86	0.00	0.27
7	3.67	3.66	1.94	0.46	2.35	0.27	0.00

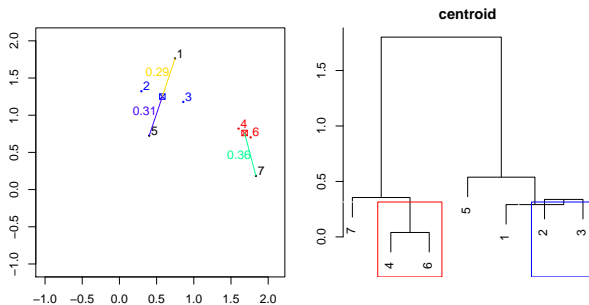
Centroid(4,6) = (1.68,0.76)

Centroid(2,3) = (0.58,1.25)

Centroid distance²: 1.46

(This is squared Euclidean distance)

CAREFUL EXAMPLE: STEP 4



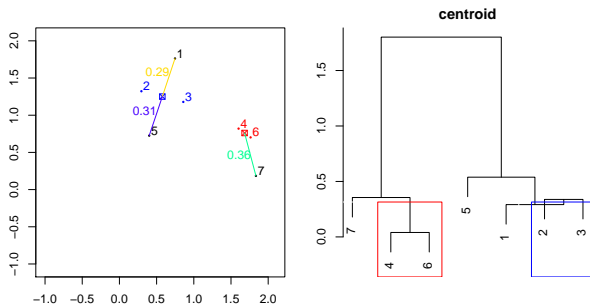
Distance matrix (Δ)

	1	2	3	4	5	6	7
1	0.00	0.40	0.35	1.62	1.20	2.16	3.67
2	0.40	0.00	0.34	1.96	0.37	2.54	3.66
3	0.35	0.34	0.00	0.68	0.42	1.05	1.94
4	1.62	1.96	0.68	0.00	1.45	0.04	0.46
5	1.20	0.37	0.42	1.45	0.00	1.86	2.35
6	2.16	2.54	1.05	0.04	1.86	0.00	0.27
7	3.67	3.66	1.94	0.46	2.35	0.27	0.00

Which one gets merged?

(This is squared Euclidean distance)

CAREFUL EXAMPLE: STEP 4



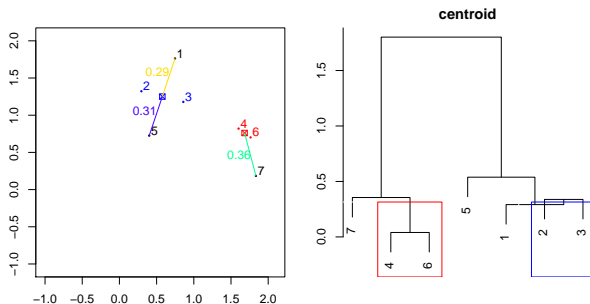
Distance matrix (Δ)

	1	2	3	4	5	6	7
1	0.00	0.40	0.35	1.62	1.20	2.16	3.67
2	0.40	0.00	0.34	1.96	0.37	2.54	3.66
3	0.35	0.34	0.00	0.68	0.42	1.05	1.94
4	1.62	1.96	0.68	0.00	1.45	0.04	0.46
5	1.20	0.37	0.42	1.45	0.00	1.86	2.35
6	2.16	2.54	1.05	0.04	1.86	0.00	0.27
7	3.67	3.66	1.94	0.46	2.35	0.27	0.00

Which one gets merged?
($\{1\}$ and $\{2,3\}$)

(This is squared Euclidean distance)

CAREFUL EXAMPLE: STEP 4



Distance matrix (Δ)

	1	2	3	4	5	6	7
1	0.00	0.40	0.35	1.62	1.20	2.16	3.67
2	0.40	0.00	0.34	1.96	0.37	2.54	3.66
3	0.35	0.34	0.00	0.68	0.42	1.05	1.94
4	1.62	1.96	0.68	0.00	1.45	0.04	0.46
5	1.20	0.37	0.42	1.45	0.00	1.86	2.35
6	2.16	2.54	1.05	0.04	1.86	0.00	0.27
7	3.67	3.66	1.94	0.46	2.35	0.27	0.00

Which one gets merged?
($\{1\}$ and $\{2,3\}$)

```
method = 'centroid'  
out = hclust(Delta,method=method)  
rect.hclust(out,k=5,  
            border=c('white','red','white','white','blue'))
```

(This is squared Euclidean distance)

Linkage Summary

LINKAGES SUMMARY

	No inversions?	Unchanged w/ monotone transforma- tion?	Cut interpre- tation?	Notes
SINGLE	✓	✓	✓	chaining
COMPLETE	✓	✓	✓	crowding
AVERAGE	✓	X	X	
CENTROID	X	X	X	inversions

Final notes:

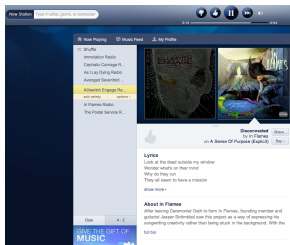
- None of this helps determine what is the **best** linkage
- Use the linkage that seems the most appropriate for the types of clusters you want to get

DESIGNING A CLEVER RADIO SYSTEM

We have a lot of songs and dissimilarities between them (d_{ij})

We want to build a clever radio system that takes a song specified by the user and produces a song of the “same” type

We ask the user how “risky” he or she wants to be



How can we use hierarchical clustering and with what linkage?

LINKAGES SUMMARY: CUT INTERPRETATIONS

Suppose we cut the tree at height $h = 1$.

SINGLE	For each point X_i , there is another point X_j in the same cluster with $d_{ij} \leq 1$ (assuming more than 1 point in cluster). Also, no points in different clusters are closer than 1.
COMPLETE	For each point X_i , every other point X_j in the same cluster has $d_{ij} \leq 1$.

DATA ANALYSIS EXAMPLE

Diffuse large B-cell lymphoma (DLBCL) is the most common type of non-Hodgkin's lymphoma

It is clinically heterogeneous:

- 40% of patients respond well
- 60% of patients succumb to the disease

The researchers propose that this difference is due to unrecognized molecular heterogeneity in the tumors

We examine the extent to which genomic-scale gene expression profiling can further the understanding of B-cell malignancies.

DATA ANALYSIS EXAMPLE

Here, we have gene expression data at 2,000 genes for 62 cancer cells.

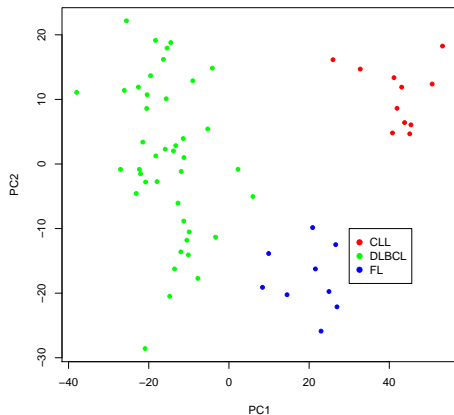
There are 3 cancer diagnoses: FL, CLL, DLBCL. Each corresponds to a type of malignant lymphoma.

We want to use hierarchical clustering to understand this data set better.

```
load(' ../data/alizadeh.RData')

genesT = alizadeh$x
genes  = t(genesT)
Yfull  = alizadeh$type
Y      = as.vector(Yfull)
Y[Yfull == "DLBCL-A"] = 'DLBCL'
Y[Yfull == "DLBCL-G"] = 'DLBCL'
Y      = as.factor(Y)
dist.mat = dist(genes)
```

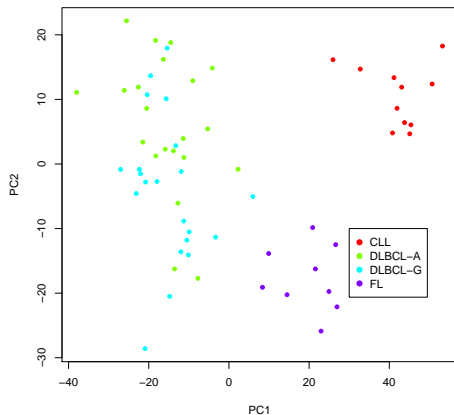
PCA PLOT



Two clear groups for
FL and CLL

DLBCL somewhat
appears to be 1 group,
but it is much more
diffuse.

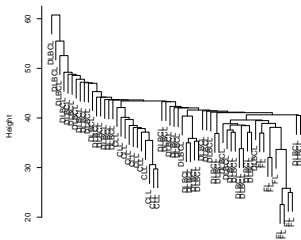
PCA PLOT



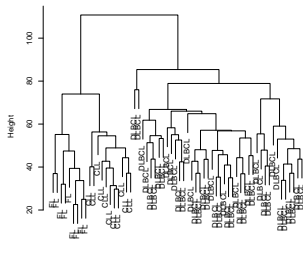
Here are the two sub-types identified by the researchers

Let's look at their results further.

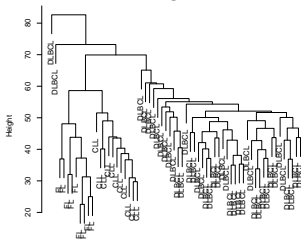
FOUR HIERARCHICAL CLUSTER SOLUTIONS



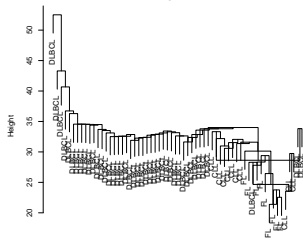
Single



Complete

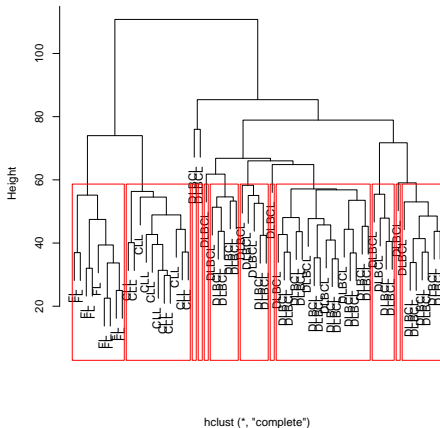


Average



Centroid

COMPLETE LINKAGE: A CLOSER LOOK



```
out.com = hclust(dist.mat,  
method='complete')  
plot(out.com,xlab='',  
main='',labels=Y)  
rect.hclust(out.com,k=12)  
  
out.cut = cutree(out.com,  
k=12)
```

Notice that FL and CLL are distinctly grouped, while there are many clusters inside the DLBCL type.

DATA ANALYSIS EXAMPLE: CONCLUSION

We find evidence of more sub-types of distinct cancers in previously diagnoses

This result is quite interesting

Clinically heterogenous outcomes could be driven by inadequate clusterings