Clustering and Classification Methods: A Brief Introduction

EPSY 905: Multivariate Analysis

Spring 2016

Lecture #14 (and last!) –

May 4, 2016

Today's Lecture

- Classification methods: Useful for when you already know which groups exist but you don't know which group to assign a new observations

- Clustering methods: Useful for when you don't know how many groups exist

- As both methods rely upon distances (more or less) we will start with a definition of distances
  - We'll also get a bonus slide or two about multidimensional scaling—another method that uses distances (but doesn't cluster or classify directly)

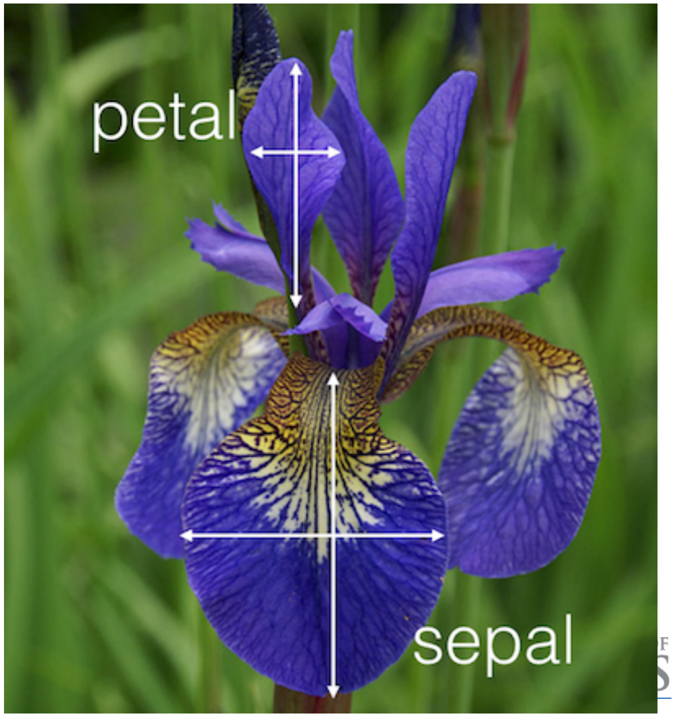
Today's Data

- We will make use of the classic Fisher's Iris Data to demonstrate clustering and classification methods

- 150 flowers; 50 from three species (Setosa, Versicolor, Virginica)

- Four measurements from each flower:
  - Sepal width
  - Sepal length
  - Petal width
  - Petal length

- These data are built into R already!



DISTANCE METRICS

Measures of Distance

- Care must be taken with choosing the metric by which similarity is quantified

- Important considerations include:
  - The nature of the variables (e.g., discrete, continuous, binary)
  - Scales of measurement (nominal, ordinal, interval, or ratio)
  - The nature of the matter under study

Euclidean Distance

- Euclidean distance is a frequent choice of a distance metric:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + \left(x_p - y_p\right)^2}$$
$$= \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T (\boldsymbol{x} - \boldsymbol{y})}$$

- Distances can be between anything you wish to cluster:
  - Observations (distance across all variables)
  - Variables (distance across all observations)

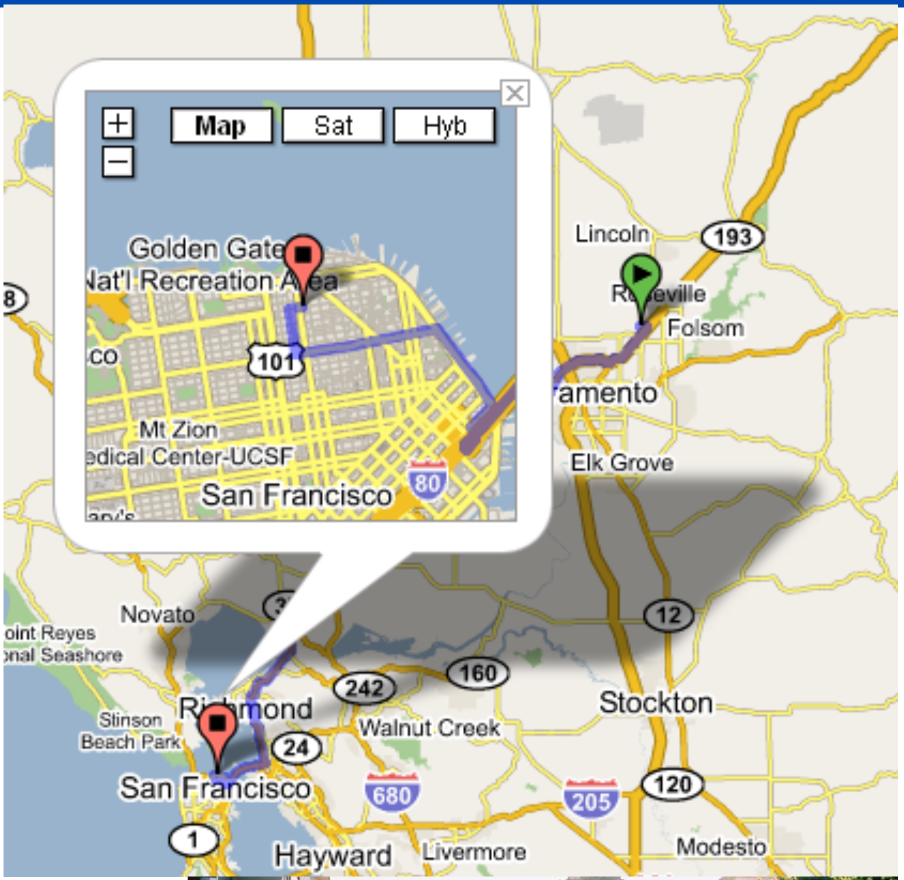
- The Euclidean distance is a frequent choice because it represents an understandable metric
  - It may not be the best choice always

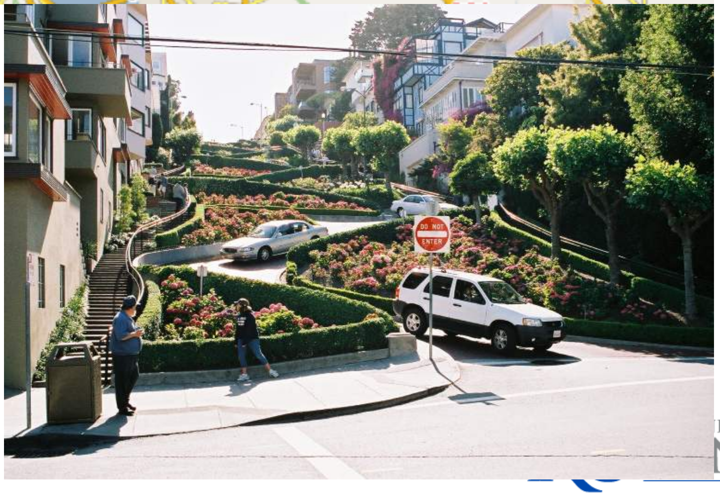
KU THE UNIVERSITY OF KANSAS

Euclidean Distance?

- Imagine I wanted to know how many miles it was from my old house in Sacramento to Lombard Street in San Francisco…

- Knowing how far it was on a straight line would not do me too much good, particularly with the number of one-way streets that exist in San Francisco





Other Distance Metrics

- Other popular distance metrics include the Minkowski metric:

$$d(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{p} \left| x_i - y_i \right|^m \right]^{1/m}$$

- The key to this metric is the choice of *m*:
  - If *m* = 2, this provides the Euclidean distance
  - If *m* = 1, this provides the "city-block" distance

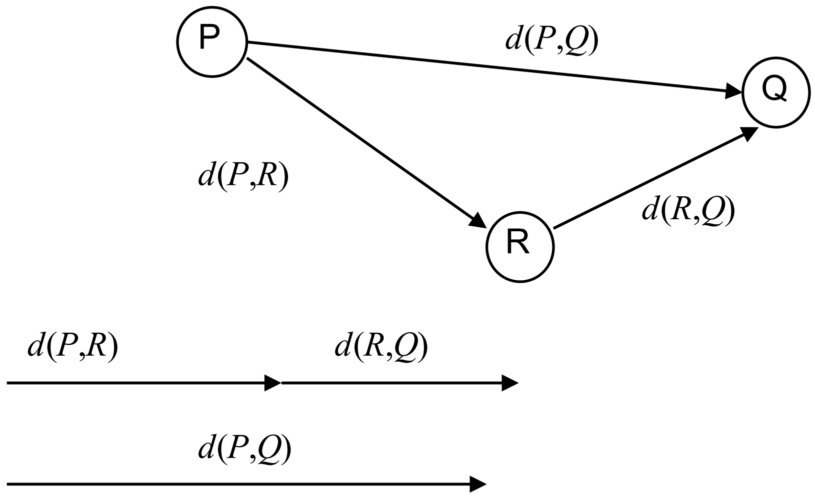
Preferred Distance Properties

- It is often desirable to use a distance metric that meets the following properties:
  - $d(P,Q) = d(Q,P)$
  - $d(P,Q) > 0$ if $P \neq Q$
  - $d(P,Q) = 0$ if $P = Q$
  - $d(P,Q) \leq d(P,R) + d(R,Q)$

- The Euclidean and Minkowski metrics satisfy these properties

Triangle Inequality

- The fourth property is called the triangle inequality, which often gets violated by non-routine measures of distance
- This inequality can be shown by the following triangle (with lines representing Euclidean distances):



Binary Variables

- In the case of binary-valued variables (variables that have a 0/1 coding), many other distance metrics may be defined
- The Euclidean distance provides a count of the number of mismatched observations:
- Here, $d(i,k) = 2$
- This is sometimes called the Hamming Distance

	Variables				
	1	2	3	4	5
Item i	1	0	0	1	1
Item k	1	1	0	1	0

Other Binary Distance Measures

- There are a number of other ways to define the distance between a set of binary variables

- Most of these measures reflect the varied importance placed on differing cells in a 2 x 2 table

General Distance Measure Properties

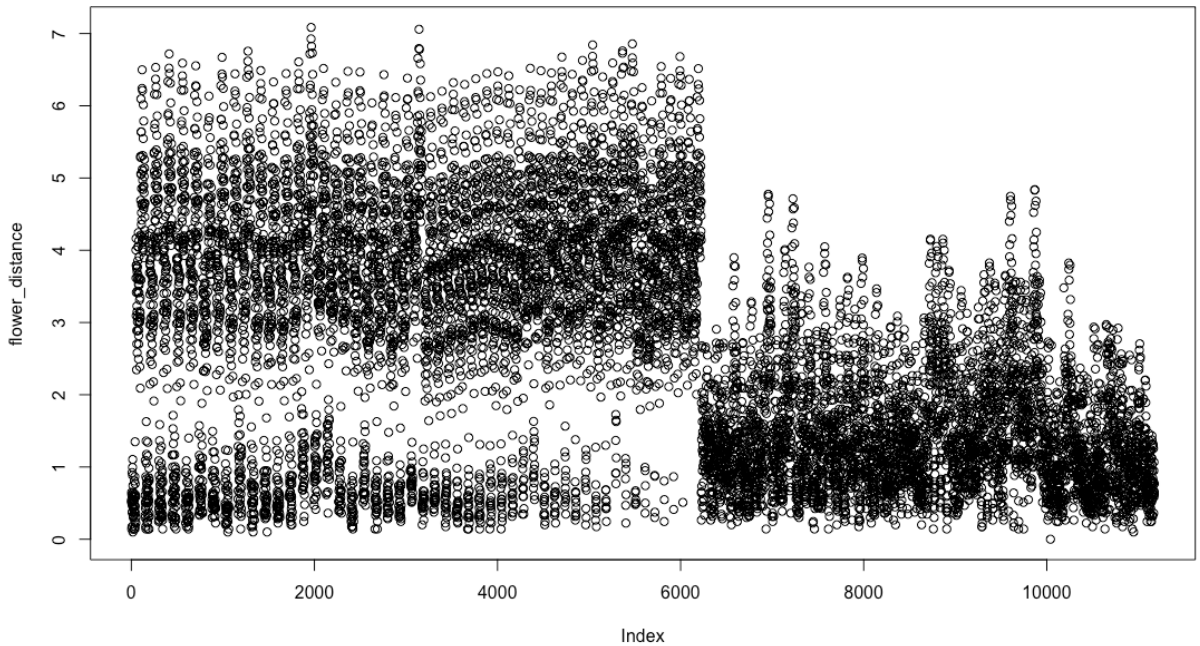
- Use of measures of distance that are monotonic in their ordering of object distances will provide identical results from clustering heuristics

- Many times this will only be an issue if the distance measure is for binary variables *or* the distance measure does not satisfy the triangle inequality

Distances in R

```r
#a distance between variables in the data set:
variable_distance = dist(t(data02), method = "euclidean")
variable_distance

flower_distance = dist(data02, method = "euclidean")
#quick graph of all the distances between each flower (index == which distance number)
plot(flower_distance)
```

```
> distance
             Sepal.Length Sepal.Width Petal.Length
Sepal.Width      36.15785                         
Petal.Length     28.96619    25.77809             
Petal.Width      57.18304    25.86407     33.86473
```



CLASSICAL CLUSTERING METHODS (USING DISTANCES)

Clustering Definitions

- Clustering objects (or observations) can provide detail regarding the nature and structure of data

- Clustering is distinct from classification in terminology
  - Classification pertains to a *known* number of groups, with the objective being to assign new observations to these groups

- Classical methods of cluster analysis is a technique where no assumptions are made concerning the number of groups or the group structure
  - Mixture models do make assumptions about the data

- Clustering algorithms make use of measures of similarity (or alternatively, dissimilarity) to define and group variables or observations

- Clustering presents a host of technical problems

- For a reasonable sized data set with *n* objects (either variables or individuals), the number of ways of grouping *n* objects into *k* groups is:

$$\left(\frac{1}{k!}\right)\sum_{j=0}^{k} -1^{j-k}\binom{k}{j} \; j^{n}$$

- For example, there are over **four trillion** ways that 25 objects can be clustered into 4 groups – which solution is best?

THE UNIVERSITY OF KANSAS

Numerical Problems

- In theory, one way to find the best solution is to try each possible grouping of all of the objects – an optimization process called integer programming
- It is difficult, if not impossible, to do such a method given the state of today's computers (although computers are catching up with such problems)
- Rather than using such brute-force type methods, a set of heuristics have been developed to allow for fast clustering of objects in to groups
- Such methods are called heuristics because they do not guarantee that the solution will be optimal (best), only that the solution will be better than most

Clustering Heuristic Inputs

- The inputs into clustering heuristics are in the form of measures of similarities or dissimilarities

- The result of the heuristic depends in large part on the measure of similarity/dissimilarity used by the procedure

Clustering Variables vs. Clustering Observations

- When variables are to be clustered, oft used measures of similarity include correlation coefficients (or similar measures for non-continuous variables)

- When observations are to be clustered, distance metrics are often used

Hierarchical Clustering Methods

- Because of the large size of possible clustering solutions, searching through all combinations is not feasible

- Hierarchical clustering techniques proceed by taking a set of objects and grouping a set at a time

- Two types of hierarchical clustering methods exist:
  - Agglomerative hierarchical methods
  - Divisive hierarchical methods

Agglomerative Clustering Methods

- Agglomerative clustering methods start first with the individual objects
- Initially, each object is it's own cluster
- The ***most similar*** objects are then grouped together into a single cluster (with two objects)

We will find that what we mean by *similar* will change depending on the method

Agglomerative Clustering Methods

- The remaining steps involve merging the clusters according the similarity or dissimilarity of the objects within the cluster to those outside of the cluster

- The method concludes when all objects are part of a single cluster

KU THE UNIVERSITY OF KANSAS

Divisive Clustering Methods

- Divisive hierarchical methods works in the opposite direction – beginning with a single, *n*-object sized cluster
- The large cluster is then divided into two subgroups where the objects in opposing groups are relatively distant from each other
- The process continues similarly until there are as many clusters as there are objects

To Summarize

- So you can see that we have this idea of steps

  - At each step two clusters combine to form one (Agglomerative)

OR…

  - At each step a cluster is divided into two new clusters (Divisive)

Methods for Viewing Clusters

- As you could imagine, when we consider the methods for hierarchical clustering, there are a large number of clusters that are formed sequentially

- One of the most frequently used tools to view the clusters (and level at which they were formed) is the dendrogram

- A dendrogram is a graph that describes the differing hierarchical clusters, and the distance at which each is formed

Example Data Set #1

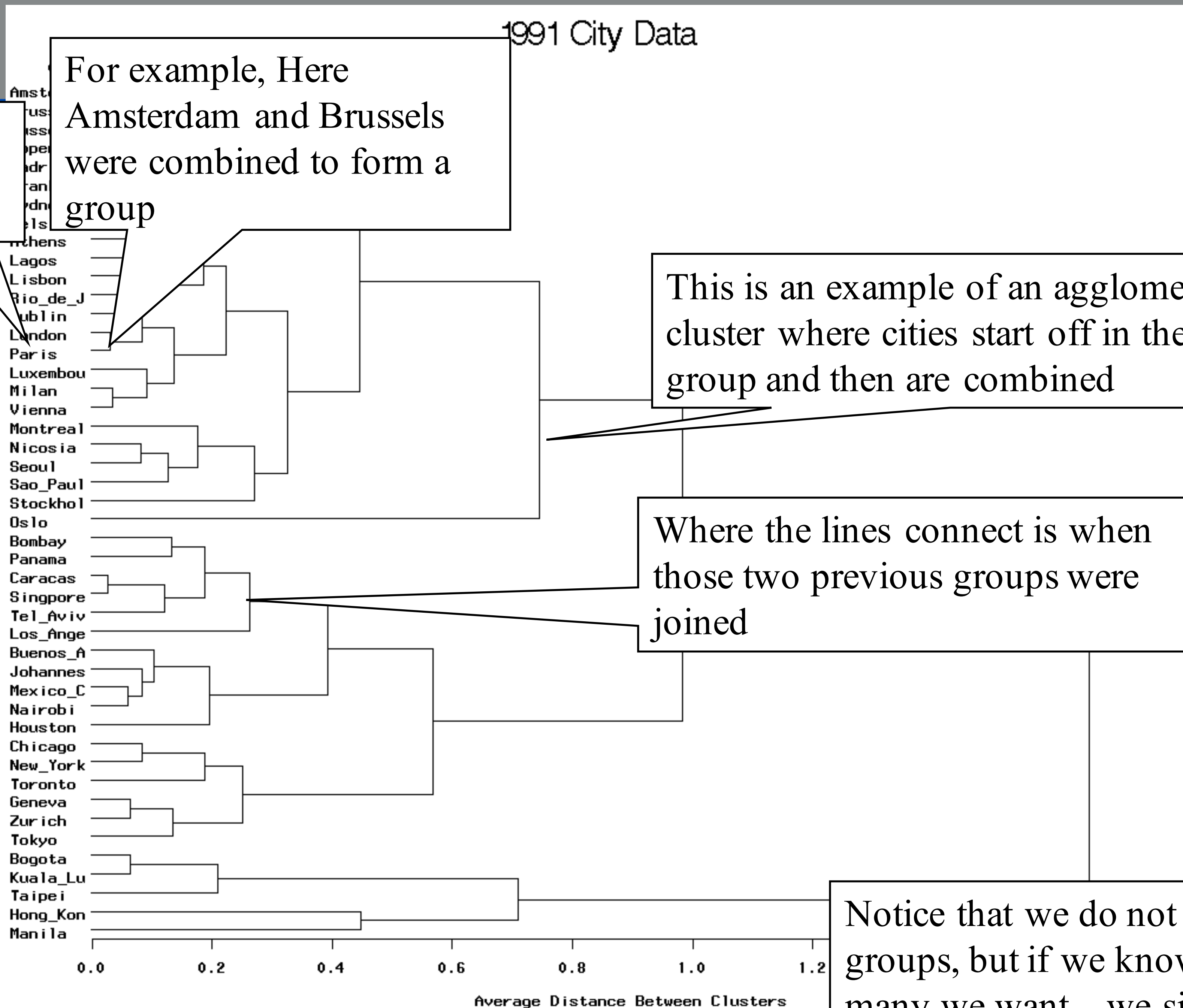
- To demonstrate several of the hierarchical clustering methods, an example data set is used

- Data come from a 1991 study by the economic research department of the union bank of Switzerland representing economic conditions of 48 cities around the world

- Three variables were collected:
  - Average working hours for 12 occupations
  - Price of 112 goods and services excluding rent
  - Index of net hourly earnings in 12 occupations

The Cities

For example, Here Amsterdam and Brussels were combined to form a group



This is an example of an agglomerate cluster where cities start off in there own group and then are combined

Where the lines connect is when those two previous groups were joined

Notice that we do not specify groups, but if we know how many we want…we simply go to the step where there are that many groups

Similarity?

- So, we mentioned that:
  - The ***most similar*** objects are then grouped together into a single cluster (with two objects)

- So the next question is how do we measure similarity between clusters
  - <u>More specifically, how do we redefine it when a cluster contains a combination of old clusters</u>

- We find that there are several ways to define similar and each way defines a new method of clustering

Agglomerative Methods

- Next we discuss several different way to complete Agglomerative hierarchical clustering:

  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid
  - Median
  - Ward Method

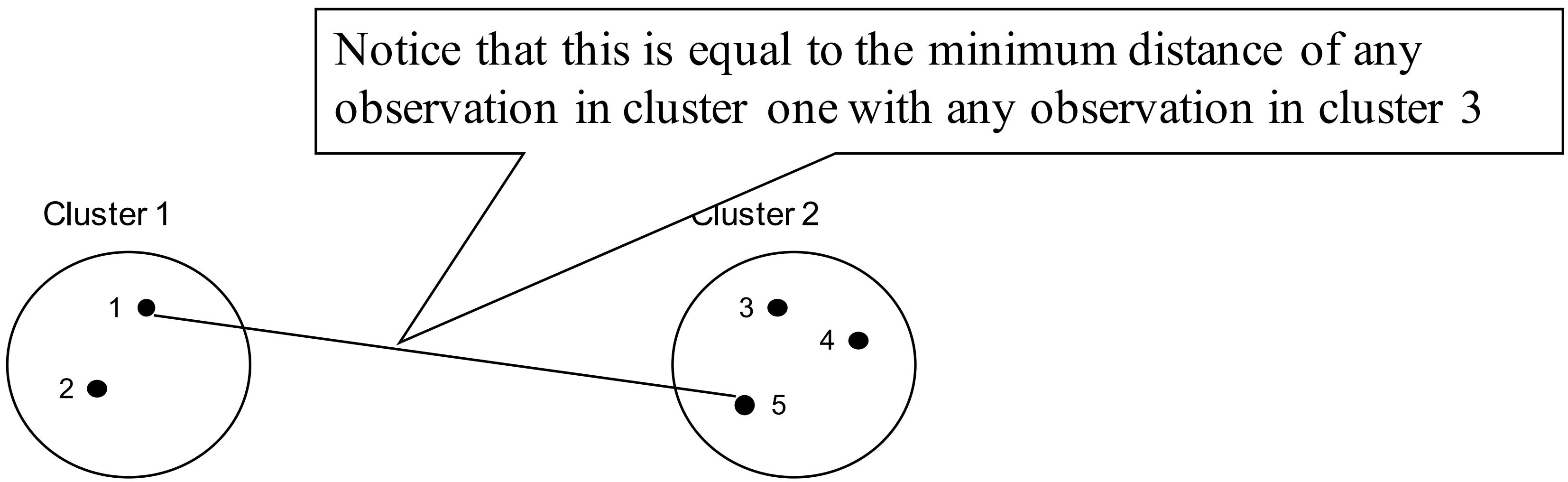
Example Distance Matrix

- The example will be based on the distance matrix below

	1	2	3	4	5
1	0	9	3	6	11
2	9	0	7	5	10
3	3	7	0	9	2
4	6	5	9	0	8
5	11	10	2	8	0

Single Linkage

- The single linkage method of clustering involves combining clusters by finding the "**nearest neighbor**" – the cluster closest to any given observation within the current cluster

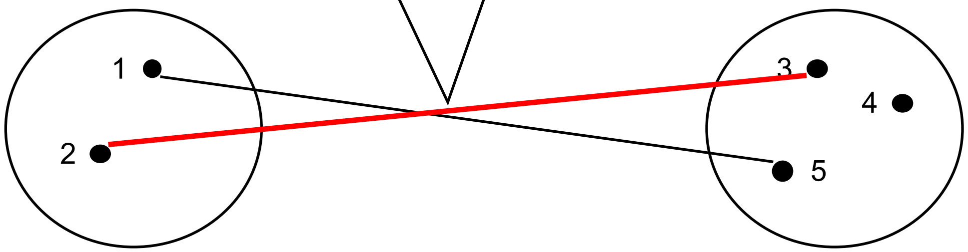


- So the distance between any two clusters is:

$$D(A,B) = \min\{d(\mathbf{y}_i, \mathbf{y}_j)\}$$

For all y_i in A and y_j in B

Notice any other distance is longer



So how would we do this using our distance matrix?

Single Linkage Example

- The first step in the process is to determine the two elements with the smallest distance, and combine them into a single cluster.
- Here, the two objects that are most similar are objects 3 and 5…we will now combine these into a new cluster, and compute the distance from that cluster to the remaining clusters (objects) via the single linkage rule.

	1	2	3	4	5
1	0	9	3	6	11
2	9	0	7	5	10
3	3	7	0	9	2
4	6	5	9	0	8
5	11	10	2	8	0

Single Linkage Example

- The shaded rows/columns are the portions of the table with th ob
- Th clu objects is given below.

These 5 with distan 1 is eq theses

These are the distances of 3 and 5 with 2. Our rule says that the distance of our new cluster with 2 is equal to the minimum of theses two values…7

	1	2	3	4	5
	0	9	3	6	11
	9	0	7	5	10
3	3	7	0	9	2
4	6	5	9	0	8
5	11	10	2	8	0

$d_{(35),1} = \min\{d_{31},d_{51}\} = \min\{3,11\} = 3$

$d_{(35),2} = \min\{d_{32},d_{52}\} = \min\{7,10\} = 7$

$d_{(35),4} = \min\{d_{34},d_{54}\} = \min\{9,8\} = 8$

In equation form our new distances are

Single Linkage Example

- Using the distance values, we now consolidate our table so that (35) is now a single row/column
- The distance from the (35) cluster to the remaining objects is given below:

$d_{(35)1} = \min\{d_{31}, d_{51}\} = \min\{3,11\} = 3$
$d_{(35)2} = \min\{d_{32}, d_{52}\} = \min\{7,10\} = 7$
$d_{(35)4} = \min\{d_{34}, d_{54}\} = \min\{9,8\} = 8$

	(35)	1	2	4
(35)	0			
1	3	0		
2	7	9	0	
4	8	6	5	0

Single Linkage Example

- We now repeat the process, by finding the smallest distance between within the set of remaining clusters
- The smallest distance is between object 1 and cluster (35)
- Therefore, object 1 joins cluster (35), creating cluster (135)

	(35)	1	2	4
(35)	0	3	7	8
1	3	0	9	6
2	7	9	0	5
4	8	6	5	0

The distance from cluster (135) to the other clusters is then computed:

$$d(135)2 = \min\{d(35)2, d12\} = \min\{7,9\} = 7$$
$$d(135)4 = \min\{d(35)4, d14\} = \min\{8,6\} = 6$$

Single Linkage Example

- Using the distance values, we now consolidate our table so that (135) is now a single row/column
- The distance from the (135) cluster to the remaining objects is given below:

	(135)	2	4
(135)	0		
2	7	0	
4	6	5	0

$$d(135)2 = \min\{d(35)2, d12\} = \min\{7,9\}$$
$$= 7$$
$$d(135)4 = \min\{d(35)4, d14\} = \min\{8,6\}$$
$$= 6$$

Single Linkage Example

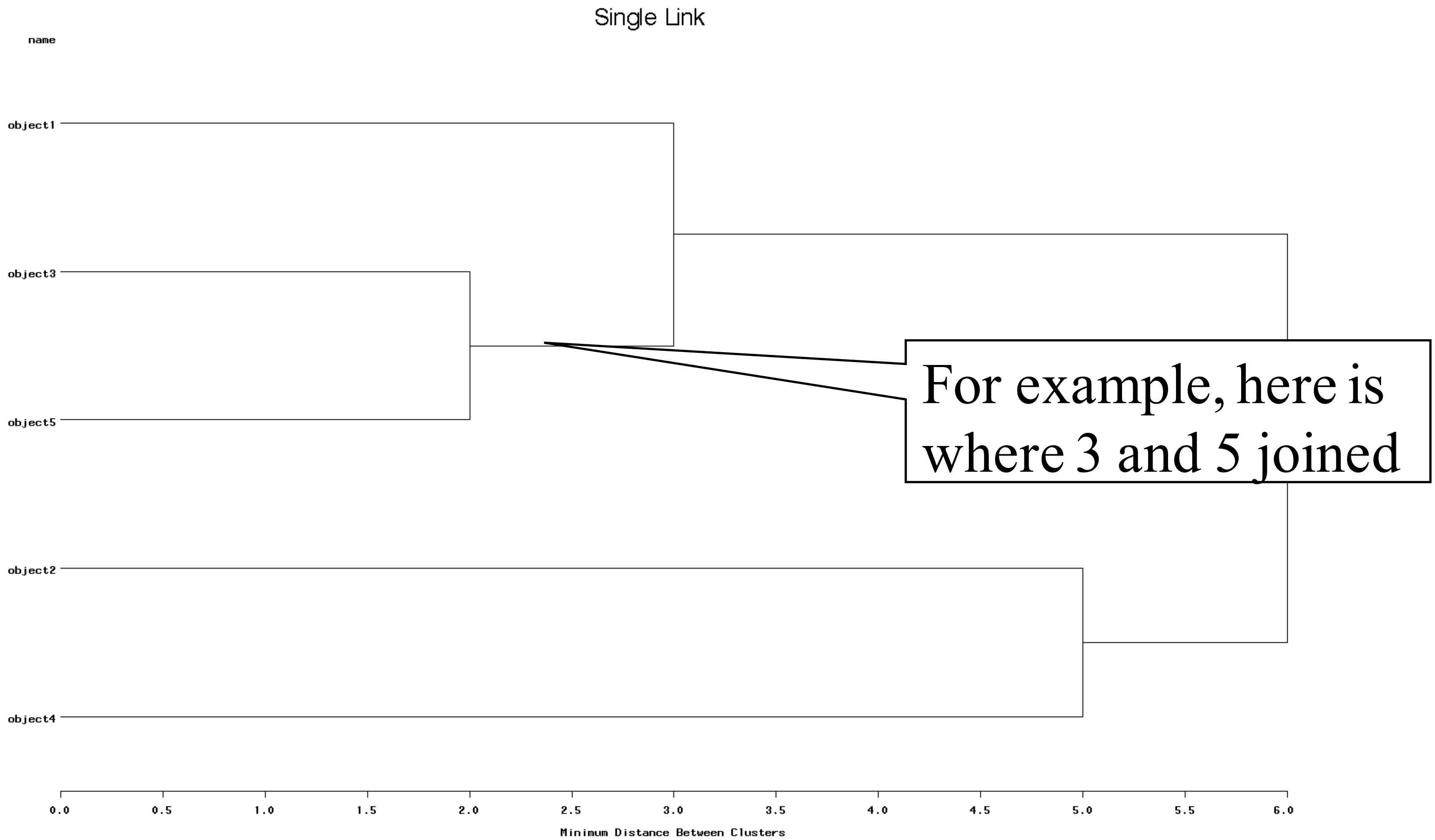
- We now repeat the process, by finding the smallest distance between within the set of remaining clusters
- The smallest distance is between object 2 and object 4
- These two objects will be joined to from cluster (24)
- The distance from (24) to (135) is then computed

$d_{(135)(24)} = \min\{d_{(135)2}, d_{(135)4}\} = \min\{7,6\} = 6$

- The final cluster is formed (12345) with a distance of 6

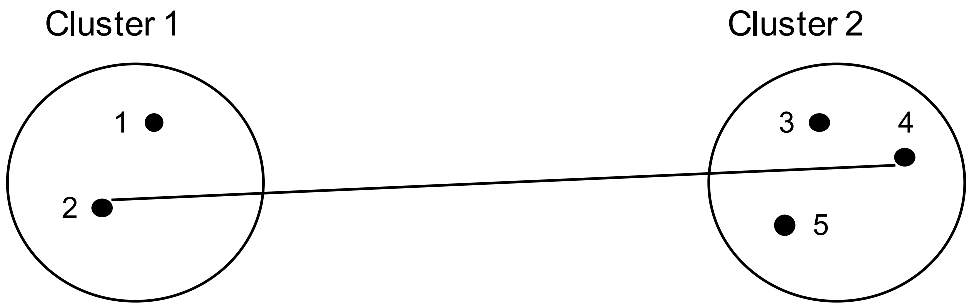
	(135)	2	4
(135)	0		
2	7	0	
4	6	5	0

The Dendogram



Complete Linkage

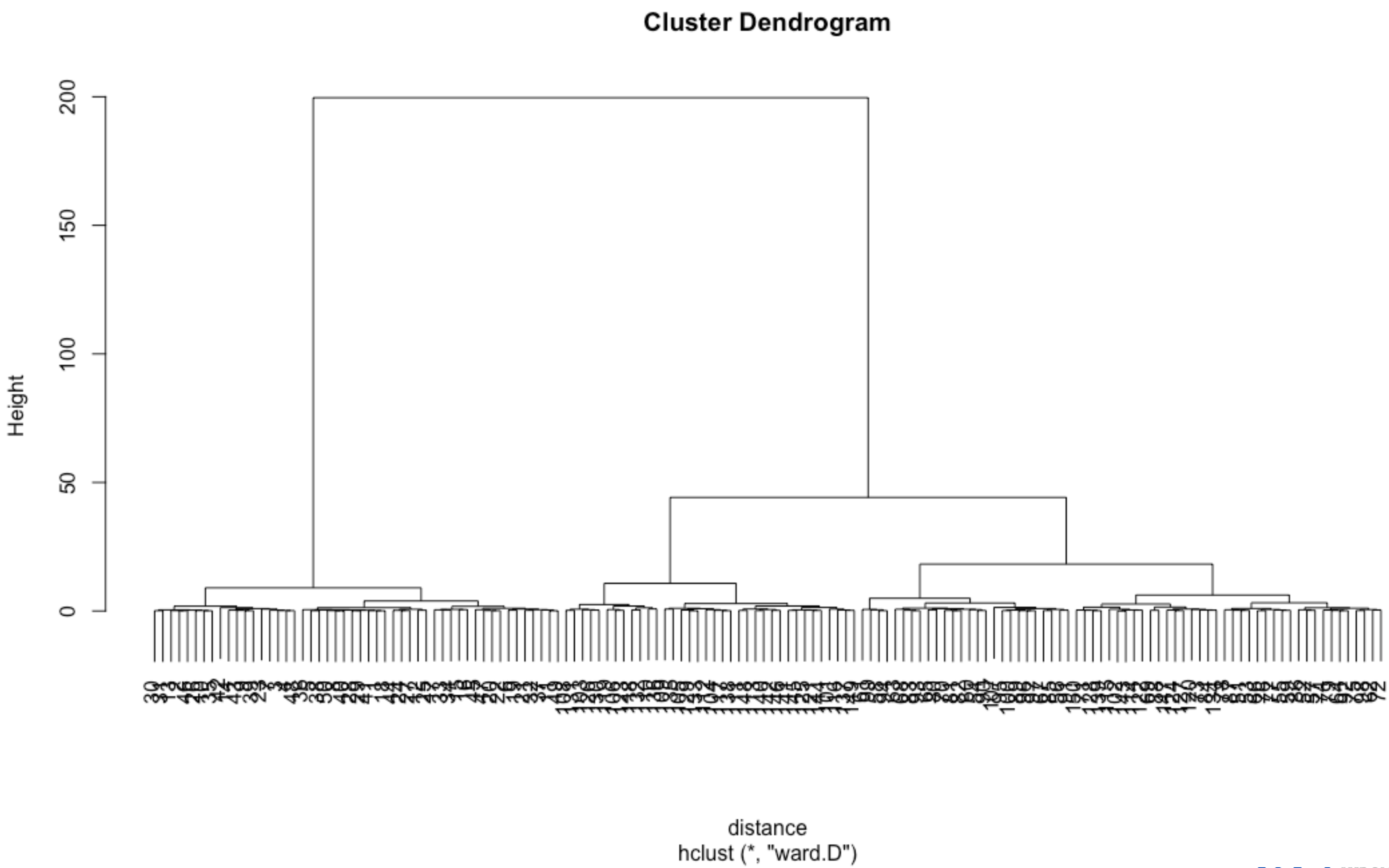
- The complete linkage method of clustering involves combining clusters by finding the "farthest neighbor" – the cluster farthest to any given observation within the current cluster
- This ensures that all objects in a cluster are within some maximum distance of each other



Clustering in R

```
#Classical method for cluster analysis: Agglomerative Hierarchical clustering ----------------------
#first, create distance matrix between flowers:
distance = dist(data02, method = "euclidean")

#second: conduct clustering
hierclust = hclust(distance, method = "ward.D")
plot(hierclust)

#cut tree into 3 clusters
groups = cutree(hierclust, k=3)
plot(groups)
```

Dendrogram of R Example with Flowers



Other Classical Clustering Methods

- K-means clustering will partition data into more than one group…but you have to specify how many groups
- The method uses the Mahalanobis distance of each observation to the group mean and iteratively switches group membership until no one switches
- R has a built-in K-means method

FINITE MIXTURE MODELS

Finite Mixture Models: Clustering with Likelihoods

- Finite mixture models are models that attempt to determine the number of clusters (called classes) in a set of data
  - Finite: countably many classes
  - Mixture: distribution of the data comes from a mixture of observations from different classes
- FMMs use likelihoods to determine class membership
  - A likelihood (pdf) is a similarity (inverse distance)
  - Higher likelihood → observation is more like class average → more likely observation comes from class
  - Lower likelihood → observation is less like class average → less likely observation comes from class
- FMMs have very specific names:
  - Latent class analysis: FMM where data are binary and independent within class
  - Factor mixture model: FMM where data are continuous and have a factor structure that yields a within-class covariance matrix

FMM Marginal Likelihood Function

- The general FMM marginal likelihood function is:

$$f(\boldsymbol{x}_i) = \sum_{c=1}^{C} \eta_c f(\boldsymbol{x}_i | c)$$

- $f(\boldsymbol{x}_i | c)$ is within class likelihood function (pdf)
  - Each class has its own distribution
- η_c is the probability any observation comes from class c
  - The "mixing" proportion
- The sum is where the mixture gets its name: the marginal data likelihood is a blend (mixture) of each class' likelihood

FMMs in R: Gaussian (MVN) Mixtures with mclust

- R has several packages for mixtures: I will only show you one – Mclust

- We will attempt to determine the number of classes in our flower data using only the four measurements

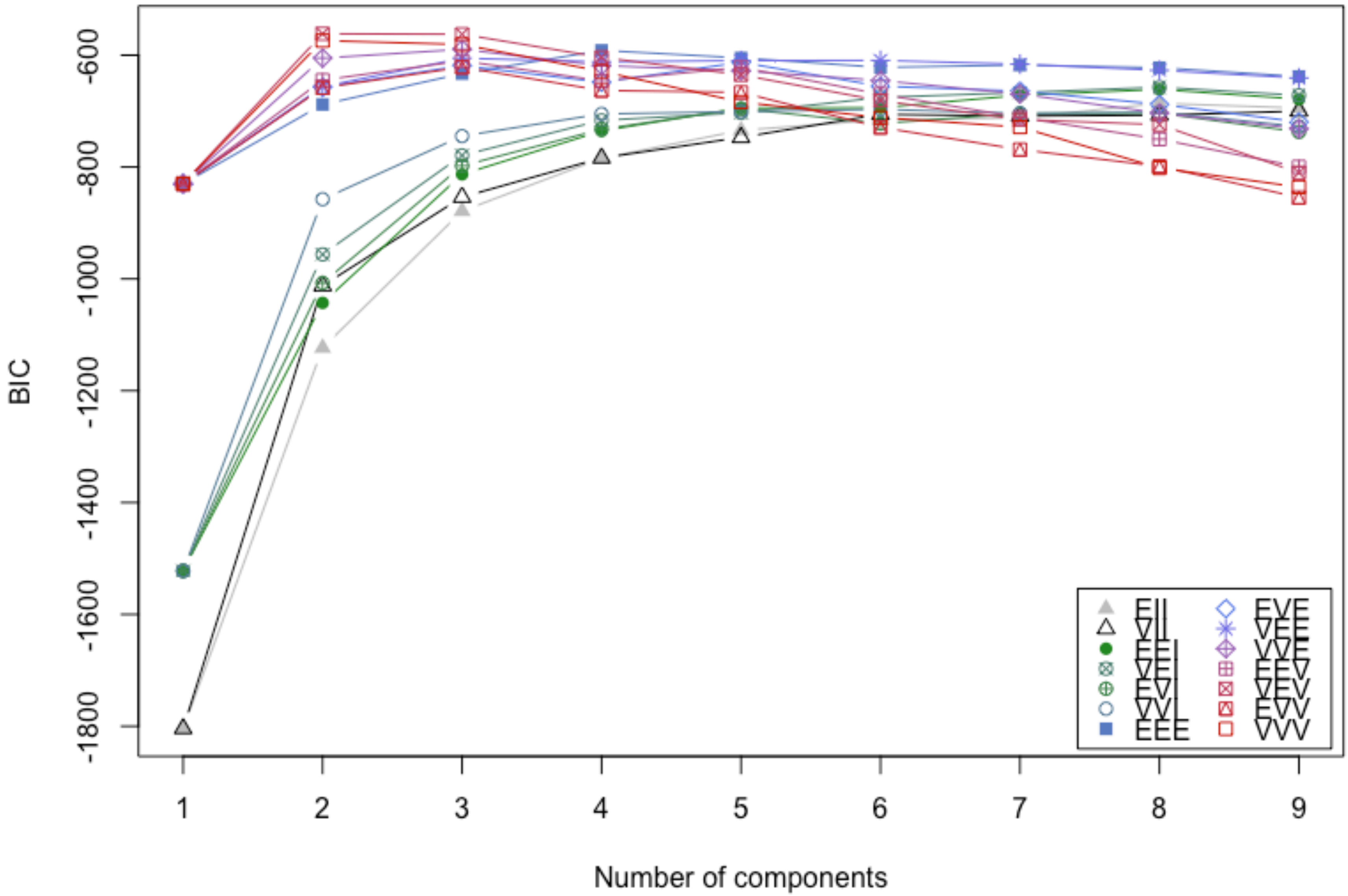
```
> library(mclust)
    __  ___________    __  _____________
   /  |/  / ____/ /   / / / / ___/_  __/
  / /|_/ / /   / /   / / / /\__ \ / /
 / /  / / /___/ /___/ /_/ /___/ // /
/_/  /_/\____/_____/\____//____//_/    version 5.2
Type 'citation("mclust")' for citing this R package in publications.
```

- First, we let mclust run a number of models to determine the "best fitting" (lowest BIC value)

```
#determine number of classes
BIC = mclustBIC(data02)
plot(BIC)
```

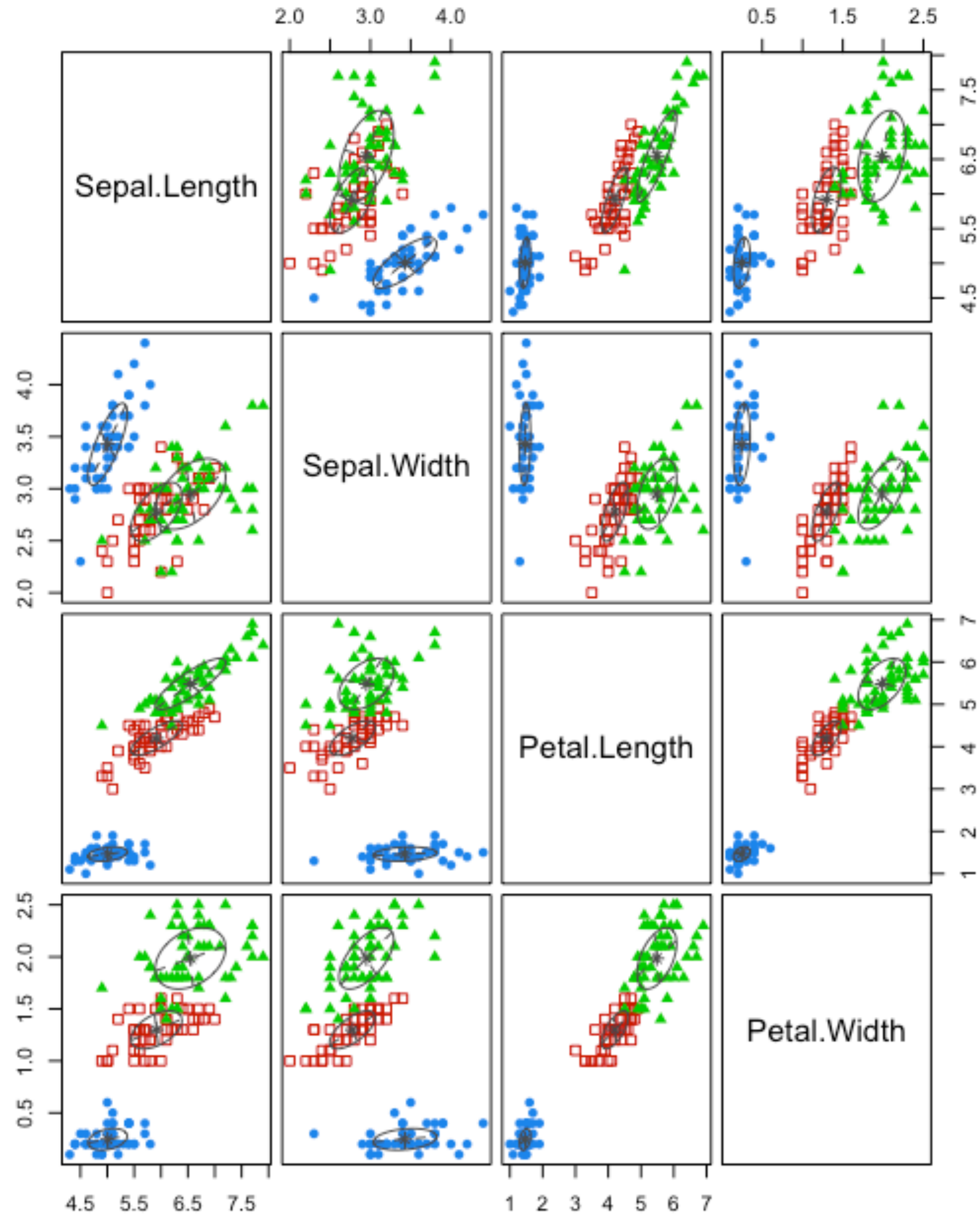
THE UNIVERSITY OF KANSAS

Plot of BICs from mclust



Example Output from One Solution

```
> summary(three_classes, parameters = TRUE)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 3 components:

 log.likelihood   n df       BIC       ICL
      -186.0736 150 38 -562.5514 -566.4577

Clustering table:
 1  2  3
50 45 55

Mixing probabilities:
        1         2         3
0.3333333 0.3002348 0.3664319

Means:
              [,1]     [,2]     [,3]
Sepal.Length 5.006 5.914717 6.546545
Sepal.Width  3.428 2.777559 2.949380
Petal.Length 1.462 4.203528 5.481568
Petal.Width  0.246 1.298712 1.985130

Variances:
[,,1]
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.13324824  0.10941877  0.019200078 0.011590068
Sepal.Width    0.10941877  0.15500101  0.012099295 0.010013052
Petal.Length   0.01920008  0.01209930  0.028278640 0.005820607
Petal.Width    0.01159007  0.01001305  0.005820607 0.010691679
[,,2]
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.22551946  0.07613511   0.14668558  0.04327572
Sepal.Width    0.07613511  0.08016383   0.07368295  0.03434262
Petal.Length   0.14668558  0.07368295   0.16588925  0.04941328
Petal.Width    0.04327572  0.03434262   0.04941328  0.03332507
[,,3]
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.42948927  0.10792653   0.33478260  0.06556364
Sepal.Width    0.10792653  0.11608122   0.08931829  0.06148507
Petal.Length   0.33478260  0.08931829   0.36479673  0.08741320
Petal.Width    0.06556364  0.06148507   0.08741320  0.08679214
> 
```



Problems with FMMs

- Exploratory (finding number of classes) uses of mixtures are often suspect as nearly anything can bring about additional (spurrious/false) classes

- For instance: if our data were not MVN within class, we would likely see more classes than we need

- Finding the right model is often difficult

WRAPPING UP

Wrapping Up

- This class only scratched the surface of what can be done to cluster or classify data

- Clustering: Determining the number of clusters in data
  - Agglomerative/Hierarchical clustering
  - K-Means
  - Finite mixture models

- Classification: Putting observations into known classes
  - Classical approach: Discriminant analysis (not shown today; lda() R function)
  - Modern approach: FMMs

- My class next semester (Diagnostic Testing) describes confirmatory uses of FMMs