FIT3152 Data analytics - S1 2026

Lecture 5

- Introduction to cluster analysis
- k-Means clustering
- Fuzzy clustering
- Hierarchical clustering, cluster analysis

Presenter: Dr Heshan Kumarage

Week-by-week outline

Week Starting	Seminar	Topic	App Ses	A1	A2	Q/P	A3	Due Date
2/3/2026	1	Introduction to Data Science, R, review of basic statistics	-					
9/3/2026	2	Data visualisation	S1					
16/3/2026	3	Data manipulation	S2					
23/3/2026	4	Regression modelling	S3					
30/3/2026	5	Clustering	S4					
6/4/2026	-	Mid-semester Break						
13/4/2026	6	Classification using decision trees	S5					17/4/2026
20/4/2026	7	Improving and evaluating classifiers. Naïve Bayes classification	S6					
27/4/2026	8	Ensemble methods, Artificial Neural Networks	S7					
4/5/2026	9	Network analysis	S8					
11/5/2026	10	Introduction to text analysis	S9					15/5/2026
18/5/2026	11	Text analysis applications	Quiz/Prac					22/5/2026
25/5/2026	12	Text Network Analysis, Review of the unit, Assignment 3	S10,11,12					
1/6/2026		SWOT VAC	-					
8/6/2026		EXAM PERIOD	-					12/6/2026

Assignment 1

MONASH University

Faculty of Information Technology

FIT3152 Data analytics – 2026: Assignment 1

Your task	• Analyse the country level predictors of confidence in **social organisations and how these change over time** using data from the World Values Survey. • This is an individual assignment.
Value	• This assignment is worth **25%** of your total marks for the unit. • It has **40** marks in total.
Suggested Length	• 8 – 10 A4 pages, approximately 2,000 words (for your report) + extra pages as appendix for your R script and report on how Generative AI used, if required. • Font size 11 or 12pt, single spacing.

Assignment 1

Due Date	**11.55pm Friday 17th April 2026**
Submission	• Submit a single PDF file **and** single video file on Moodle. • Note that submission of a video report is a hurdle requirement. • Use the naming convention: *FirstnameSecondnameID.{pdf, mp4, mov etc.}* • Turnitin will be used for similarity checking of all written submissions.
Generative AI Use	• In this assessment, you can use generative artificial intelligence (AI) in order to search for R functions and examples to perform tasks that you specify only. Any use of generative AI must be appropriately acknowledged (see Learn HQ).
Late Penalties	• 5% (**2** mark) deduction per calendar day for up to one week. • Submissions more than 7 calendar days after the due date will receive a mark of zero (0) and no assessment feedback will be provided.

Assignment 1

Questions

The World Values Survey (WVS) is an international research program that studies the social, political, economic, religious and cultural attitudes and values of people around the world. You can read more here: https://www.worldvaluessurvey.org/WVSContents.jsp.

For this assignment you will analyse data collected over Waves 1 - 7, from 1981 to 2022. The aim of this assignment is to understand country-level differences in participant responses and the predictors of confidence in social organisations, and how these responses and predictors of confidence have changed over time.

Social organisations include aspects of society such as religion, armed forces, the press, television, trade unions, police, the courts, government, banks, and international and environmental organisations etc. They are indicated in your data by column names having the prefix "C". Predictor variables (**attributes**) include personal information such as age and gender, happiness indicators, attitudes and values towards others, political and social views and participation.

Each student will be assigned a **different** subset of organisations and attributes to study. Your task is to analyse **all** the survey data assigned to you, with a **focus** on the country you have been allocated.

Assignment 1

1. Descriptive analysis. (5 Marks)

(a) Describe the data overall, including things such as dimension, data types, distribution of numerical responses, variety of non-numerical (text) responses, missing values, and anything else of interest or relevance.

Assignment 1

2. Focus country vs all other countries as a group (independent of time). (13 Marks)

For Question 2 ignore the effect of time. That is, do not separate your data by years or waves when answering the questions below.

(a) Identify your focus country from the accompanying list (**WVSFocusCountry.pdf**). How do participant responses for your focus country differ from the other countries in the survey (treating them as a group)?

(b) How well do participant responses (attributes) predict confidence in social organisations in your **focus country**? Which attributes seem to be the best predictors? Confidence in which social organisations can be more reliably predicted? Explain your reasoning.

(c) Repeat Question 2(b) for the **other countries** as a group. Which attributes are the strongest predictors? Confidence in which social organisations can be more reliably predicted? How do these results compare to those of your focus country?

Assignment 1

3. Focus country vs all other countries as a group (over time). (12 Marks)

For Question 3 study the effect of time by separating your data by years or waves when answering the questions below.

(a) How do participant responses for your **focus country** vary over time (using either years or successive waves)? Describe these changes over time and comment on whether they are significant or not. Perform the same analysis for the **other countries** (as a group) and compare the results with your focus country. What are the major differences between the two groups? Create a graphic enabling a reader to compare results (focus *vs* other countries) over time, for the most interesting results. Describe your reasoning for the design of the graphic.

(b) How does the ability of participant responses (attributes) to predict confidence in social organisations in your **focus country** change over time? Do the important attributes for predicting confidence change over time? Perform the same analysis for the **other countries** (as a group) and compare the results. What are the major differences between the two groups? Create a graphic enabling a reader to compare results (focus *vs* other countries) over time, for the strongest predictors. Describe your reasoning for the design of the graphic.

Assignment 1

4. Video Presentation: (Submission Hurdle and 4 Marks)

Record a short presentation using your smartphone, Zoom, or similar method. Your presentation should be approximately 5 minutes in length and summarise your main findings for Sections 1 – 3, as well as describing how you conducted your research, any assumptions made, and how you designed your graphics.

5 Overall considerations (6 Marks)

This includes: the quality and clarity of your reasoning and assumptions; the strength of support for your findings; the quality of your writing in general and communication of results;-the quality of your graphics throughout; the quality of your R coding.

Assignment 1

Data

The data for this assignment is a reduced version of the World Values Survey Waves 1 -7 data. The filename is "WVSExtract.csv". The data includes ordinal data coded on a numerical scale. For this assignment assume it is reasonable to treat these responses as numerical.

Create your individual data as follows:

```
rm(list = ls())
set.seed(12345678) # Your Student Number
VCData = read.csv("WVSExtract.csv")
VC = VCData[sample(1:nrow(VCData),100000, replace=FALSE),]
VC = VC[,c(1:3,sort(sample(4:50,25,replace=FALSE)),
sort(sample(51:65,8,replace=FALSE)))]
#write.csv(VC, "FIT3152A1Data_YourName.csv", row.names = FALSE)
```

You can save the “VC” file you created by uncommenting the last line above. You can then delete the WVSExtract.csv file you downloaded.

Locate your focus country using the accompanying document FocusCountryByID.pdf. A list matching country names with three letter code is in WVSCountryCodes.pdf.

Assignment 1

Data fields and brief descriptor

Most fields are on integer scales over varying range. The convention is that larger numbers generally indicate greater agreement with statement or frequency of occurrence. Some exceptions given below. Fields in bold indicate confidence in social organisations.

You can access more detail on each field in your data from the *WVS-7 Master Questionnaire 2017-2020 English.pdf*, linked from https://www.worldvaluessurvey.org/WVSDocumentationWV7.jsp. Use the question ID given in the **WVS Wave 7 Reference** in the table below.

Column Name	**Original Descriptor**	**WVS Wave 7 Reference**
Wave	Chronology of EVS-WVS waves	A_WAVE
Country	ISO 3166-1 alpha-3 country code	B_COUNTRY_ALPHA
Year	Year survey	A_YEAR
ILFam	Important in life: Family	Q1
ILFriends	Important in life: Friends	Q2
ILLeisure	Important in life: Leisure time	Q3
ILPolitics	Important in life: Politics	Q4
ILWork	Important in life: Work	Q5

Assignment 1

- Students who joined the unit late (and are not on the FocusCountryByID.pdf) need to email john.betts@monash.edu to be assigned a focus country.
- Data may contain missing/NA values. Check the survey documentation: WVS-7 Master Questionnaire 2017-2020 English.pdf
- It is likely many attributes will have low predictive power. The aim of the analysis is to find the “best” ones.

Clustering

In this lecture we'll cover:

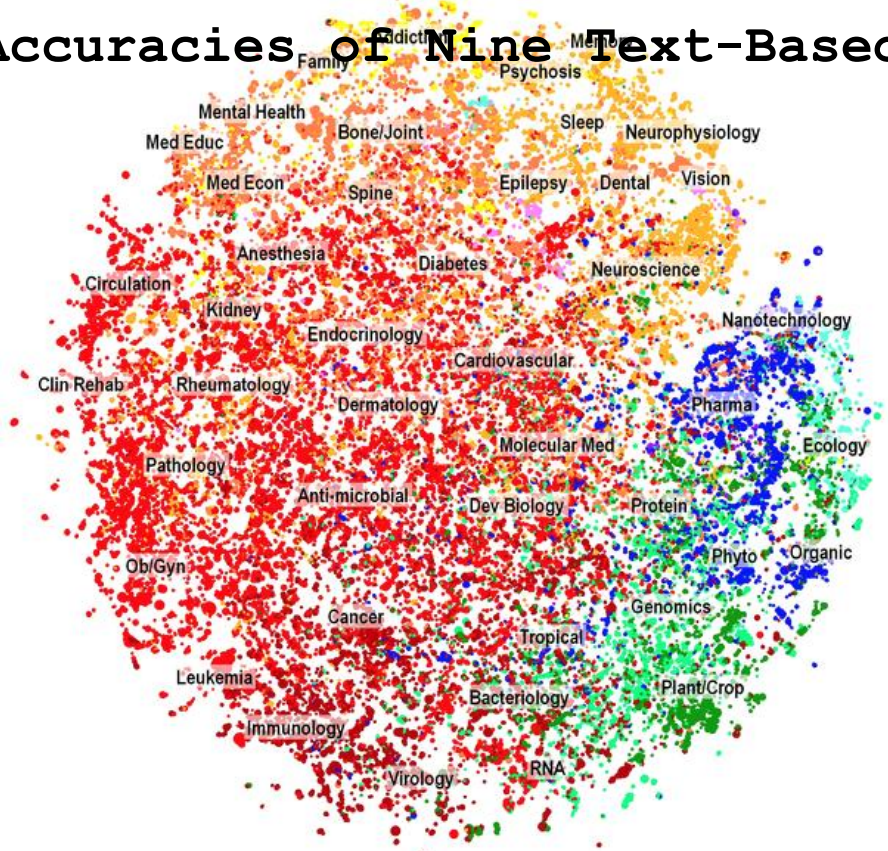
- Introduction to cluster analysis
- k-Means clustering in R
- Fuzzy clustering
- Hierarchical clustering
- Cluster analysis

Food groups

https://www.muralsyourway.com/p/food-groups-mural/

Document clustering

Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches

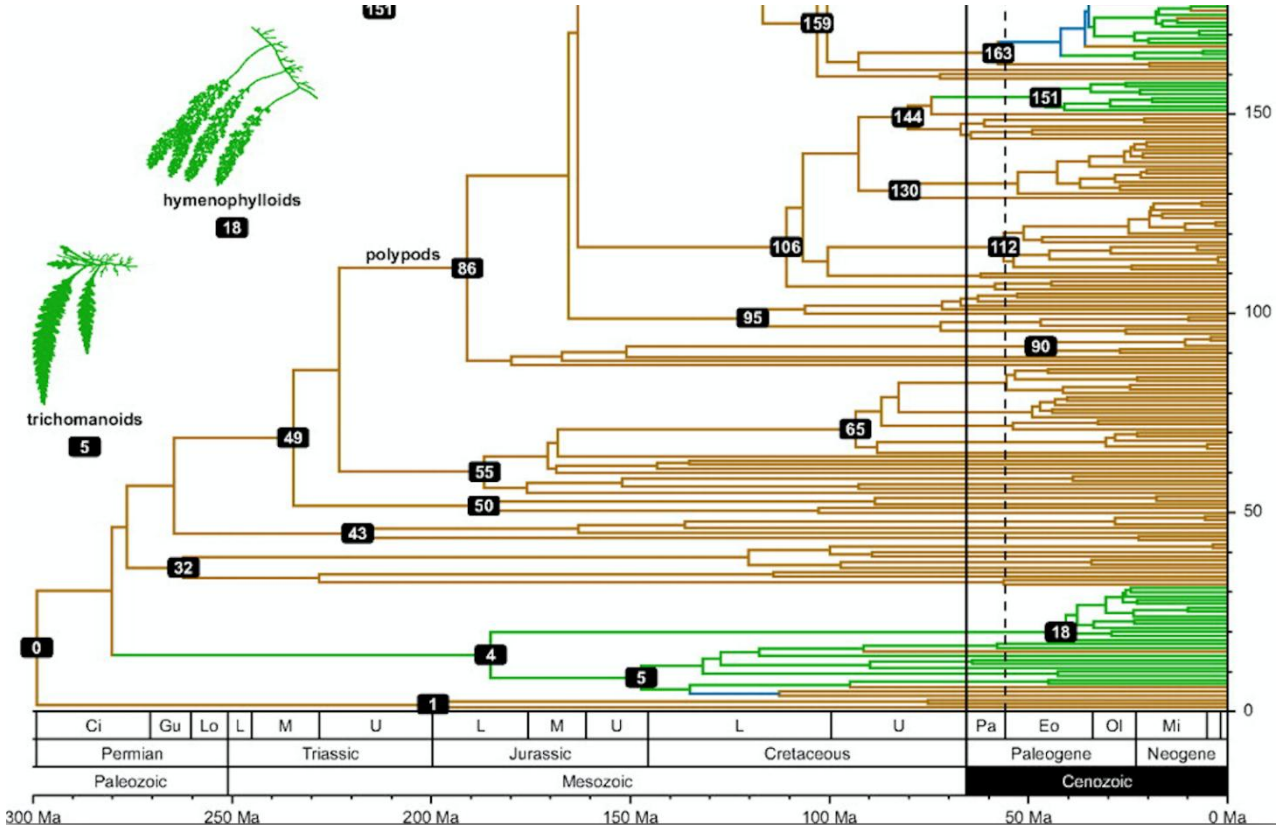


https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0018029

7 Australian political personas

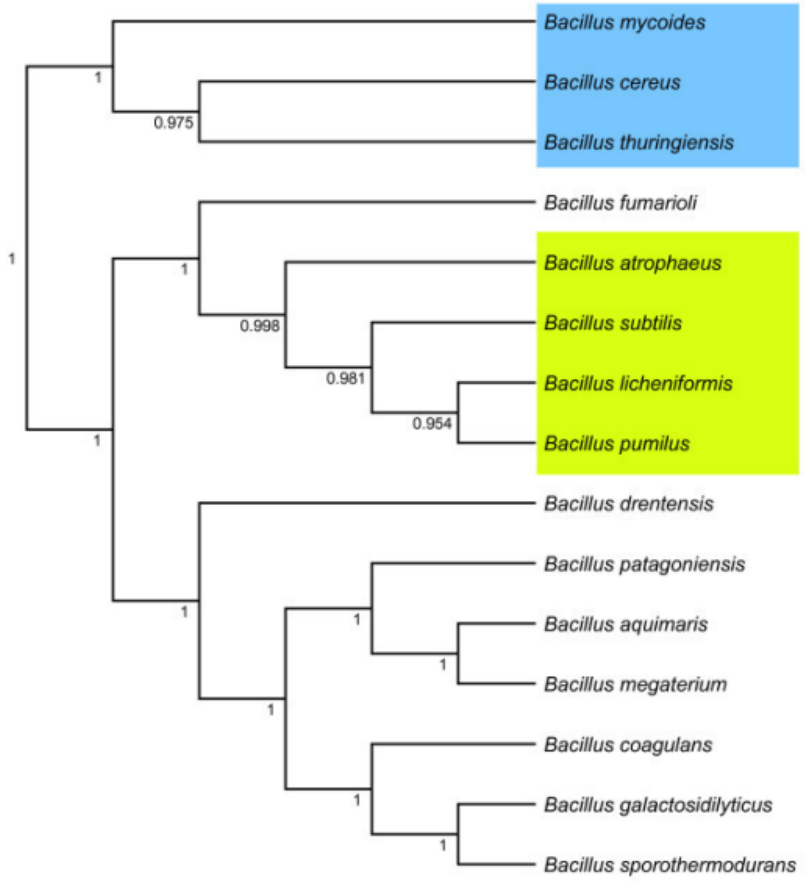
https://www.smh.com.au/

Phylogenetic tree, fern evolution



https://www.pnas.org/content/106/27/11200/F1.expansion.html

Phylogenetic tree, Bacillus species



https://openi.nlm.nih.gov/detailedresult.php?img=PMC2828439_1471-2105-11-69-1&req=4

COVID-19

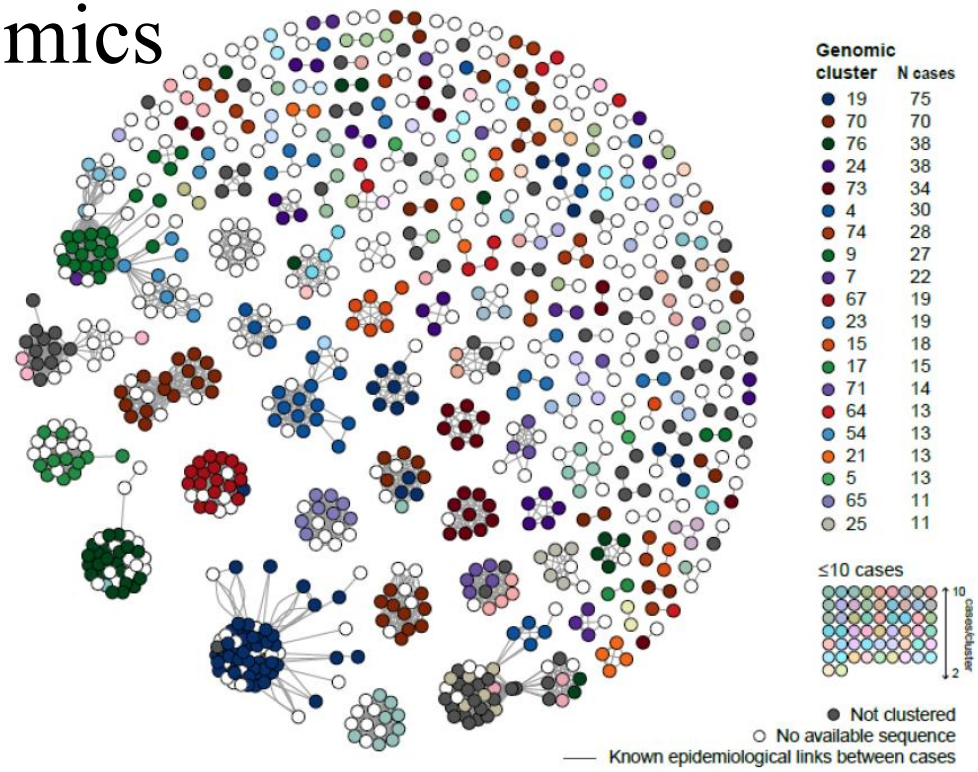
Tracking the COVID-19 pandemic in Australia using genomics

> **Sequenced samples from Australia were representative of the global diversity of SARS-CoV-2, ... In total, 76 distinct genomic clusters were identified; these included large clusters associated with social venues, healthcare facilities and cruise ships. Sequencing of sequential samples from 98 patients revealed minimal intra-patient SARS-CoV-2 genomic diversity.**

https://www.medrxiv.org/content/10.1101/2020.05.12.20099929v1

COVID-19

Tracking the COVID-19 pandemic in Australia using genomics



https://www.medrxiv.org/content/10.1101/2020.05.12.20099929v1

SARS-CoV-2 antigenic variants

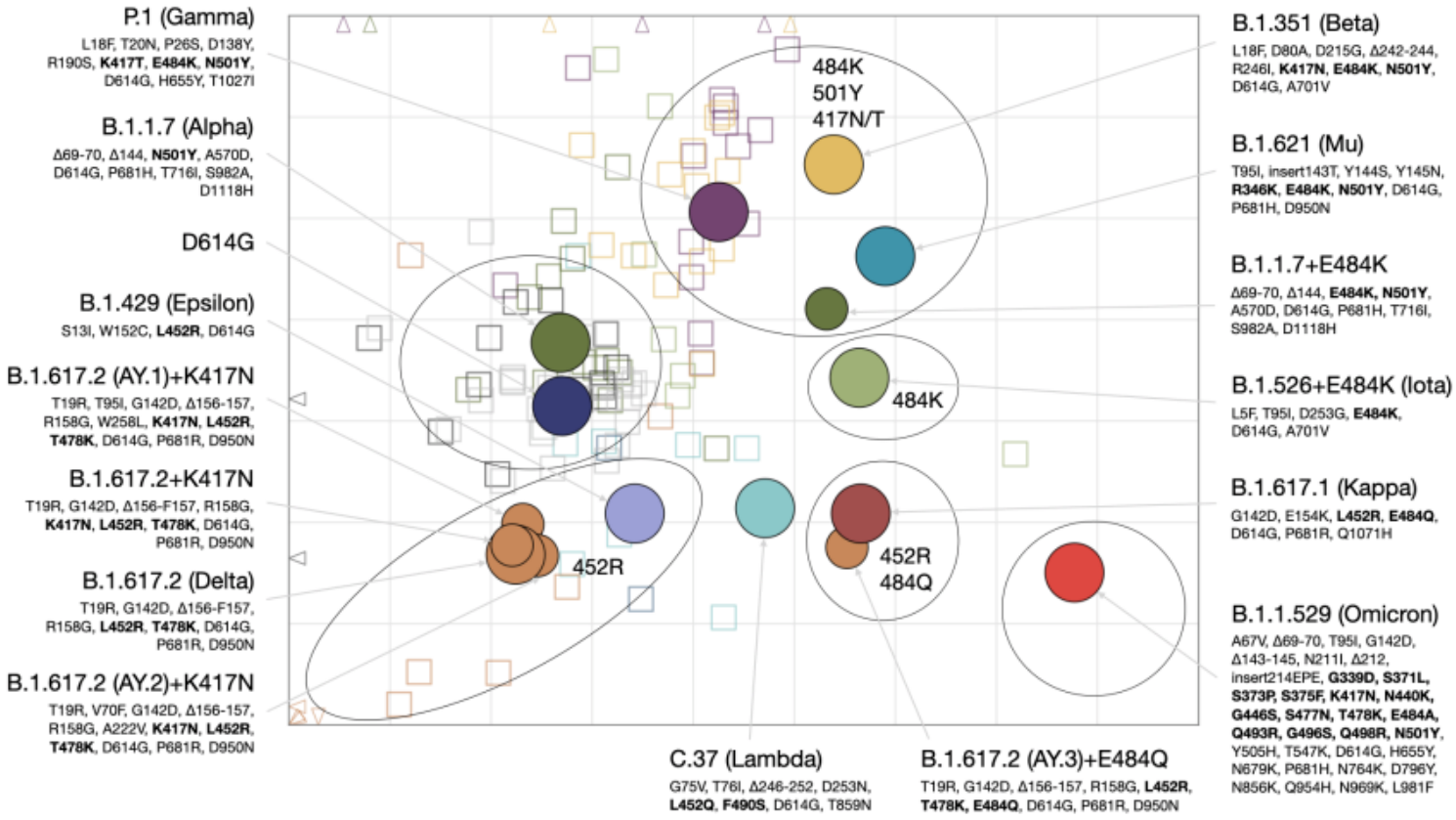


Fig. 2: Antigenic map of SARS-CoV-2 variants and selected substitutions. Variants are shown as circles, sera as squares.

https://www.biorxiv.org/content/10.1101/2022.01.28.477987v1.full.pdf

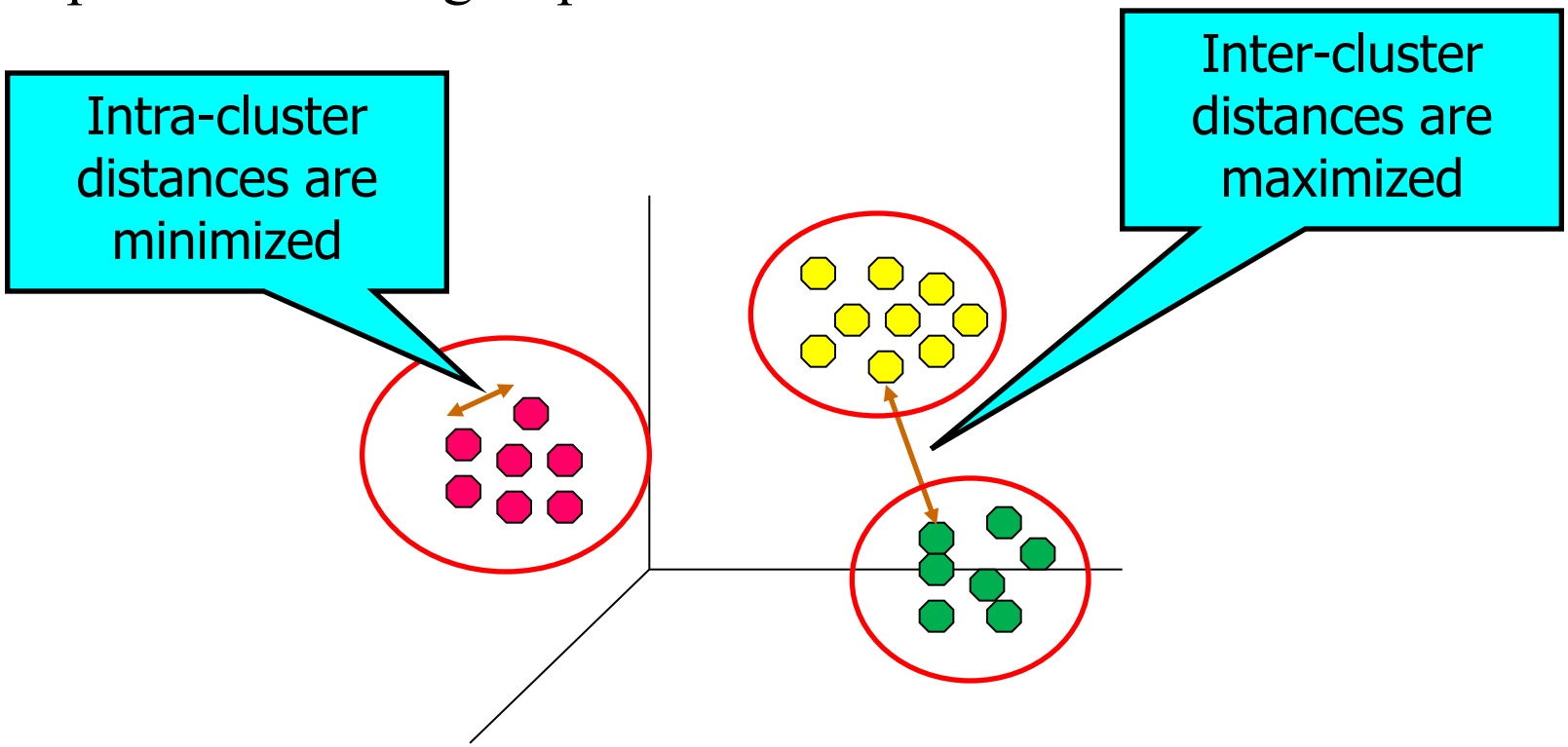
Supervised *vs* unsupervised learning

There are two main approaches to machine learning:

- Supervised learning algorithms:
  - > Algorithms are given labelled examples (target class) for the various types of data that need to be learned.
  - > For example: regression.
- Unsupervised learning algorithms:
  - > Data is unlabeled (has no predefined classes), and the learning algorithms attempt to find patterns within the data to put into groups or sets.
  - > For example, clustering algorithms.

What is Cluster Analysis?

Finding groups of points such that the points in a group will be similar (or related) to one another and different from (or unrelated to) the points in other groups



Clustering – applications

Examples:

- Segment customer database based on similar buying patterns.
- Group houses in a town into neighborhoods based on similar features.
- Identify similar Internet usage patterns.
- Clustering emails by content.
- Gene clustering in biology.
- Group documents that have similar content.

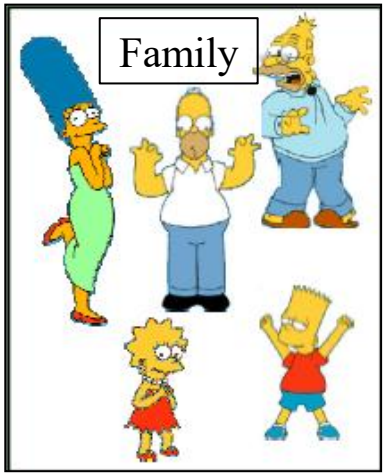
Are these clusters pre-defined?

- No, it depends how the distance between points are measured. There are no class labels.

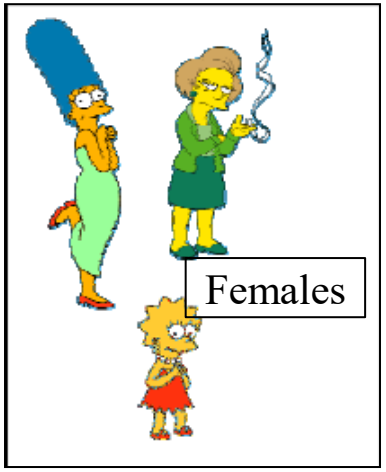
Illustrating clustering

Are there natural groupings amongst this group?

Possible clusters



or



Clustering Definition

- Clustering identifies natural groups in a data set:
  - > Given a set of data points, each having a set of attributes, and a similarity measure, find clusters such that:
  - > Data points in each cluster are more similar to each other.
  - > Data points in separate clusters are less similar.
- Similarity Measures:
  - > Euclidean Distance (e.g., Pythagoras' theorem).
  - > Other distance-based measures (for example, Manhattan).
  - > Other measures if the attribute values are not continuous, e.g., cosine distance for text.

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

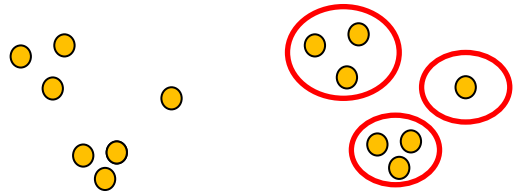
Two Clusters

Four Clusters

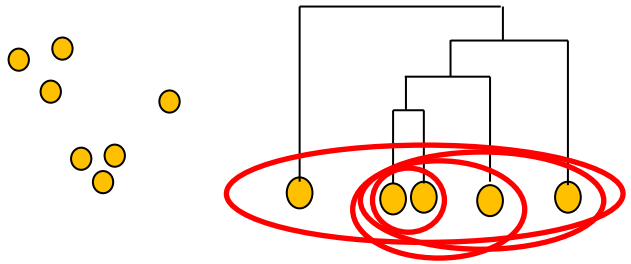
Types of clustering

Two main approaches: partitional and hierarchical.

- Partitional: the division of data points into non-overlapping subsets (clusters) such that each data point is in exactly one subset.



- Hierarchical: a set of nested clusters organized as a hierarchical tree.



k-Means clustering

k-Means Clustering

Partitional clustering approach

Each cluster is associated with a **centroid** (center point)

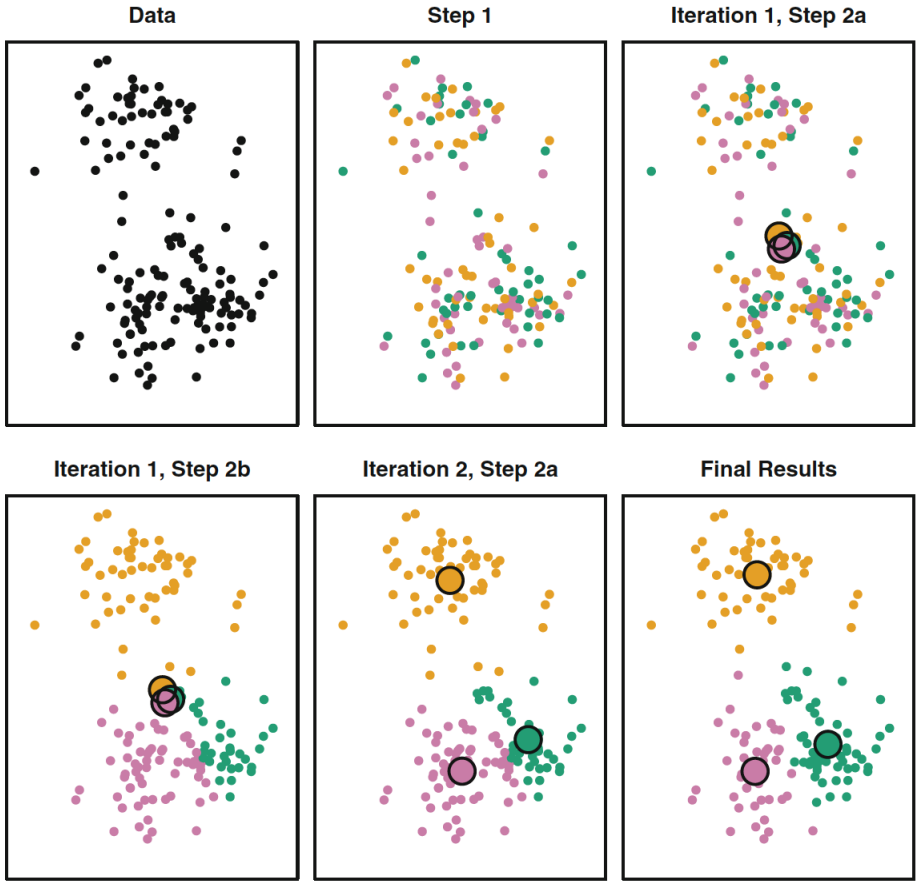
Each point is assigned to the cluster with the closest centroid

Number of clusters, *k*, must be specified

The basic algorithm is very simple:

1. Select *k* points (at random) as the initial centroids
2. **Repeat**
    3. Form *k* clusters by assigning all points to the closest centroid
    4. Re-compute the centroid of each cluster
5. **Until** the centroids don't change

k-Means demonstration



In Step 1 each point is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. In Step 2(b), each point is assigned to the nearest centroid. Bottom center: Step 2(a) is once again performed, leading to new cluster centroids. Bottom right: the results obtained after ten iterations.

James et al., An Introduction to Statistical Learning

Finding the centroids

How do we decide which is the **closest centroid**?

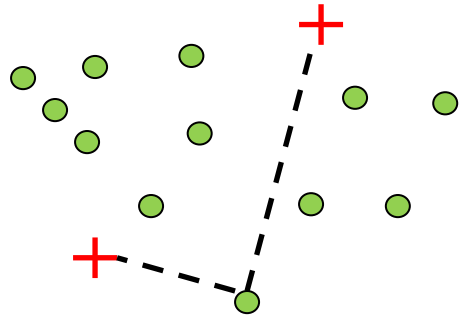
We need to find the ‘distance’ between each point and all the centroids

What does ‘distance’ mean?

There are many ways of defining ‘distance’. We need to use a distance metric.

Data points

Centroids



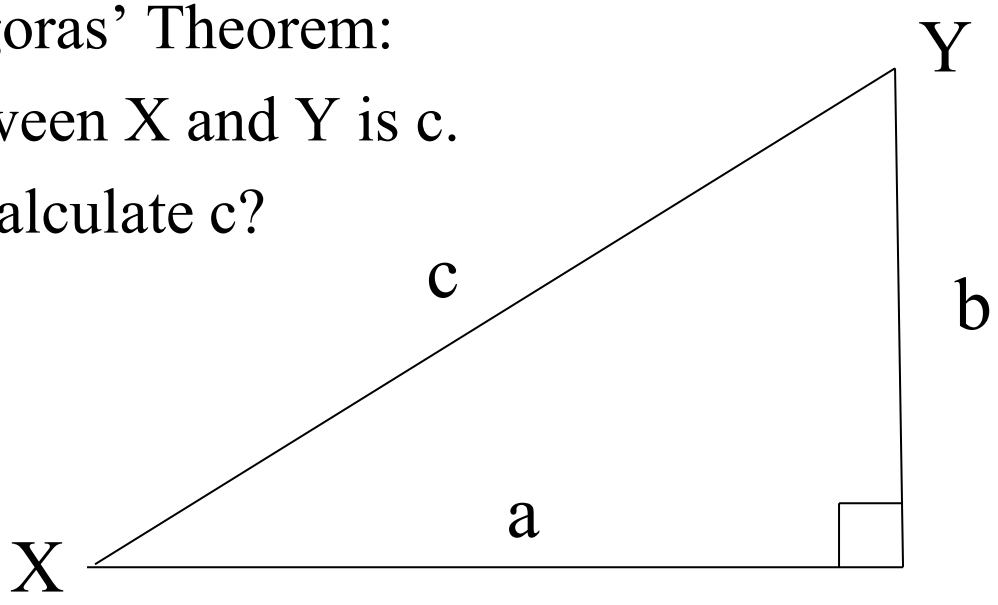
Euclidean distance

Euclidean distance is the shortest distance between two points.

Using Pythagoras' Theorem:

Distance between X and Y is c.

How do we calculate c?



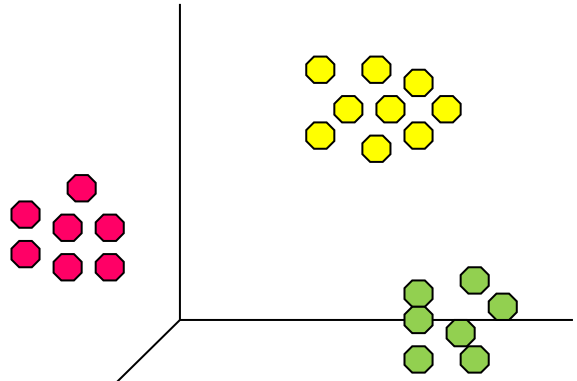
$$c^2 = a^2 + b^2, therefore\ c = \sqrt{(a^2 + b^2)}$$

This model can be applied to multiple dimensions!

What k-Means is aiming to do

The objective of the k-Means algorithm is to minimise the total squared distance of each point to its centroid:

$\sum_{i=1}^{k} \sum_{j=1}^{n} d(c_i, x_{i,j})^2$ where:



- k is the number of clusters
- c_i is the centroid of each cluster for i=1,…,k
- n_i is the number in cluster i
- $x_{i,j}$ is the jth point of cluster i
- $d(c_i, x_{i,j})$ is the distance between c_i and $x_{i,j}$.

Evaluating k-Means Clusters

Most common measure: Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest cluster.
- To get SSE, we square these errors and sum them.
- x_i is a data point in cluster C_i and c_i is the centroid of cluster C_i .
- From previous slide: SSE = $\sum_{i=1}^{k} \sum_{j=1}^{n} d\left(c_i, x_{i,j}\right)^2$
- Given two sets of clusters, we can choose the one with the smallest error.
- Note: the easiest way to reduce SSE is to increase *k*, the number of clusters.
- We look at some alternative squared measures using R in the following examples.

Normalising attributes

It is a good idea to normalise the data before clustering, otherwise large valued attributes will exert greater influence on the clustering.

This is achieved by rescaling each attribute to fit within the same range (for example, between 0 and 1). To normalize attribute A:

MaxA and *MinA* are the maximum and minimum of *A*. Then, the normalized values of *A* are: $x_{new}=\frac{x-MinA}{MaxA-MinA}$

R software has a function (scale) which performs a similar – but not identical function.

Pre-processing and post-processing

Pre-processing

- Normalise the data
- Eliminate outliers

Post-processing

- Eliminate small clusters that may represent outliers
- Split ‘loose’ clusters, i.e., clusters with relatively high SSE.
- Merge clusters that are ‘close’ and that have relatively low SSE.

k-Means clustering in R

The k-Means function is built into the Stats package, which is loaded by default.

Using the iris data:

```
> set.seed(9999) # makes “random” method repeatable
> # clone and scale numerical data
> niris = iris
> niris[,1:4] = scale(niris[,1:4])
```

k-Means clustering in R

Using sepals (Cols 1 & 2), create 3 clusters, taking the best out of 20 starting configurations.

```
> ikfit = kmeans(niris[,1:2], 3, nstart = 20)
> ikfit
> table(actual = niris$Species, fitted = ikfit$cluster)
```

```
            fitted
actual        1  2  3
  setosa     50  0  0
  versicolor  0 12 38
  virginica   0 35 15
```

k-Means clustering in R

Looking at the ikfit object:

> ikfit

```
K-means clustering with 3 clusters of sizes 50,
47, 53

Cluster means:
  Sepal.Length Sepal.Width
1     5.006000    3.428000
2     6.812766    3.074468
3     5.773585    2.692453

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 ...
```

k-Means clustering in R

Looking at the ikfit object:

```
...
Within cluster sum of squares by cluster:
[1] 13.1290 12.6217 11.3000
 (between_SS / total_SS =  71.6 %)

Available components:

[1] "cluster"      "centers"      "totss"
[4] "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

k-Means clustering in R

Looking at the sums of squares calculations:

> ikfit$totss # Total SS from a single centroid (treats data as one cluster).

```
[1] 130.4753
```

> ikfit$withinss # SS within each cluster.

```
[1] 13.1290 12.6217 11.3000
```

> ikfit$tot.withinss # Total within clusters. (Sum of Squared Error)

```
[1] 37.0507
```

> ikfit$betweenss # Total sum of squares - total SS within clusters.

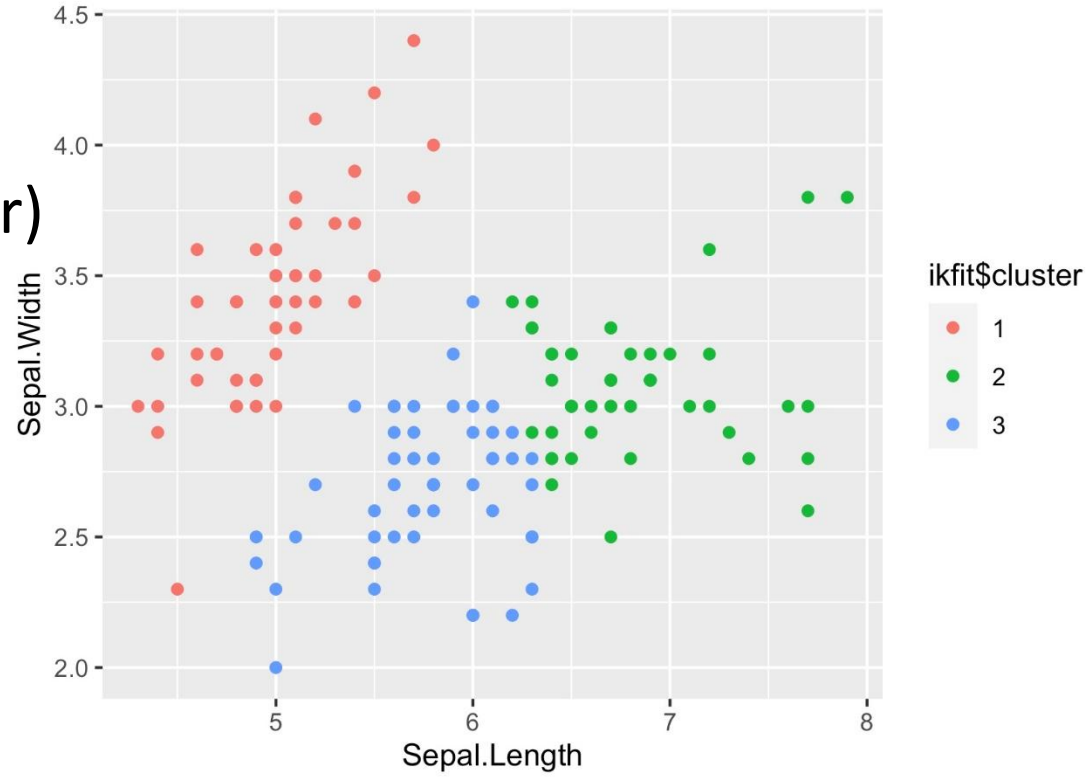
```
[1] 93.42456
```

k-Means clustering in R

Plotting the clusters:

> ikfit$cluster = as.factor(ikfit$cluster)

> ggplot(iris, aes(Sepal.Length, Sepal.Width, color = ikfit$cluster)) + geom_point()



? kmeans

- Description

  Perform k-means clustering on a data matrix.

- Usage

```
kmeans(x, centers, iter.max = 10, nstart = 1,
algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
"MacQueen"), trace=FALSE)
x           data
centers     number of clusters (k)
nstart      random starting positions to test
iter.max    maximum number of iterations
...
```

k-Means clustering in R

Note that using both petals and sepals improves the accuracy of the clustering for these data:

```
> ikfit = kmeans(niris[,1:4], 3, nstart = 20)
> table(actual = niris$Species, fitted = ikfit$cluster)
                fitted
actual           1  2  3
  setosa        50  0  0
  versicolor     0 39 11
  virginica      0 14 36
> ikfit$tot.withinss
[1] 138.8884
```

k-Means for classification…

From previous slide, using both petals and sepals for the clustering:

> table(actual = niris$Species, fitted = ikfit$cluster)

```
                fitted
actual           1  2  3
  setosa        50  0  0
  versicolor     0 39 11
  virginica      0 14 36
```

- If we classify Setosa = Group 1, Versicolor = Group 2 and Virginica = Group 1, this has an accuracy of:

> (50 + 39 + 36)/150 = 0.83.

k-Means clustering in R

But the number of clusters is arbitrary, for example:

> ikfit = kmeans(niris[,1:4], 5, nstart = 20)

> ttable(actual = niris$Species, fitted = ikfit$cluster)

```
                fitted
actual           1  2  3  4  5
  setosa        22  0  0  0 28
  versicolor     0 21 27  2  0
  virginica      0  2 21 27  0
```

> ikfit$tot.withinss

```
[1] 90.20221
```

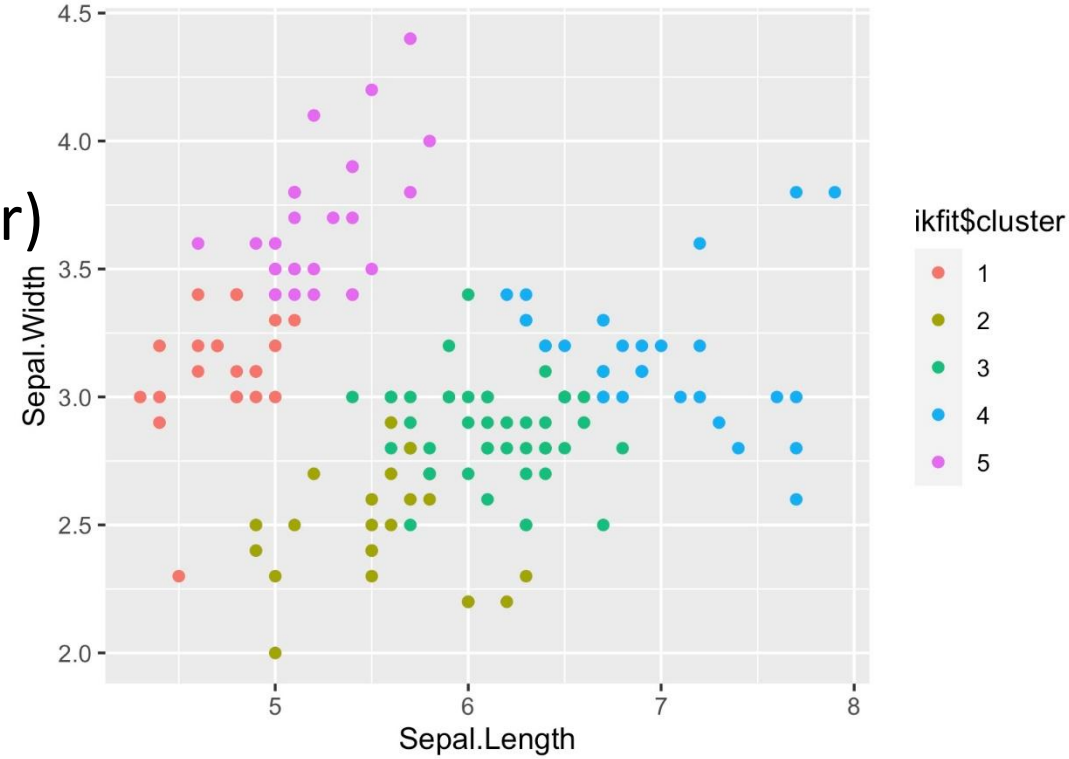
compared to 138.88 for 3 clusters!

k-Means clustering in R

Plotting the clusters:

> ikfit$cluster = as.factor(ikfit$cluster)

> ggplot(iris, aes(Sepal.Length, Sepal.Width, color = ikfit$cluster)) + geom_point()



Countries data

Sample socio-economic data for 19 countries.

Country	Per capita income	Literacy	Infant mortality	Life expectancy
Brazil	10326	90	23.6	75.4
Germany	39650	99	4.08	79.4
Mozambique	830	38.7	95.9	42.1
Australia	43000	99	4.57	81.2
China	5300	90.9	23	73
Argentina	13308	97.2	13.4	75.3
United Kingdom	34105	99	5.01	79.4
South Africa	10600	82.4	44.8	49.3
Zambia	1000	68	92.7	42.4
Namibia	5249	85	42.3	52.9
Georgia	4200	100	17.36	71
Pakistan	3320	49.9	67.5	65.5
India	2972	61	55	64.7
Turkey	12888	88.7	27.5	71.8
Sweden	34735	99	3.2	80.9
Lithuania	19730	99.6	8.5	73
Greece	36983	96	5.34	79.5
Italy	26760	98.5	5.94	80
Japan	34099	99	3.2	82.6

Countries data: scaling

```
> summary(CD)
      Country   Per.capita.income    Literacy      Infant.mortality Life.expectancy
 Argentina: 1   Min.   :  830     Min.   : 38.70   Min.   : 3.200   Min.   :42.10
 Australia: 1   1st Qu.: 4724     1st Qu.: 83.70   1st Qu.: 5.175   1st Qu.:65.10
 Brazil   : 1   Median :12888     Median : 96.00   Median :17.360   Median :73.00
 China    : 1   Mean   :17845     Mean   : 86.36   Mean   :28.574   Mean   :69.44
 Georgia  : 1   3rd Qu.:34102     3rd Qu.: 99.00   3rd Qu.:43.550   3rd Qu.:79.45
 Germany  : 1   Max.   :43000     Max.   :100.00   Max.   :95.900   Max.   :82.60
 (Other)  :13

> # scale numerical data
> CD[,2:5] = scale(CD[,2:5])

> summary(CD)
      Country   Per.capita.income    Literacy      Infant.mortality  Life.expectancy
 Argentina: 1   Min.   :-1.1367   Min.   :-2.5773   Min.   :-0.8459   Min.   :-2.0659
 Australia: 1   1st Qu.:-0.8765   1st Qu.:-0.1440   1st Qu.:-0.7800   1st Qu.:-0.3281
 Brazil   : 1   Median :-0.3312   Median : 0.5211   Median :-0.3738   Median : 0.2688
 China    : 1   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 Georgia  : 1   3rd Qu.: 1.0861   3rd Qu.: 0.6833   3rd Qu.: 0.4993   3rd Qu.: 0.7562
 Germany  : 1   Max.   : 1.6805   Max.   : 0.7374   Max.   : 2.2444   Max.   : 0.9942
 (Other)  :13
```

Countries data: k-Means

k-Means for the scaled data set

```
> set.seed(9999)
> CD <- read.csv("CountriesData.csv")
> # clone and scale numerical data
> CDS = CD
> CDS[,2:5] = scale(CD[,2:5])
> CDSkfit = kmeans(CDS[,2:5], 3, nstart = 20)
> CDSkfit
> table(actual = CDS$Country, fitted = CDSkfit$cluster)
```

Non–scaled *v* scaled clusters

Not-scaled

actual	fitted 1	2	3
Argentina	1	0	0
Australia	0	0	1
Brazil	1	0	0
China	1	0	0
Georgia	1	0	0
Germany	0	0	1
Greece	0	0	1
India	0	1	0
Italy	0	0	1
Japan	0	0	1
Lithuania	1	0	0
Mozambique	0	1	0
Namibia	0	1	0
Pakistan	0	1	0
South Africa	0	1	0
Sweden	0	0	1
Turkey	1	0	0
United Kingdom	0	0	1
Zambia	0	1	0

Scaled

actual	fitted 1	2	3
Argentina	1	0	0
Australia	0	0	1
Brazil	1	0	0
China	0	1	0
Georgia	0	1	0
Germany	0	0	1
Greece	0	0	1
India	0	1	0
Italy	0	0	1
Japan	0	0	1
Lithuania	1	0	0
Mozambique	0	1	0
Namibia	0	1	0
Pakistan	0	1	0
South Africa	1	0	0
Sweden	0	0	1
Turkey	1	0	0
United Kingdom	0	0	1
Zambia	0	1	0

Scaling changes the clusters of these countries.

Analysing the clusters

Using scaled cluster means

See difference in indicators

> > CDkfit

> K-means clustering with 3 clusters of sizes 6, 6, 7

> Cluster means:

	Income	Lit	I.mortality	L.exp
1	-0.925	-1.200	1.259	-1.256
2	-0.460	0.434	-0.322	0.287
3	1.187	0.656	-0.803	0.830

Analysing the clusters

If you want to compare the original (un-scaled) data then you can adapt the following:

```
> by(CD$Per.capita.income, CDSkfit$cluster, mean)
Cluster means:CDSkfit$cluster: 1
[1] 3995.167
------------------------------------------------
CDSkfit$cluster: 2
[1] 35618.86
------------------------------------------------
CDSkfit$cluster: 3
[1] 10958.67
```

k-Means: some considerations

How do we decide which k to use?

- Trial and error?
- There is no single best way of doing this. One reference with some good approaches is https://uc-r.github.io/kmeans_clustering.
- The first method shows, the average silhouette, adapted from Giordani et al., An Introduction to Clustering with R.
- The second is the “elbow method”.

K-Means: Silhouette

- The average silhouette calculates how well each data point sits within its cluster. It is a proxy measure for the quality of the clustering. For each point, i ,

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}, i = 1, 2, 3, \ldots$$

- where a_i is the average distance between that point and all other points in the same cluster, and
- b_i is smallest average distance to any cluster it does not belong to. Ideally a_i is small and b_i is large.
- The average s can then be evaluated across all i. at different values of k.

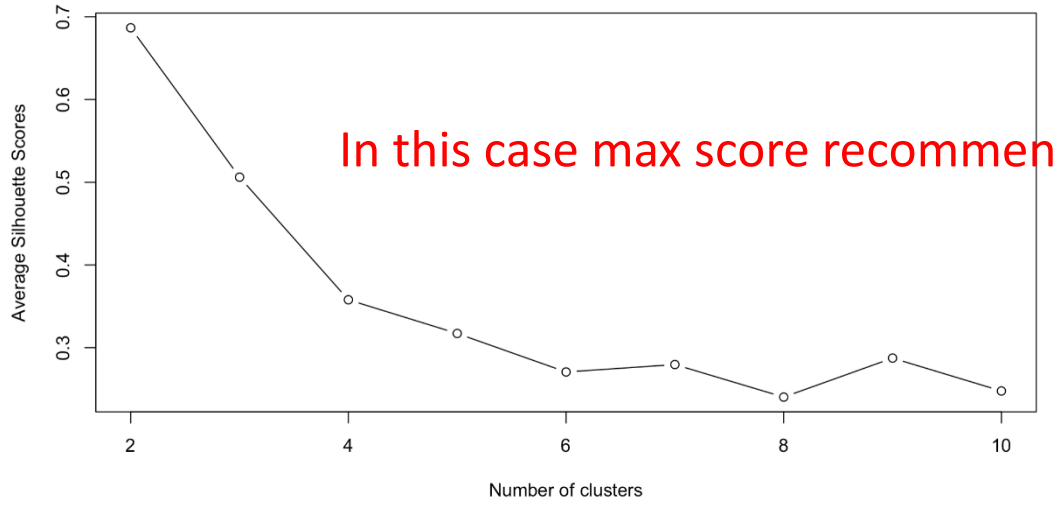
K-Means: Silhouette in R

For the iris data, using sepals and petals:

```
> library(cluster)
> #make function to get average silhouette score
> i_silhouette_score <- function(k){
>   km <- kmeans(iris[,1:4], centers = k, nstart=25)
>   ss <- silhouette(km$cluster, dist(iris[,1:4]))
>   mean(ss[, 3])
> }
```

K-Means: Silhouette in R

```
> #calc and plot average silhouette for 2-10 clusters
> k <- 2:10
> avg_sil <- sapply(k, i_silhouette_score)
> plot(k, type='b', avg_sil, xlab='Number of clusters',
  ylab='Average Silhouette Scores')
```



In this case max score recommends 2 clusters!

K-Means: elbow method

- … Using the "elbow" or "knee of a curve" as a cutoff point is a common heuristic in mathematical optimization to choose a point where diminishing returns are no longer worth the additional cost. In clustering, this means one should choose a number of clusters so that adding another cluster doesn't give a much better modeling of the data. …

- https://en.wikipedia.org/wiki/Elbow_method_(clustering)

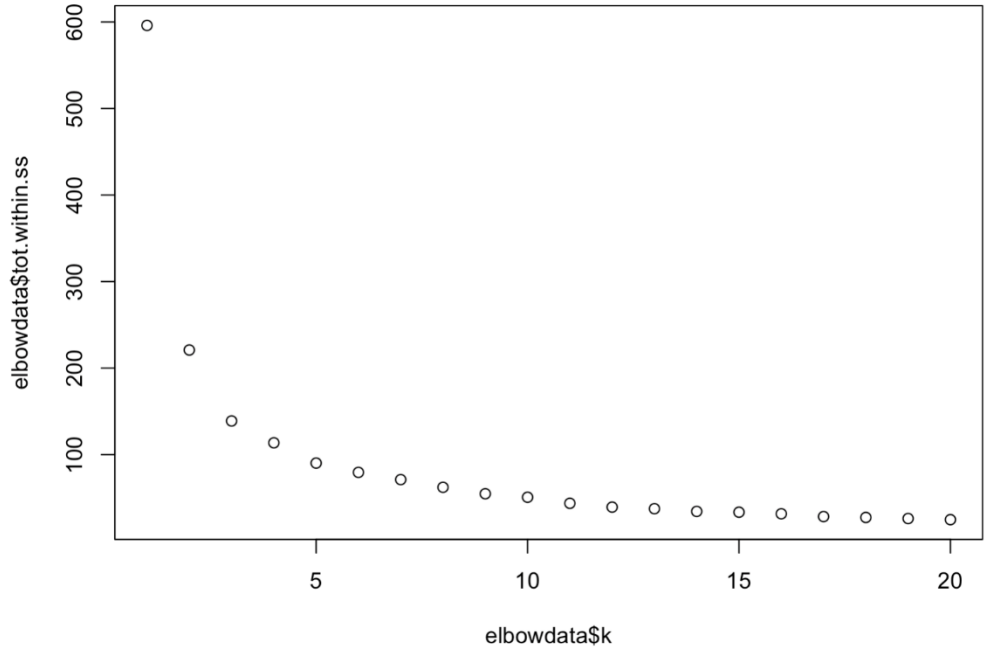
K-Means: elbow method

- The following code evaluates the total within sum of squares as a proxy for explained variation as number of clusters changes:

```
> elbowdata = data.frame()
> for (k in 1:20){
>  kfit = kmeans(niris[,1:4], centers = k, nstart = 10)
>  print(kfit$tot.withinss)
>  elbowdata = rbind(elbowdata, t(c(k,kfit$tot.withinss) ))
>  }
> colnames(elbowdata) = c("k", "tot.within.ss")
> plot(elbowdata$k, elbowdata$tot.within.ss)
```

K-Means: elbow method

- The “elbow” or “knee of a curve” suggests after 5 clusters, say, adding another cluster does not reduce within sum of squares significantly!



K-Means: further thoughts…

- Advantages:
  - > Relatively simple to implement. Scales to large data sets. Guarantees convergence. Can warm-start the positions of centroids. Easily adapts to new examples. Generalizes to clusters of different shapes and sizes.
- Disadvantages:
  - > Dependent on initial values. Clusters of varying sizes and density. Centroids can be dragged by outliers. Outliers might become their own cluster. Non-globular clusters hard to identify.

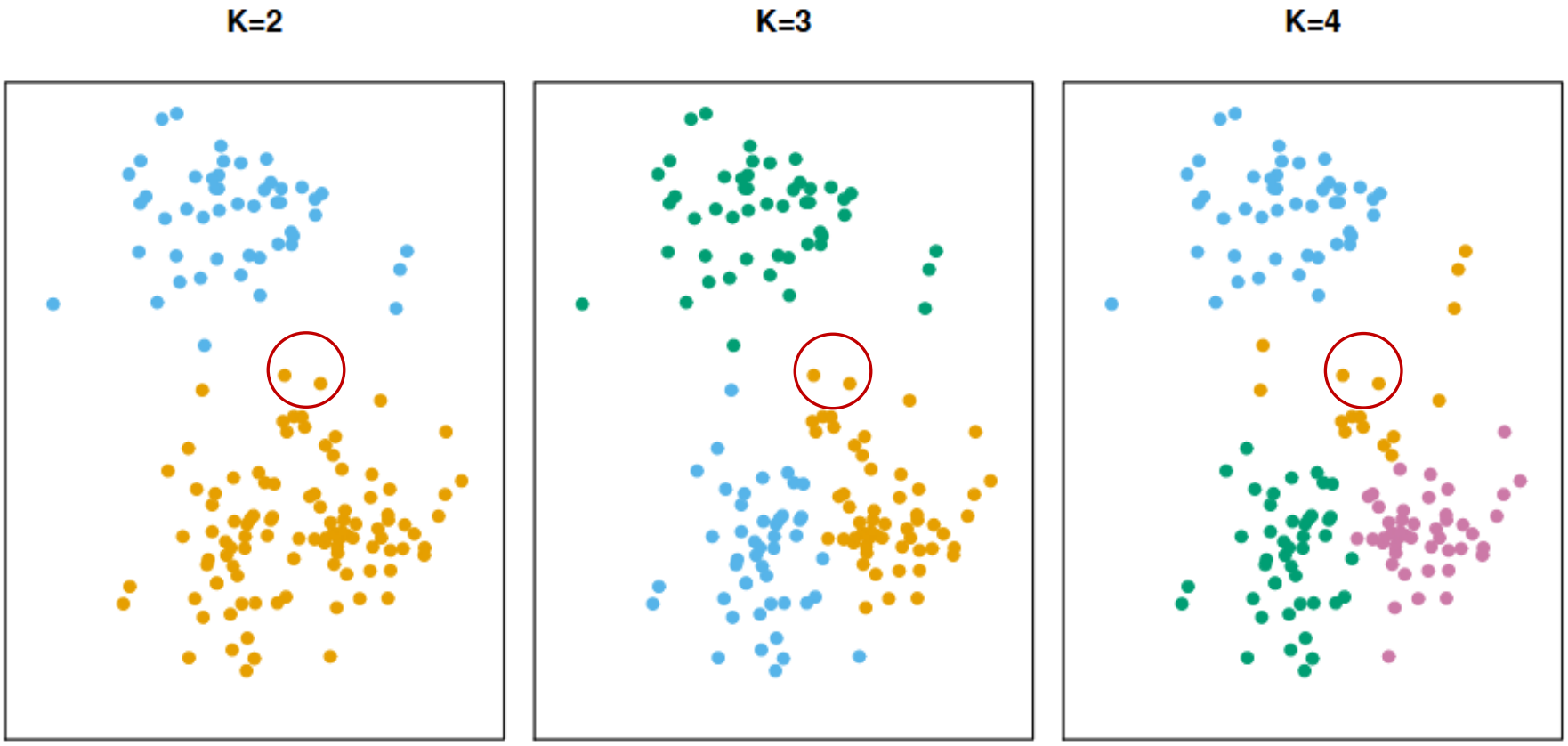
https://developers.google.com/(inactive)

Fuzzy Clustering

Fuzzy Clustering

- Produces a ‘soft’ partitioning as opposed to a ‘crisp’ partitioning.
- Objects are not assigned to a single cluster.
- Assigns fuzzy membership values [0,1] instead.
- Ability to indicate “second best” cluster where “crisp” partitioning is unrealistic.
- Fuzzy memberships can be calculated using a variety of iterative optimization methods.
- Once scaled to [0,1], fuzzy membership can be interpreted as probabilities for belonging in different clusters.

Fuzzy Clustering



Fuzzy K-Means

A fuzzy generalization of the K-Means algorithm.

Introduces a membership degree matrix (U) to the standard objective function.

$$\sum_{t=1}^{g} \sum_{i=1}^{n} u_{it}^{v} \, d^2(\mathbf{x}_i, \mathbf{m}_t),$$

where $\mathbf{m}_t$ is the centre of cluster t, $u_{it} \geq 0$ for all $i = 1, \ldots, n$ and $\Sigma_{t=1}^{g} u_{it} = 1$. The memberships u_{it} are unknown; the $d(\mathbf{x}_i, \mathbf{m}_t)$ are Euclidean distances between the data point and the cluster centres; v is called the *fuzzifier* and affects the final membership distribution; typically it is 2 (setting $v = 1$ leads to the crisp solution).

Fuzzy Clustering in R

The FKM function is built into the fclust package.

Using the scaled countries data:

```
> library(fclust)
> CDSffit <- FKM(CDS[, 2:5], 3, RS = 20)
> CDSffit
> table(actual = CDS$Country , fitted = CDSffit$clus [ , 1])
```

	fitted		
actual	1	2	3
Argentina	0	0	1
Australia	0	1	0
Brazil	0	0	1
China	0	0	1
Georgia	0	0	1
Germany	0	1	0
Greece	0	1	0
India	1	0	0
Italy	0	1	0
Japan	0	1	0
Lithuania	0	0	1
Mozambique	1	0	0
Namibia	0	0	1
Pakistan	1	0	0
South Africa	1	0	0
Sweden	0	1	0
Turkey	0	0	1
United Kingdom	0	1	0
Zambia	1	0	0

Fuzzy Clustering in R

Looking at the CDSffit object:

> CDSffit

```
Fuzzy clustering object of class 'fclust'

Number of objects:
19

Number of clusters:
3

Clustering index values:
SIL.F k=3
0.7888557


Closest hard clustering partition:
Obj 1  Obj 2  Obj 3  Obj 4  Obj 5  Obj 6  Obj 7  Obj 8  Obj 9 Obj 10 Obj 11 Obj 12
    1      3      2      3      1      1      3      2      2      1      1      2
Obj 13 Obj 14 Obj 15 Obj 16 Obj 17 Obj 18 Obj 19
     2      1      3      1      3      3      3
```

Fuzzy Clustering in R

Looking at the CDSffit object:

```
...
 Membership degree matrix (rounded):
       Clus 1 Clus 2 Clus 3
Obj 1    0.96   0.01   0.03
Obj 2    0.02   0.00   0.98
Obj 3    0.09   0.86   0.05
Obj 4    0.04   0.01   0.94
Obj 5    0.97   0.01   0.02
Obj 6    0.86   0.02   0.12
Obj 7    0.00   0.00   1.00
Obj 8    0.41   0.45   0.14
Obj 9    0.08   0.88   0.04
Obj 10   0.50   0.37   0.13
Obj 11   0.91   0.03   0.06
Obj 12   0.13   0.80   0.06
Obj 13   0.23   0.69   0.08
Obj 14   0.96   0.01   0.03
Obj 15   0.00   0.00   1.00
Obj 16   0.59   0.04   0.37
Obj 17   0.01   0.00   0.99
Obj 18   0.13   0.02   0.86
Obj 19   0.01   0.00   0.99

 Available components:
 [1] "U"         "H"         "F"         "clus"      "medoid"    "value"
 [7] "criterion" "iter"      "k"         "m"         "ent"       "b"
[13] "vp"        "delta"     "stand"     "Xca"       "X"         "D"
[19] "call"
```

Fuzzy Clustering in R

> CDSffit$U #membership matrix for all clusters

```
            Clus 1       Clus 2     Clus 3
Obj 1  0.957986151 0.0111989131 0.03081494
Obj 2  0.018366702 0.0042516061 0.97738169
Obj 3  0.089267199 0.8581479540 0.05258485
Obj 4  0.043796634 0.0115279242 0.94467544
Obj 5  0.967198420 0.0114330811 0.02136850
Obj 6  0.859508512 0.0220896580 0.11840183
Obj 7  0.002615322 0.0004761824 0.99690850
Obj 8  0.408969031 0.4530927248 0.13793824
Obj 9  0.082674495 0.8762008816 0.04112462
Obj 10 0.503529217 0.3697745218 0.12669626
Obj 11 0.908242357 0.0280531592 0.06370448
Obj 12 0.134553341 0.8035158511 0.06193081
Obj 13 0.229105143 0.6918108485 0.07908401
Obj 14 0.957988726 0.0124407343 0.02957054
Obj 15 0.002423221 0.0004641798 0.99711260
Obj 16 0.592651848 0.0361148920 0.37123326
Obj 17 0.007733518 0.0016242273 0.99064225
Obj 18 0.126417939 0.0156938904 0.85788817
Obj 19 0.011644105 0.0022224971 0.98613340
```

> CDSffit$H #cluster centres

```
       Per.capita.income   Literacy Infant.mortality Life.expectancy
Clus 1        -0.5465558  0.3149728       -0.1883201       0.1411461
Clus 2        -1.0148587 -1.5608549        1.5889643      -1.3235609
Clus 3         1.1602138  0.6475296       -0.7896332       0.8023638
```

Fuzzy Clustering in R

> CDSffit$clus #assigned clusters and associated membership

```
        Cluster Membership degree
Obj 1         1           0.9579862
Obj 2         3           0.9773817
Obj 3         2           0.8581480
Obj 4         3           0.9446754
Obj 5         1           0.9671984
Obj 6         1           0.8595085
Obj 7         3           0.9969085
Obj 8         2           0.4530927
Obj 9         2           0.8762009
Obj 10        1           0.5035292
Obj 11        1           0.9082424
Obj 12        2           0.8035159
Obj 13        2           0.6918108
Obj 14        1           0.9579887
Obj 15        3           0.9971126
Obj 16        1           0.5926518
Obj 17        3           0.9906423
Obj 18        3           0.8578882
Obj 19        3           0.9861334
> 
```

Fuzzy Clustering: Cluster Validity

- In addition to previously discussed methods for k means the following fuzziness measures can be used to evaluate partition quality.
- Partition Coefficient (PC) $\mathrm{PC} = \sum_{i=1}^{n}\sum_{g=1}^{k}\frac{(u_{ig})^2}{n}.$
- The optimal number of cluster k is achieved when the value is maximized over different values of k.

- Partition Entropy (PE) $\mathrm{PE} = -\sum_{i=1}^{n}\sum_{g=1}^{k}\frac{u_{ig}\log(u_{ig})}{n}$
- The optimal value of k corresponds to the minimum achieved over different values of k.
- R functions PC() and PE() from the fclust library.

Fuzzy Clustering: Further thoughts

The FKM function can also be run without specifying the number of clusters.

Partition Entropy (PE), Partition Coefficient (PC) or Silhouette (SIL) can be used as an index to optimize and automatically select the number of clusters.

> CDSffit = FKM(CDS[,2:5], RS = 20, index = "PE")

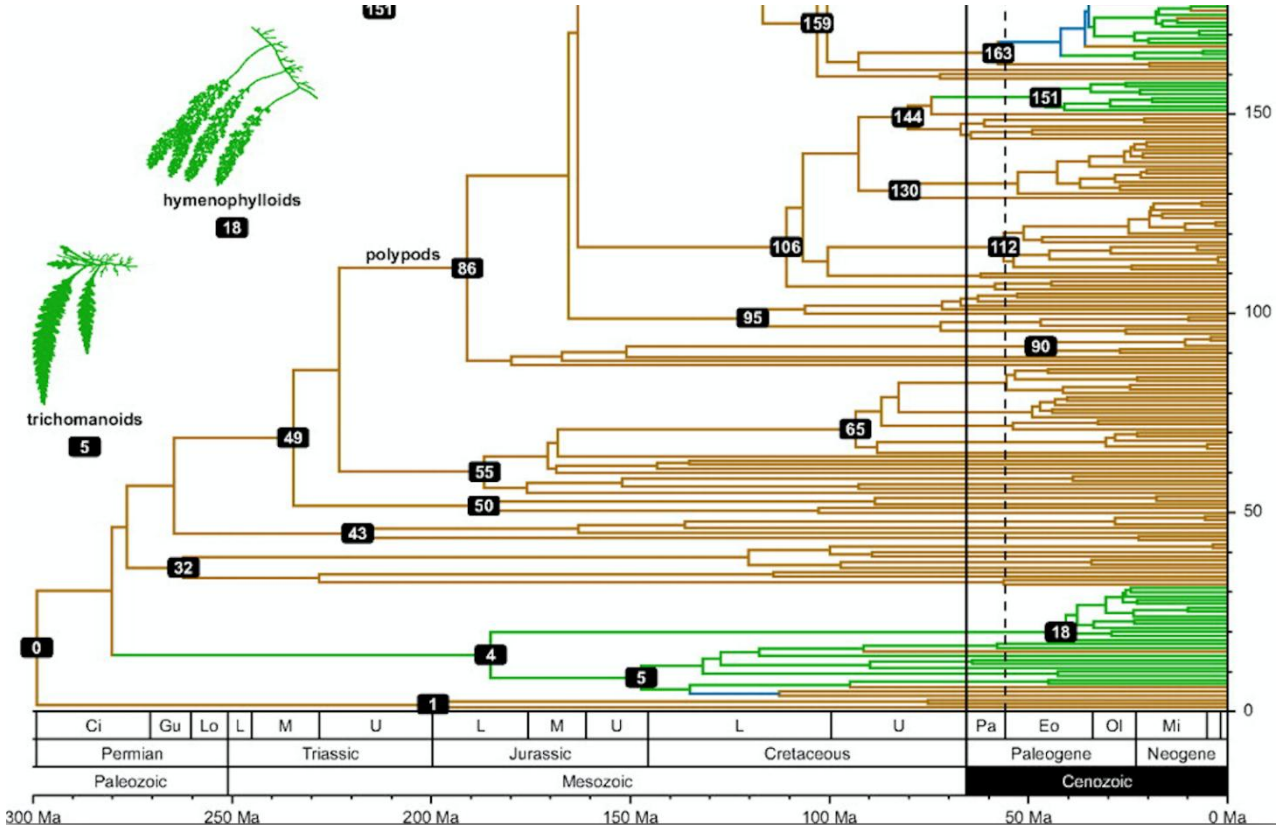
>CDSffit$H

	Per.capita.income	Literacy	Infant.mortality	Life.expectancy
Clus 1	0.5425486	0.5663053	-0.6121265	0.599181
Clus 2	-0.9281045	-1.1991432	1.2465400	-1.170742

Many other fuzzy clustering algorithms available, including Fuzzy c- means, Fuzzy k-mediods, Entropic fuzzy k-means etc.

Hierarchical Clustering

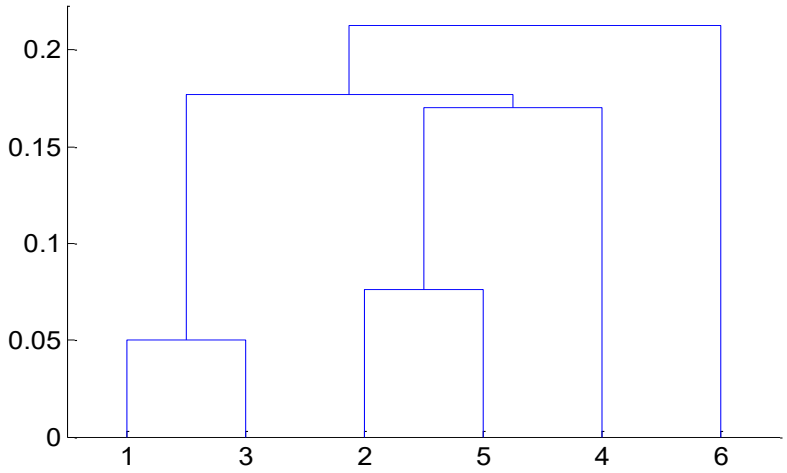
Phylogenetic tree, fern evolution

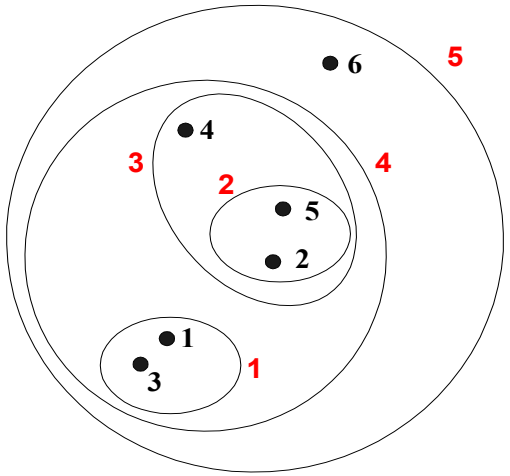


https://www.pnas.org/content/106/27/11200/F1.expansion.html

Hierarchical clustering

- Creates a set of nested clusters organized as a hierarchical tree that:
  - > Records the sequences of merges or splits
  - > Can be visualized as a dendrogram or enclosure diagram





Advantages of hierarchical clustering

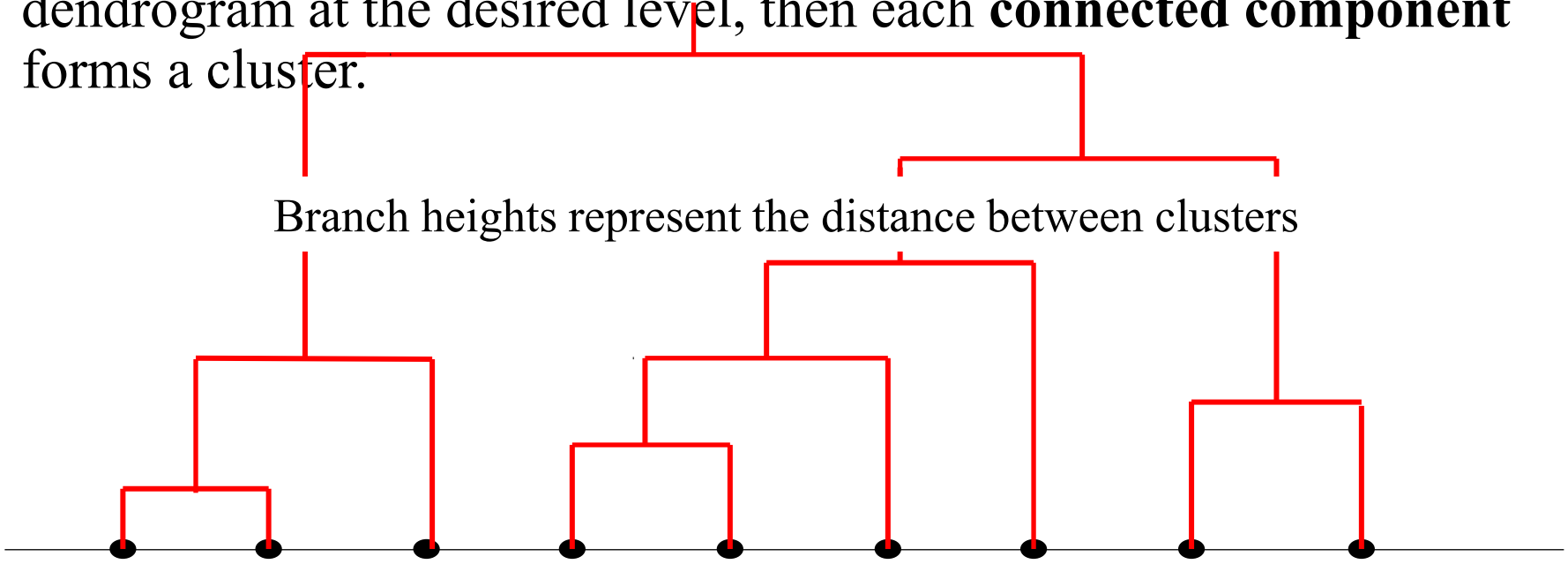
- Do not have to assume any particular number of clusters:
  - > Any desired number of clusters can be obtained by 'cutting' the dendrogram at the appropriate level.

- They may correspond to meaningful taxonomies:
  - > For example, in biological sciences (e.g., plant and animal kingdoms).

Dendrogram and hierarchies

Decompose data objects into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**. But we use agglomerative.

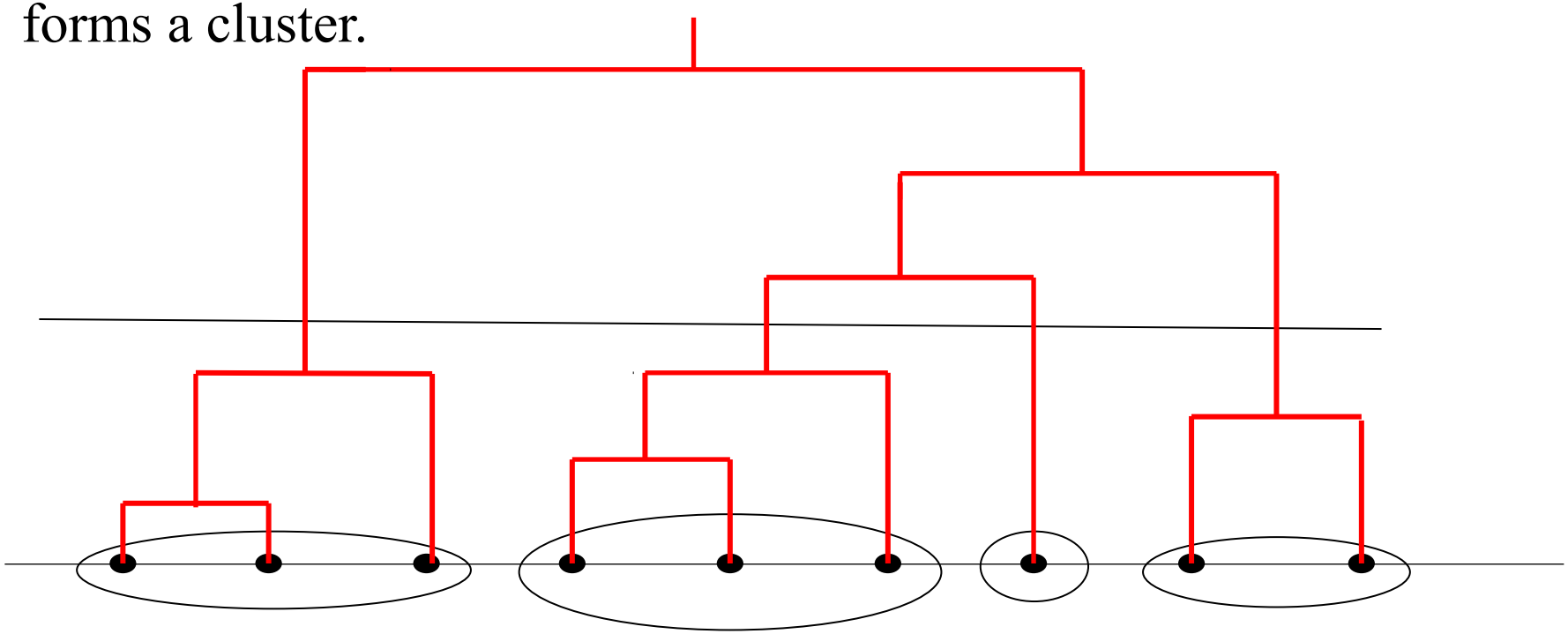
A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Dendrogram and hierarchies

Decompose data points into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

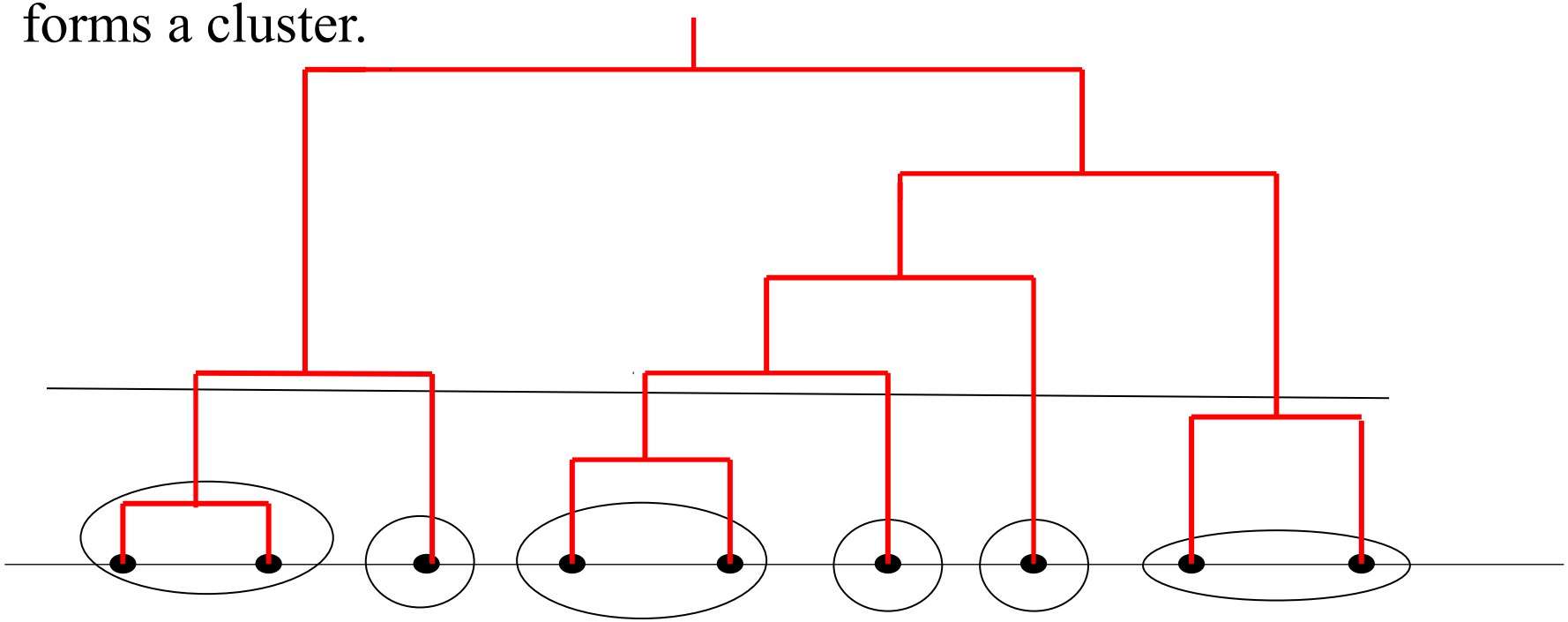
A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Dendrogram and hierarchies

Decompose data points into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Hierarchical Clustering

Two main types of hierarchical clustering:

- Agglomerative (the more usual method):
  - > Start with the points as individual clusters.
  - > At each step, merge the closest pair of clusters until only one cluster (or k clusters) left.
- Divisive:
  - > Start with one, all-inclusive cluster.
  - > At each step, split a cluster until each cluster contains a point (or there are k clusters).
- Traditional hierarchical algorithms use a similarity or distance matrix and merge or split one cluster at a time

Agglomerative Clustering Algorithm

More popular hierarchical clustering technique

Distance matrix stores the distances between each cluster

Basic algorithm is straightforward

1. Compute the distance matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the distance matrix
6. **Until** only a single cluster remains

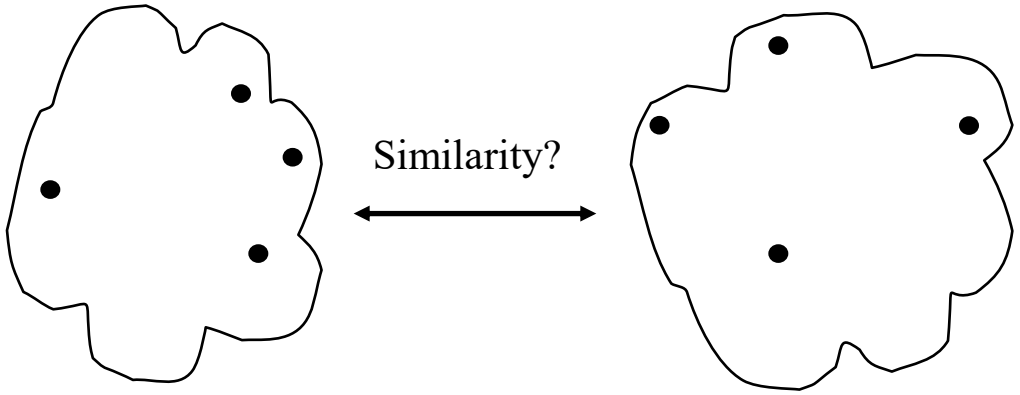
Key operation is the computation of distance between two clusters

Different approaches to defining the distance distinguish the different algorithms.

Agglomerative clustering

The tree is built from the “ground” up…

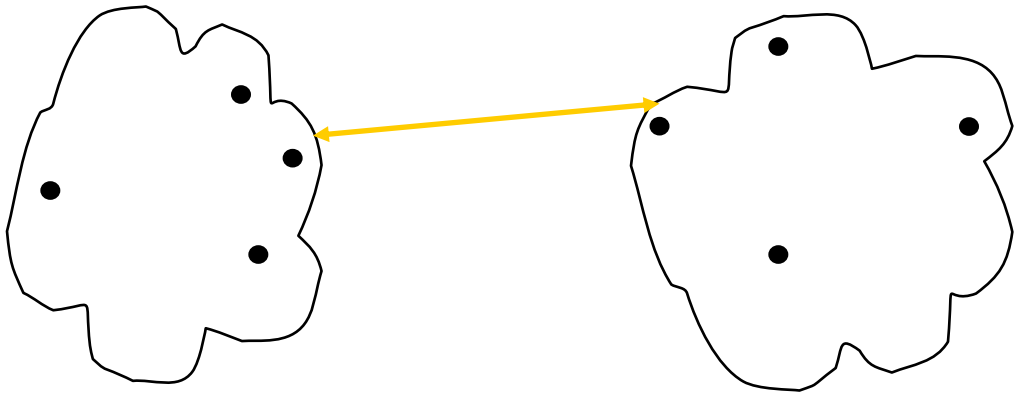
Defining inter-cluster similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average/Median
- Distance Between Centroids

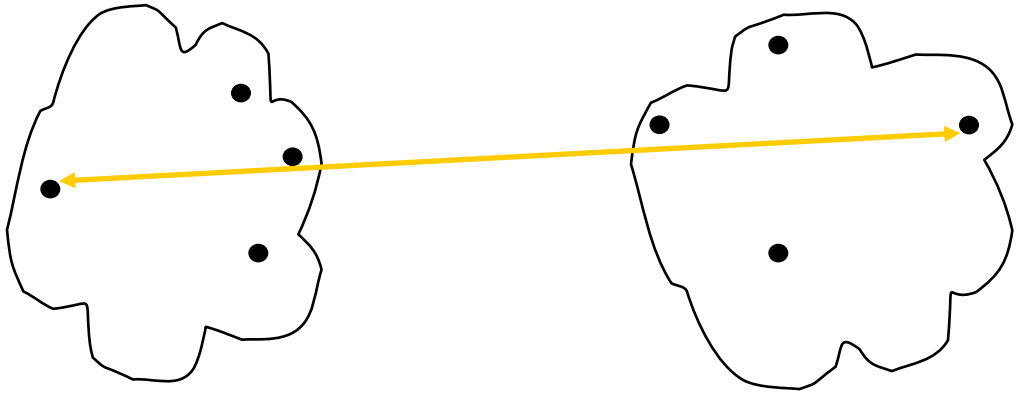
Defining inter-cluster similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average/Median
- Distance Between Centroids

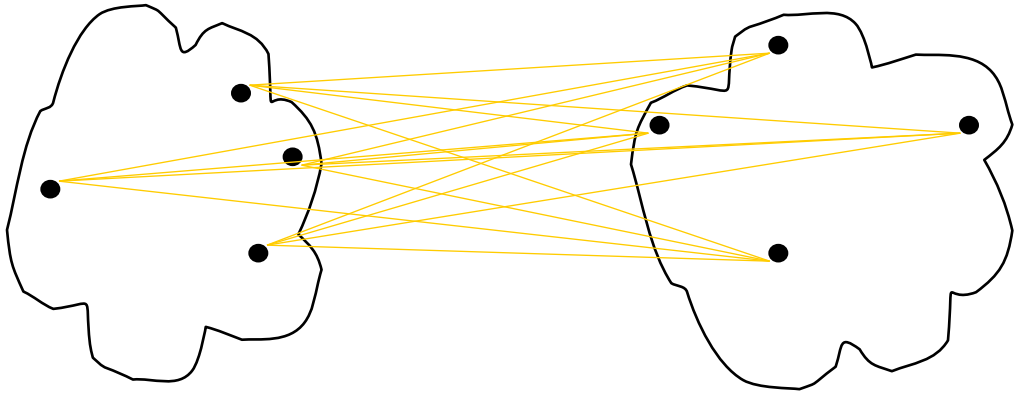
Defining inter-cluster similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average/Median
- Distance Between Centroids

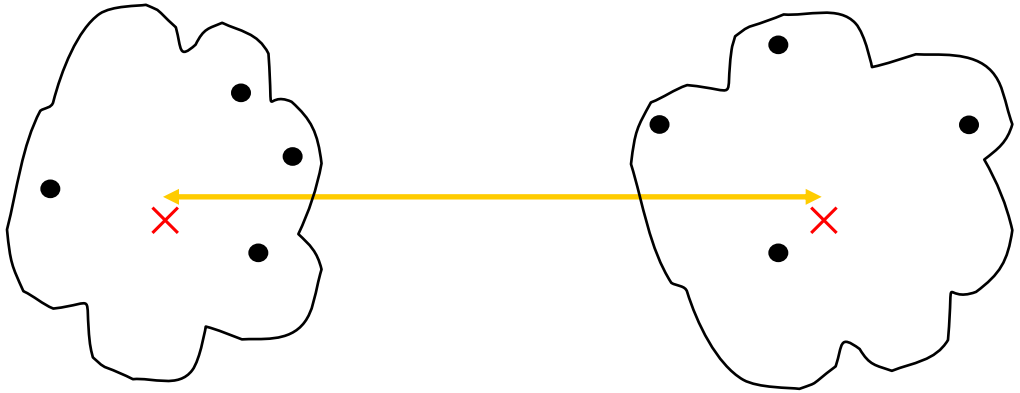
Defining inter-cluster similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average/Median
- Distance Between Centroids

Defining inter-cluster similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average/Median
- Distance Between Centroids

Quick demonstration

Merging with MIN, let's try first merge for a hypothetical data set using distance matrix on the left...

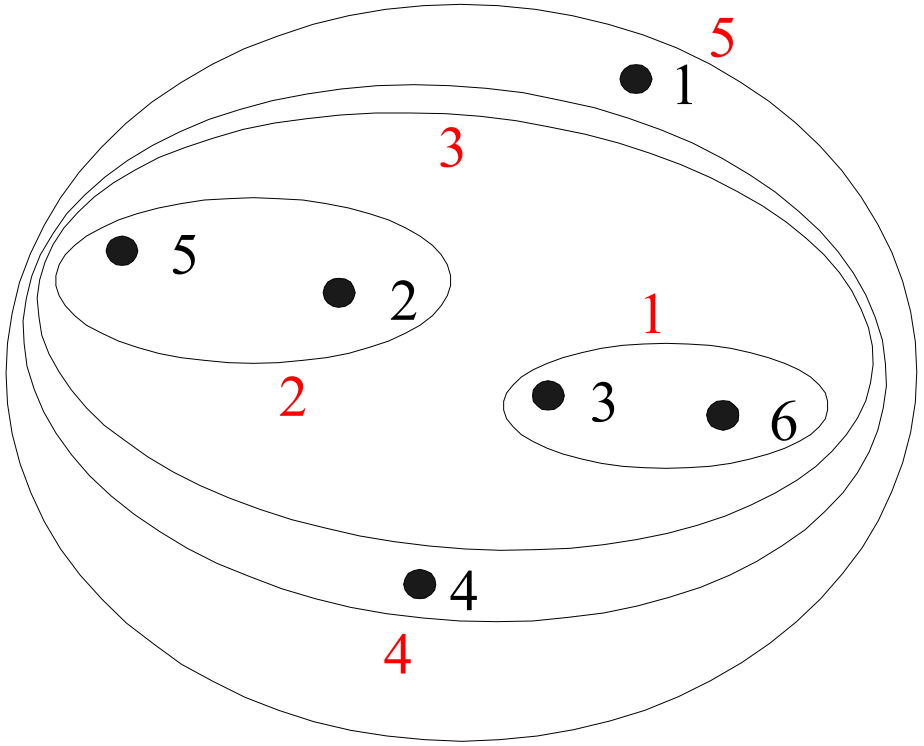
	P1	P2	P3	P4	P5	P6
P1	0	0.24	0.22	0.37	0.34	0.23
P2	0.24	0	0.15	0.2	0.14	0.25
P3	0.22	0.15	0	0.15	0.28	0.11
P4	0.37	0.2	0.15	0	0.29	0.22
P5	0.34	0.14	0.28	0.29	0	0.39
P6	0.23	0.25	0.11	0.22	0.39	0

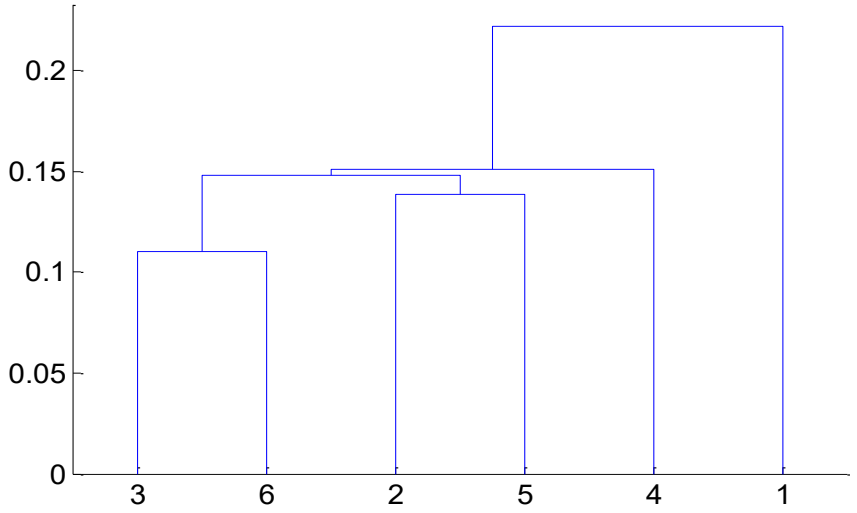
First□oin					
	P1	P2	P36	P4	P5
P1		0.24	0.22	0.37	0.34
P2			0.15	…	
P36					
P4					
P5					

Effect of clustering method

The following slides show the slightly different clustering obtained by MIN, MAX and Group Average distance measures…

Hierarchical Clustering: MIN

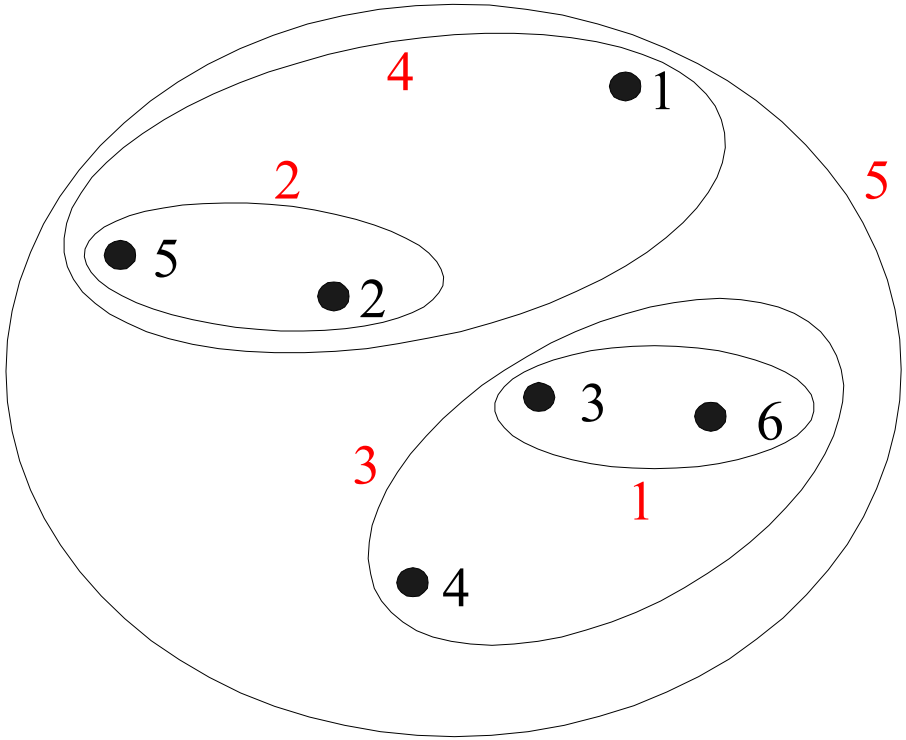




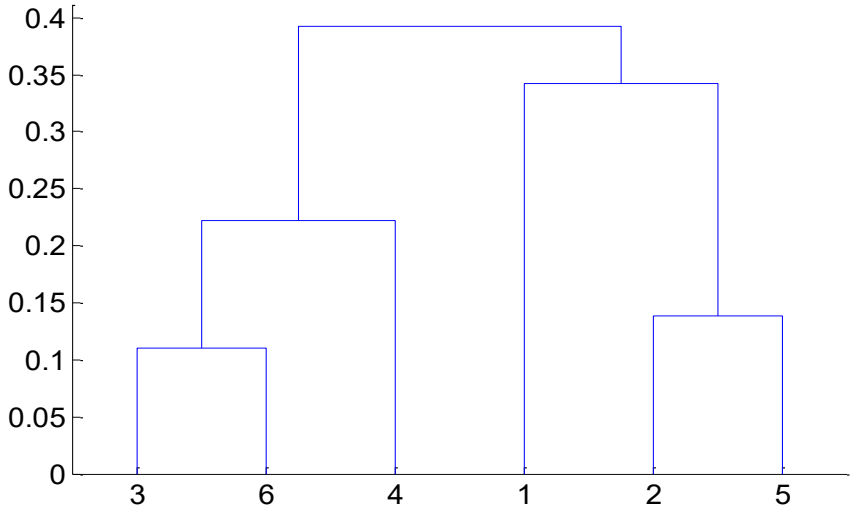
Nested Clusters

Dendrogram

Hierarchical Clustering: MAX

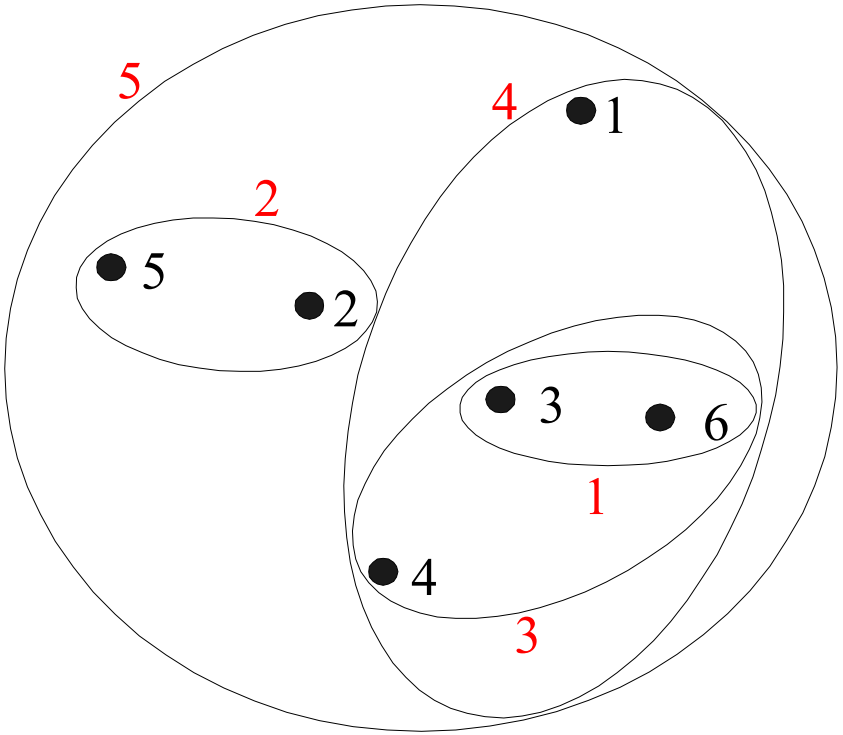


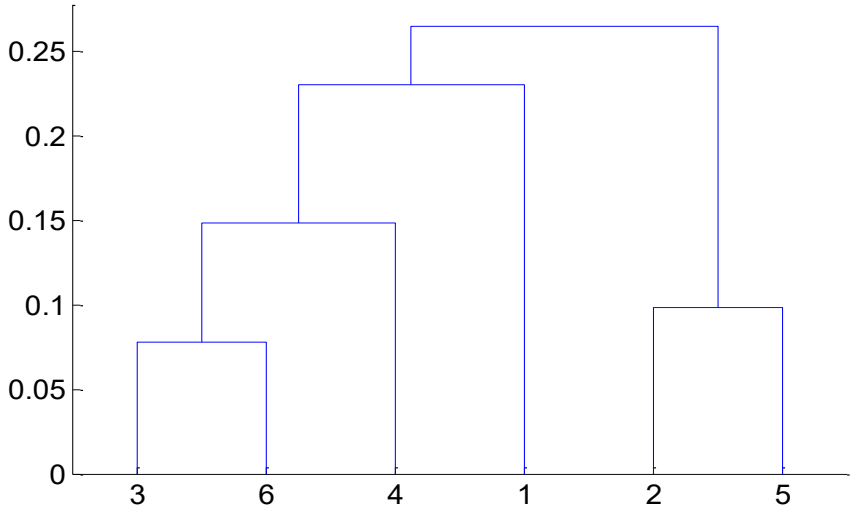
Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average

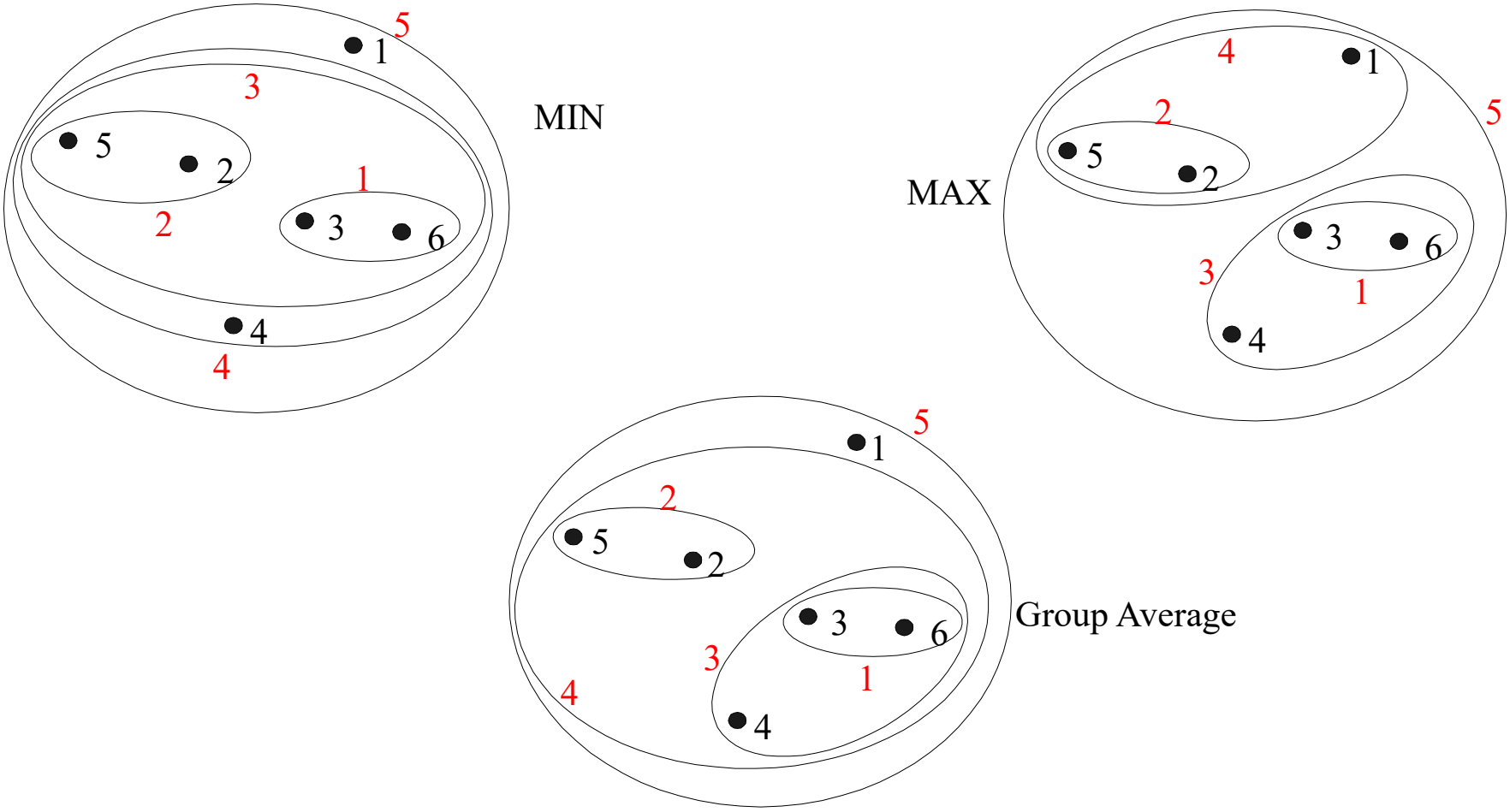




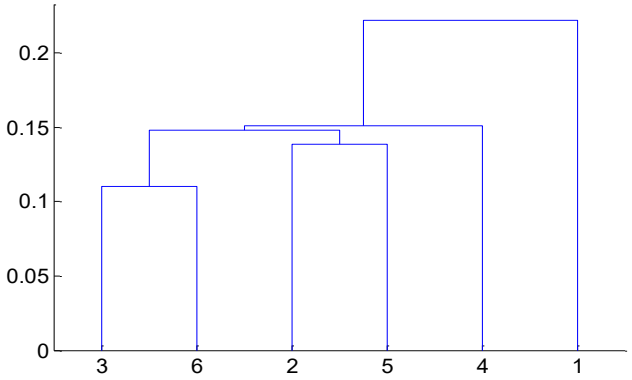
Nested Clusters

Dendrogram

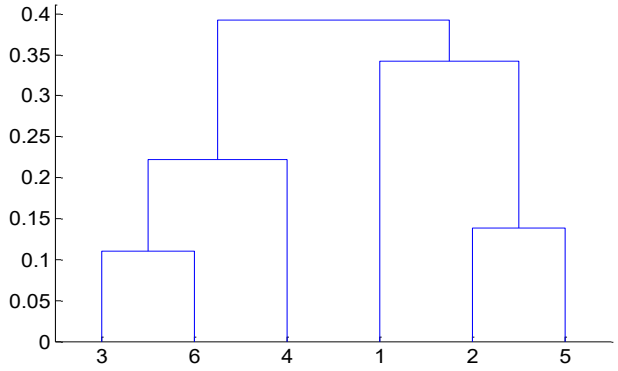
Hierarchical Clustering: Comparison



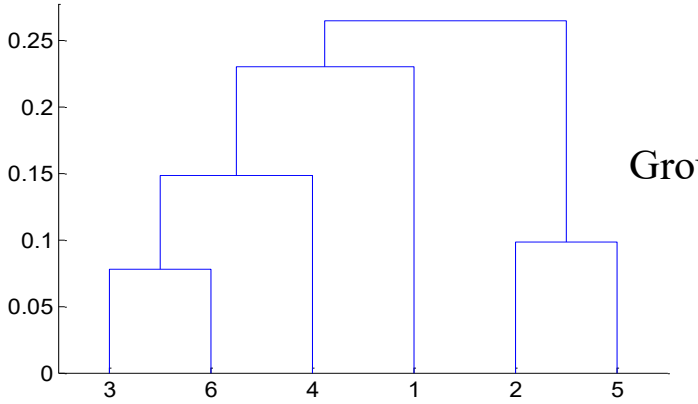
Hierarchical Clustering: Comparison



MIN



MAX



Group Average

Similarity measures: pros and cons

- MIN
  - > Can handle non-elliptical shapes
  - > Sensitive to noise and outliers
- MAX
  - > Less susceptible to noise and outliers
  - > Tends to break large clusters, biased towards elliptical shapes
- Group Average
  - > Compromise between Single and Complete Link
  - > Less susceptible to noise and outliers
  - > Biased towards globular clusters

Hierarchical clustering: considerations

- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized, unlike k-Means.
- Different schemes have problems with one or more of the following:
  - > Sensitivity to noise and outliers.
  - > Difficulty handling different sized clusters and convex shapes.
  - > Breaking large clusters.

Hierarchical clustering in R

Hierarchical clustering of the Iris data.

We use hclust, built into the Stats package and loaded by default.

```
> set.seed(9999) # make results repeatable
> niris = iris
> #scale numerical data
> #niris[,1:4] = scale(niris[,1:4])
> ihfit = hclust(dist(niris[,1:4]), "ave")
> plot(ihfit, hang = -1)
```

? hclust

- Description

  Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

- Usage

```
hclust(d, method = "complete", members = NULL)
```

d	**dissimilarity structure**
method	**agglomeration method to be used. This should be (... one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC)...)**

Hierarchical clustering in R

The fitted object:

```
> ihfit
Call:
hclust(d = dist(niris[, 1:4]), method = "ave")

Cluster method   : average Distance          :
euclidean
Number of objects: 150
```

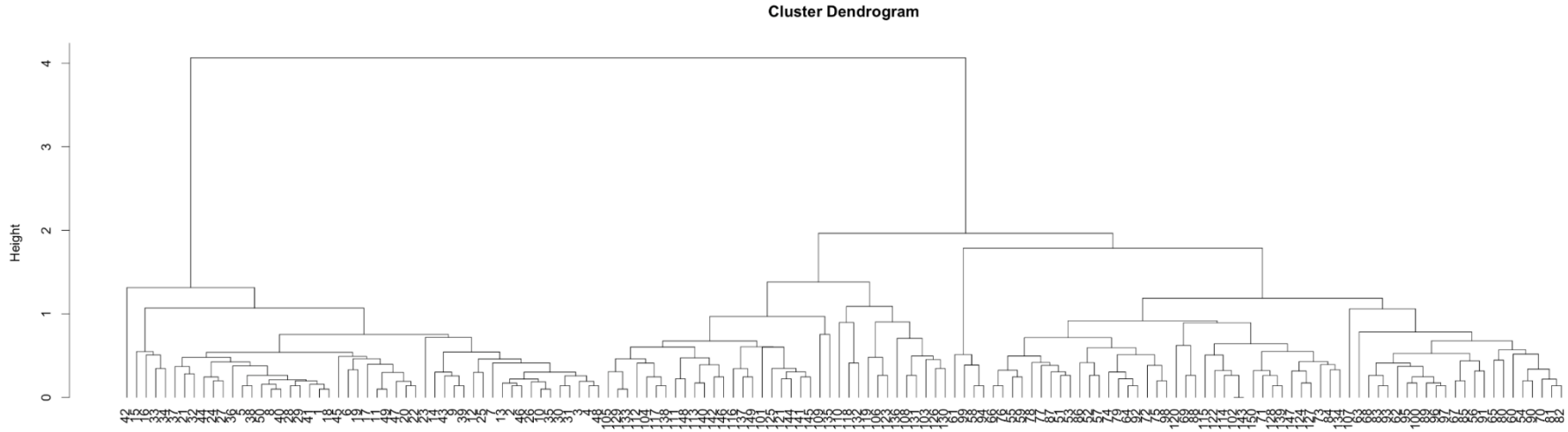
Hierarchical clustering in R

Viewing in the environment browser:

ihfit	List of 7
merge : int [1:149, 1:2] -102 -8 -1 -10 -129 -11 -5 -20 -30 -58 ...	
height : num [1:149] 0 0.1 0.1 0.1 0.1 ...	
order : int [1:150] 42 15 16 33 34 37 21 32 44 24 ...	
labels : NULL	
method : chr "average"	
call : language hclust(d = dist(niris[, 1:4]), method = "ave")	
dist.method: chr "euclidean"	
attr(*, "class")= chr "hclust"	

Hierarchical clustering in R

Dendrogram:



Where are the clusters?

How many do you want?

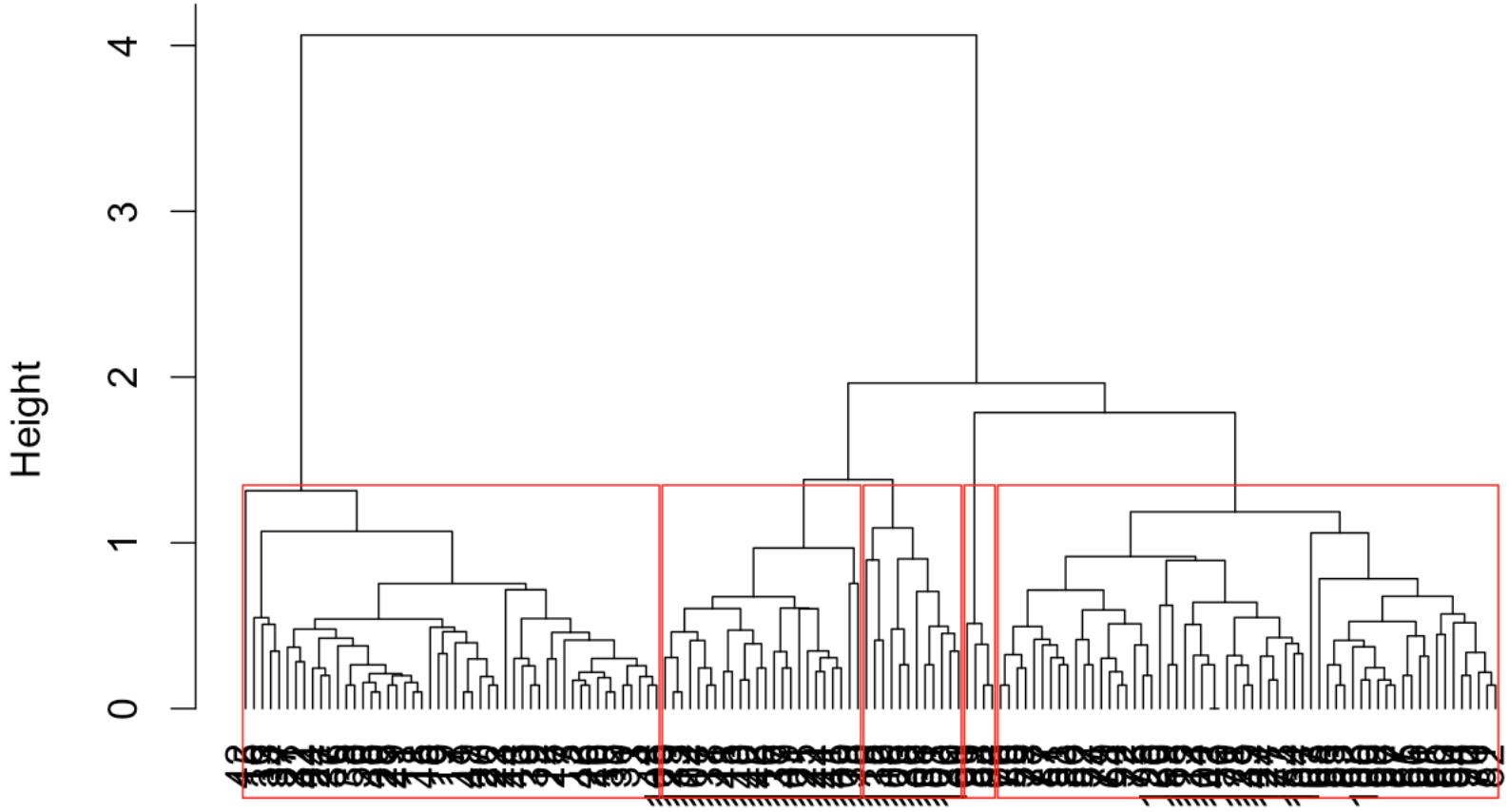
Hierarchical clustering in R

Setting a particular number of clusters:

```
> # pruning the tree into 5 clusters
> cutihfit = cutree(ihfit, k = 5) #associate each iris to a cluster
> rect.hclust(ihfit, k = 5, border = "red")
> table(actual = niris$Species, fitted = cutihfit)
            fitted
actual        1  2  3  4  5
  setosa     50  0  0  0  0
  versicolor  0 46  4  0  0
  virginica   0 14  0 24 12
```

Hierarchical clustering in R

Dendrogram showing 5 clusters:



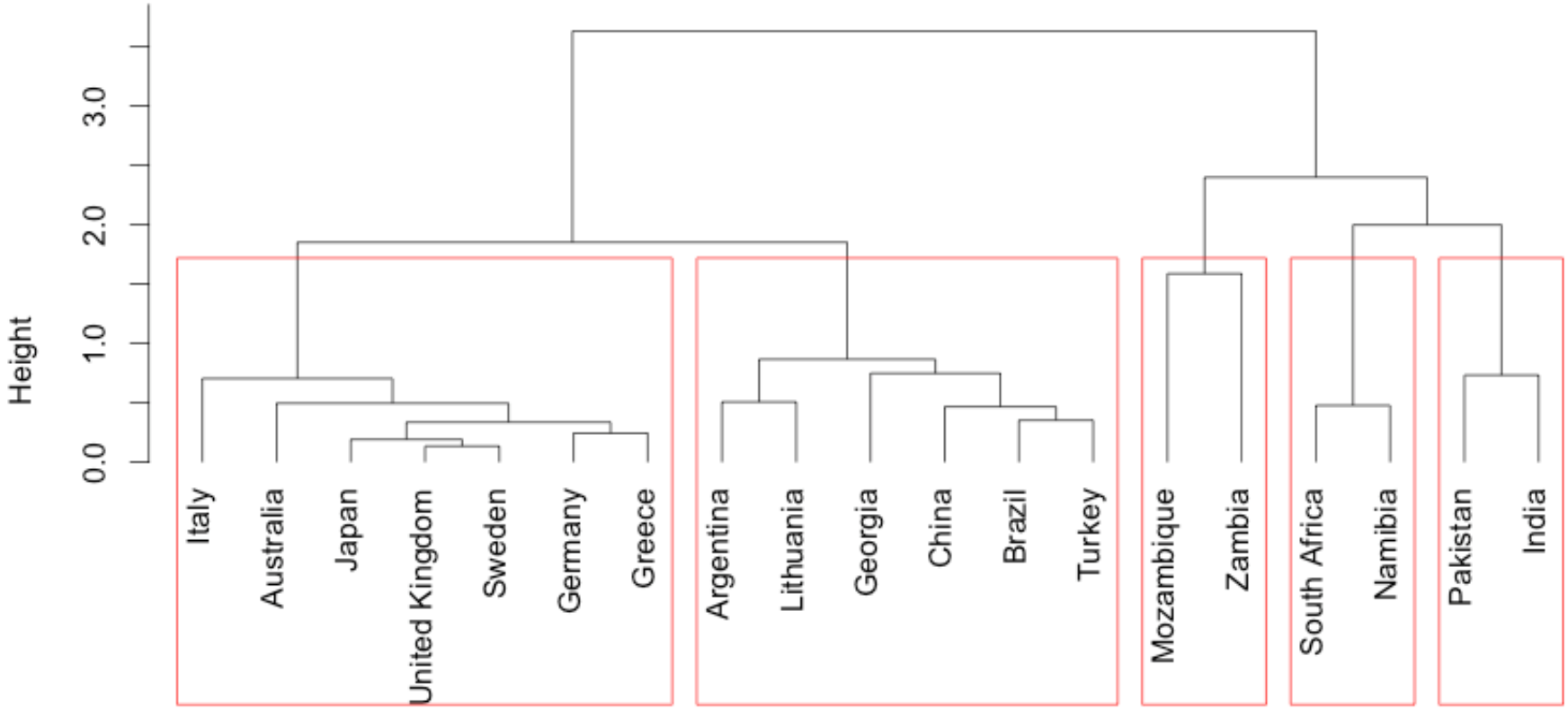
Countries data (scaled)

Reading data, scaling, setting row names to country names (to appear in dendrogram)

```
> CD <- read.csv("CountriesData.csv")
> CD[,2:5] = scale(CD[,2:5])
> rownames(CD) = CD$Country
> hfit = hclust(dist(CD[,2:5]), "average")
> plot(hfit, hang = -1)
> cut.hfit = cutree(hfit, k = 5) #Pruning
> rect.hclust(hfit, k = 5, border = "red")
```

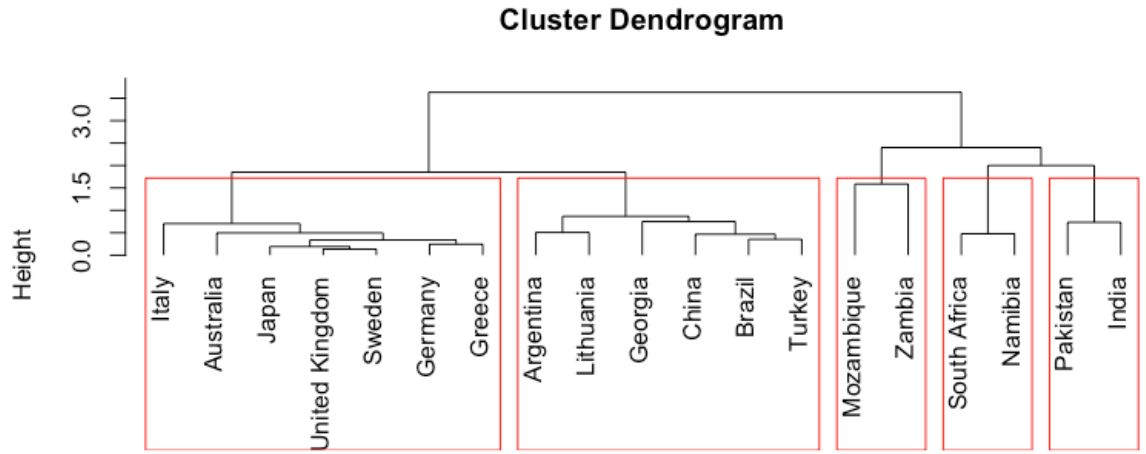
Countries data (scaled)

Dendrogram

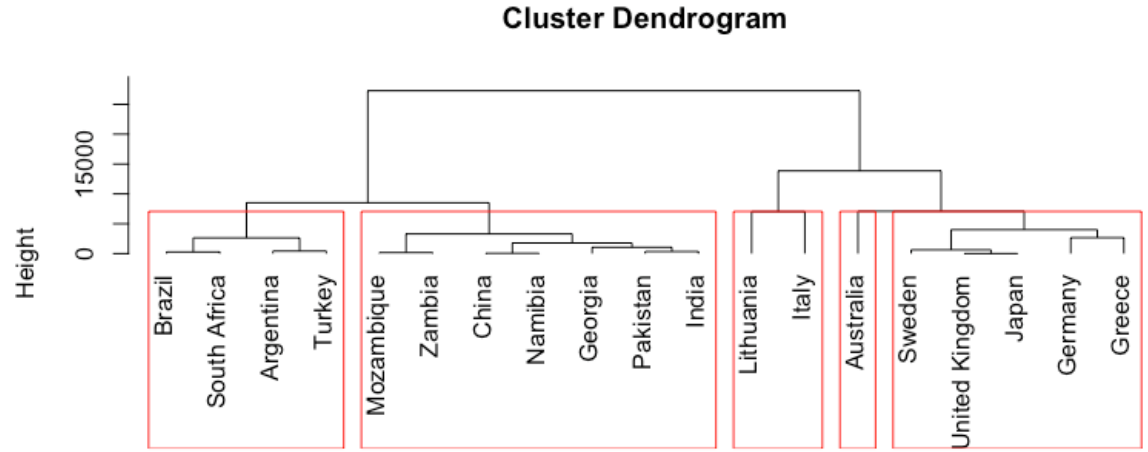


Countries data: effect of scaling

Scaled



Not-scaled



Countries data (normalised)

Normalised input gives similar tree to scaled data:

```
> CD <- read.csv("CountriesData.csv")
> # for loop to normalise cols 2 - 5
> for (i in 2:5){
> CD[,i] = (CD[,i]-min(CD[,i]))/(max(CD[,i])-min(CD[,i]))
> }
> rownames(CD) = CD$CountryCD
> hfit = hclust(dist(CD[,2:5]), "average")
> ...
```

Analysing the clusters

To compare the original (un-scaled) data:

> CD <- read.csv("CountriesData.csv") # reread unscaled

> as.table(by(CD$Per.capita.income, cut.hfit, mean))

```
       1        2        3        4        5
10958.67 35618.86   915.00  7924.50  3146.00
```

> cutCDhfit

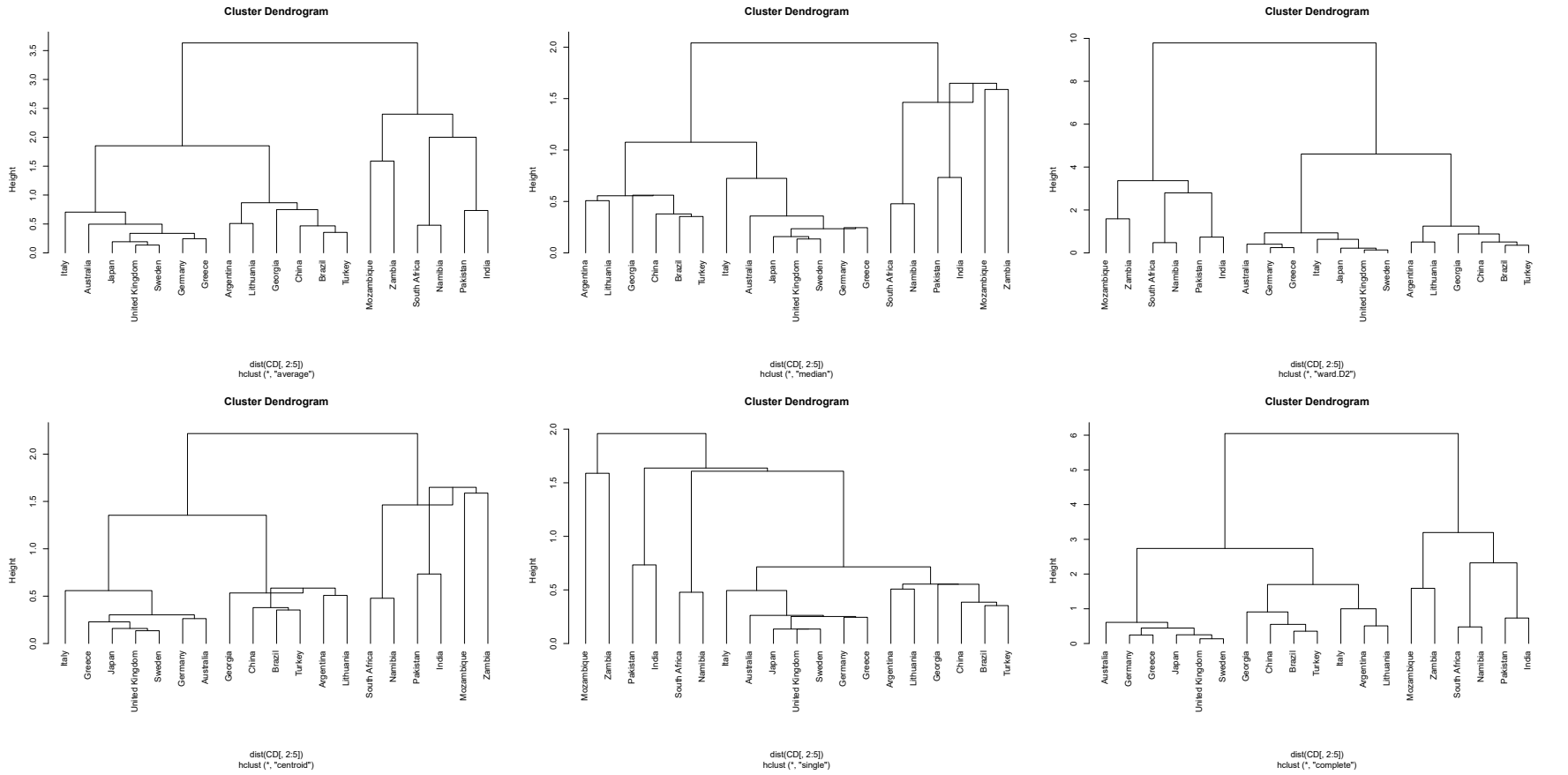
```
    Brazil    Germany Mozambique  Australia      China
         1          2          3          2          1
 Argentina United Kingdom South Africa     Zambia    Namibia
         1          2          4          3          4
   Georgia   Pakistan      India     Turkey     Sweden
         1          5          5          1          2
 Lithuania     Greece      Italy      Japan
         1          2          2          2
```

Effect of clustering rule

Different clustering rules give slightly different results. See:

```
> CD <- read.csv("CountriesData.csv")
> CD[,2:5] = scale(CD[,2:5]) # scaled; rownames(CD) = CD$Country
> pdf("H Cluster methods.pdf", width=20, height=10)
> par(mfrow = c(2, 3))
> hfit = hclust(dist(CD[,2:5]), "average"); plot(hfit, hang = -1)
> hfit = hclust(dist(CD[,2:5]), "median"); plot(hfit, hang = -1)
> hfit = hclust(dist(CD[,2:5]), "ward.D2"); plot(hfit, hang = -1)
> hfit = hclust(dist(CD[,2:5]), "centroid"); plot(hfit, hang = -1)
> hfit = hclust(dist(CD[,2:5]), "single"); plot(hfit, hang = -1)
> hfit = hclust(dist(CD[,2:5]), "complete"); plot(hfit, hang = -1)
> dev.off()
```

Effect of clustering rule



Closing remarks

Clustering:

- An important unsupervised learning tool for grouping data.
- Enables data reduction, by identifying representative subsets of the data.
- You can experiment with different clustering rules.

Many R packages for cluster analysis:

- Cluster – is one of these which gives more control over clustering algorithm and more analysis tools.

References to this lecture

- James et al., An Introduction to Statistical Learning with Applications in R, 2nd Ed. Springer, 2021. Section 12.4.
- Giordani, Ferraro and Martella, An Introduction to Clustering with R. Springer, 2020.
- Everitt, Landau, Leese and Stah, Cluster Analysis, 5th Edition, John Wiley, 2011.

Notes on the presentation

This presentation contains slides created to accompany: *Introduction to Data Mining*, Tan, Steinbach, Kumar. Pearson Education Inc., 2006.

Presentation contains some material originally created by Dr. Sue Bedingfield, with additions by Rui Jie Chow & Dr. Parthan Kasarapu.