wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

**Dr Wenjuan Zhang**

**Associate Professor in OR & Applied Statistics**

# IB98D
# Advanced Data Analysis

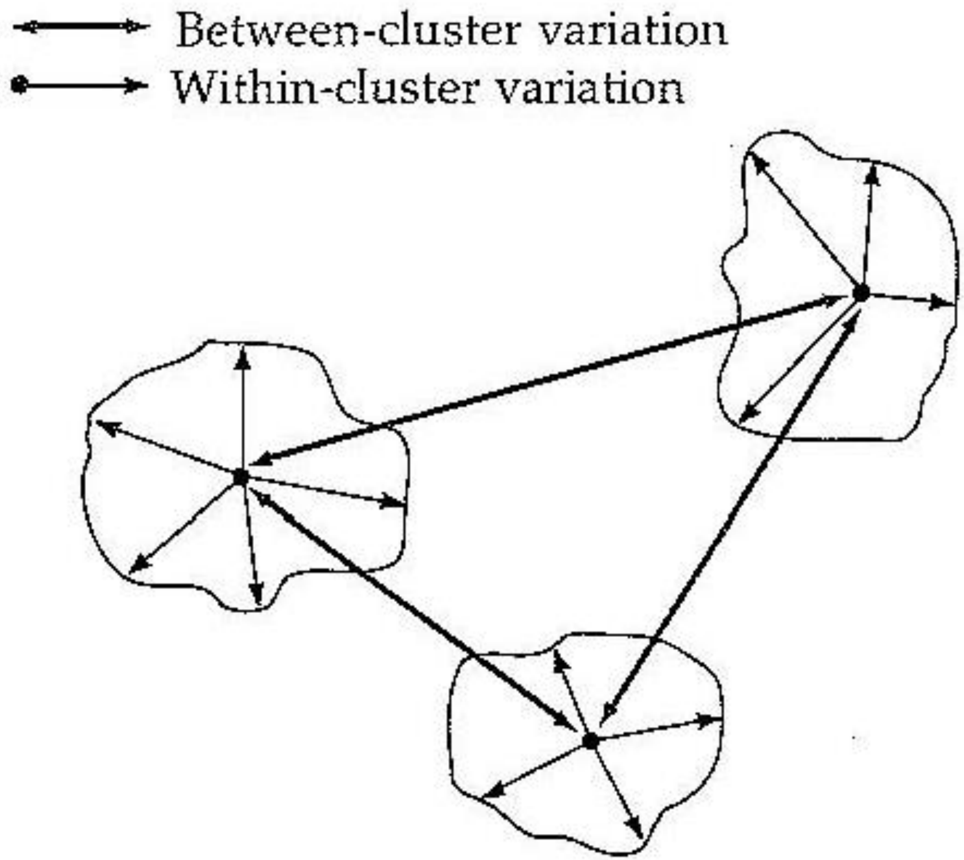
# Cluster Analysis

# Cluster Analysis

- Interdependence technique

- The purpose of a cluster analysis is to group objects based on their characteristics
  - Objects = cases = observations (e.g. individuals, firms, countries, products, behaviours,…)
  - If grouping variables then we use factor analysis

- Cluster analysis groups objects into clusters such that objects in the same cluster are more similar to each other than they are to objects in other clusters
  - Minimise within cluster variation
  - Maximise between cluster variation




Cluster Diagram source: Hair et al., Prentice Hall.

# Cluster Analysis

- The classification is suggested by natural groupings in the sample data, hence aims to identify existing groups within the population.

- Cluster analysis uses distances (between points) to group objects.

- Cluster analysis is used in many different disciplines
  - Target marketing (Business), Classification of living organisms (Biology), Analysis of psychiatric profiles (Psychology), etc…

(Often called by different names: Q analysis, typology construction, classification analysis, numerical taxonomy,..)

# The Clustering Process

- Need to tackle 3 basic questions:
  - How do we measure similarity?
  - How do we form clusters?
  - How many clusters (groups) do we form?

| | |
|---|---|
| Define the Problem | Objectives? Aim?<br>Select cluster variables. |
| Make pre-analysis decisions | Sample size? Outliers? |
| Check assumptions | Is the sample representative of the population? Multicollinearity? |
| Create clusters | Standardize data?<br>Select clustering method &<br>Similarity measure. |
| Comparing results & choosing solution | Choose final cluster solution.<br>Interpret the clusters. |
| Validate & Profile cluster solution | Is the cluster solution stable?<br>Does it represent the population?<br>What does it tell me? |

# Define the Problem (1)

**What are my objectives? Why am I doing this!?**

**Cluster Analysis can be used to explore these main objectives:**

- Data reduction: simplify the data
  - Analyse groups rather than individual observations
- To identify relationships
  - Reveal structure and relationships in data
- To define an empirically based classification or to confirm a theoretically based classification.

# Define the Problem (2)

**Which variables (characteristics) should I use as cluster variables?**

Selected variables must

(a) characterize the objects being clustered

(b) relate directly to the objectives of the cluster analysis

NB. Cluster analysis cannot distinguish between relevant and irrelevant variables – so <u>you</u> must !

- Use judgement, theory, past research knowledge, etc.…when selecting clustering variables.
- During cluster analysis eliminate any variables that are not distinctive (do not differ much between clusters).

# Example: Children's intelligence

- The Wechsler Intelligence Scale for Children combines the scores from a number of different subtests:
  - Information (info)
  - Comprehension (comp)
  - Arithmetic (arith)
  - Similarities (simil)
  - Vocabulary (vocab)
  - Digit Span (digit)
  - Picture Completion (pitcomp)
  - Paragraph Arrangement (parang)
  - Block Design (block)
  - Object Assembly (object)
  - Coding (coding)
- The data set from Tabachnick & Fidell (1996) contains 175 observations with values for each of the 11 tests plus age and subject number

# Make pre-analysis decisions (1)

**Is the sample size adequate?**

- What group size is relevant for the questions being tackled by the analysis?
- Is it important to have sufficient representation of small groups?
  - Related to questions of outliers:
    Are they outliers or representative of a meaningful small group?

# Make pre-analysis decisions (2)

**Are there outliers in the data?**

- Outlier: Case with a **unique combination of characteristics (variable values)** making it **distinctly different** from other cases
- Cluster analysis is sensitive to outliers.
- Outliers can represent:
  - Non-representative observations
    - Problematic as they distort analysis - remove
  - Small or insignificant segments
    - Remove
  - Under-sampling of groups
    - Keep

# Outliers can be identified using (combinations of)

- Univariate detection methods
  - Graphical methods (Box plots, individual value plots,…)
  - Calculate standardized values (z scores): Subtract mean & divide by standard deviation (for each variable).

    Rule of thumb – possible outlier if |z-score| is > 2.5 for small samples, or > 4 for large samples

- Multivariate detection methods
  - Graphical methods (profile plots, scatter plots,…)
  - Mahalanobi's $D^2$ : Measures each case's distance in multidimensional space from the mean centre of all cases. Rule of Thumb – possible outliers if $D^2$/ (no. of variables involved) > 2.5 for small samples and > 4 for large samples

- Outliers may also become apparent through measures of similarities – large distances from all other objects......
- and the cluster analysis itself – single-member or small clusters

# Multi dimension identification of outliers – Mahalanobis distance

```
Maha <- mahalanobis(Intelligence,colMeans(Intelligence),cov(Intelligence))
print(Maha)
```

```
##   [1] 12.668886 12.149312 17.057589 14.169191 17.855060 17.103911  9.699832
##   [8] 11.921506  5.667787  5.706725 11.487460  9.695600 17.922046 16.486924
##  [15]  8.836291 14.517783 11.573951  6.964889 11.376192 15.917549  6.494665
##  [22] 17.023135 14.144248  6.877761  9.808713  9.388069 12.560736 12.180952
##  [29] 15.269696 14.499420 11.603871 10.343768  4.631166  6.867921 13.383374
##  [36] 23.830990  9.017185 21.125027  7.788939  3.067359  5.949061 17.252544
##  [43] 16.370716 11.712378 22.551796  8.830683 10.413639 14.113851 14.634875
##  [50]  7.193981  9.854201  7.802749  7.678914  6.897980 11.550946  8.582619
##  [57] 11.490353 11.764988 13.225221  4.874921  3.324658  9.739868  9.373887
##  [64] 12.555988  7.532015 27.357509  8.030235  8.152272  6.397183 17.659943
##  [71] 14.195502 10.930442 24.892896  4.934785 12.298897 27.804045 12.731436
##  [78]  8.075169  5.950763  9.923560 15.195444 11.026661  8.861127  8.171114
```

# The p value for each Mahalanobis distance

```
MahaPvalue <-pchisq(Maha,df=10,lower.tail = FALSE)
print(MahaPvalue)
```

In general, a p-value that is less than 0.001 is considered to be an outlier.

```
##   [1] 0.242778380 0.275187927 0.073099351 0.165413215 0.057458356 0.072096079
##   [7] 0.467210926 0.290344207 0.842359524 0.839271708 0.320821214 0.467592912
##  [13] 0.056292939 0.086516373 0.547707533 0.150656445 0.314581407 0.728755409
##  [19] 0.328969819 0.102017107 0.772134129 0.073853617 0.166513117 0.736932468
##  [25] 0.457432918 0.495712403 0.249278482 0.273128619 0.122534210 0.151405632
##  [31] 0.312442124 0.410868499 0.914417146 0.737852384 0.203019643 0.008061856
##  [37] 0.530473446 0.020238417 0.649444413 0.979794094 0.819524559 0.068959298
##  [43] 0.089500651 0.304767120 0.012527512 0.548244388 0.404982808 0.167861554
##  [49] 0.145951499 0.707013818 0.453376596 0.648097157 0.660167401 0.735039902
##  [55] 0.316233053 0.572118363 0.320611191 0.301092875 0.211350687 0.899375140
##  [61] 0.972719339 0.463604451 0.497025602 0.249566759 0.674444118 0.002285680
##  [67] 0.625883489 0.613966053 0.780863270 0.060978100 0.164259489 0.362962889
```

# Check assumptions (1)

- Assumptions are more mathematical than statistical.
- Linearity, normality and homoscedasticity **not** important.

**Is the sample representative of the population?**

- Satisfy yourself that all relevant groups are sufficiently sampled
  - Enough observations in each group

# Check assumptions (2)

**Is there substantial multicollinearity in the clustering variables?**

- Multicollinearity = extent to which a variable can be explained by the other variables in the analysis.
- If multicolinearity present, the correlated variables affect the clusters the most.

- Check for...
  - High correlations (>0.8) between variables
  - Variance Inflation Factor (VIF): indicates whether a variable has a strong linear relationship with other variables. Rule of thumb: values of around 10 + are good indications of multicollinearity.
  - Tolerance statistic = 1/(VIF): less than 0.1 indicates a serious problem.
- If substantial multicollinearity is present:
  - Use a set of cluster variables that are not highly correlated with one another (i.e. drop problematic variables).
  - Use Mahalanobi's distance measure.
  - Do Cluster Analysis on Principal Components/Factors instead of the original variables.

# Correlation matrix

```
lowerCor(Intelligence)
```

```
##          info  comp  arith simil vocab digit pctcm parng block objct codng
## info     1.00
## comp     0.47  1.00
## arith    0.49  0.39  1.00
## simil    0.51  0.51  0.37  1.00
## vocab    0.63  0.53  0.39  0.54  1.00
## digit    0.35  0.24  0.27  0.26  0.29  1.00
## pictcomp 0.23  0.41  0.16  0.37  0.29  0.08  1.00
## parang   0.20  0.19  0.23  0.30  0.13  0.15  0.25  1.00
## block    0.23  0.37  0.27  0.26  0.30  0.07  0.38  0.35  1.00
## object   0.18  0.32  0.04  0.27  0.19  0.03  0.36  0.25  0.40  1.00
## coding   0.01  0.06  0.09 -0.04  0.10  0.17 -0.07  0.04  0.11  0.05  1.00
```

# Standardise the data?

**Should you standardize the data before calculating the similarities?**

- Most of the distance measures are sensitive to differences in scales/magnitudes between variables
  - Larger st.deviation → more impact on similarity value

- Solution:
  - Use Mahalanobi's distance
  - Or standardise variable values
    - Subtract mean and divide by standard deviation (z-scores)

# Descriptive statistics

```
describe(Intelligence)
```

```
##          vars   n  mean   sd median trimmed  mad min max range  skew kurtosis
## info        1 175  9.50 2.91     10    9.50 2.97   3  19    16  0.08    -0.08
## comp        2 175 10.00 2.97     10    9.95 2.97   0  18    18  0.09     0.33
## arith       3 175  9.00 2.31      9    8.89 2.97   4  16    12  0.39    -0.18
## simil       4 175 10.61 3.18     11   10.62 2.97   2  18    16  0.02    -0.23
## vocab       5 175 10.70 2.93     10   10.61 2.97   2  19    17  0.27     0.29
## digit       6 175  8.73 2.70      8    8.65 1.48   0  16    16  0.27     0.07
## pictcomp    7 175 10.68 2.93     11   10.70 2.97   2  19    17 -0.07     0.29
## parang      8 175 10.37 2.66     10   10.43 2.97   2  17    15 -0.20    -0.06
## block       9 175 10.31 2.71     10   10.36 2.97   2  18    16 -0.22     0.50
## object     10 175 10.90 2.84     11   10.94 2.97   3  19    16 -0.12     0.15
## coding     11 175  8.55 2.87      9    8.55 2.97   0  15    15 -0.05    -0.45
```

# Choose similarity measure

**How should we measure the similarity between objects?**

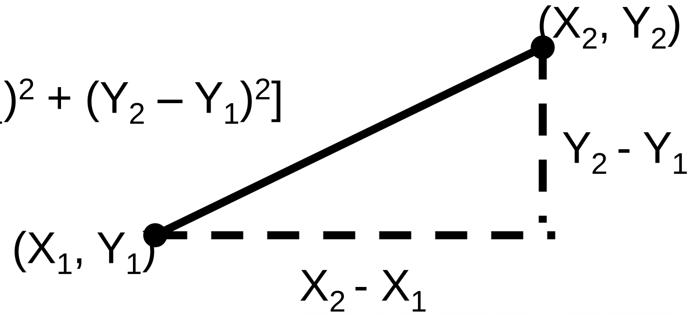
- Cluster analysis uses a distance measure
  - many different distance measures

Examples:

- Euclidean distance: Square root of the sum of the squared differences between the values for two cases (i.e. straight line distance).

$$\sqrt{[(X_2 - X_1)^2 + (Y_2 - Y_1)^2]}$$

e.g. for two cases
with just two variables



- Euclidean distance: straight-line distance

- Squared (or absolute) Euclidean distance: Sum of the squared differences between the values for two cases. (recommended for use with the centroid and Ward's methods of clustering)

- Chebychev: The maximum absolute difference between the values for the two cases.

- Block or Manhattan distance: The sum of the absolute differences between the values of the two cases.

- Mahalanobis distance:  Accounts for correlation among variables.  Standardises data.

- Using different distance measures can give different results
  - Try different ones and compare results
  - Suggest: Experiment with different combinations of similarity measures and linkages (cluster methods).

- Note that we only consider metric variables here
  - Non-metric variables (nominal or ordinal) need different measures called measures of association

# Create clusters

**Select Clustering Algorithm:**

a) **Hierarchical** – a step procedure that combines (or divides) the objects producing N -1 possible cluster solutions (where N = number of objects).

b) **Nonhierarchical** – Number of clusters set by analyst, therefore produces a single cluster solution.

c) **Combination of both.**

# Create clusters (1)

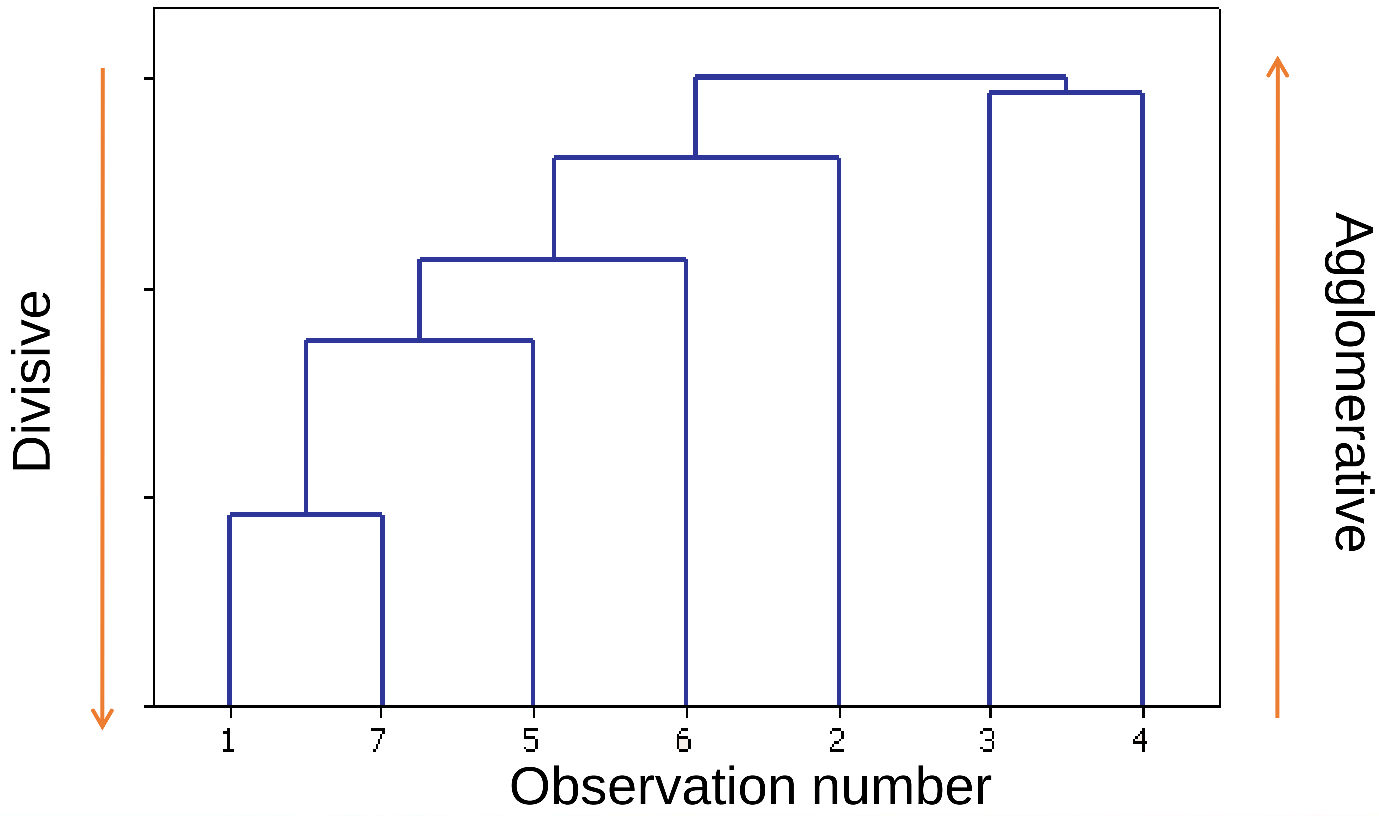
**a) Hierarchical procedures:**

- Agglomerative methods – Each object starts out as its own cluster; at each step of the procedure the two most similar clusters are combined into one cluster, until all objects belong to one large cluster.

- Divisive methods – Opposite to Agglomerative methods. All objects start out in one large cluster; at each step of the procedure the clusters are divided to produce 2, 3, 4 etc.. separate clusters until each object is on its own in a cluster.

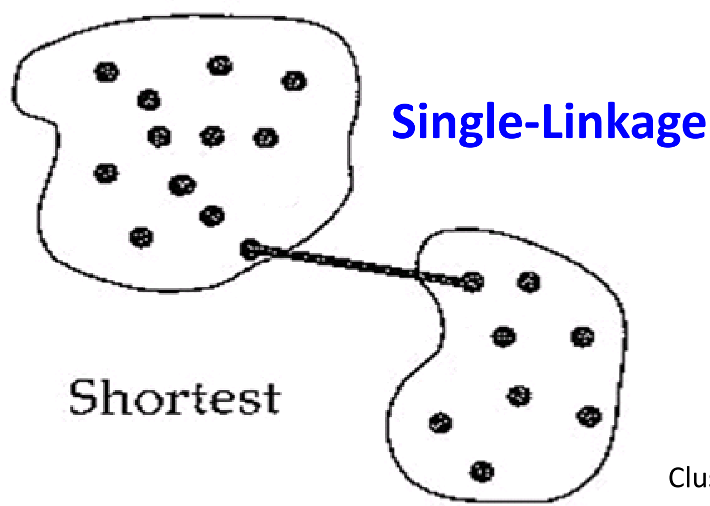
# Dendrogram

- From Greek:
- *"Dendron"* = tree
- "G*ramma"* = drawing



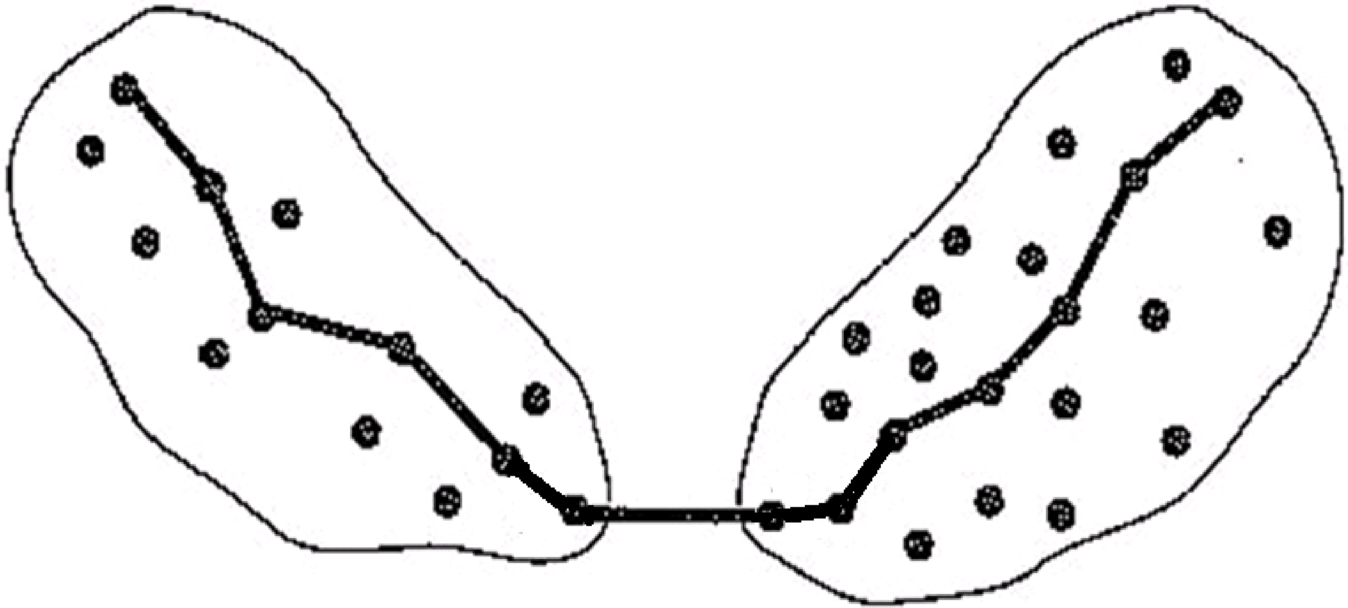
## Agglomerative methods for combining clusters.

- **Single-Linkage** (or nearest-neighbour) method – combines the 2 clusters that have the minimum similarity value (shortest distance) from any object in one cluster to any object in the other cluster.
  - Can form undesirable long chain–like clusters.



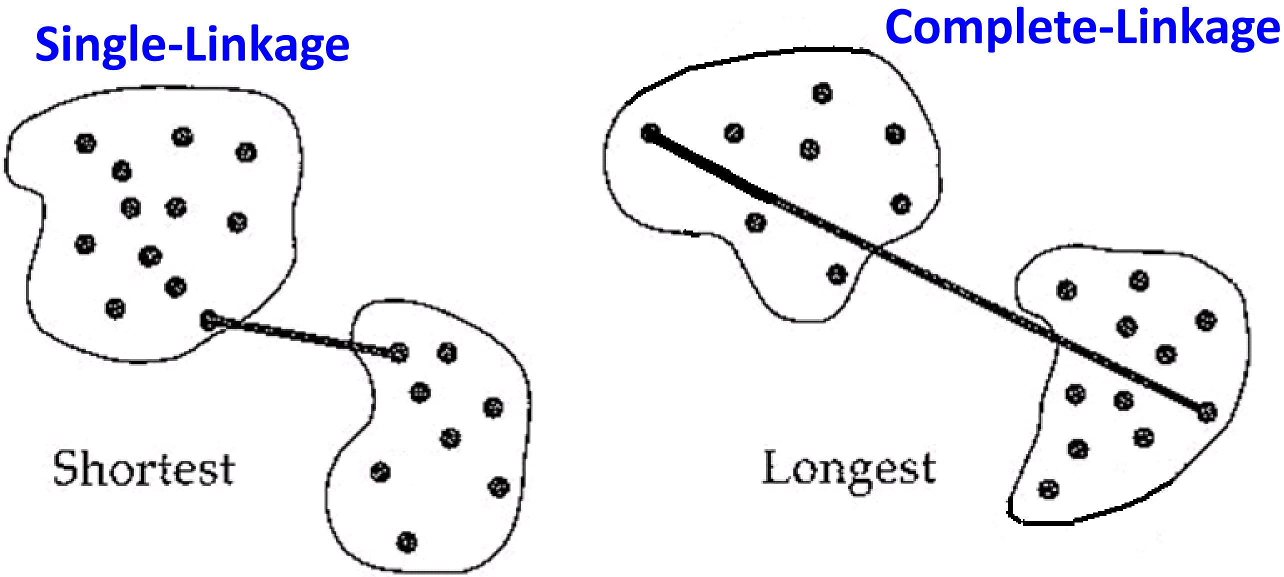

Cluster Diagram source: Hair et al., Prentice Hall.

Example of single linkage joining dissimilar points and therefore forming an undesirable chain.



Cluster Diagram source: Hair et al., Prentice Hall.

- **Complete-Linkage** (or farthest-neighbour or diameter) method – combines the 2 clusters that have the smallest sphere (minimum diameter) that can enclose both clusters.
  - Tends to create good compact clusters.




Cluster Diagram source: Hair et al., Prentice Hall.

- **Average-Linkage** method – combines the 2 clusters that have the minimum average similarity of all objects in one cluster with all objects in the other cluster.
  - Tend to produce clusters with small within cluster variation.

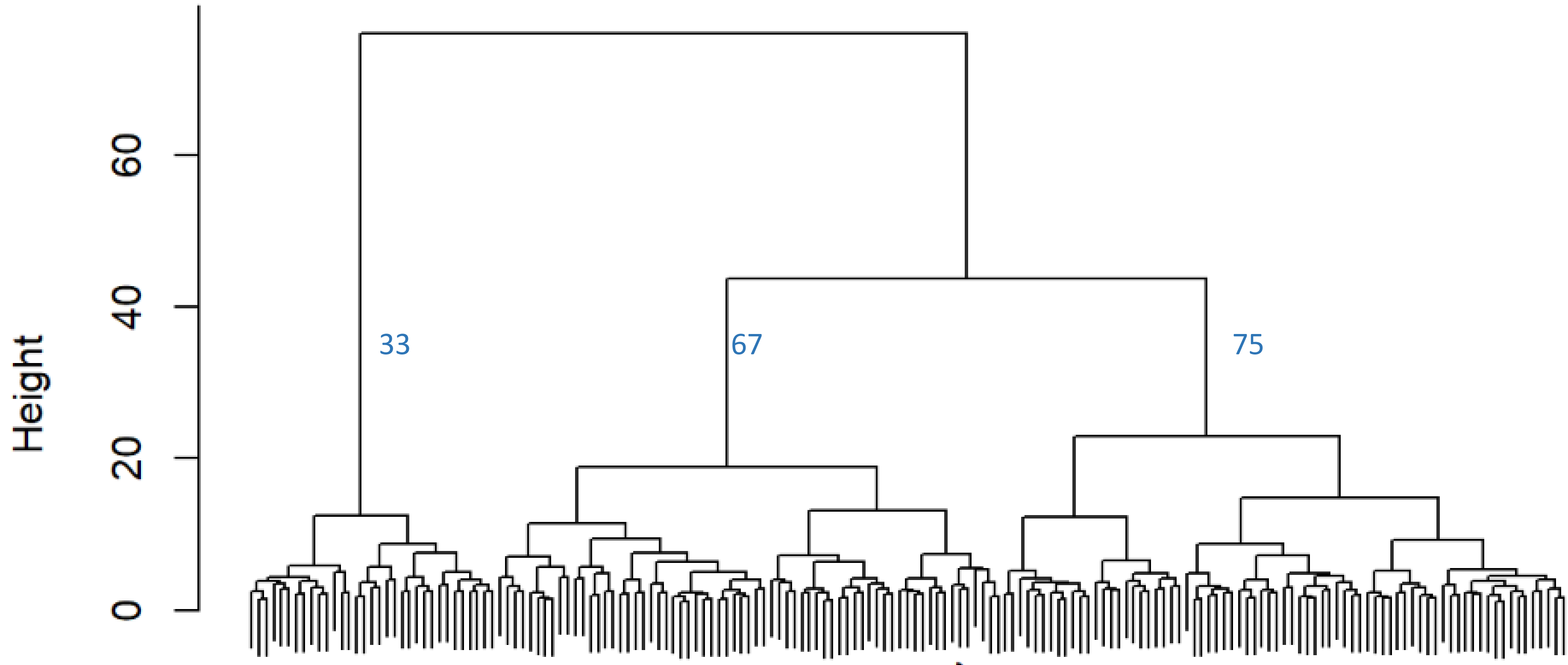
- **Centroid** method – combines the 2 clusters that have the minimum distance between their centroids.
  - Centroid = mean value of cluster variables for all objects in the cluster.
  - Less affected by outliers than other methods.

- **Ward's** method – looks at the sum of squares within the clusters, and combines the 2 clusters that minimizes the increase in the total sum of squares across all variables
  - Badly effected by outliers.
  - Tends to produce clusters with approximately the same number of objects in them.

# Comparing results & choosing solution

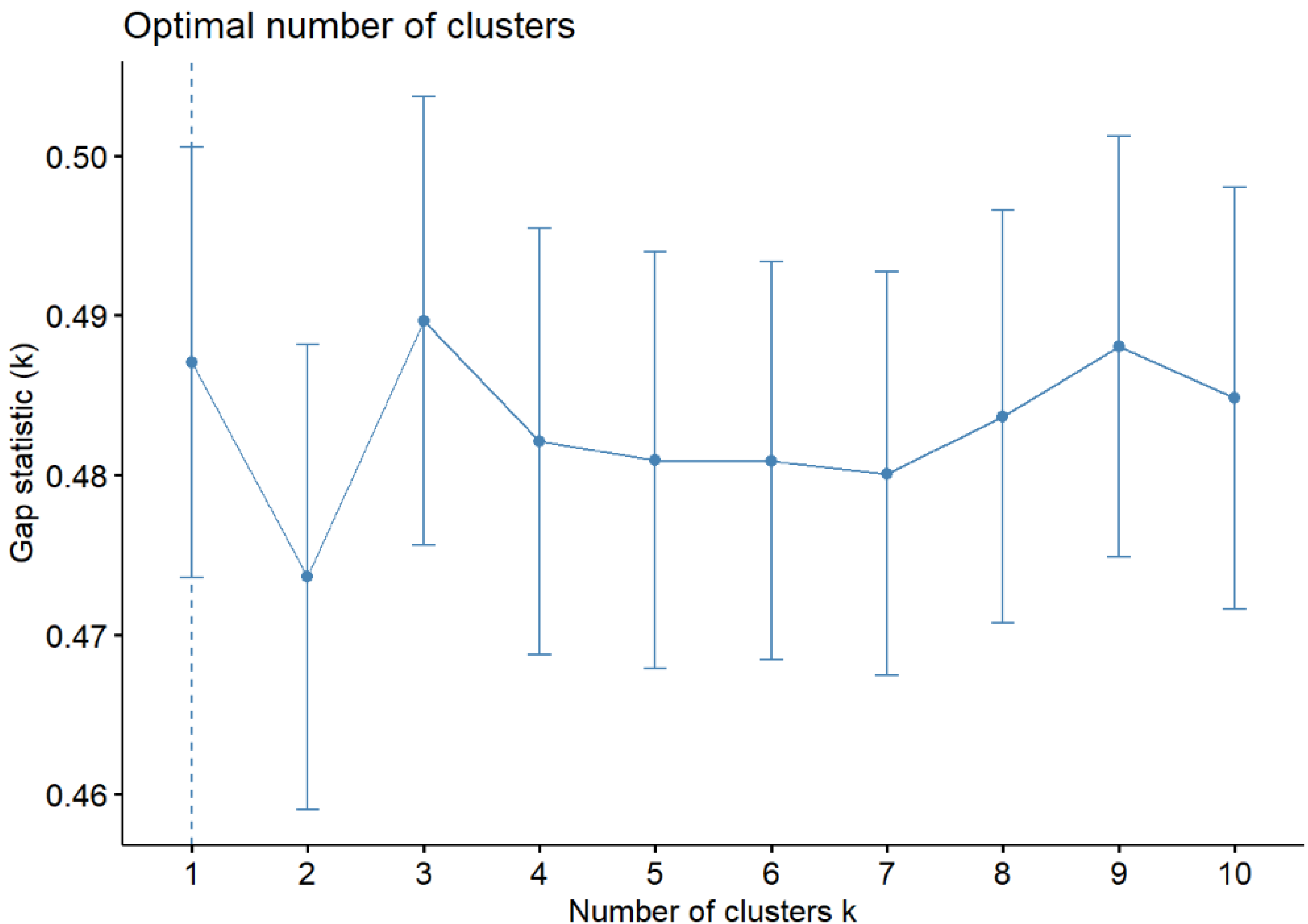
- **Check structure of cluster solutions:**
  - Small (one or two-object clusters) not ideal
  - Ideally not extreme differences in cluster sizes
  - Possibly delete outliers and re-run

# Cluster Dendrogram



# Comparing results & choosing solution

- **How many clusters** (re. Hierarchical method)**?**
  - Stopping rules ad-hoc rather than standard
  - Consider:
    - Changes in heterogeneity – (i.e. How different the objects in a cluster are from one another) – when a large increase occurs select the prior cluster solution.
    - Interpretation of clusters
      - Practical considerations limiting cluster number
      - Conceptual aspects
      - Practical significance


Optimal number of clusters
Gap statistic (k)
0.50
0.49
0.48
0.47
0.46
1
2
3
4
5
6
7
8
9
10
Number of clusters k

# Cluster centroids

```
hcentres<-aggregate(x=final_data, by=list(cluster=fit), FUN="mean")
print(hcentres)
```

```
##   cluster        info        comp      arith        simil        vocab
## 1       1 -0.50895615 -0.58386453 -0.1682161 -0.6233647808 -0.565372870
## 2       2 -0.03793474 -0.01798571 -0.2947664  0.0005982987  0.005845376
## 3       3  1.11955054  1.22629855  1.0114535  1.2642596336  1.134593307
##         digit   pictcomp     parang      block     object     coding cluster
## 1 -0.17113295 -0.6386805 -0.6054244 -0.6171951 -0.6900752 -0.2325720       1
## 2 -0.06339425  0.1681309  0.2313204  0.2186044  0.4701664  0.3800082       2
## 3  0.49152960  0.9145991  0.7034668  0.7562650  0.3325018 -0.3914633       3
```

# Create clusters (2)

**b) Non-Hierarchical procedures:**

- Assigns objects to clusters once the number of clusters (k) have been specified by the analyst.
- Usually a two stage process:
  - Set starting points (cluster seeds) for each cluster.
  - Assign each object to one of the cluster seeds based on similarity.
- Objects can swap between clusters until best k-cluster solution is found.

# How do you select the seed points?

- Specified by Analyst, based on:
  - Previous research/knowledge.
  - Previous multivariate analysis e.g. Hierarchical clustering.

## How to assign objects to clusters?

- Many algorithms usually known as K-means algorithms.

# Non-hierarchical clustering - Kmeans

```
set.seed(55)
k_cl <- kmeans(Intelligence,3,nstart=25)
k_cl
```

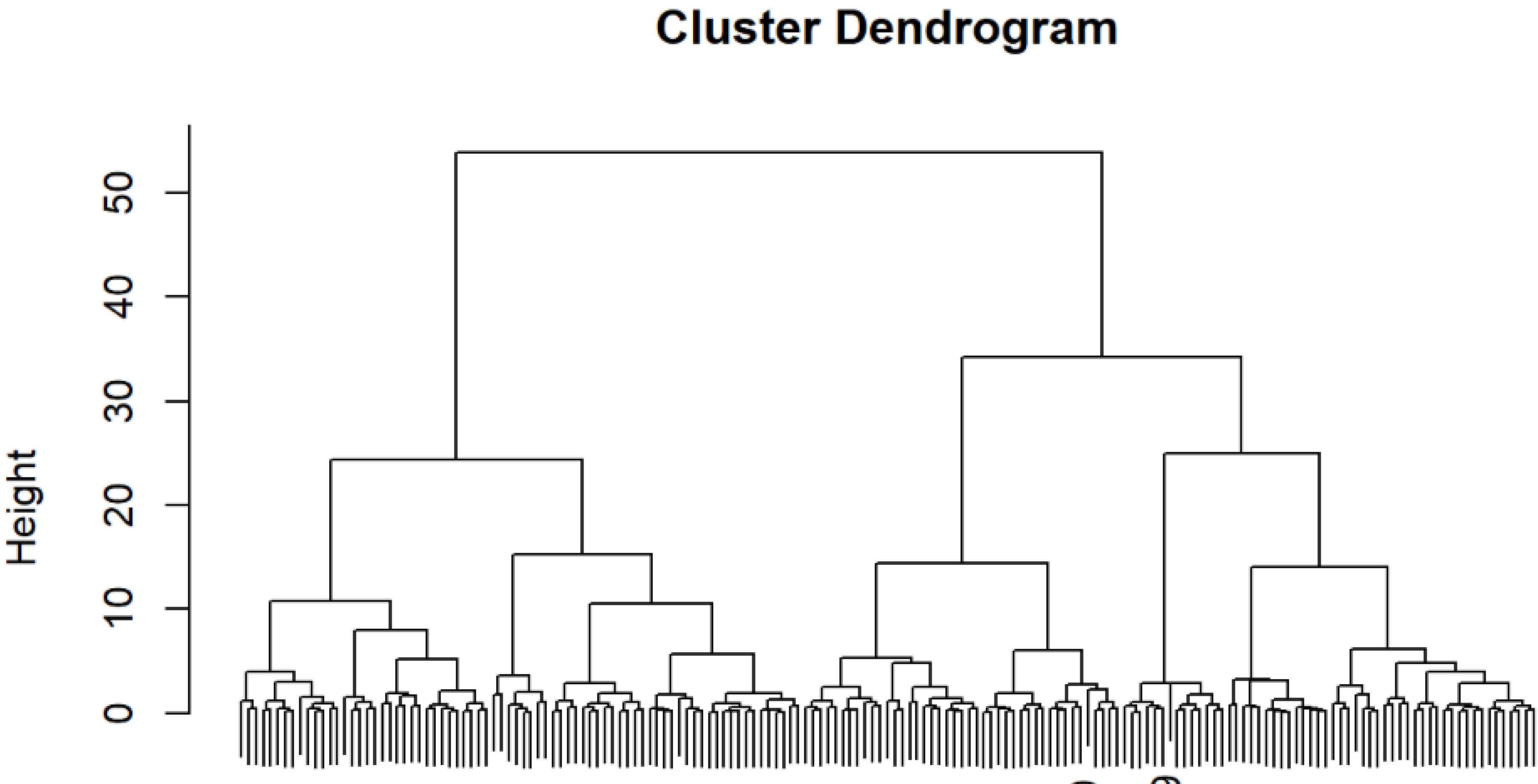
```
## K-means clustering with 3 clusters of sizes 59, 48, 68
##
## Cluster means:
##         info        comp      arith      simil      vocab      digit   pictcomp
## 1 -0.5664604 -0.69161160 -0.2644963 -0.6392569 -0.6442132 -0.1701975 -0.7516214
## 2  0.9953373  0.96954229  0.8218060  0.9858468  1.0390155  0.7002424  0.6628675
## 3 -0.2111034 -0.08430803 -0.3506089 -0.1412425 -0.1744730 -0.3466174  0.1842357
##       parang      block     object      coding
## 1 -0.5793632 -0.7226847 -0.8955502 -0.22050525
## 2  0.5574876  0.4683124  0.3198488  0.14992252
## 3  0.1091621  0.2964618  0.5512459  0.08549307
```

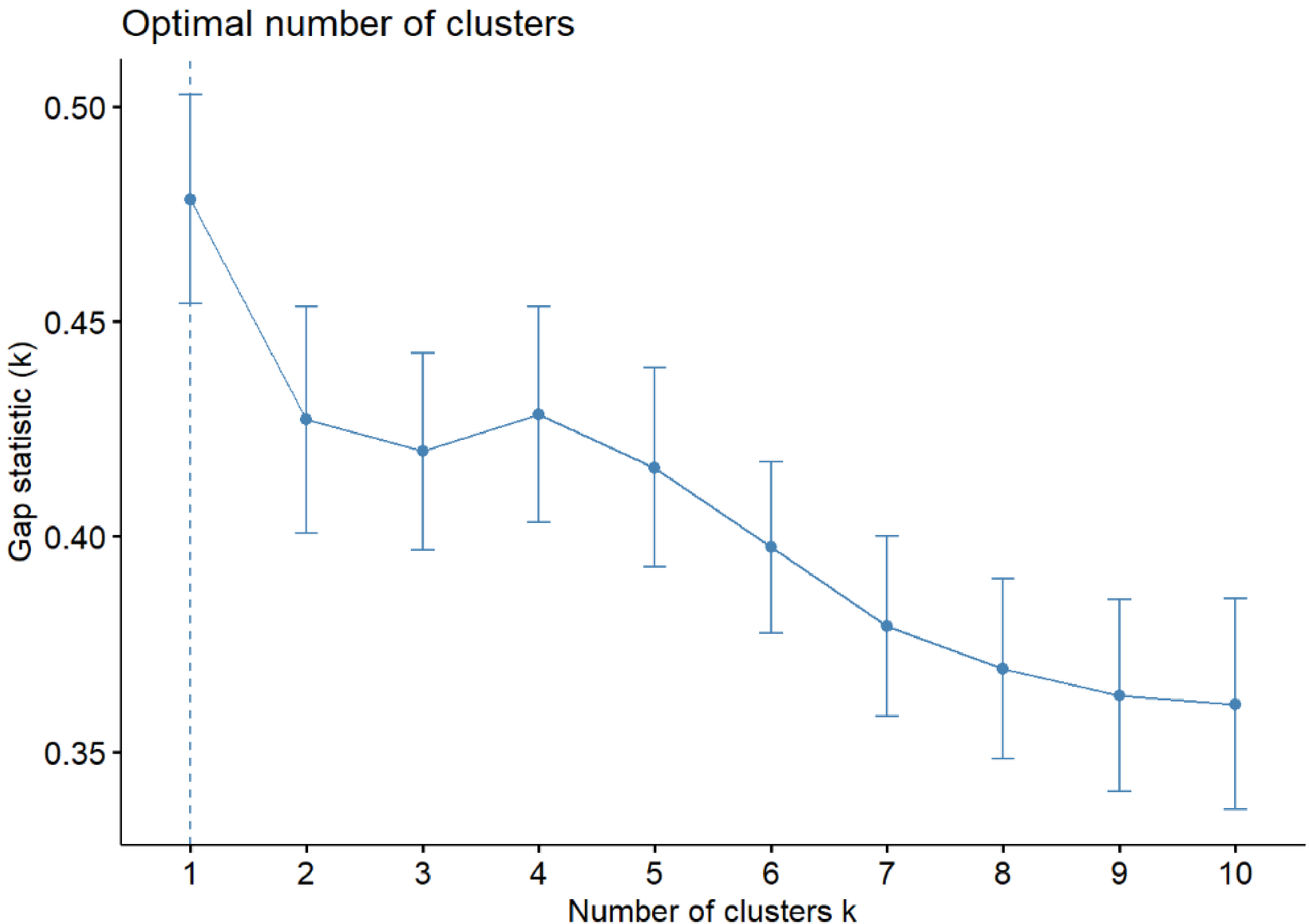
# Non-hierarchical clustering - Kmeans

```
set.seed(55)
k_cl <- kmeans(Intelligence,3,nstart=25)
k_cl
```

```
## K-means clustering with 3 clusters of sizes 59, 48, 68
##
## Cluster means:
##         info        comp      arith      simil      vocab      digit   pictcomp
## 1 -0.5664604 -0.69161160 -0.2644963 -0.6392569 -0.6442132 -0.1701975 -0.7516214
## 2  0.9953373  0.96954229  0.8218060  0.9858468  1.0390155  0.7002424  0.6628675
## 3 -0.2111034 -0.08430803 -0.3506089 -0.1412425 -0.1744730 -0.3466174  0.1842357
##       parang      block     object      coding
## 1 -0.5793632 -0.7226847 -0.8955502 -0.22050525
## 2  0.5574876  0.4683124  0.3198488  0.14992252
## 3  0.1091621  0.2964618  0.5512459  0.08549307
```

# Hierarchical clustering using factors




Optimal number of clusters
Gap statistic (k)
0.50
0.45
0.40
0.35
1
2
3
4
5
6
7
8
9
10
Number of clusters k

Find mean values for each cluster

```
hcentres<-aggregate(x=final_data, by=list(cluster=fit), FUN="mean")
print(hcentres)
```

```
##   cluster         RC1        RC2         RC3 cluster
## 1       1  0.08695974 -0.4866914  0.80592950       1
## 2       2  1.52516607  0.4369993 -1.24806471       2
## 3       3 -0.43207445 -0.3619036 -1.02758432       3
## 4       4 -0.22823577  1.0714071 -0.01439988       4
```

Kmeans clustering

```
set.seed(55)
k_cl <- kmeans(fscores,4,nstart=25)
k_cl
```

```
## K-means clustering with 4 clusters of sizes 35, 53, 40, 47
##
## Cluster means:
##          RC1        RC2        RC3
## 1  1.41301067  0.4608320 -0.3805837
## 2 -0.64388534  0.8141054  0.3763251
## 3 -0.46085098 -0.4263288 -1.0914388
## 4  0.06605508 -0.8983735  0.7879308
```

# Pros and cons of Hierarchical & Non-Hierarchical methods

## Hierarchical

- Pros
  - Simple & fast,
  - extensive development of similarity measures.
- Cons
  - undesirable early combinations may lead to misleading results,
  - generally sensitive to outliers & deletion of cases,
  - not good for analysing large samples or large numbers of variables (e.g. 500 cases requires storage of around 125,000 similarities!)

# Non-Hierarchical

- Pros
  - With well chosen seeds, the results are not very sensitive to outliers and analysis decisions.
  - Can easily analyse extremely large data sets.
- Cons
  - Using random seeds can produce inferior results to Hierarchical techniques.
  - Different seeds will usually give different clustering solutions.

# Create clusters (3)

**c) Combining Hierarchical & Non-Hierarchical procedures:**

i. Use a Hierarchical technique to decide on the appropriate number of clusters

ii. Use Non-Hierarchical technique to cluster the observations into that number of clusters

Combines the advantages of the Hierarchical technique with the Non-Hierarchical technique's ability to allow objects to move between clusters.

# Validate cluster solution

**Is the cluster solution representative of the general population and stable over time?**

- Ideally validate by running the cluster analysis on different (random) subsets of the total data set and comparing the results: (*Rule of Thumb*)
  - Very stable solution – less than 10% of cases assigned to different cluster
  - Stable solution – 10 to 20% assigned to different clusters
  - Fairly stable solution – 20 to 25% assigned to different clusters

# Profile cluster solution

- Describe the characteristics of each cluster to explain how they may differ from each other.
- Use data not previously used in the clustering procedure, e.g. Demographics, psychological profiles, purchase patterns etc...
- Look for practical importance that could predict membership in a particular cluster.
- Useful in management strategic decisions

# Cluster Analysis Summary

- Objectives & Aim?   Cluster variables?
- Sample size?   Outliers?
- Sample represents population?   Multicollinearity?
- Standardize data?

**Hierarchical clustering:**

- Clustering method & similarity measure?

**K-Means clustering:**

- Cluster number?  Initial cluster centres?

- Interpret cluster results – choose a solution.
- Stable solution?  Represents population?
  Management/policy implications?