

# CLUSTERING

## -STATISTICAL MACHINE LEARNING-

Lecturer: Darren Homrighausen, PhD

# K-MEANS

1. Select a number of clusters  $K$ .
2. Let  $C_1, \dots, C_K$  partition  $\{1, 2, 3, \dots, n\}$  such that
  - ▶ All observations belong to some set  $C_j$ .
  - ▶ No observation belongs to more than one set.
3. K-means attempts to form these sets by making **within-cluster variation**,  $W(C_k)$ , as small as possible.

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K W(C_k).$$

4. To Define  $W$ , we need a concept of distance. By far the most common is Euclidean

$$W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \|X_i - X_{i'}\|_2^2.$$

That is, the average (Euclidean) distance between all cluster members.

# K-MEANS

It turns out

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K W(C_k). \quad (1)$$

is too hard of a problem to solve computationally ( $K^n$  partitions!).

So, we make a greedy approximation:

1. Randomly assign observations to the  $K$  clusters
2. Iterate until the cluster assignments stop changing:
  - ▶ For each of  $K$  clusters, compute the **centroid**, which is the  $p$ -length vector of the means in that cluster.
  - ▶ Assign each observation to the cluster whose centroid is closest (in Euclidean distance).

This procedure is guaranteed to decrease (1) at each step.

# K-MEANS: A SUMMARY

To fit K-means, you need to

1. Pick  $K$  (inherent in the method)
2. Convince yourself you have found a good solution (due to the randomized approach to the algorithm).

It turns out that 1. is difficult to do in a principled way. We will discuss this next

For 2., a commonly used approach is to run K-means many times with different starting points. Pick the solution that has the smallest value for

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K W(C_k)$$

# Choosing $K$

# CHOOSING THE NUMBER OF CLUSTERS

Why is it important?

- It might make a big difference (concluding there are  $K = 2$  cancer sub-types versus  $K = 3$ ).
- One of the major goals of statistical learning is automatic inference. A good way of choosing  $K$  is certainly a part of this.

## REMINDER: WHAT DOES $K$ -MEANS DO?

Given a number of clusters  $K$ , we (approximately) minimize:

$$\sum_{k=1}^K W(C_k) = \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i,i' \in C_k} \|X_i - X_{i'}\|_2^2.$$

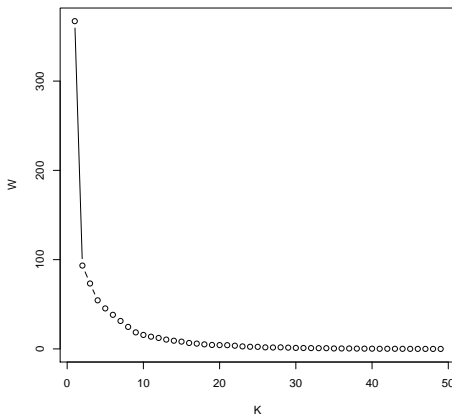
We can rewrite this in terms of the **centroids** as

$$W(K) = \sum_{k=1}^K \sum_{i \in C_k} \|X_i - \bar{X}_k\|_2^2,$$

## MINIMIZING $W$ IN $K$

Of course, a lower value of  $W$  is better. Why not minimize  $W$ ?

```
plotW = rep(0,49)
for(K in 1:49){
  plotW[K] = kmeans(x,centers=K,nstart=20)$tot.withinss
}
```

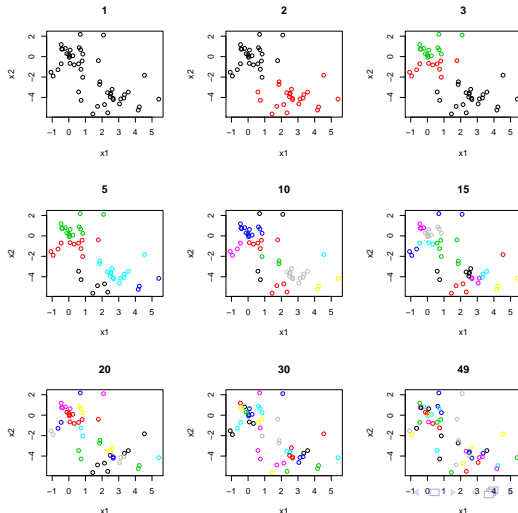




# MINIMIZING $W$ IN $K$

Of course, a lower value of  $W$  is better. Why not minimize  $W$ ?

A look at the cluster solution



## BETWEEN-CLUSTER VARIATION

Within-cluster variation measures how **tightly grouped** the clusters are. As we increase  $K$ , this will always decrease.

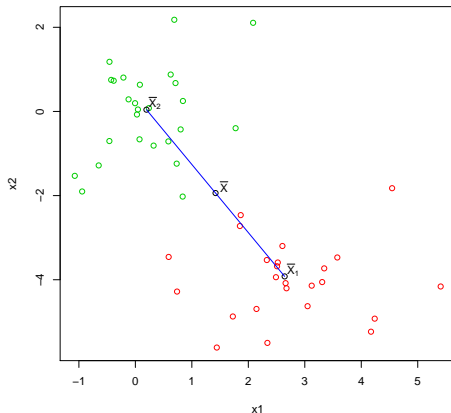
What we are missing is **between-cluster variation**, ie: how spread apart the groups are

$$B = \sum_{k=1}^K |C_k| \|\bar{X}_k - \bar{X}\|_2^2,$$

where  $|C_k|$  is the number of points in  $C_k$ , and  $\bar{X}$  is the grand mean of all observations:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i.$$

# BETWEEN-CLUSTER VARIATION: EXAMPLE

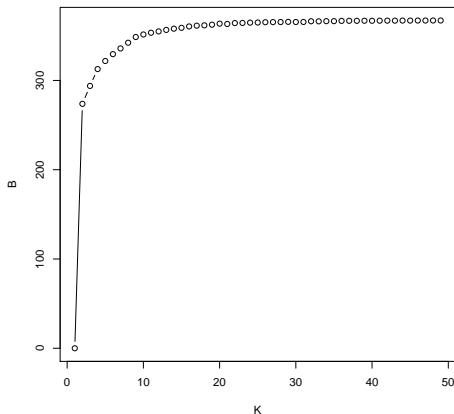


$$B = |C_1| \|\bar{X}_1 - \bar{X}\|_2^2 + |C_2| \|\bar{X}_2 - \bar{X}\|_2^2$$

$$W = \sum_{i \in C_1} |C_1| \|\bar{X}_1 - X_i\|_2^2 + \sum_{i \in C_2} |C_2| \|\bar{X}_2 - X_i\|_2^2$$

## CAN WE JUST MAXIMIZE $B$ ?

Sadly, no. Just like  $W$  can be made arbitrarily small,  $B$  will always be increasing with increasing  $K$ .



## CH INDEX

Ideally, we would like our cluster assignment to **simultaneously** have small  $W$  and large  $B$ .

This is the idea behind **CH index**. For clustering assignments coming from  $K$  clusters, we record  $CH$  score:

$$CH(K) = \frac{B(K)/(K-1)}{W(K)/(n-k)}$$

To choose  $K$ , pick some maximum number of clusters to be considered ( $K_{\max} = 20$ , for example) and choose the value of  $K$  that

$$\hat{K} = \arg \max_{K \in \{2, \dots, K_{\max}\}} CH(K).$$

**Note:**  $CH$  is undefined for  $K = 1$ .

(Calinski, Harabasz (1974))

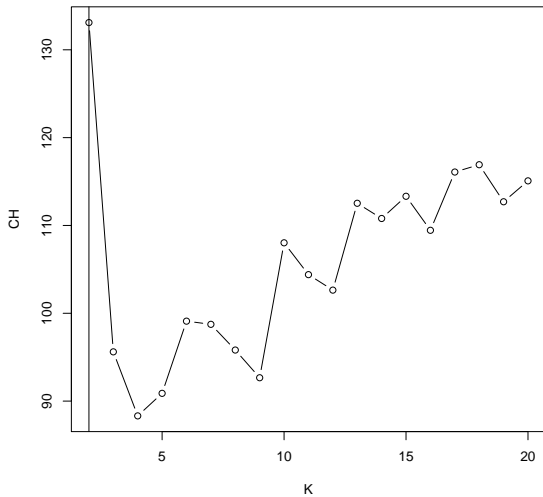
```
ch.index = function(x,kmax,iter.max=100,nstart=10,
                    algorithm="Lloyd")
{
  ch = numeric(length=kmax-1)
  n = nrow(x)
  for (k in 2:kmax) {
    a = kmeans(x,k,iter.max=iter.max,nstart=nstart,
              algorithm=algorithm)
    w = a$tot.withinss
    b = a$betweenss
    ch[k-1] = (b/(k-1))/(w/(n-k))
  }
  return(list(k=2:kmax,ch=ch))
}
```

# A SIMULATED EXAMPLE

```
x = matrix(rnorm(50*2),ncol=2)
x[1:25,1] = x[1:25,1] + 3
x[1:25,2] = x[1:25,2] -4
```

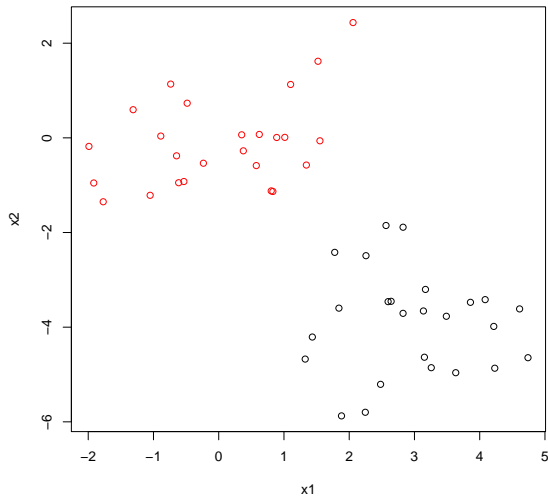
We want to cluster this data set using K-means with  $K$  chosen via  $CH$  index.

# CH PLOT





# CORRESPONDING SOLUTION



## ALTERNATE APPROACH: GAP STATISTIC

While true that  $W(K)$  keeps dropping in  $K$ , **how much it drops** might be informative.

The **gap statistic** is based on this idea. We compare the observed within-cluster variation  $W(K)$  to the within-cluster variation we would observe if the data were uniformly distributed  $W_{unif}(K)$ .

$$Gap(K) = \log W(K) - \log W_{unif}(K)$$

After simulating many  $\log W_{unif}(K)$ , we compute its standard deviation  $s(K)$ . Then, we choose  $K$  by

$$\hat{K} = \min\{K \in \{1, \dots, K_{\max}\} : Gap(K) \geq Gap(K+1) - s(K+1)\}.$$

(Tibshirani et al. (2001))

# Hierarchical clustering

# 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.

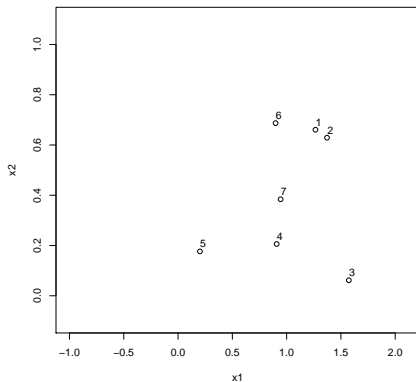
At the other, all points are in **one** cluster.

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

# AGGLOMERATIVE EXAMPLE

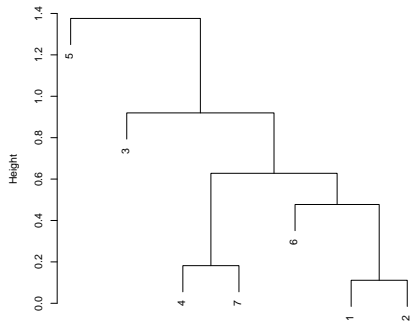
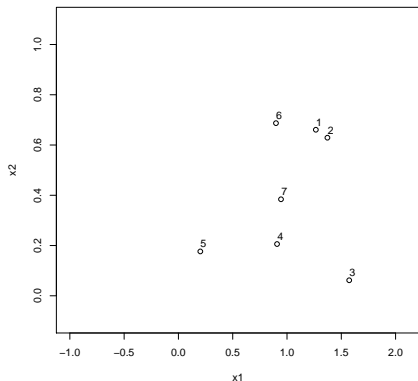
Given these data points, an agglomerative algorithm **might** decide on the following clustering sequence:

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



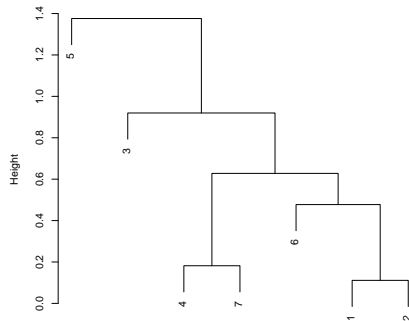
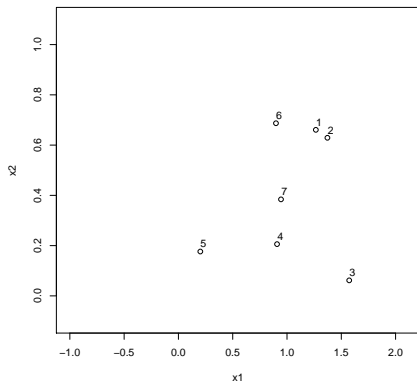
1.  $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2.  $\{1, 2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
3.  $\{1, 2\}, \{3\}, \{5\}, \{4, 7\}$
4.  $\{1, 2, 6\}, \{3\}, \{5\}, \{4, 7\}$
5.  $\{1, 2, 4, 6, 7\}, \{3\}, \{5\}$
6.  $\{1, 2, 3, 4, 6, 7\}, \{5\}$
7.  $\{1, 2, 3, 4, 5, 6, 7\}$

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 dissimilarity 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 dissimilarity score between them.

Agglomerative clustering, given the linkage:

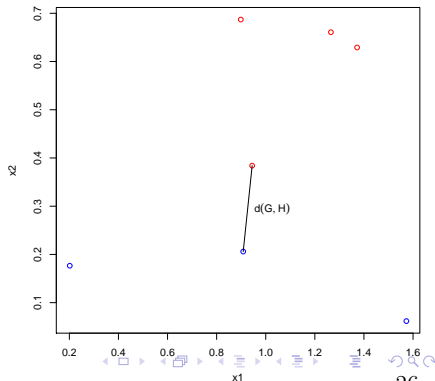
- Start with each point in its own group
- Until there is only one cluster, repeatedly merge the two groups  $G, H$  that minimize  $d(G, H)$ .

# SINGLE LINKAGE

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

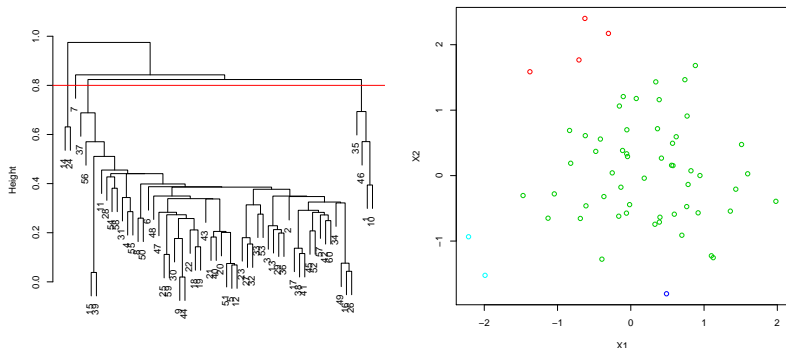
$$d_{\text{single}}(G, H) = \min_{i \in G, j \in H} d_{ij}$$

**EXAMPLE:** There are two clusters  $G$  and  $H$  (red and blue).  
The single linkage score  
(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:

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

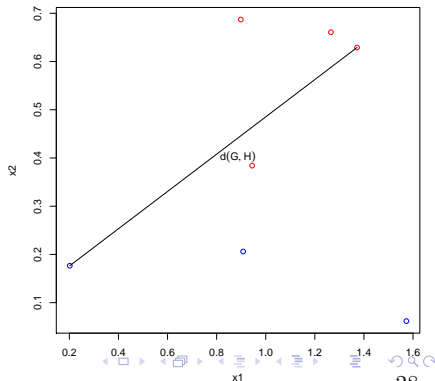
**EXAMPLE:** There are two clusters  $G$  and  $H$  (red and blue).

The complete linkage score

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

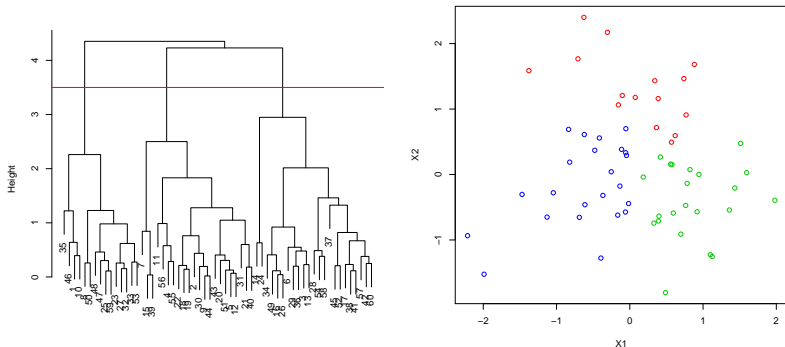
is the dissimilarity between the **farthest** pair

(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 dissimilarity between  $G, H$  is the **average** dissimilarity over all points in different clusters:

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

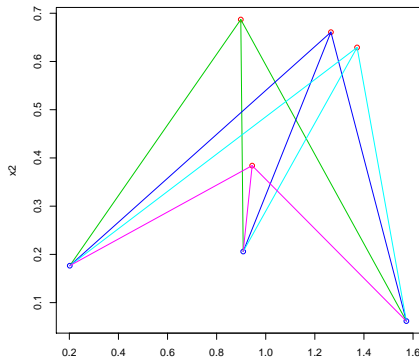
**EXAMPLE:** There are two clusters  $G$  and  $H$  (**red** and **blue**).

The average linkage score

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

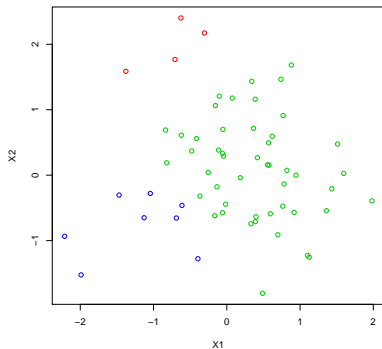
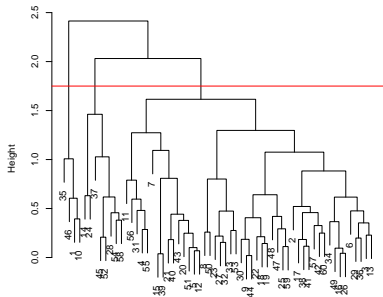
is the average dissimilarity between **all** points in different clusters

(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 dissimilarity scores 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

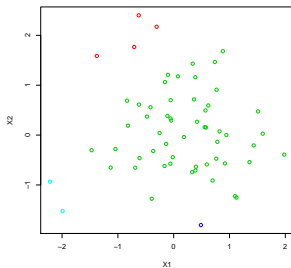
Single and complete linkage have practical problems:

**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.

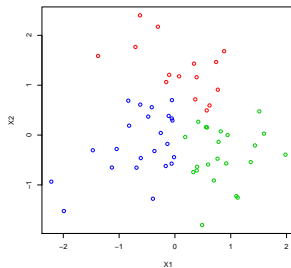
**COMPLETE LINKAGE:** Often suffers from **crowding**, that is, a point can be closer to points in other clusters than to points in its own cluster. Therefore, the clusters are compact, but not far enough apart.

Average linkage tries to strike a balance between these two.

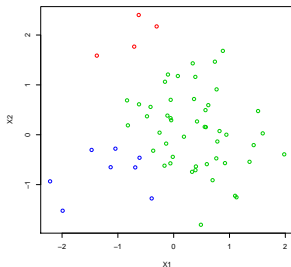
# EXAMPLE OF CHAINING AND CROWDING



Single linkage



Complete linkage



Average linkage

# SHORTCOMINGS OF AVERAGE LINKAGE

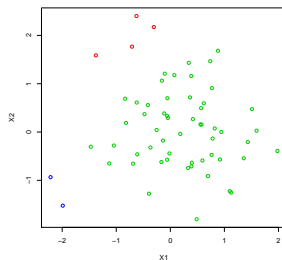
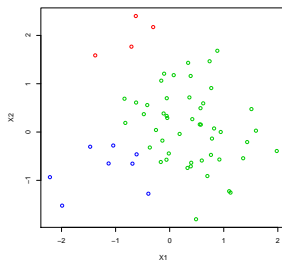
Average linkage isn't perfect.

- 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).

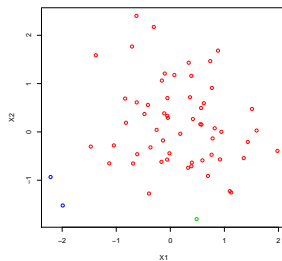
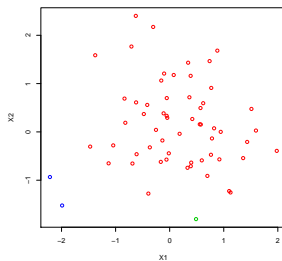
Neither of these problems afflict single or complete linkage.

# 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

The function `hclust` in base R performs the necessary computations. E.g.

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

# RECAP

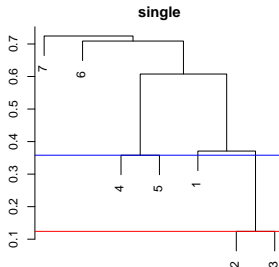
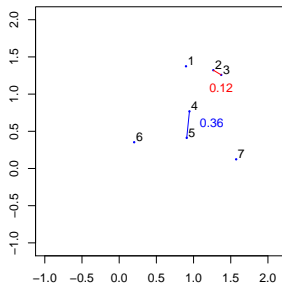
**HIERARCHICAL AGGLOMERATIVE CLUSTERING:** Start with all data points in their own groups, and repeatedly merge groups, based on linkage function. Stop when points are in one group (this is agglomerative; there is also divisive)

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 dissimilarity 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

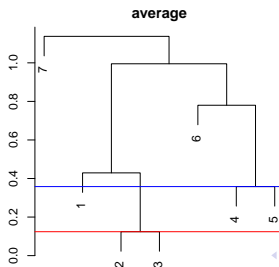
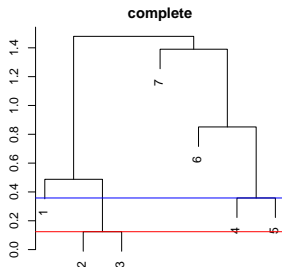


Distance matrix ( $\Delta$ )

	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.

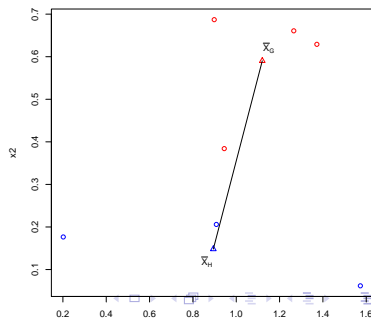
Assume

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

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 score ( $d_{\text{centroid}}(G, H)$ ) is the distance between the centroids (black line segment).





# CENTROID LINKAGE

Centroid linkage is

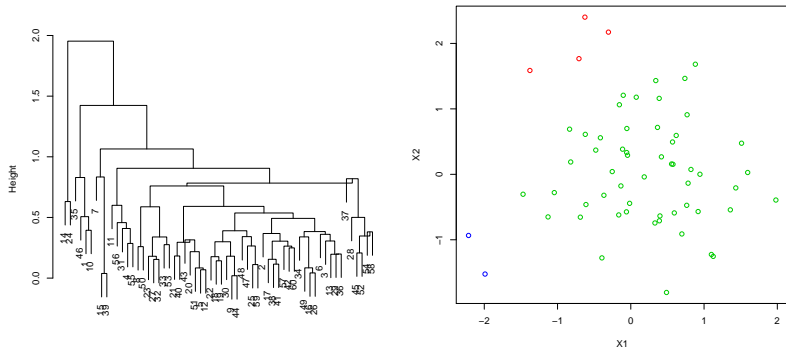
- ... quite intuitive
- ... widely used
- ... nicely analogous to  $K$ -means.
- ... very related to average linkage (and much, much faster)

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

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

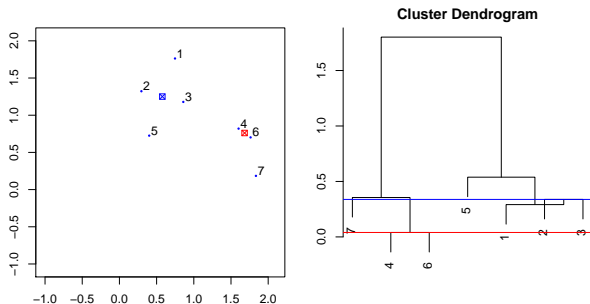
# CENTROID LINKAGE EXAMPLE

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 ( $\Delta$ )

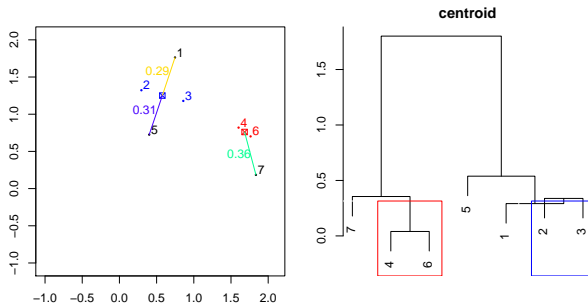
	1	2	3	4	5	6	7
1	0.00	0.40	0.35	1.62	1.20	2.16	3.67
2	0.40	0.00	0.34	1.96	0.37	2.54	3.66
3	0.35	0.34	0.00	0.68	0.42	1.05	1.94
4	1.62	1.96	0.68	0.00	1.45	0.04	0.46
5	1.20	0.37	0.42	1.45	0.00	1.86	2.35
6	2.16	2.54	1.05	0.04	1.86	0.00	0.27
7	3.67	3.66	1.94	0.46	2.35	0.27	0.00

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

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

(This is squared Euclidean distance)

# CAREFUL EXAMPLE: STEP 4



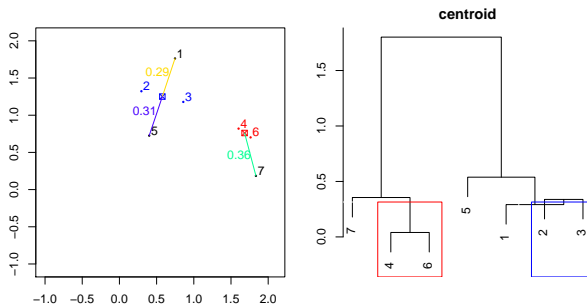
Distance matrix ( $\Delta$ )

	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



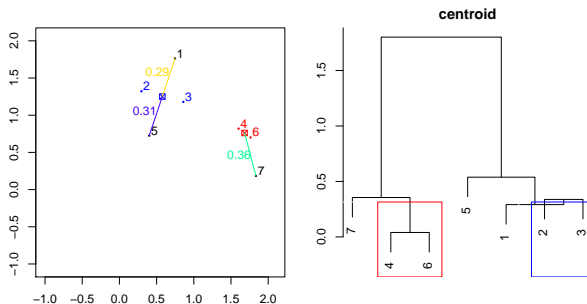
Distance matrix ( $\Delta$ )

	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 ( $\Delta$ )

	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)

# 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:

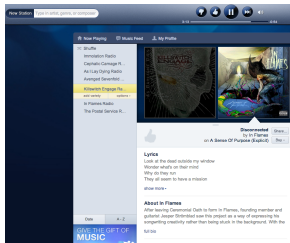
- None of this helps determine what is the **best** linkage
- Use the linkage that seems the most appropriate for the types of clusters you want to get

# DESIGNING A CLEVER RADIO SYSTEM

We have a lot of songs and dissimilarity scores 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.

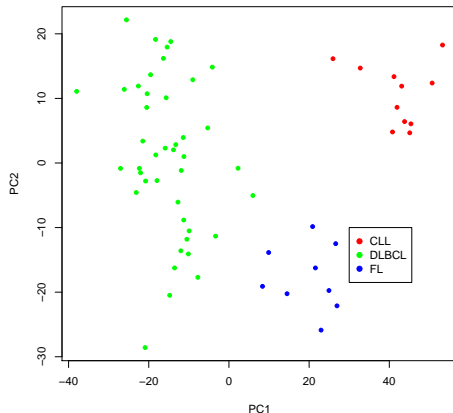
There are 3 cancer diagnoses: FL, CLL, DLBCL. Each corresponds to a type of malignant lymphoma.

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)
```

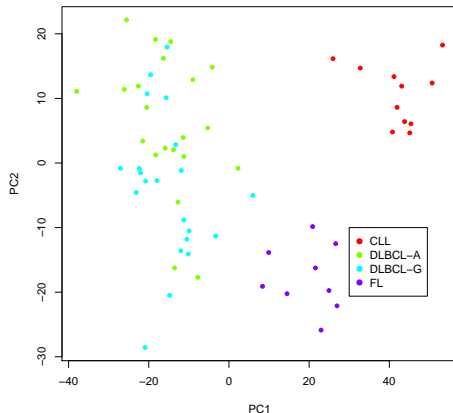
# PCA PLOT



Two clear groups for  
FL and CLL

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

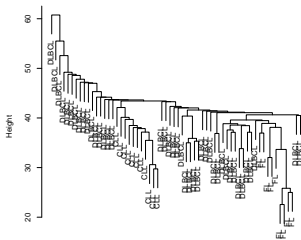
# PCA PLOT



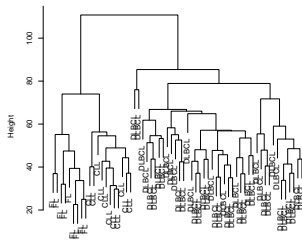
Here are the two sub-types identified by the researchers

Let's look at their results further.

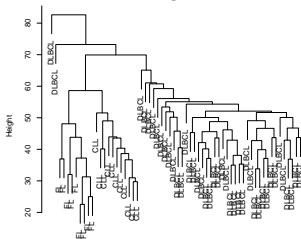
## FOUR HIERARCHICAL CLUSTER SOLUTIONS



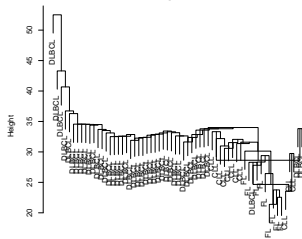
Single



Complete

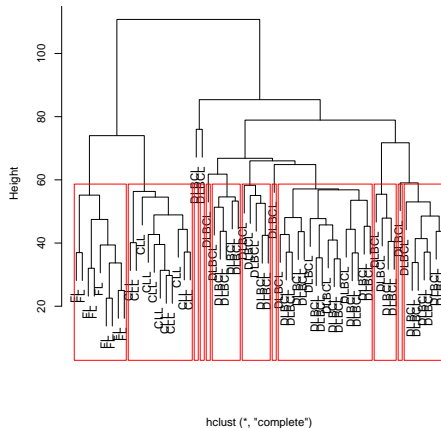


Average



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.

# CLUSTER METHODOLOGY

In the paper, the researchers used hierarchical clustering to find cancer subtypes

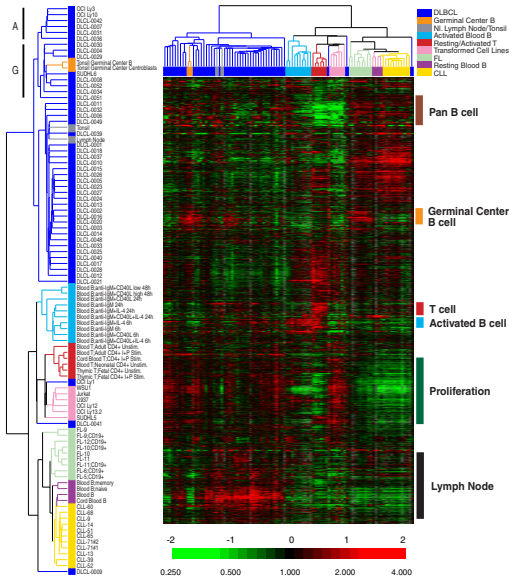
They used a variation on the methodology we have discussed

For a distance, they used correlation

They used a centroid linkage (oddly, they do not encounter any inversions...)



# FULL MICROARRAY



Each row is a cDNA clone (essentially a gene) and each column is a mRNA sample (essentially an observation).

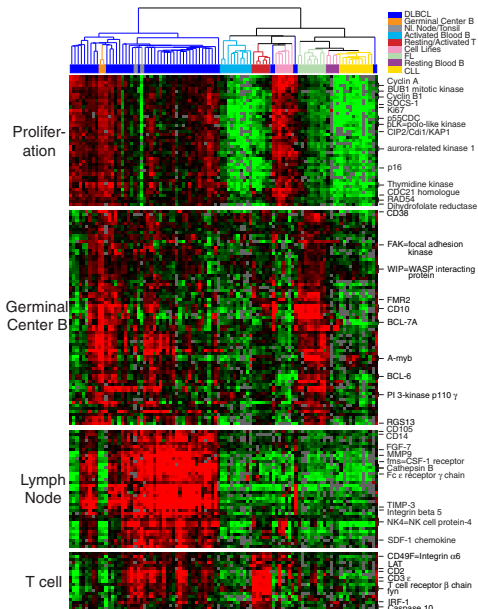
The green/red colors on the microarray indicate mRNA expression relative to a reference.

The color coding for labels was added *after* the clustering. The non-cancer related colors are experimental controls (other tissues).

Both FL and CLL are near resting B-cells (*indolence*)

DLBCL is highly expressed in the area where proliferation is controlled. However, there is substantial variation

# SUBSET OF MICROARRAY



The variable presence of T lymphocytes in DLBCL biopsies was readily discernible by a T-cell gene expression signature

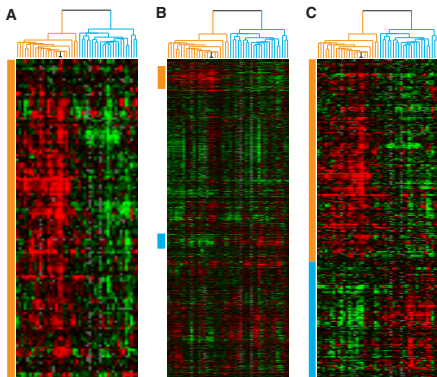
Again, notice the difference in expression for DLBCL in proliferation region

Likewise, FL and CLL are unexpressed in proliferation region.

FL is highly expressed in Germinal Center B while CLL is not.

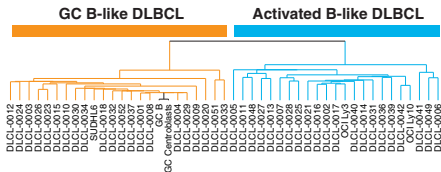
Some of the DLBCL is equally expressed in Lymph Node biology as normal lymph node/tonsil cells (grey).

# CREATING SUB-TYPES FOR DLBCL

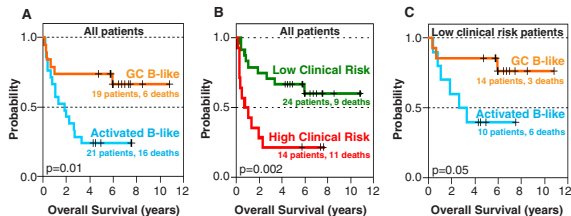


The vertical blue/orange bars indicate high expression levels associated with GC B-like and Activated B-like DLBCL, respectively.

- A The DLBCL type clustered into two groups based only on genes in Germinal Center B region.
- B A subset of the genes (excluding Proliferation, T-cell, and Lymph-node) ordered by hierarchical clustering.
- C Only looking at the differentially expressed regions from B.



# COMPARING SUB-TYPES TO MORTALITY DATA



Survival curve of patients, grouped by...

**A** clustered sub-type

**B** International Prognostication Index (IPI)  
( $0 \leq \text{IPI} \leq 2$  vs.  $3 \leq \text{IPI} \leq 5$ ).

**C** Low clinical risk ( $0 \leq \text{IPI} \leq 2$ ), by clustered sub-type.

(Note: these are known as **Kaplan-Meier plots**)

# DATA ANALYSIS EXAMPLE: CONCLUSION

The researchers found convincing sub-types of distinct cancers in previously diagnoses

This result is quite interesting:

- Compelling evidence that differential response to cancer treatments is the result (on average) of insufficiently differentiated cancer types
- Calls into question fundamental definitions such as what is a disease?
- Provides an alternative method to generate clinical diagnosis categories

# DATA ANALYSIS EXAMPLE: CONCLUSION

The researchers found convincing sub-types of distinct cancers in previously diagnoses

This result is quite interesting:

- Compelling evidence that differential response to cancer treatments is the result (on average) of insufficiently differentiated cancer types
- Calls into question fundamental definitions such as what is a disease?
- Provides an alternative method to generate clinical diagnosis categories

# DATA ANALYSIS EXAMPLE: CONCLUSION

The researchers found convincing sub-types of distinct cancers in previously diagnoses

This result is quite interesting:

- Compelling evidence that differential response to cancer treatments is the result (on average) of insufficiently differentiated cancer types
- Calls into question fundamental definitions such as what is a disease?
- Provides an alternative method to generate clinical diagnosis categories

# DATA ANALYSIS EXAMPLE: CONCLUSION

The researchers found convincing sub-types of distinct cancers in previously diagnoses

This result is quite interesting:

- Compelling evidence that differential response to cancer treatments is the result (on average) of insufficiently differentiated cancer types
- Calls into question fundamental definitions such as what is a disease?
- Provides an alternative method to generate clinical diagnosis categories

Conclusion: Qualitative *symptoms-based* categorization of disease is inadequate for diagnosis and treatment



# WRAP-UP

Just like in  $K$ -means, we can use CH-index to pick the number of clusters:

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

There are many, many other clustering approaches to take: KNN, Factor analysis, level-sets of density estimators, mixture of normals, ...