Multivariate analysis

- Multiple regression
- Multiple/partial correlation
- **Cluster analysis**
- **Discriminant analysis**
- **Principal component analysis**
- **Factor analysis**
- **Correspondence analysis**
- **Redundancy analysis**
- **Canonical correspondence analysis**
- **Principal coordinate analysis (multidimensional scaling)**

History of ordination methods

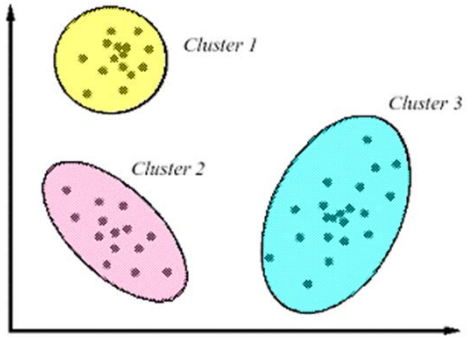
- In 1930, Ramensky began to use informal ordination techniques for vegetation. Such informal and largely subjective methods became widespread in the early 1950's (Whittaker 1967).
- In 1951, Curtis and McIntosh (1951) developed the 'continuum index', which later lead to conceptual links between species responses to gradients and multivariate methods. Shortly thereafter, Goodall (1954) introduced the term 'ordination' in an ecological context for Principal Components Analysis.
- Bray and Curtis (1957) developed polar ordination, which became the first widely-used ordination technique in ecology.
- Austin (1968) used canonical correlation to assess plant-environment relationships in what may have been the first example of multivariate direct gradient analysis in ecology.
- In 1973, Hill introduced correspondence analysis, a technique originating in the 1930's, to ecologists. Correspondence analysis gradually supplanted polar ordination, which today has few practitioners.
- Fasham (1977) and Prentice (1977) independently discovered and demonstrated the utility of Kruskal's (1964) nonmetric multidimensional scaling, originally intended as a psychometric technique, for community ecology.
- Hill (1979) corrected some of the flaws of Correspondence Analysis and thereby created Detrended Correspondence Analysis, which is the most widely used indirect gradient analysis technique today. The software to implement Detrended Correspondence Analysis, DECORANA, became the backbone of many later software packages.
- Gauch's (1982) book "Multivariate Analysis in Community Ecology" described ordination in non-technical terms to the average practitioner, and allowed ordination techniques to enter the mainstream.
- Fuzzy set theory, introduced to ecologists by Roberts (1986), is a promising approach with ties to polar ordination, but has yet to gain many adherents.
- Ter Braak (1986) ushered in the biggest modern revolution in ordination methods with Canonical Correspondence Analysis. This technique coupled Correspondence Analysis with regression methodologies, and provides for hypothesis testing.
- Ter Braak and Prentice (1988) developed a theoretical unification of ordination techniques, hence placing gradient analysis on a firm theoretical foundation.

From Wikipedia

Cluster analysis

What is cluster analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters



- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What is good clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on the similarity measure.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Measure the quality of clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, which is typically metric:

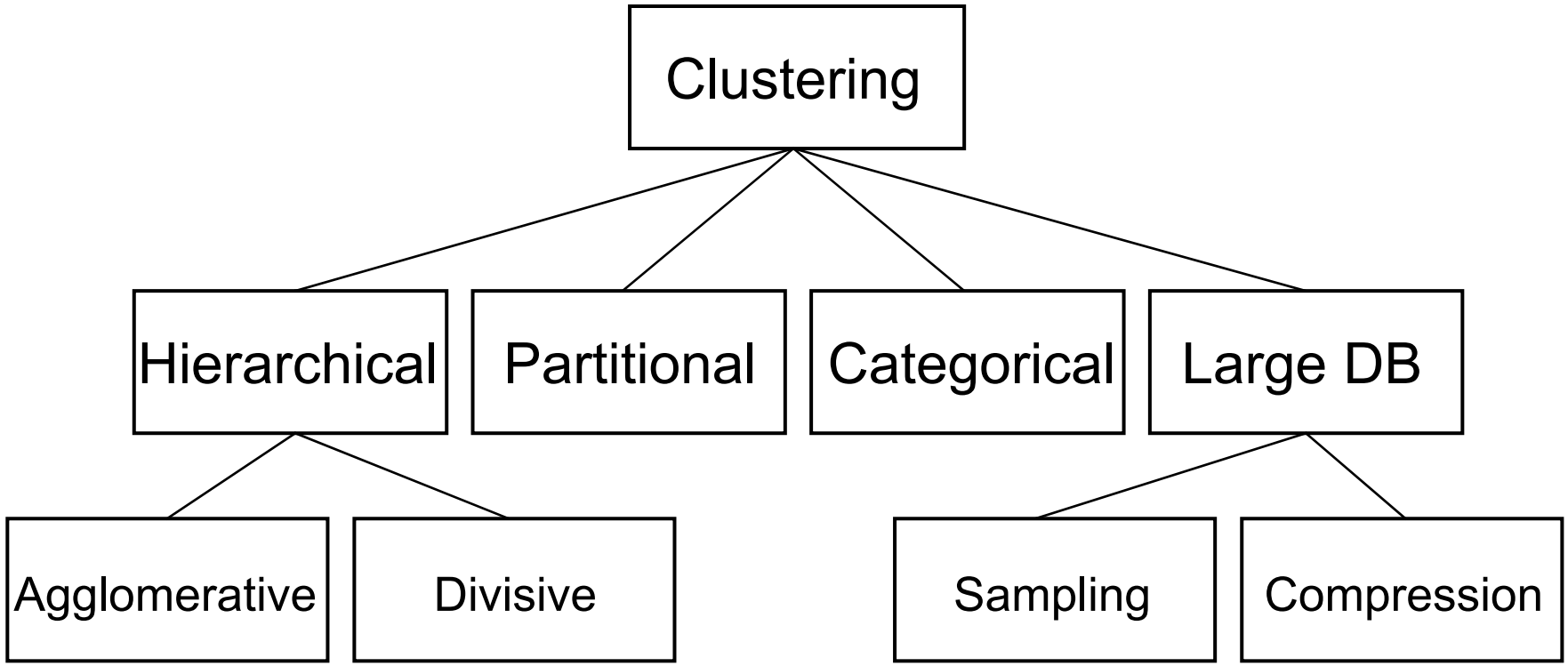
  $d(i, j)$

- The definitions of distance functions are usually very different for boolean, categorical, ordinal, interval-scaled, and ratio variables.

- Weights should be associated with different variables based on applications and data semantics.

- It is hard to define “similar enough” or “good enough”
  - the answer is typically highly subjective.

Clustering approaches



Data structures

- Data matrix

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Partitioning algorithms: basic concept

- Partitioning method: construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
    - *k-means* (MacQueen 1967): Each cluster is represented by the center of the cluster
    - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw 1987): Each cluster is represented by one of the objects in the cluster

K-Means clustering

- Basic ideas : using cluster centre (means) to represent cluster
- Assigning data elements to the closet cluster (centre).
- Goal: Minimise square error (intra-class dissimilarity):

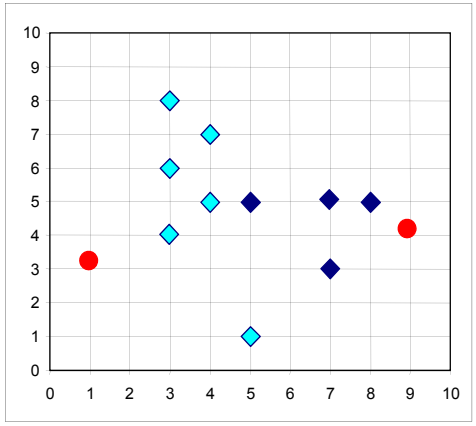
$$\sum_{i} d(\vec{x}_i, C(\vec{x}))^2$$

The K-Means clustering method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster) $\vec{C}(S) = \sum_{i=1}^{n} \vec{X}_i / n, \vec{X}_1, ..., \vec{X}_n \in S$
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment

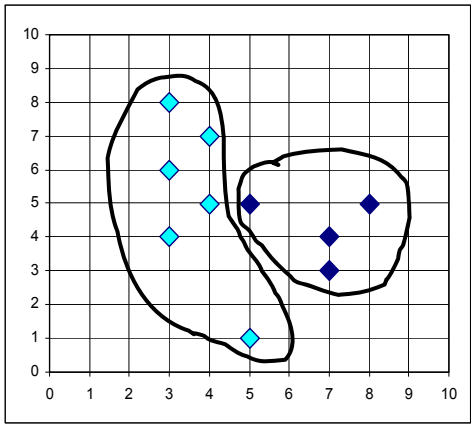
The K-Means clustering method

Example

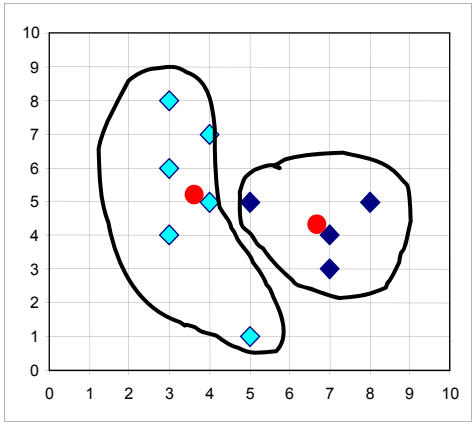


K=2

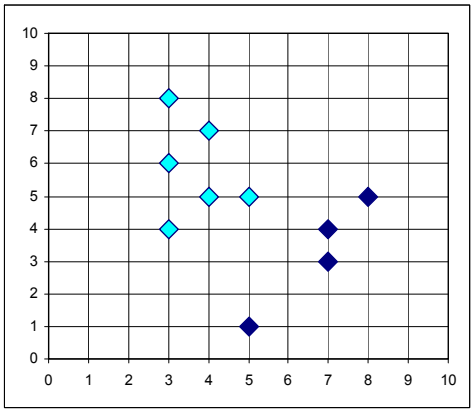
Arbitrarily choose K object as initial cluster center



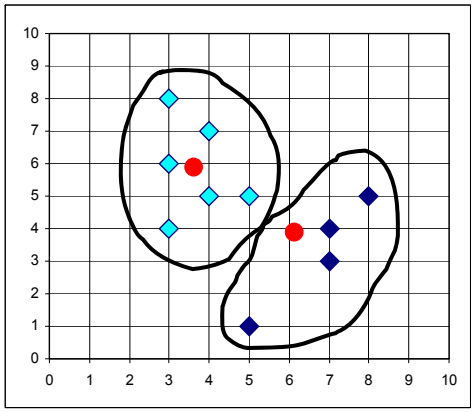
Assign each objects to most similar center



Update the cluster means



reassign



Update the cluster means

reassign

k-means Clustering : Procedure (1)

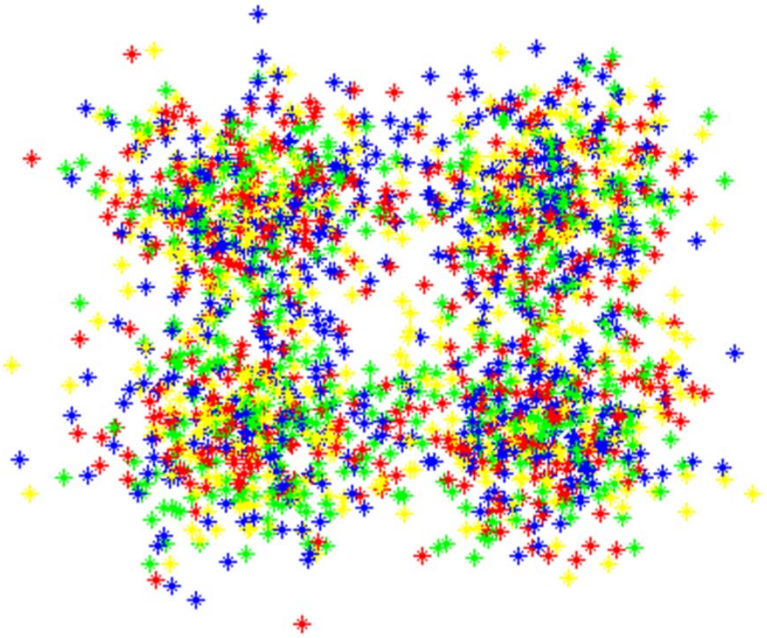
Initialization 1

Specify the number of cluster k :
for example, $k = 4$

k-means Clustering : Procedure (2)

Initialization 2

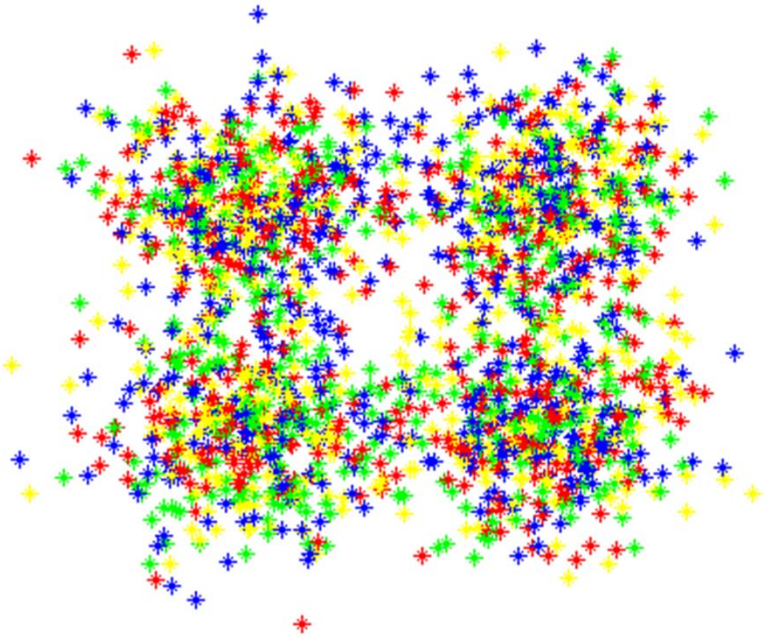
Points are randomly assigned to one of *k* clusters

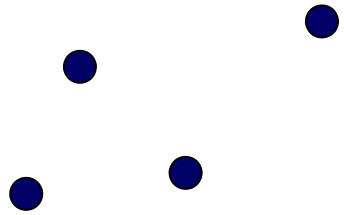


From Han and Kamber's slides on cluster analysis

k-means Clustering : Procedure (3)

Calculate the mean of each cluster

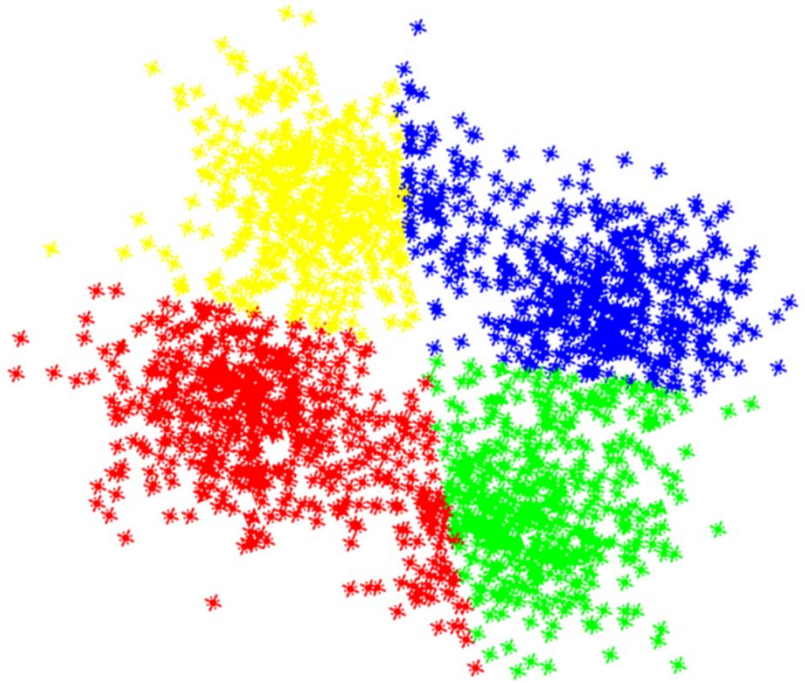




From Han and Kamber's slides on cluster analysis

k-means Clustering : Procedure (4)

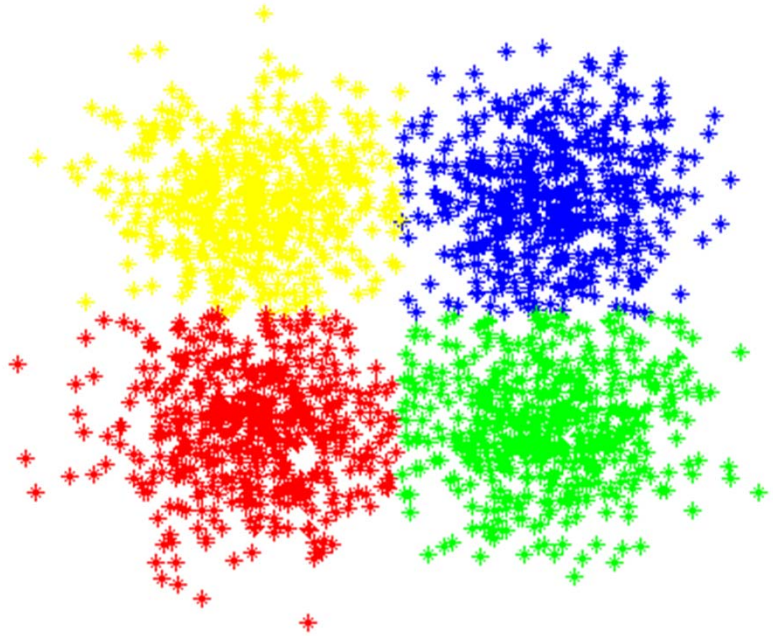
Each point is reassigned to the nearest cluster



From Han and Kamber's slides on cluster analysis

k-means Clustering : Procedure (5)

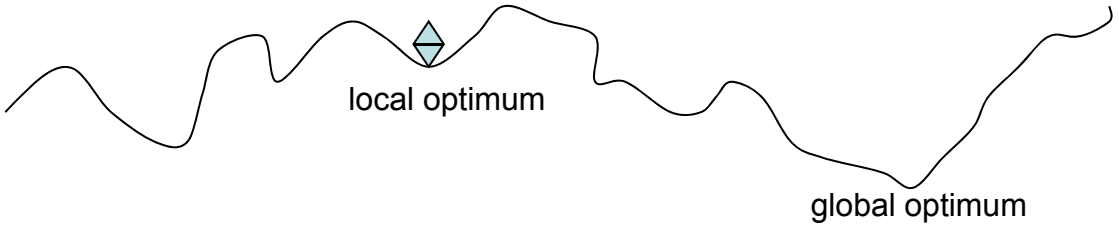
Iterate until the means are converged



From Han and Kamber's slides on cluster analysis

Comments on the K-Means Method

- Strength: *relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Comment: often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*



- Weakness
  - Applicable only when *mean* is defined, not applicable to categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the K-Means Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang 1998)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

The K-Medoids Clustering Method

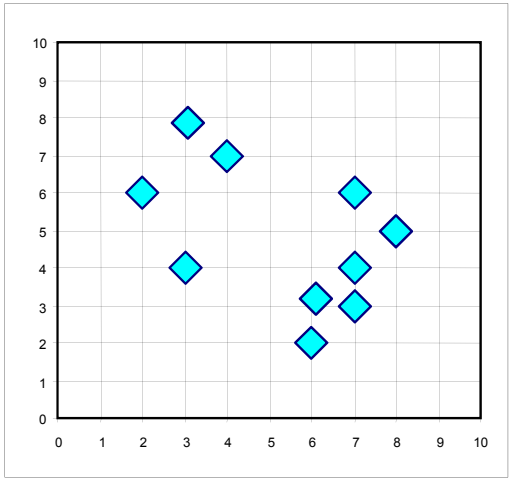
Find representative objects, called medoids, in clusters

- PAM (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - PAM works effectively for small data sets, but does not scale well for large data sets
- CLARA (Kaufmann & Rousseeuw, 1990)
- CLARANS (Ng & Han, 1994): randomized sampling

PAM (Partitioning Around Medoids) (1987)

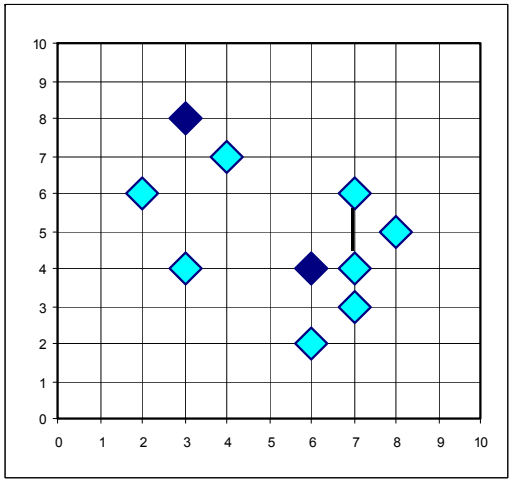
- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  1. Select ***k*** representative objects arbitrarily
  2. For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  3. For each pair of ***i*** and ***h***,
     - ✓If $TC_{ih} < 0$, ***i*** is replaced by ***h***
     - ✓Then assign each non-selected object to the most similar representative object
  4. repeat steps 2-3 until there is no change

Typical k-medoids algorithm (PAM)



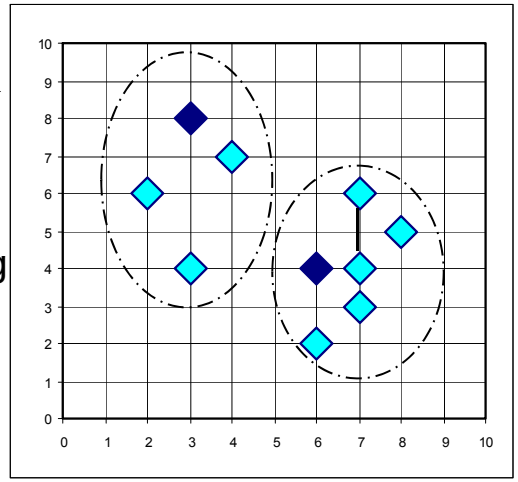
K=2

Arbitrary choose k object as initial medoids

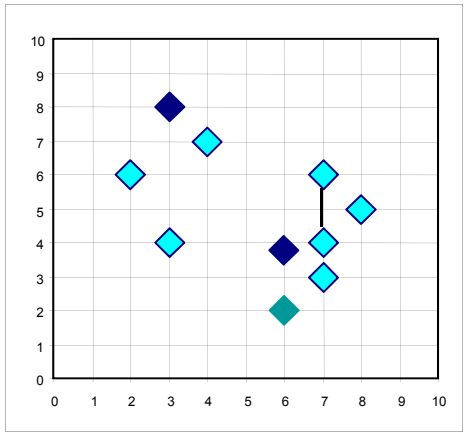


Total Cost = 20

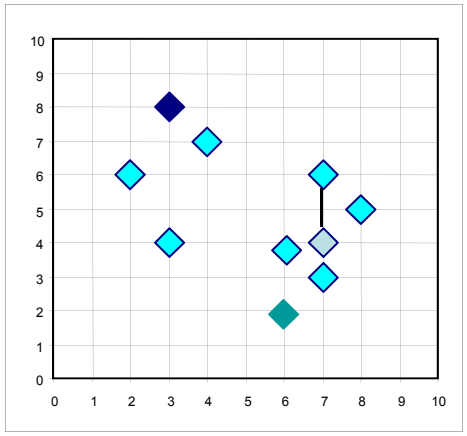
Assign each remaining object to nearest medoids



Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping



Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

From Han and Kamber's slides on cluster analysis

Comments on PAM?

- PAM is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean

- PAM works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

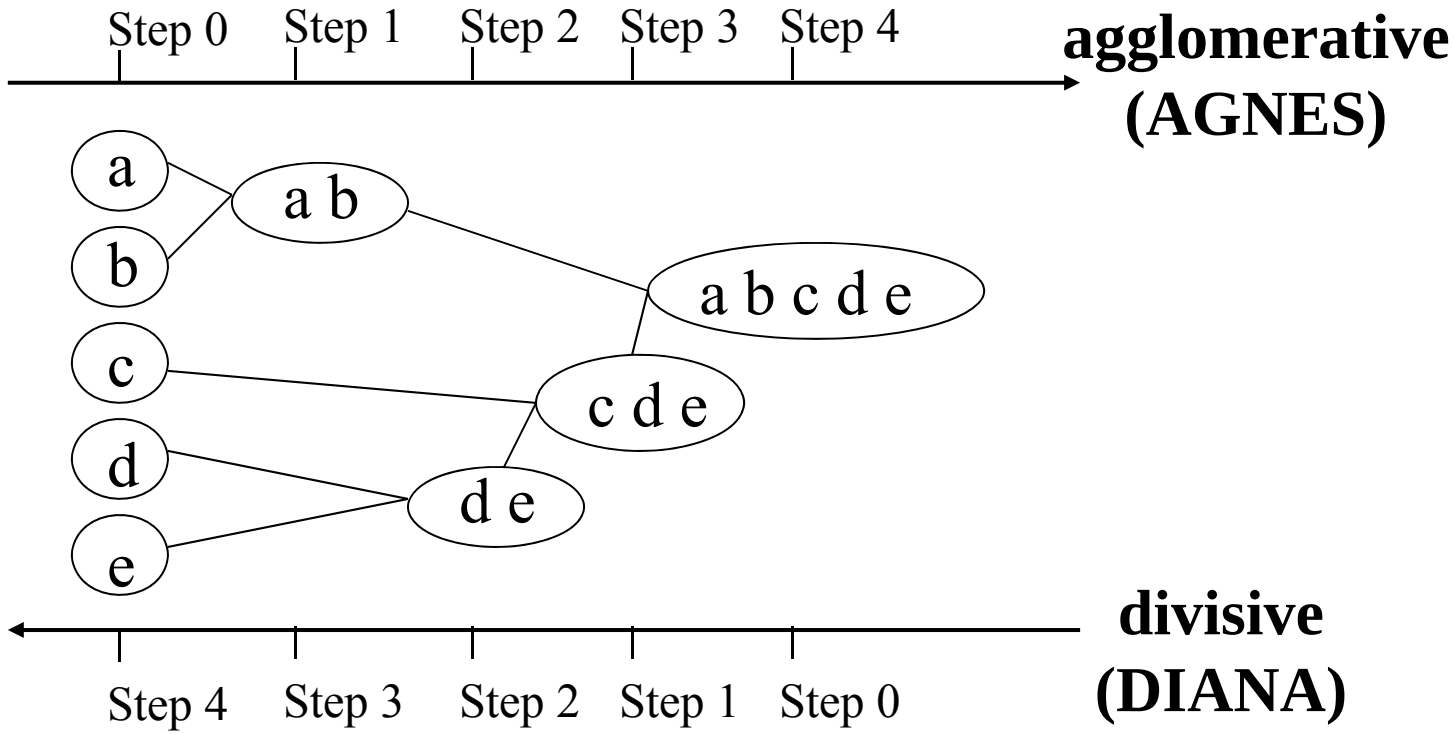
    where n is # of data,k is # of clusters

CLARA (Clustering Large Applications) (1990)

- CLARA (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws multiple samples of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



From Han and Kamber's slides on cluster analysis

Hierarchical Clustering

Given a set of N items to be clustered, and an NxN distance (or similarity) matrix, the basic process hierarchical clustering is this:

1. Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.

2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

3. Compute distances (similarities) between the new cluster and each of the old clusters.

4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Amalgamation or linkage rules

Single linkage (nearest neighbor). The distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters. This rule will, in a sense, *string* objects together to form clusters, and the resulting clusters tend to represent long "chains."

Complete linkage (furthest neighbor). The distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors"). This method usually performs quite well in cases when the objects actually form naturally distinct "clumps." If the clusters tend to be somehow elongated or of a "chain" type nature, then this method is inappropriate.

Unweighted pair-group average. The distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters. This method is also very efficient when the objects form natural distinct "clumps," however, it performs equally well with elongated, "chain" type clusters. Note that in their book, Sneath and Sokal (1973) introduced the abbreviation UPGMA to refer to this method as *unweighted pair-group method using arithmetic averages*.

Weighted pair-group average. This method is identical to the *unweighted pair-group average* method, except that in the computations, the size of the respective clusters (i.e., the number of objects contained in them) is used as a weight. Thus, this method (rather than the previous method) should be used when the cluster sizes are suspected to be greatly uneven. Note that in their book, Sneath and Sokal (1973) introduced the abbreviation *WPGMA* to refer to this method as *weighted pair-group method using arithmetic averages*.

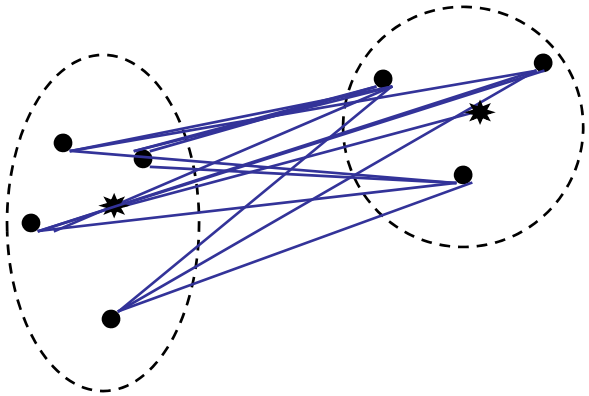
Unweighted pair-group centroid. The *centroid* of a cluster is the average point in the multidimensional space defined by the dimensions. In a sense, it is the *center of gravity* for the respective cluster. In this method, the distance between two clusters is determined as the difference between centroids. Sneath and Sokal (1973) use the abbreviation *UPGMC* to refer to this method as *unweighted pair-group method using the centroid average*.

Weighted pair-group centroid (median). This method is identical to the previous one, except that weighting is introduced into the computations to take into consideration differences in cluster sizes (i.e., the number of objects contained in them). Thus, when there are (or one suspects there to be) considerable differences in cluster sizes, this method is preferable to the previous one. Sneath and Sokal (1973) use the abbreviation *WPGMC* to refer to this method as *weighted pair-group method using the centroid average*.

Ward's method. This method is distinct from all other methods because it uses an analysis of variance approach to evaluate the distances between clusters. In short, this method attempts to minimize the Sum of Squares (SS) of any two (hypothetical) clusters that can be formed at each step. Refer to Ward (1963) for details concerning this method. In general, this method is regarded as very efficient, however, it tends to create clusters of small size.

Distance between clusters

- ***Single Link***: smallest distance between points
- ***Complete Link:*** largest distance between points
- ***Average Link:*** average distance between points
- ***Centroid:*** distance between centroids



Distance measures

Euclidean distance. This is probably the most commonly chosen type of distance. It simply is the geometric distance in the multidimensional space. It is computed as: $\text{distance}(x,y) = \{ \sum_i (x_i - y_i)^2 \}^{½}$

Note that Euclidean (and squared Euclidean) distances are usually computed from raw data, and not from standardized data. This method has certain advantages (e.g., the distance between any two objects is not affected by the addition of new objects to the analysis, which may be outliers). However, the distances can be greatly affected by differences in scale among the dimensions from which the distances are computed. For example, if one of the dimensions denotes a measured length in centimeters, and you then convert it to millimeters (by multiplying the values by 10), the resulting Euclidean or squared Euclidean distances (computed from multiple dimensions) can be greatly affected (i.e., biased by those dimensions which have a larger scale), and consequently, the results of cluster analyses may be very different. Generally, it is good practice to transform the dimensions so they have similar scales.

Squared Euclidean distance. You may want to square the standard Euclidean distance in order to place progressively greater weight on objects that are further apart. This distance is computed as (see also the note in the previous paragraph):

$\text{distance}(x,y) = \sum_i (x_i - y_i)^2$

City-block (Manhattan) distance. This distance is simply the average difference across dimensions. In most cases, this distance measure yields results similar to the simple Euclidean distance. However, note that in this measure, the effect of single large differences (outliers) is dampened (since they are not squared). The city-block distance is computed as:

$\text{distance}(x,y) = \sum_i |x_i - y_i|$

Chebychev distance. This distance measure may be appropriate in cases when one wants to define two objects as "different" if they are different on any one of the dimensions. The Chebychev distance is computed as:

$\text{distance}(x,y) = \text{Maximum}|x_i - y_i|$

Power distance. Sometimes one may want to increase or decrease the progressive weight that is placed on dimensions on which the respective objects are very different. This can be accomplished via the *power distance*. The power distance is computed as:

$\text{distance}(x,y) = (\sum_i |x_i - y_i|^p)^{1/r}$

where *r* and *p* are user-defined parameters. A few example calculations may demonstrate how this measure "behaves." Parameter *p* controls the progressive weight that is placed on differences on individual dimensions, parameter *r* controls the progressive weight that is placed on larger differences between objects. If *r* and *p* are equal to 2, then this distance is equal to the Euclidean distance.

Percent disagreement. This measure is particularly useful if the data for the dimensions included in the analysis are categorical in nature. This distance is computed as: $\text{distance}(x,y) = (\text{Number of } x_i \neq y_i)/ i$.

Distance Measures: Minkowski Metric

Suppose two objects x and y both have p features :

$$x = (x_1 x_2 \cdots x_p)$$

$$y = (y_1 y_2 \cdots y_p)$$

The Minkowski metric is defined by

$$d(x, y) = \sqrt[r]{\sum_{i=1}^{p} | x_i - y_i |^r}$$

Commonly Used Minkowski Metrics

$$1, r = 2 \text{ (Euclidean distance)}$$

$$d(x, y) = \sqrt[2]{\sum_{i=1}^{p} | x_i - y_i |^2}$$

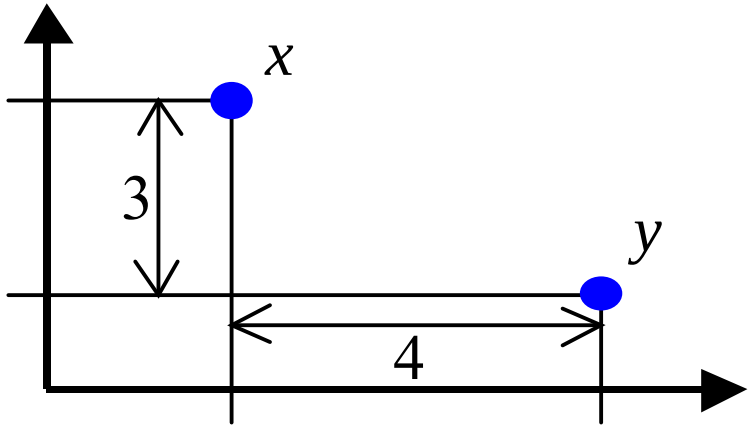
$$2, r = 1 \text{ (Manhattan distance)}$$

$$d(x, y) = \sum_{i=1}^{p} | x_i - y_i |$$

$$3, r = +\infty \text{ ("sup" distance)}$$

$$d(x, y) = \max_{1 \le i \le p} | x_i - y_i |$$

An Example



1, Euclidean distance: $\sqrt[2]{4^2+3^2}=5.$

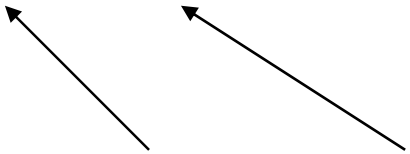
2, Manhattan distance: $4+3=7.$

3, "sup" distance: $\max\{4,3\}=4.$

Manhattan distance is called *Hamming distance* when all features are binary.

Gene expression levels under 17 conditions (1-High,0-Low)

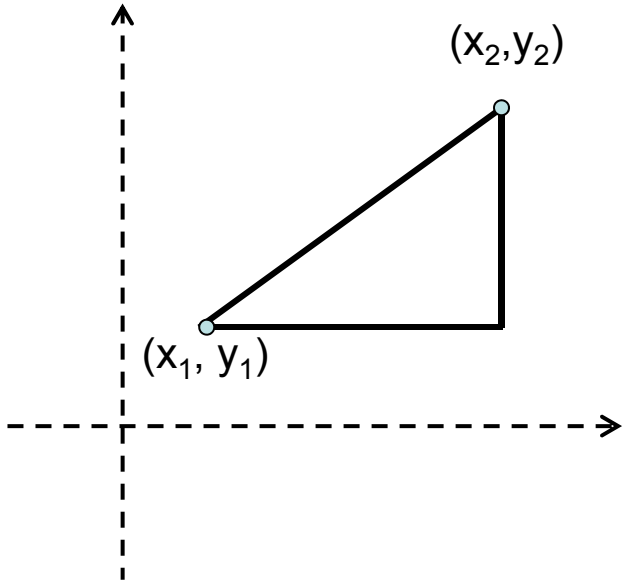
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
GeneA	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
GeneB	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1



Hamming Distance : $\#(01)+\#(10)=4+1=5$**.**

Measuring similarity

- Euclidean (L_2) distance
- Manhattan (L_1) distance
- L_m: $(|x_1-x_2|^m+|y_1-y_2|^m)^{1/m}$
- L_∞: $\max(|x_1-x_2|,|y_1-y_2|)$
- Inner product: $x_1x_2+y_1y_2$
- Correlation coefficient (Pearson)
- Spearman rank correlation coefficient



Similarity measures: correlation coefficient

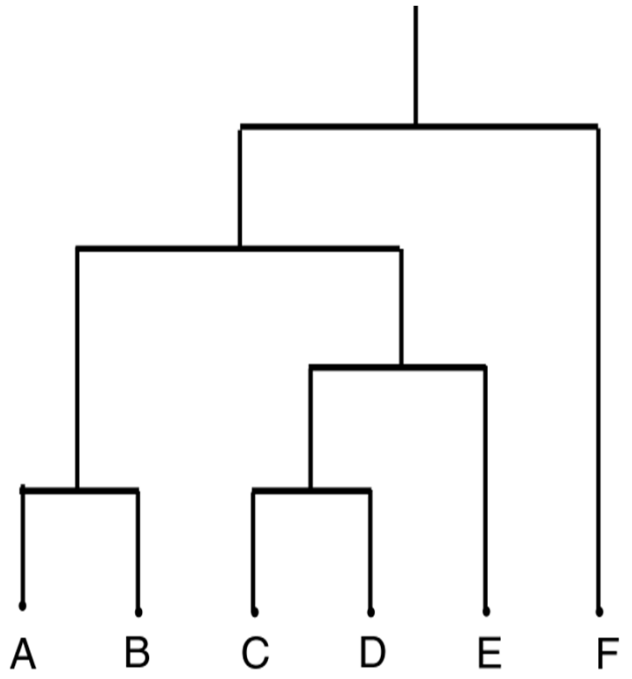
$$s(x,y) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \times \sum_{i=1}^{p}(y_i - \bar{y})^2}}$$

$$\left|s(x,y)\right| \leq 1$$

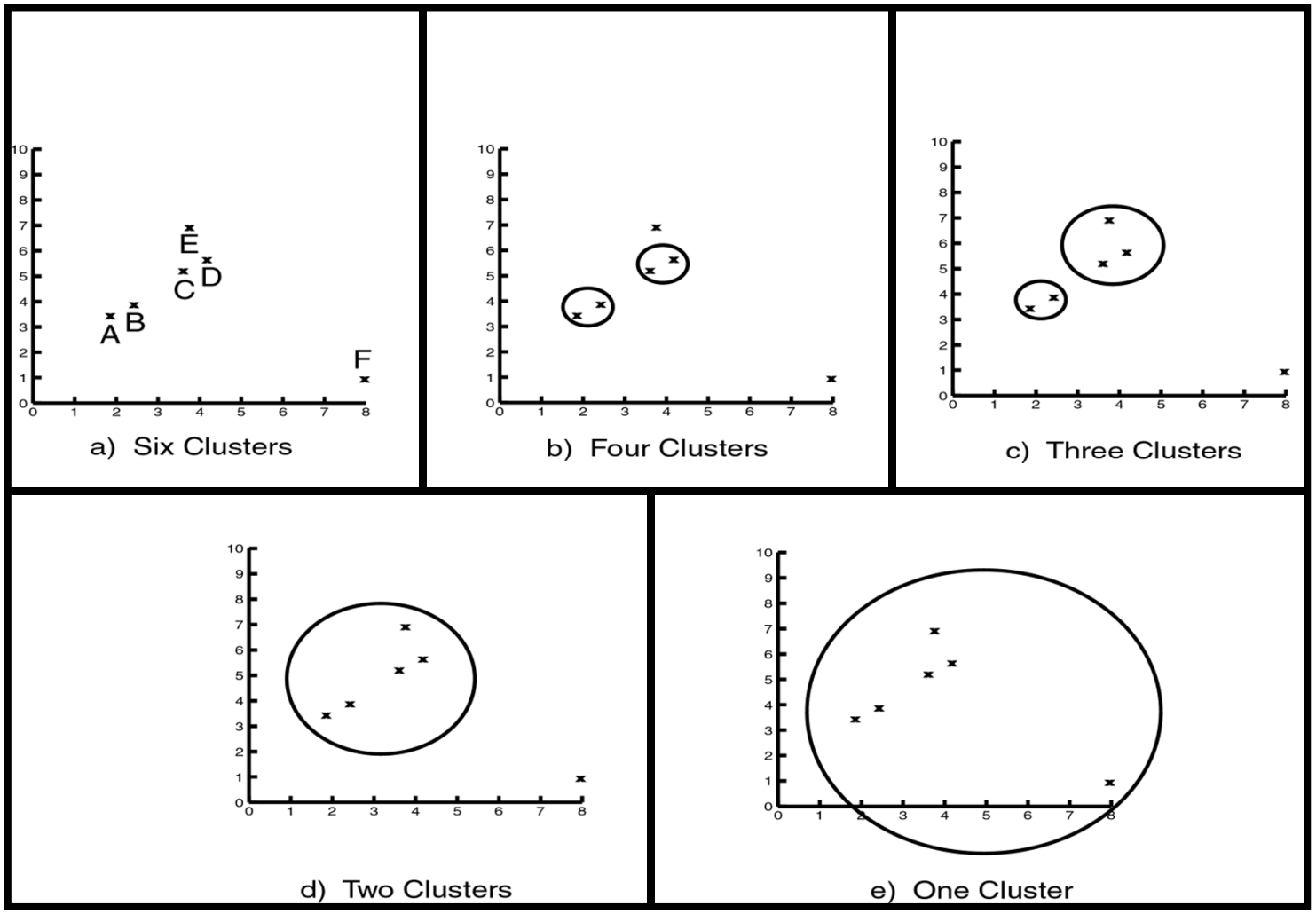
Dendrogram

Dendrogram: a tree data structure which illustrates hierarchical clustering techniques.

- Each level shows clusters for that level.
  - Leaf – individual clusters
  - Root – one cluster
- A cluster at level i is the union of its children clusters at level i+1.

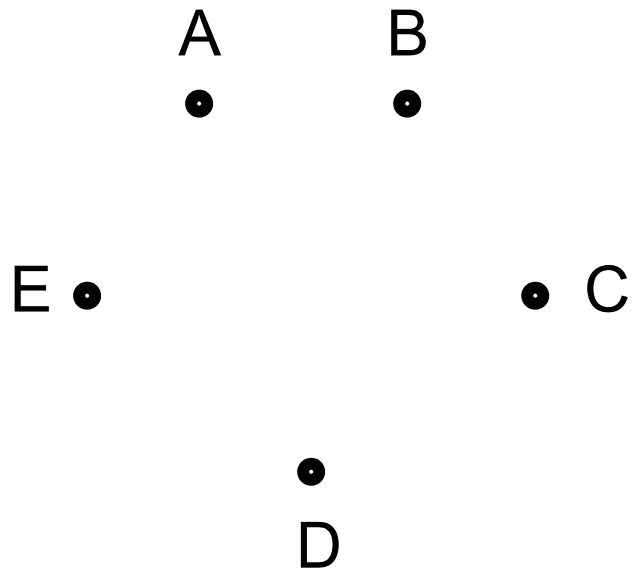


Levels of clustering



Agglomerative example

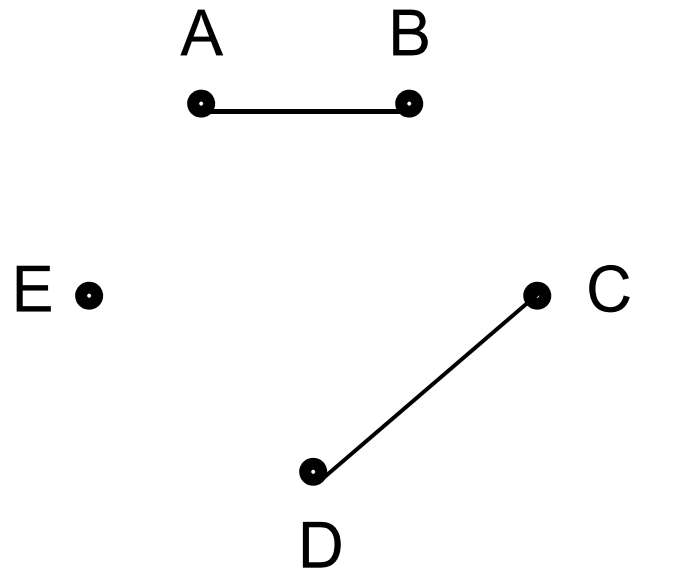
	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

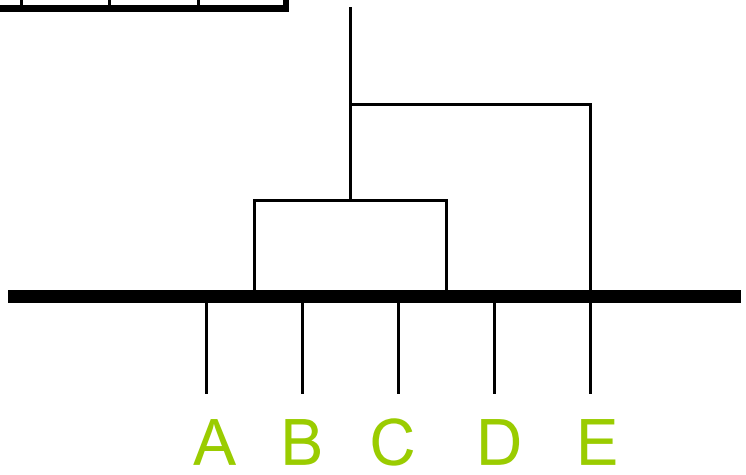


Threshold of

Agglomerative example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



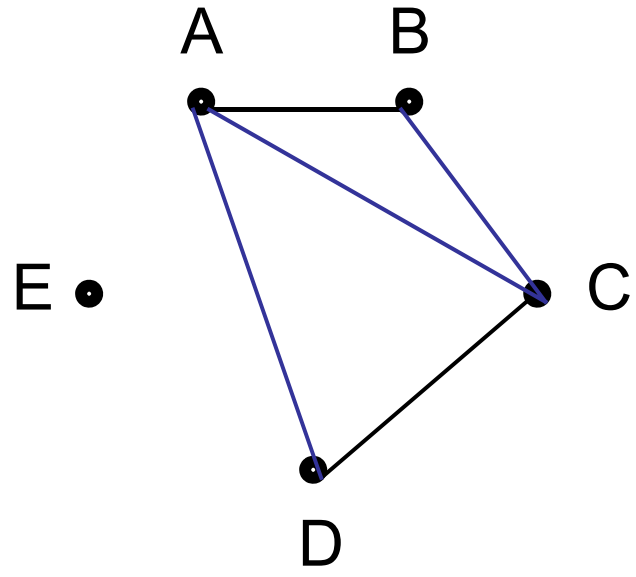


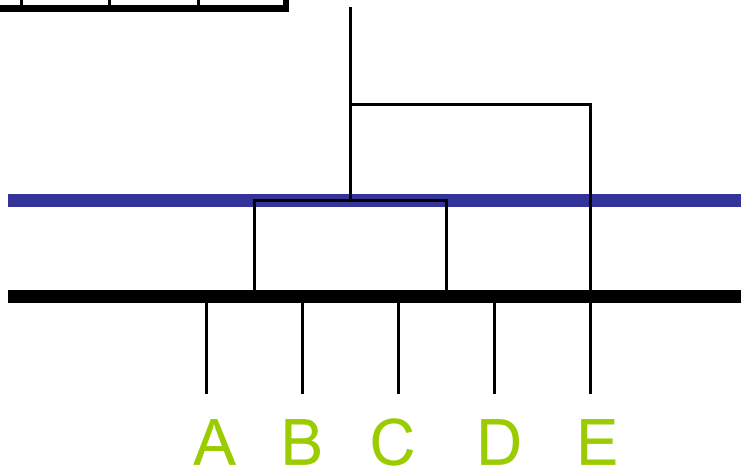
Threshold of

1

Agglomerative example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



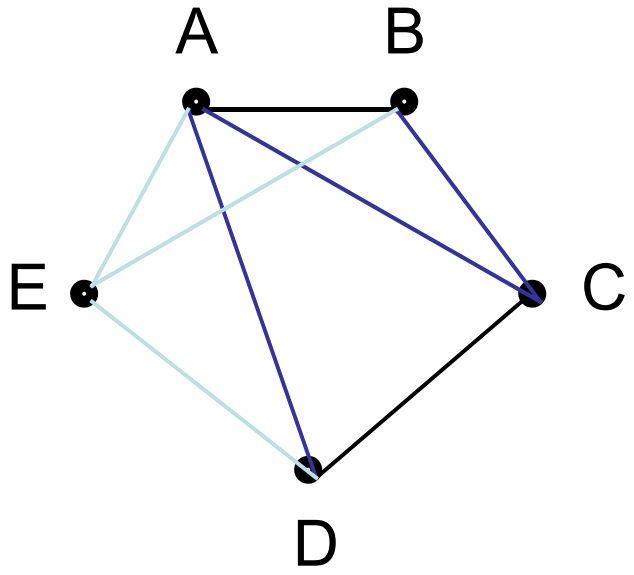


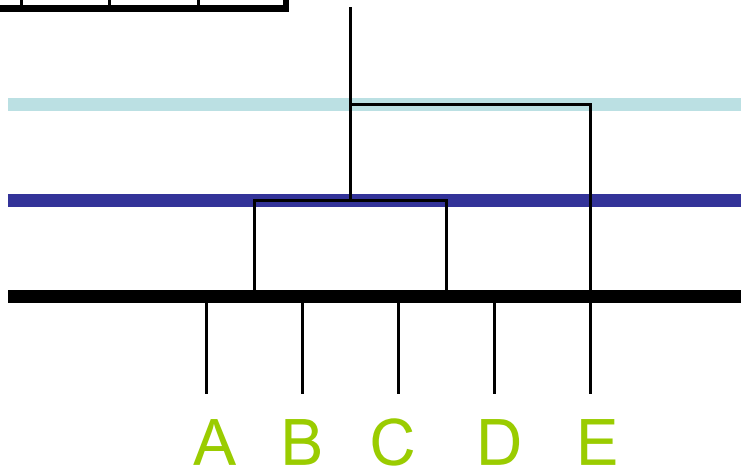
Threshold of

1 2

Agglomerative example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



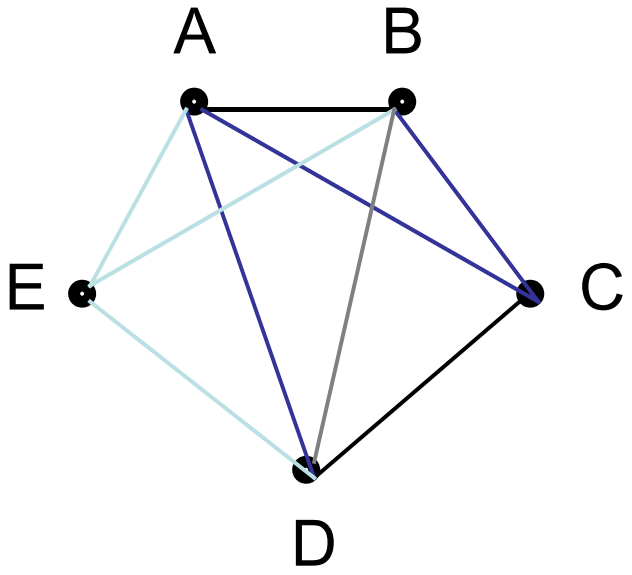


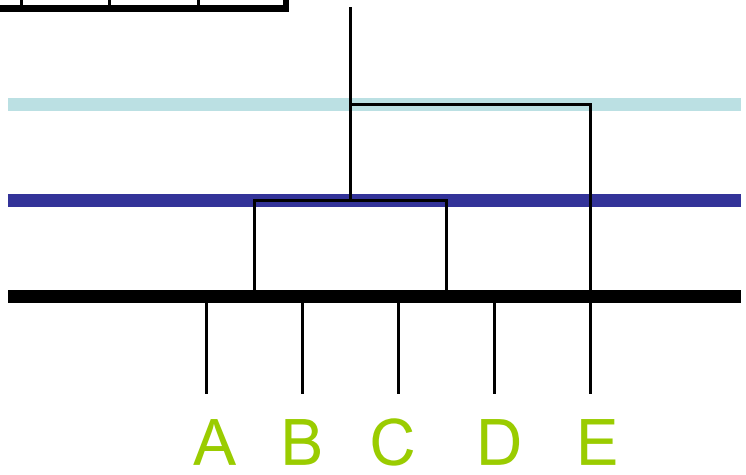
Threshold of

1 2 3

Agglomerative example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



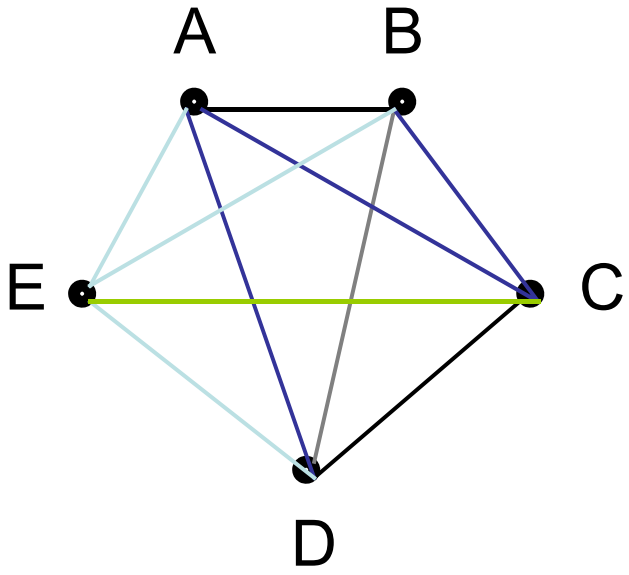


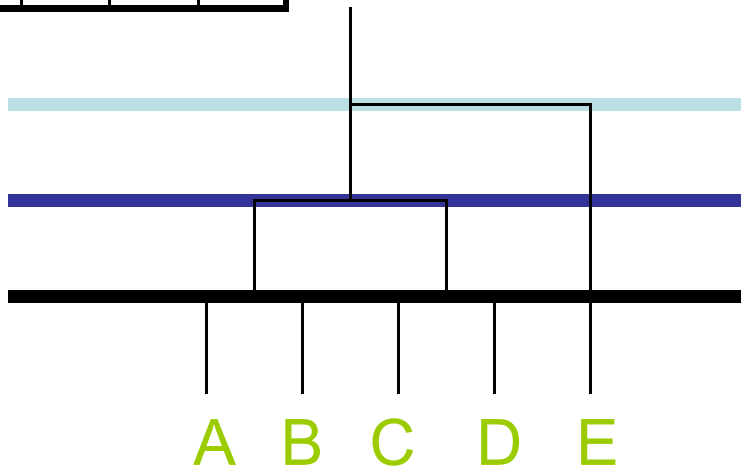
Threshold of

1 2 3 4

Agglomerative example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

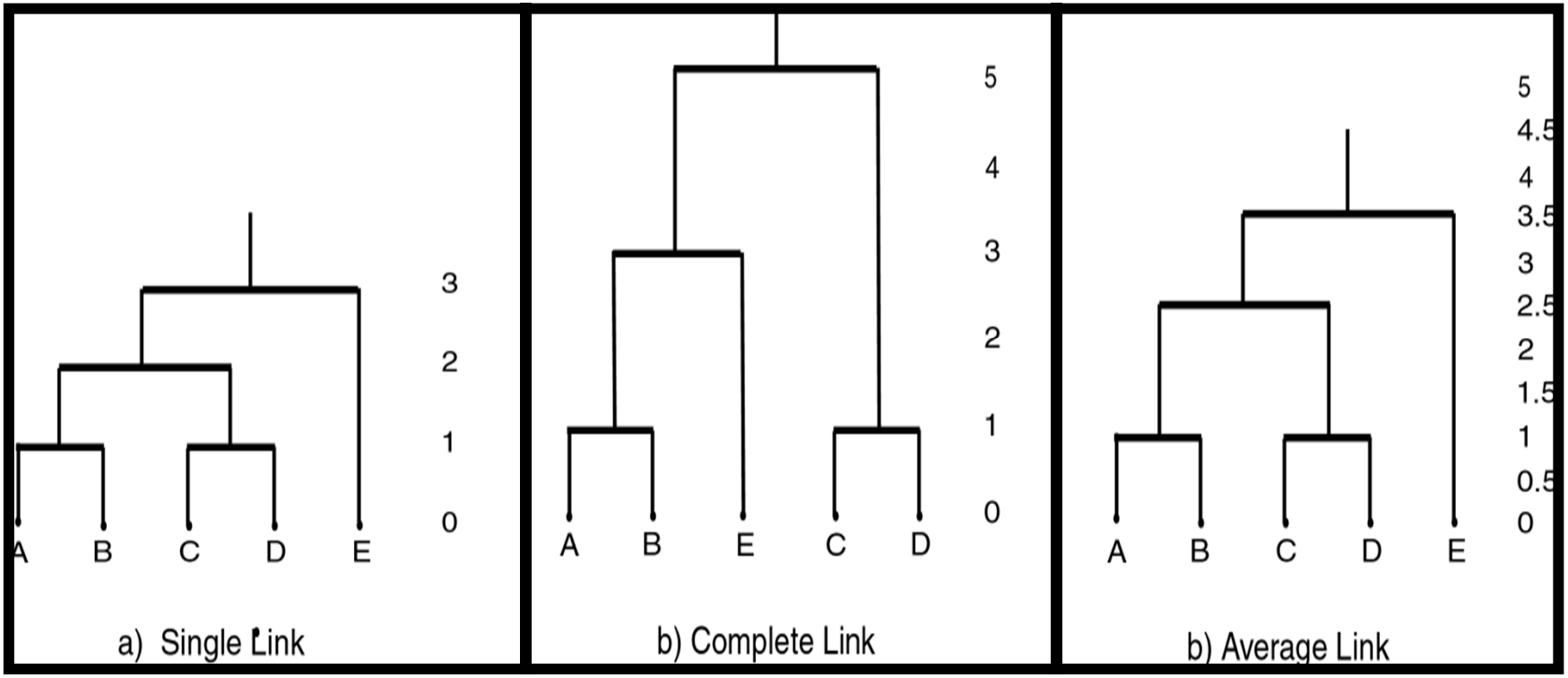




Threshold of

1 2 3 4 5

Single-Link, Complete-Link & Average-Link Clustering



Issues in cluster analysis

- A lot of clustering algorithms
- A lot of distance/similarity metrics
- Which clustering algorithm runs faster and uses less memory?
- How many clusters after all?
- Are the clusters stable?
- Are the clusters meaningful?

Statistical significance testing

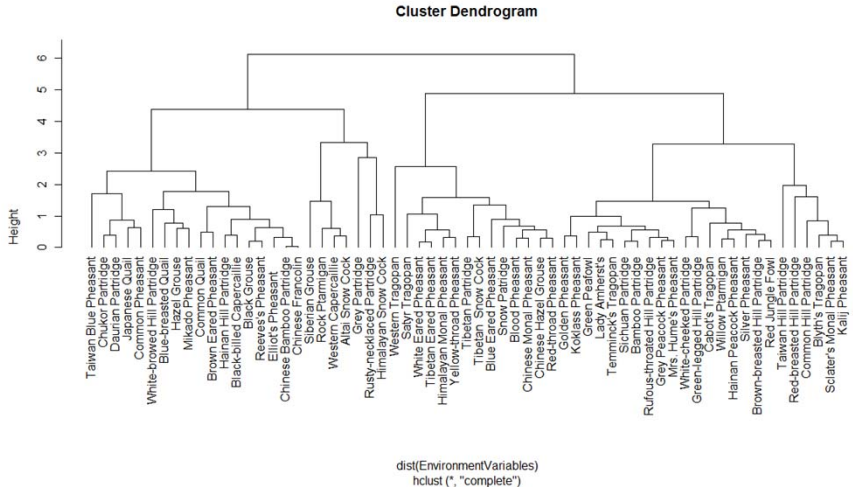
- Cluster analysis is a "collection" of different algorithms that "put objects into clusters according to well defined similarity rules." not as much a typical statistical test

- Cluster analysis methods are mostly used when we do **not have any a priori hypotheses**, but are still in the exploratory phase of our research. In a sense, cluster analysis finds the "most significant solution possible."

- Statistical significance testing is not appropriate here, even in cases when p-levels are reported (as in *k*-means clustering).

R code: Hierarchical cluster analysis

```
library(RODBC)
channel = odbcConnectAccess('D:/text/pheasant/pheasants_points_feature.mdb')
Galliform63 = sqlFetch(channel, 'species63')
EnvironmentVariables = Galliform63[, c('elevation_mean', 'footprint_mean', 'veg_mean')]
row.names(EnvironmentVariables)=Galliform63[,2]

for(i in 1:length(EnvironmentVariables)){
            EnvironmentVariables[, i] =
           (EnvironmentVariables[, i] - mean(EnvironmentVariables[,i], na.rm = T))
           /sd(EnvironmentVariables[,i], na.rm = T)
           } # standarization

# Hierarchical cluster analysis
hi.cluster = hclust(dist(EnvironmentVariables)) #using default configuration

x11() #create a new window
plot(hi.cluster, hang = -1)#plot cluster dendrogram
```



Discriminant analysis (DA)

Discriminant analysis (DA)

- DA is used to identify boundaries between groups of objects by a measure of distance.

  For example:

  (a) What taxa do some insects belong to on basis of a number of measures.

  (b) Is someone a good credit risk or not?

  (c) Should a student be admitted to college?

- Similar to regression, except that criterion (or dependent variable) is categorical rather than continuous.
- Alternatively, discriminant function analysis is multivariate analysis of variance (MANOVA) **reversed**.

In MANOVA, the independent variables are the groups and the dependent variables are the continuous measures. In DA, the independent variables are the continuous measures and the dependent variables are the groups.

Example

Discriminant analysis of remote sensing data on five crops

crop	band1	band2	band3	band4
CORN	16	27	31	33
CORN	15	23	30	30
CORN	16	27	27	26
CORN	18	20	25	23
CORN	15	15	31	32
CORN	15	32	32	15
CORN	12	15	16	73
SOYBEANS	20	23	23	25
SOYBEANS	24	24	25	32
SOYBEANS	21	25	23	24
SOYBEANS	27	45	24	12
SOYBEANS	12	13	15	42
SOYBEANS	22	32	31	43
COTTON	31	32	33	34
COTTON	29	24	26	28
COTTON	34	32	28	45
COTTON	26	25	23	24
COTTON	53	48	75	26
COTTON	34	35	25	78
SUGARBEETS	22	23	25	42
SUGARBEETS	25	25	24	26
SUGARBEETS	34	25	16	52
SUGARBEETS	54	23	21	54
SUGARBEETS	25	43	32	15
SUGARBEETS	26	54	2	54
CLOVER	12	45	32	54
CLOVER	24	58	25	34
CLOVER	87	54	61	21
CLOVER	51	31	31	16
CLOVER	96	48	54	62
CLOVER	31	31	11	11
CLOVER	56	13	13	71
CLOVER	32	13	27	32
CLOVER	36	26	54	32
CLOVER	53	08	06	54
CLOVER	32	32	62	16

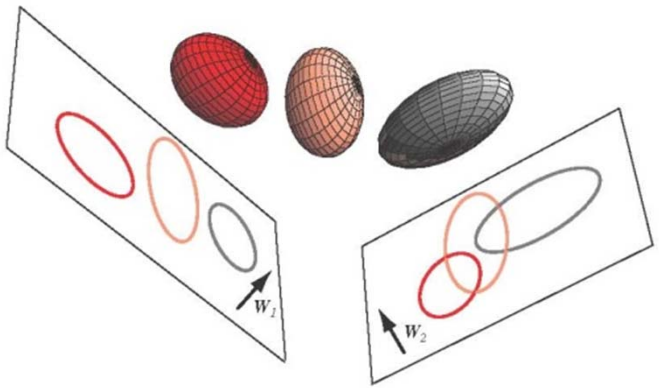
R code: linear discriminant analysis

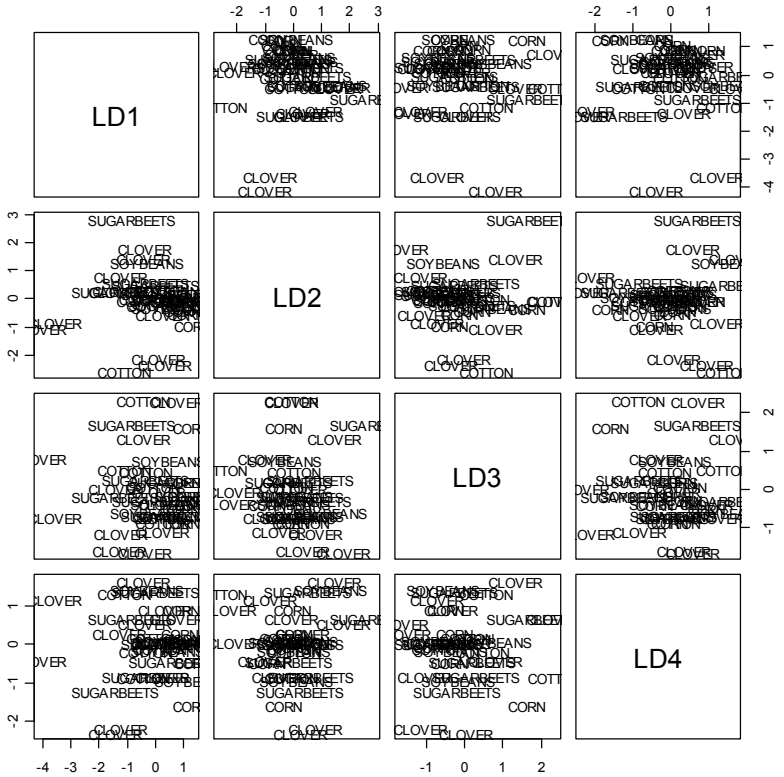
```
library(MASS);  remote.sensing = read.csv("remote.sensing.csv", header = T);
table(remote.sensing$crop)
```

CLOVER	CORN	COTTON	SOYBEANS	SUGARBEETS
11	7	6	6	6

```
nrow(remote.sensing) # 36

lda.result <- lda(crop ~ band1+band2+band3+band4, remote.sensing)

plot(lda.result, cex = 1)
```

> ?lda





R: linear discriminant analysis

lda.result

```
Call:
lda(crop ~ band1 + band2 + band3 + band4, data = remote.sensing)

Prior probabilities of groups:
    CLOVER       CORN     COTTON   SOYBEANS SUGARBEETS
 0.3055556  0.1944444  0.1666667  0.1666667  0.1666667

Group means:
              band1    band2    band3    band4
CLOVER     46.36364 32.63636 34.18182 36.63636
CORN       15.28571 22.71429 27.42857 33.14286
COTTON     34.50000 32.66667 35.00000 39.16667
SOYBEANS   21.00000 27.00000 23.50000 29.66667
SUGARBEETS 31.00000 32.16667 20.00000 40.50000

Coefficients of linear discriminants:
                LD1           LD2          LD3           LD4
band1 -6.147360e-02  0.009215431 -0.02987075 -0.014680566
band2 -2.548964e-02  0.042838972  0.04631489  0.054842132
band3  1.642126e-02 -0.079471595  0.01971222  0.008938745
band4  5.143616e-05 -0.013917423  0.05381787 -0.025717667

Proportion of trace:
   LD1    LD2    LD3    LD4
0.7364 0.1985 0.0576 0.0075
```

R: linear discriminant analysis

lda.predict <- predict(lda.result, remote.sensing)

table(lda.predict$class) ***#number of observation for each crop***

lda.predict$class ***#predicted crop types***

```
 [1] CORN      CORN      CORN      CORN      CORN      SOYBEANS
 [7] CORN      SOYBEANS  SOYBEANS  SOYBEANS  SUGARBEETS CORN
…
```

lda.predict$posterior ***#predicted crop types values***

	CLOVER	CORN	COTTON	SOYBEANS	SUGARBEETS
1	0.08935	0.4054295	0.17631	0.239184	0.08971
2	0.07690	0.4558027	0.14209	0.253010	0.07219

…

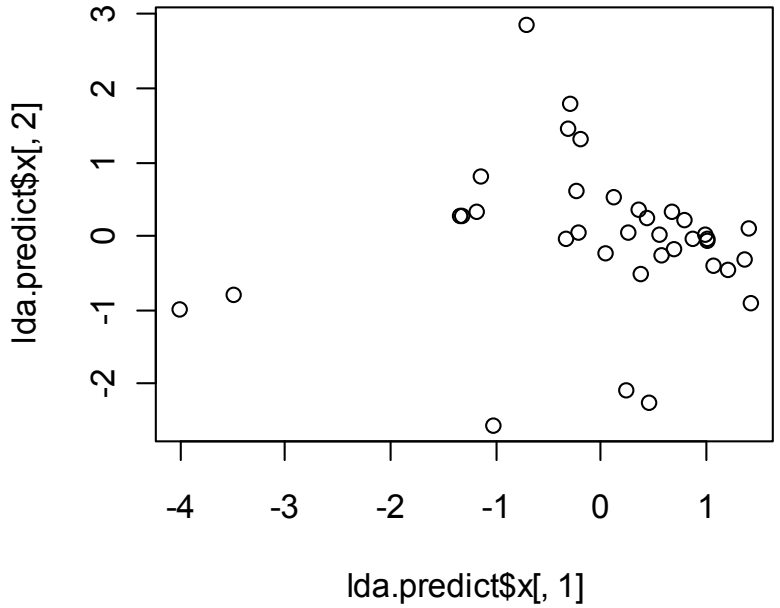
lda.predict$x ***#linear discriminant scores for each observation***

	LD1	LD2	LD3	LD4
1	1.05991249	-0.38894000	0.22804664	0.17329536
2	1.20676907	-0.44828745	-0.10850799	0.03682165

…

R - linear discriminant analysis

```
plot(lda.predict$x[,1], lda.predict$x[,2])

compare = data.frame(remote.sensing$crop,
                     lda.predict$class)
```



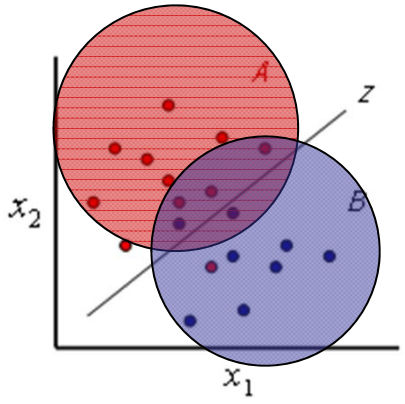
	remote.sensing.crop	lda.predict.class
1	CORN	CORN
2	CORN	CORN
3	CORN	CORN
4	CORN	CORN
5	CORN	CORN
6	CORN	SOYBEANS
7	CORN	CORN
8	SOYBEANS	SOYBEANS
9	SOYBEANS	SOYBEANS
10	SOYBEANS	SOYBEANS
11	SOYBEANS	SUGARBEETS
12	SOYBEANS	CORN
13	SOYBEANS	COTTON
14	COTTON	CLOVER
15	COTTON	SOYBEANS
16	COTTON	CLOVER
17	COTTON	SOYBEANS
18	COTTON	CLOVER
19	COTTON	COTTON
20	SUGARBEETS	CORN
21	SUGARBEETS	SOYBEANS
22	SUGARBEETS	SUGARBEETS
23	SUGARBEETS	CLOVER
24	SUGARBEETS	SOYBEANS
25	SUGARBEETS	SUGARBEETS
26	CLOVER	COTTON
27	CLOVER	SUGARBEETS
28	CLOVER	CLOVER
29	CLOVER	CLOVER
30	CLOVER	CLOVER
31	CLOVER	SUGARBEETS
32	CLOVER	CLOVER
33	CLOVER	CLOVER
34	CLOVER	COTTON
35	CLOVER	CLOVER
36	CLOVER	COTTON

Discriminant analysis vs. clustering

Discriminant Analysis	Clustering
• known number of classes	• unknown number of classes
• based on a training set	• no prior knowledge
• used to classify future observations	• used to understand (explore) data
• classification is a form of supervised learning	• clustering is a form of unsupervised learning
• Y = X1 + X2 + X3……	• X1 + X2 + X3……

Linear discriminant analysis

- Linear discriminant analysis attempts to find the linear combination of the selected measures that best separate the population



$$Z = b_1 x_1 + b_2 x_2$$

b = discriminant coefficients
x = input variables

Procedure

Discriminant function analysis is broken into a 2-step process:

1. testing significance of a set of discriminant functions

- The first step is computationally identical to MANOVA. There is a matrix of total variances and covariances; likewise, there is a matrix of pooled within-group variances and covariances.
- The two matrices are compared via multivariate F tests in order to determine whether or not there are any significant differences (with regard to all variables) between groups.
- One first performs the multivariate test, and, if statistically significant, proceeds to see which of the variables have significantly different means across the groups.

Procedure

2. classification

- Once group means are found to be statistically significant, classification of variables is undertaken.
- Discriminant analysis automatically determines some optimal combination of variables so that the first function provides the most overall discrimination between groups, the second provides second most, and so on.
- Moreover, the functions will be independent or orthogonal, that is, their contributions to the discrimination between groups will not overlap.

Assumptions

Sample size:
Unequal sample sizes are acceptable. The sample size of the smallest group needs to exceed the number of predictor variables. As a “rule of thumb”, the smallest sample size should be at least 20 for a few (4 or 5) predictors. The maximum number of independent variables is n - 2, where n is the sample size. While this low sample size may work, it is not encouraged, and generally it is best to have 4 or 5 times as many observations and independent variables.

Normal distribution:
It is assumed that the data (for the variables) represent a sample from a multivariate normal distribution. You can examine whether or not variables are normally distributed with histograms of frequency distributions. However, note that violations of the normality assumption are not "fatal" and the resultant significance test are still reliable as long as non-normality is caused by skewness and not outliers (Tabachnick and Fidell 1996).

Homogeneity of variances/covariances:
Discriminant analysis is very sensitive to heterogeneity of variance-covariance matrices. Before accepting final conclusions for an important study, it is a good idea to review the within-groups variances and correlation matrices. Homoscedasticity is evaluated through scatterplots and corrected by transformation of variables.

Assumptions

Outliers:

- Discriminant analysis is highly sensitive to the inclusion of outliers.
- Run a test for univariate and multivariate outliers for each group, and transform or eliminate them.
- If one group in the study contains extreme outliers that impact the mean, they will also increase variability. Overall significance tests are based on pooled variances, that is, the average variance across all groups. Thus, the significance tests of the relatively larger means (with the large variances) would be based on the relatively smaller pooled variances, resulting erroneously in statistical significance.

Non-multicollinearity:

- If one of the independent variables is very highly correlated with another, or one is a function (e.g., the sum) of other independents, then the matrix will not have a unique discriminant solution.
- To the extent that independents are correlated, the standardized discriminant function coefficients will not reliably assess the relative importance of the predictor variables.

Principal Component Analysis (PCA)

Principal component analysis

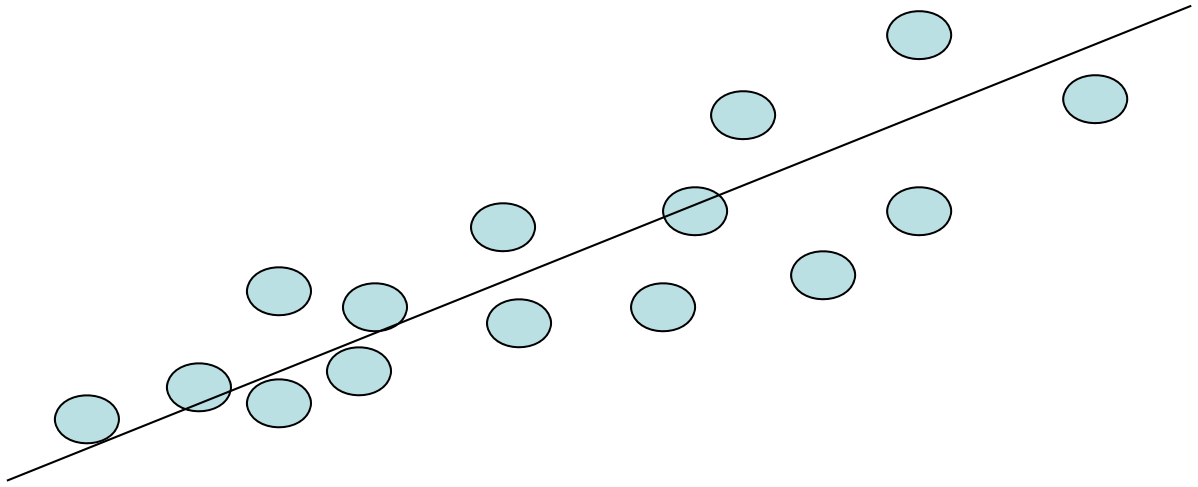
Principal component analysis (PCA) is a technique that is useful for the compression and classification of data. The purpose is to **reduce the dimensionality of a data set** (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information.

By information we mean the variation present in the sample, given by the correlations between the original variables. The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Principal Component Analysis

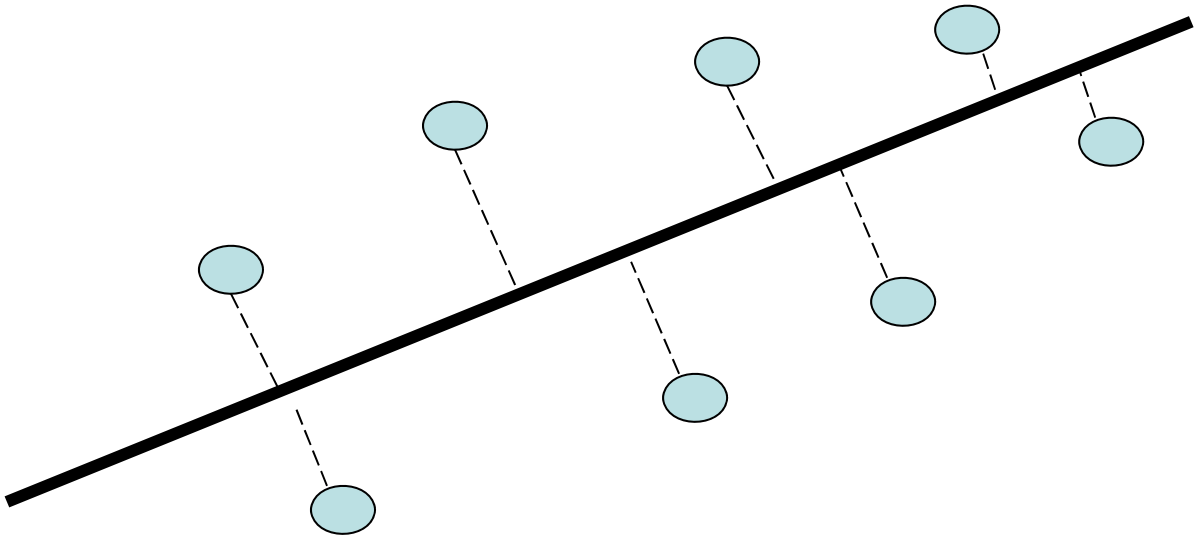
- Given m points in a n dimensional space, for large n, how does one project on to a 1 dimensional space?

- Choose a line that fits the data so the points are spread out well along the line.



Principal Component Analysis

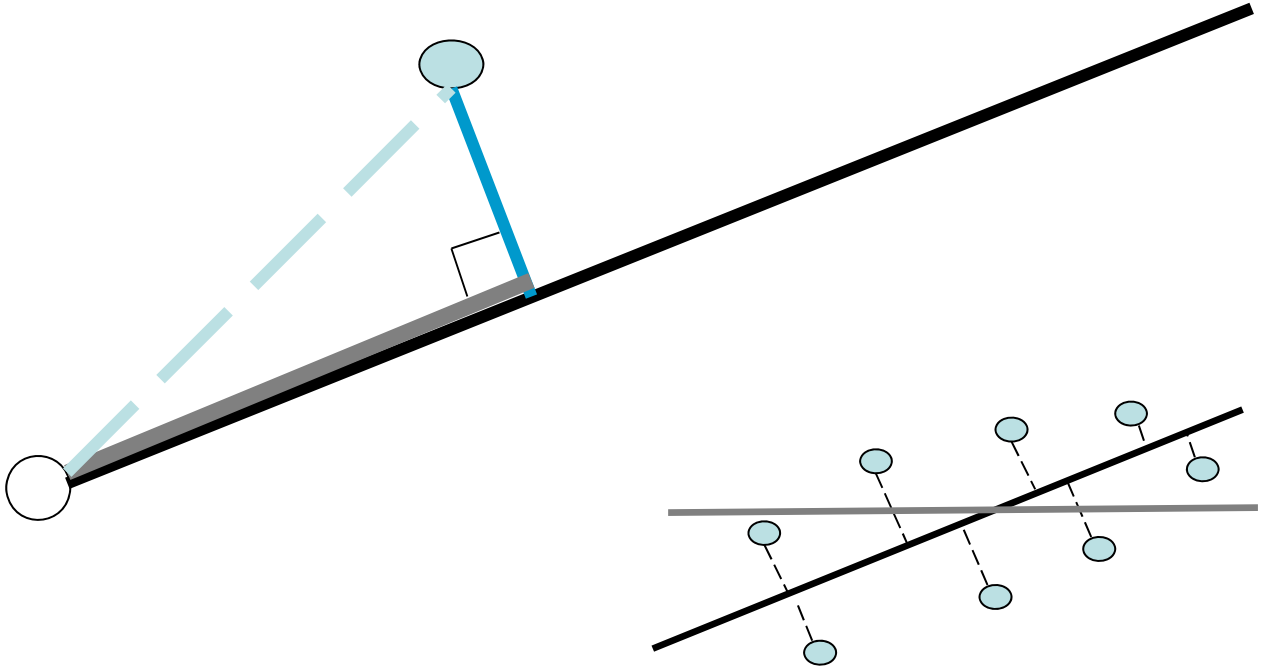
- Formally, minimize sum of squares of distances to the line.



- Why sum of squares? Because it allows fast minimization.

Principal Component Analysis

- For one data point and a line through point (0,0), minimizing sum of squares of distances to the line is the same as maximizing the sum of squares of the projections on that line (Pythagoras, long ago)



PCA: General methodology

From k original variables: $x_1,x_2,\ldots,x_k$:

Produce k new variables: $y_1,y_2,\ldots,y_k$:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$$
$$\ldots$$
$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$$

y_k's are
Principal Components

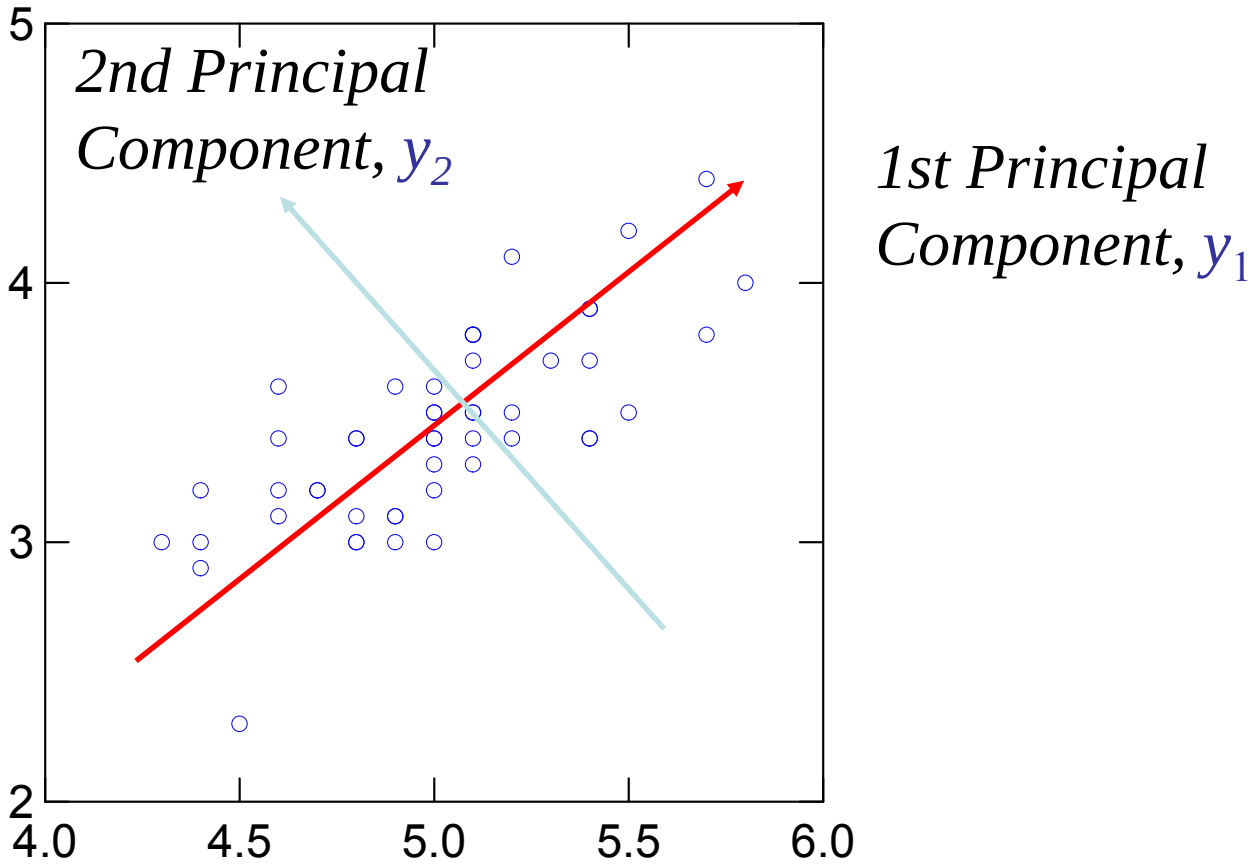
such that:

y_k's are uncorrelated (orthogonal)

y_1 explains as much as possible of original variance in data set

y_2 explains as much as possible of remaining variance

Principal Components Analysis



Principal Components Analysis

$\{a_{11},a_{12},...,a_{1k}\}$ is 1st **Eigenvector** of covariance matrix, and **coefficients** of first principal component

$\{a_{21},a_{22},...,a_{2k}\}$ is 2nd **Eigenvector** of covariance matrix, and **coefficients** of 2nd principal component

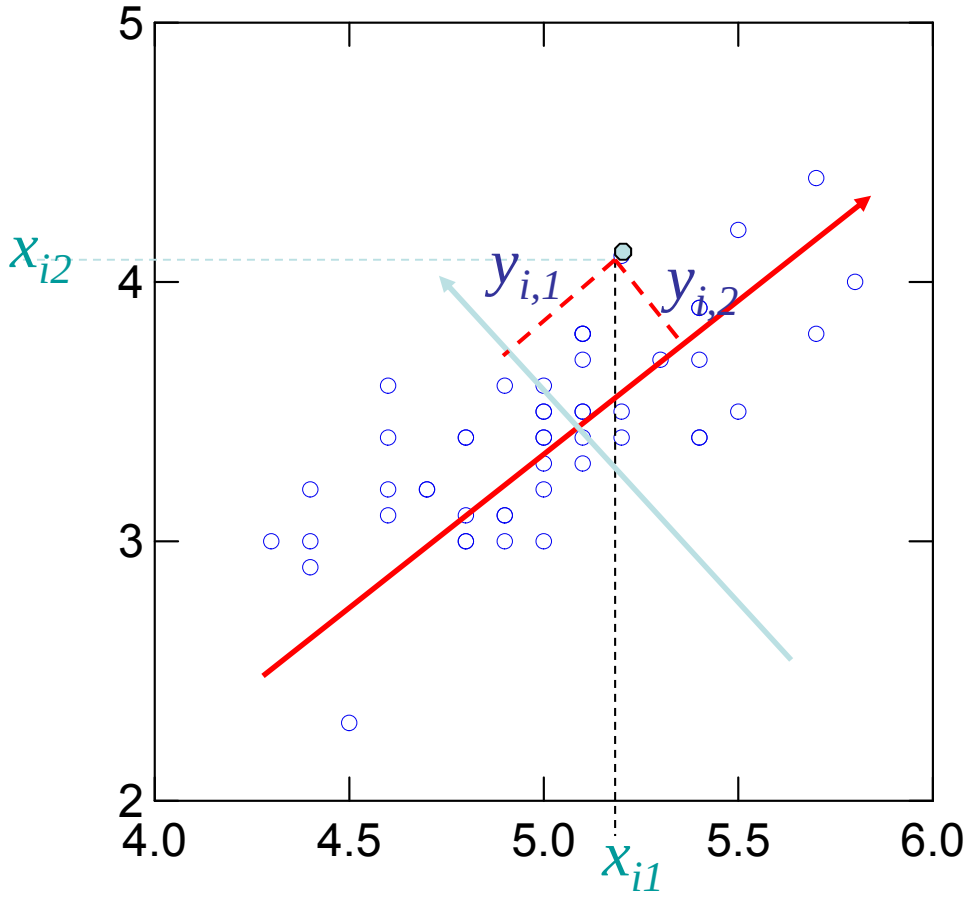
…

$\{a_{k1},a_{k2},...,a_{kk}\}$ is *k*th **Eigenvector** of covariance matrix, and **coefficients** of *k*th principal component

Scores

Score of *i*th unit on *j*th principal component

$$y_{i,j} = a_{j1}x_{i1} + a_{j2}x_{i2} + \ldots + a_{jk}x_{ik}$$



Principal Components Analysis

Amount of **variance accounted for** by:

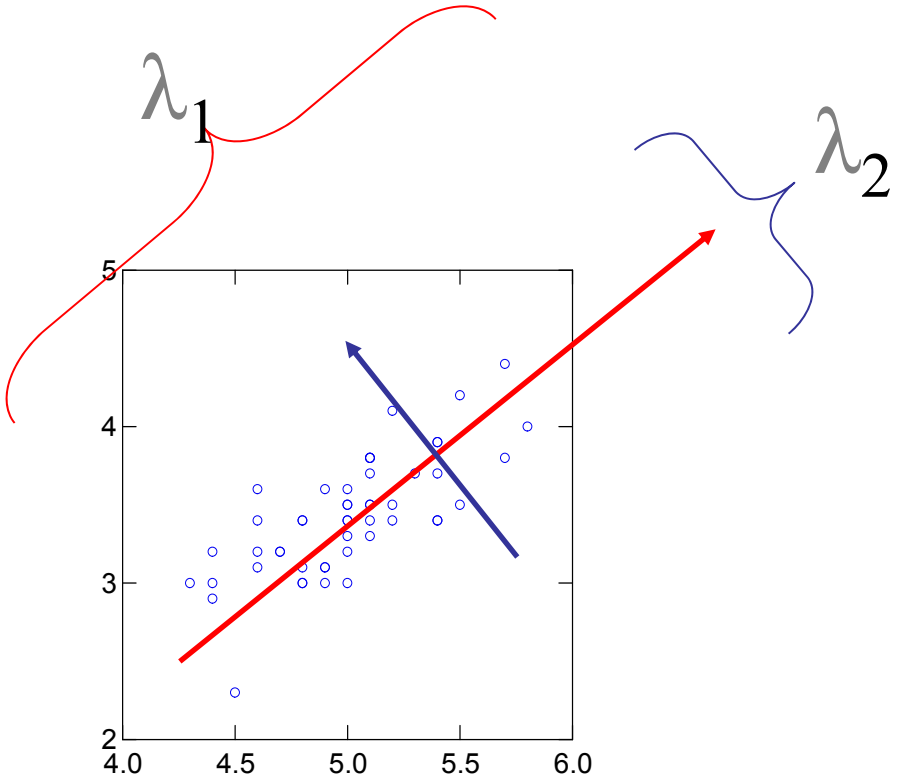
1st principal component, λ_1, 1st **eigenvalue**

2nd principal component, λ_2, 2nd **eigenvalue**

...

$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \ldots$

Average $\lambda_j = 1$



PCA: Terminology

- *j*th **principal component** is linear combination of all variables
- **coefficients**, a_{jk}, are elements of eigenvectors and relate original variables (standardized if using correlation matrix) to components
- **scores** are values of units on components (produced using coefficients)
- **amount of variance accounted for** by component is given by eigenvalue, λ_j
- **proportion of variance accounted for** by component is given by $\lambda_j / \Sigma \lambda_j$
- **loading** of *k*th original variable on *j*th component is given by $a_{jk}\lambda_j$ --correlation between variable and component

R code: PCA

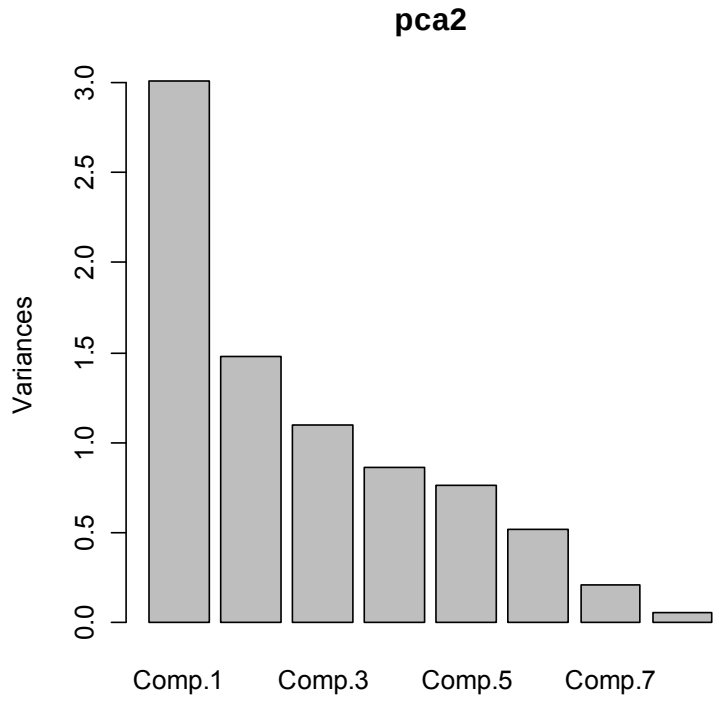
```
ibis = read.csv('D:/database/ibis2010.csv', header=T)
head(ibis)
ibis.pre = ibis[ibis$use==1,c(3:6,8,9,11,12)]
head(ibis.pre)
```

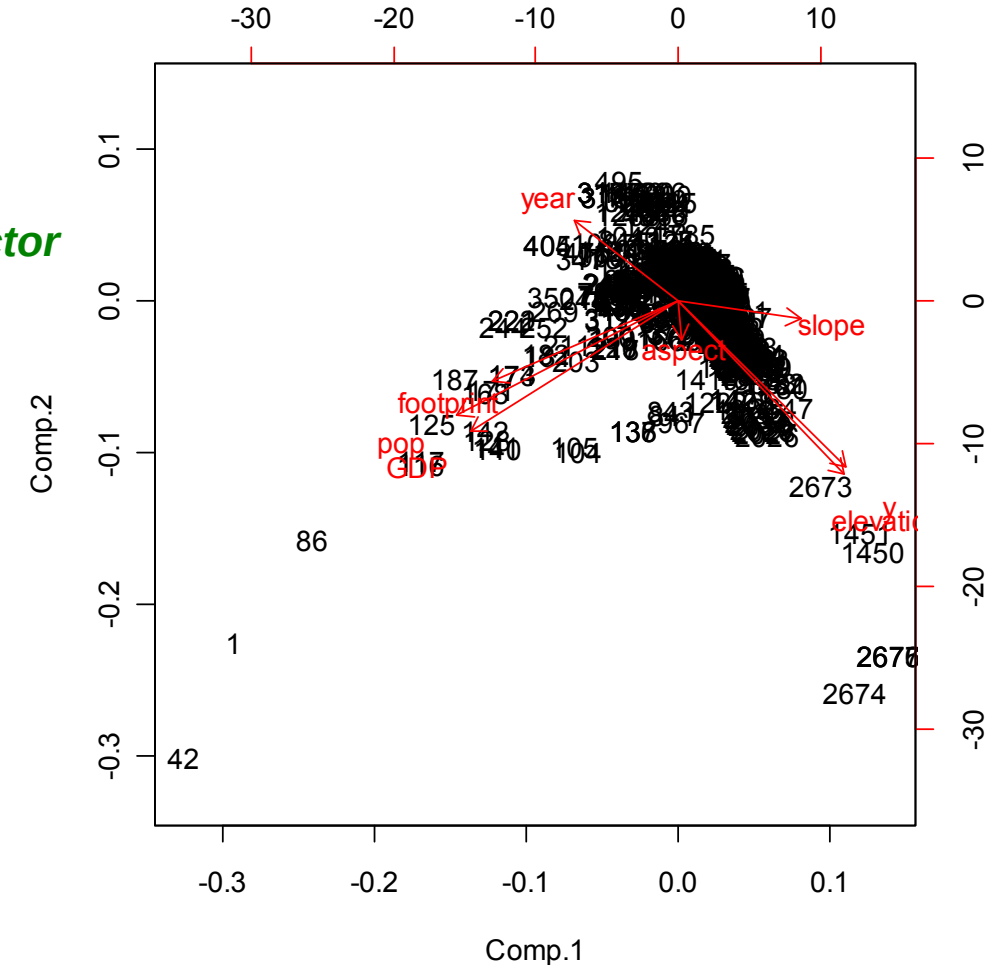
	latitude	aspect	elevation	footprint	year	GDP	pop	slope
1	33.1	0.893	476	61	2008	333	2032	0.503
42	33.3	0.798	484	38	2007	420	3049	0.685
86	33.1	0.56	473	60	2008	256	1485	0.812
104	33.4	0.502	942	20	2006	186	488	5.002
105	33.4	0.502	942	20	2008	186	488	5.002
116	33.2	0.201	476	44	2006	169	1321	2.275

```
################ PCA ################
## The variances of the variables in the ibis data
## vary by orders of magnitude, so scaling is appropriate
pca1 <- princomp(ibis.pre) # inappropriate
pca2 <- princomp(ibis.pre, cor = TRUE) # =^= prcomp(ibis.pre, scale=TRUE)
```

R code: PCA

```
plot(pca2) # shows a scree plot
biplot(pca2)
summary(pca2)
pca2$loadings
pca2$scores # principal component  vector
```





R code: PCA

pca2$loadings

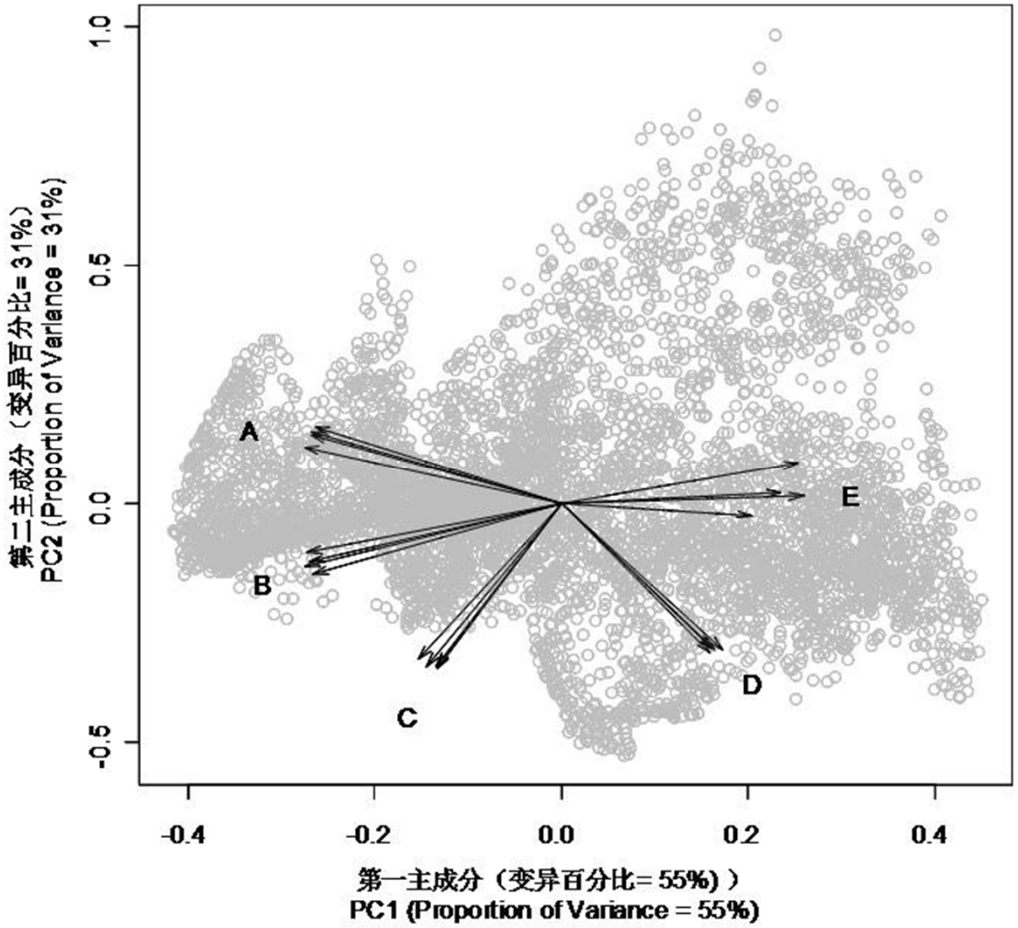
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
latitude	0.366	-0.519	0.19	0.206	0.143		0.696	-0.113
aspect		-0.119	-0.836	-0.119	0.499	-0.138		
elevation	0.363	-0.541	0.133	0.127	0.211		-0.686	0.165
footprint	-0.406	-0.252			0.178	0.853		
year	-0.23	0.254		0.859	0.372			
GDP	-0.457	-0.406			-0.212	-0.338	-0.14	-0.665
pop	-0.485	-0.357			-0.211	-0.228	0.137	0.716
slope	0.267		-0.489	0.415	-0.653	0.293		

R code: PCA

pca2$scores

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
1	-11.732	-6.269	-1.487	-0.016	-0.722
42	-13.055	-8.397	-1.325	0.868	-2.854
86	-9.641	-4.353	-0.257	-0.166	-0.521
104	-2.673	-2.746	0.447	0.537	-0.290
105	-2.762	-2.646	0.466	0.839	-0.160
116	-6.773	-2.977	1.060	-0.025	-1.138
117	-6.814	-2.932	1.071	0.128	-1.072

Example: study on climate change consequences to the crested ibis



The loadings of 20 variables (five climate variables, i.e. annual total precipitation (A), annual minimum temperature (B), annual maximum temperature (C), seasonal variance of temperature (D), and seasonal variance of precipitation (E), at four time periods, i.e. present, 2020, 2050, and 2080 at the first and second principal components space. The grey circles are the scores of 5751 sites in Yang county at the first and second principal components space.

PCA: potential problems

- Lack of independence
  - **No problem**
- Lack of normality
  - Normality desirable but not essential
- Lack of precision
  - Precision desirable but not essential
- Many zeroes in data matrix
  - **Problem** (use correspondence analysis)

Note

- The principal components are dependent on the units used to measure the original variables as well as on the range of values they assume.

- We **usually** standardize the data prior to using PCA.

Hourly records of sperm whale behaviour

- Variables:
  - Mean cluster size
  - Max. cluster size
  - Mean speed
  - Heading consistency
  - Fluke-up rate
  - Breach rate
  - Lobtail rate
  - Spyhop rate
  - Sidefluke rate
  - Coda rate
  - Creak rate
  - High click rate
- Data collected:
  - Off Galapagos Islands
  - 1985 and 1987
- Units:
  - hours spent following sperm whales
  - 440 hours

Whitehead, Hal and Weilgart, Linda. 1991. Patterns of Visually Observable Behaviour and Vocalizations in Groups of Female Sperm Whales. Behaviour 118 (3): 275-296

Principal Components

	Principal Components:			
	1	2	3	4
% of variance accounted for	31.09	13.41	12.08	10.52
Loadings:				
Mean cluster size	0.82	0.35	0.01	-0.14
Max. cluster size	0.83	0.24	0.17	-0.12
Mean speed	-0.38	0.30	0.44	-0.09
Heading consistency	-0.48	0.39	0.19	-0.04
Fluke-up rate	-0.65	-0.19	0.30	0.25
Breach rate	0.24	0.24	-0.13	0.74
Lobtail rate	0.29	0.30	-0.09	0.71
Spyhop rate	0.46	-0.60	-0.23	0.01
Sidefluke rate	0.49	-0.57	-0.20	0.07
Coda rate	0.68	-0.08	0.53	0.03
Creak rate	0.57	-0.11	0.70	0.02
High click rate	-0.41	-0.55	0.47	0.31

Whitehead, Hal and Weilgart, Linda. 1991. Patterns of Visually Observable Behaviour and Vocalizations in Groups of Female Sperm Whales. Behaviour 118 (3): 275-296

	Principal Components:			
	1	2	3	4
% of variance accounted for	31.09	13.41	12.08	10.52
Loadings:				
Mean cluster size	0.82	0.35	0.01	-0.14
Max. cluster size	0.83	0.24	0.17	-0.12
Mean speed	-0.38	0.30	0.44	-0.09
Heading consistency	-0.48	0.39	0.19	-0.04
Fluke-up rate	-0.65	-0.19	0.30	0.25
Breach rate	0.24	0.24	-0.13	0.74
Lobtail rate	0.29	0.30	-0.09	0.71
Spyhop rate	0.46	-0.60	-0.23	0.01
Sidefluke rate	0.49	-0.57	-0.20	0.07
Coda rate	0.68	-0.08	0.53	0.03
Creak rate	0.57	-0.11	0.70	0.02
High click rate	-0.41	-0.55	0.47	0.31
Principal Components meanings	"Socializing/ foraging"	"Directed movement"	"Vocal"	"Aerial"

Whitehead, Hal and Weilgart, Linda. 1991. Patterns of Visually Observable Behaviour and Vocalizations in Groups of Female Sperm Whales. Behaviour 118 (3): 275-296

Example

In this example wildlife (moose) population density was measured over time (once a year) in three areas.

Year	Area 1	Area 2	Area 3	Year	Area 1	Area 2	Area 3
1	11.3	14.1	6.9	13	6.1	9.9	6.8
2	10.4	14	11.2	14	9.7	13.2	6.6
3	9.9	13	8.7	15	8.1	9.4	4
4	8.2	11.4	3.3	16	11.3	11.8	4.9
5	10.1	11.9	8.7	17	8.8	11.5	8.8
6	10.7	13.8	12.5	18	9.4	11.6	5.7
7	11	14.9	8.9	19	7.5	11.4	4.9
8	7.1	8.5	3.7	20	8.8	10.7	7.2
9	14.7	14.5	12.1	21	7.5	11.1	7
10	5.4	9	4.1	22	9.1	13.2	8.9
11	7.3	7.6	5.6	23	6.8	9.8	7.6
12	10.2	10.9	7.3				

From Laverty's presentation "Techniques for studying correlation and covariance structure"

Habitats

From Laverty's presentation "Techniques for studying correlation and covariance structure"

The Sample Statistics

$$\vec{\bar{x}} = \begin{bmatrix} 9.10 \\ 11.62 \\ 7.19 \end{bmatrix}$$

The mean vector

$$S = \begin{bmatrix} 4.297 & 3.307 & 3.295 \\ & 3.527 & 3.527 \\ & & 6.566 \end{bmatrix}$$

The covariance matrix

$$R = \begin{bmatrix} 1 & .796 & .620 \\ & 1 & .687 \\ & & 1 \end{bmatrix}$$

The correlation matrix

From Laverty's presentation "Techniques for studying correlation and covariance structure"

Principal component Analysis

$$S = \begin{bmatrix} 4.297 & 3.307 & 3.295 \\ & 3.527 & 3.527 \\ & & 6.566 \end{bmatrix}$$

The eigenvalues of S

$$\lambda_1 = 11.85974, \quad \lambda_2 = 2.204232, \quad \lambda_3 = 0.814249$$

The eigenvectors of S

$$\vec{a}_1 = \begin{bmatrix} .522 \\ .523 \\ .674 \end{bmatrix}, \ \vec{a}_2 = \begin{bmatrix} .582 \\ .359 \\ -.730 \end{bmatrix}, \ \vec{a}_3 = \begin{bmatrix} .624 \\ -.733 \\ .117 \end{bmatrix}$$

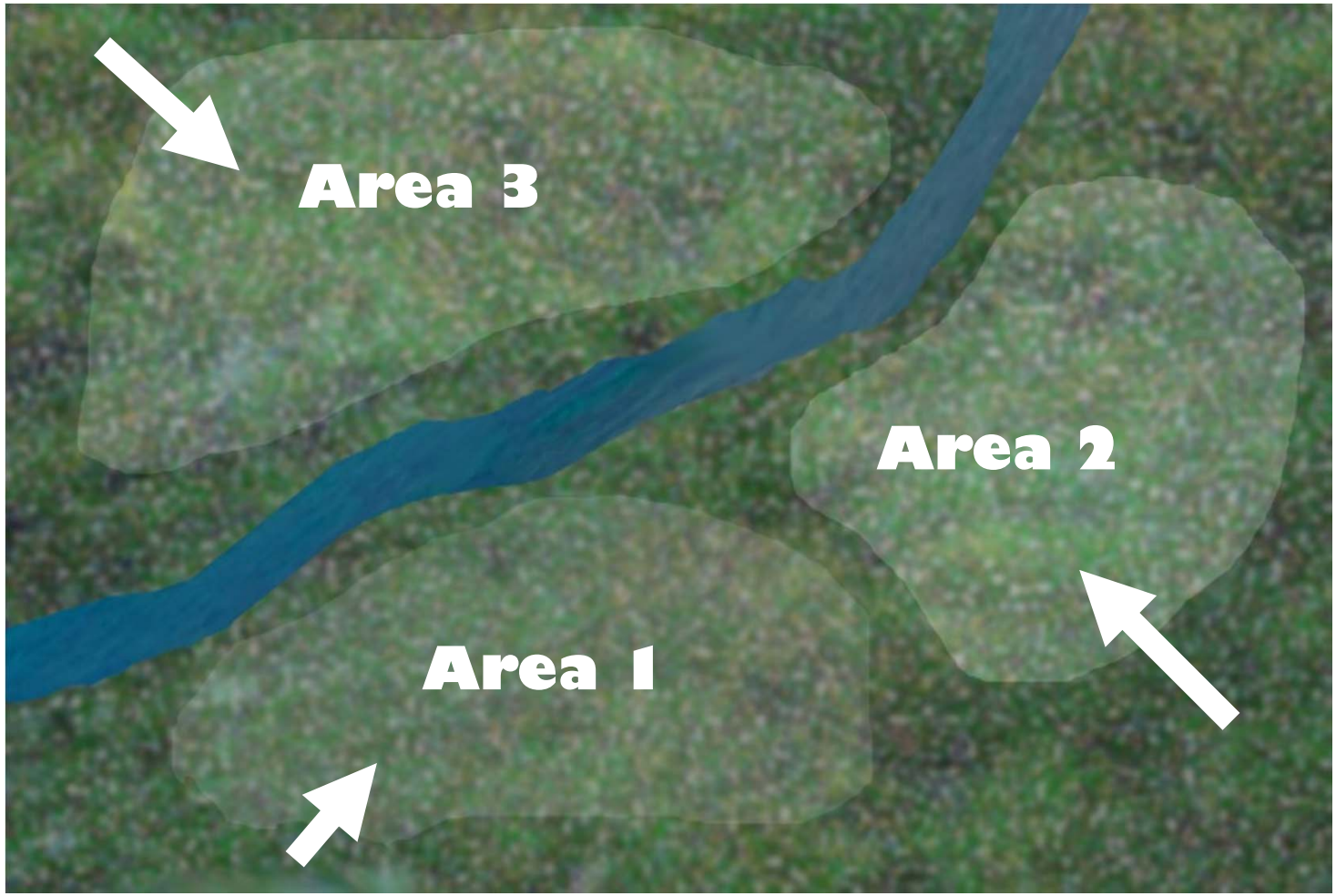
The principal components

$$C_1 = .522x_1 + .523x_2 + .674x_3$$
$$C_2 = .582x_1 + .359x_2 - .730x_3$$
$$C_3 = .624x_1 - .733x_2 + .117x_3$$

From Laverty's presentation "Techniques for studying correlation and covariance structure"

Example 1/3

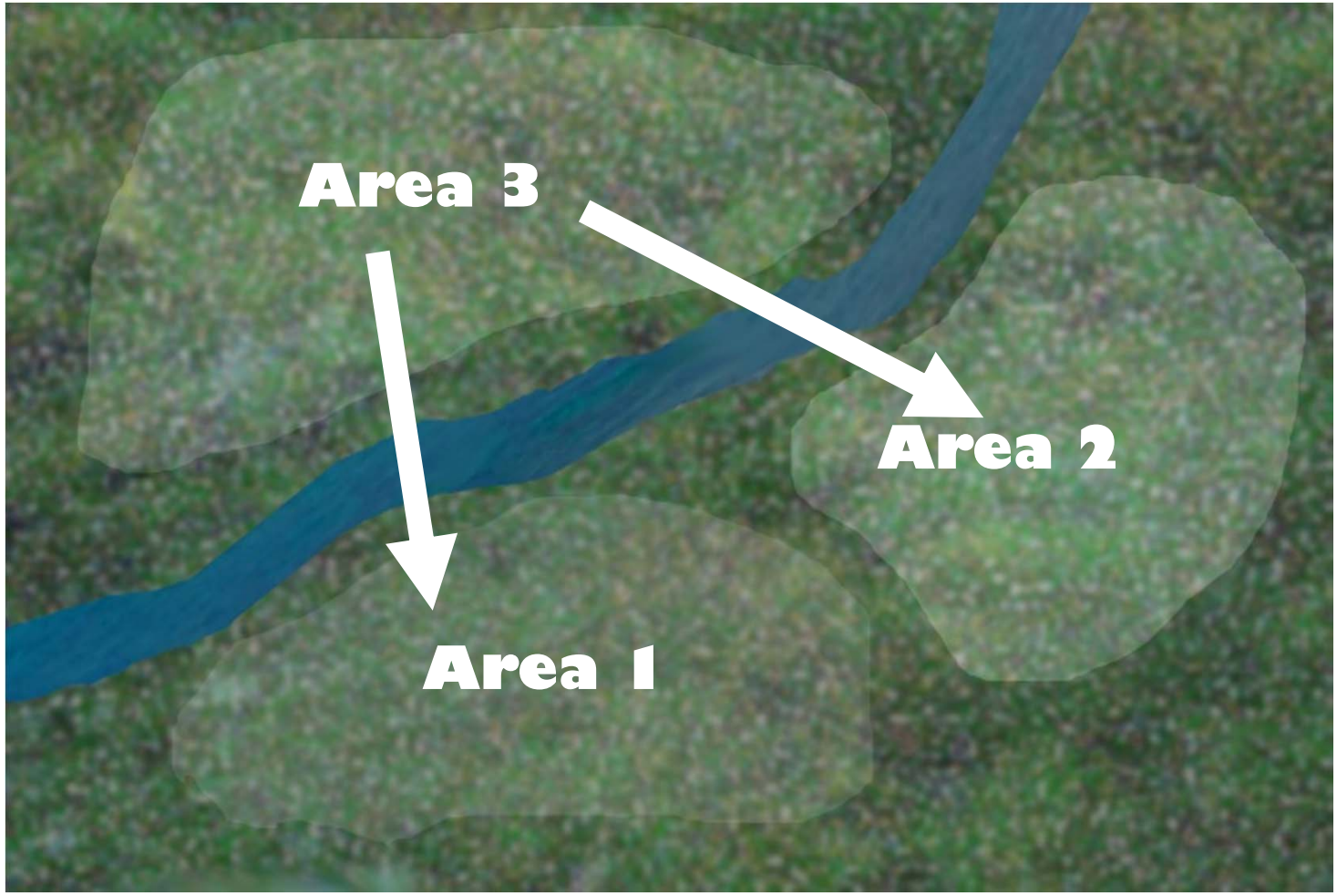
$$C_1 = .522x_1 + .523x_2 + .674x_3$$



From Laverty's presentation "Techniques for studying correlation and covariance structure"

Example 2/3

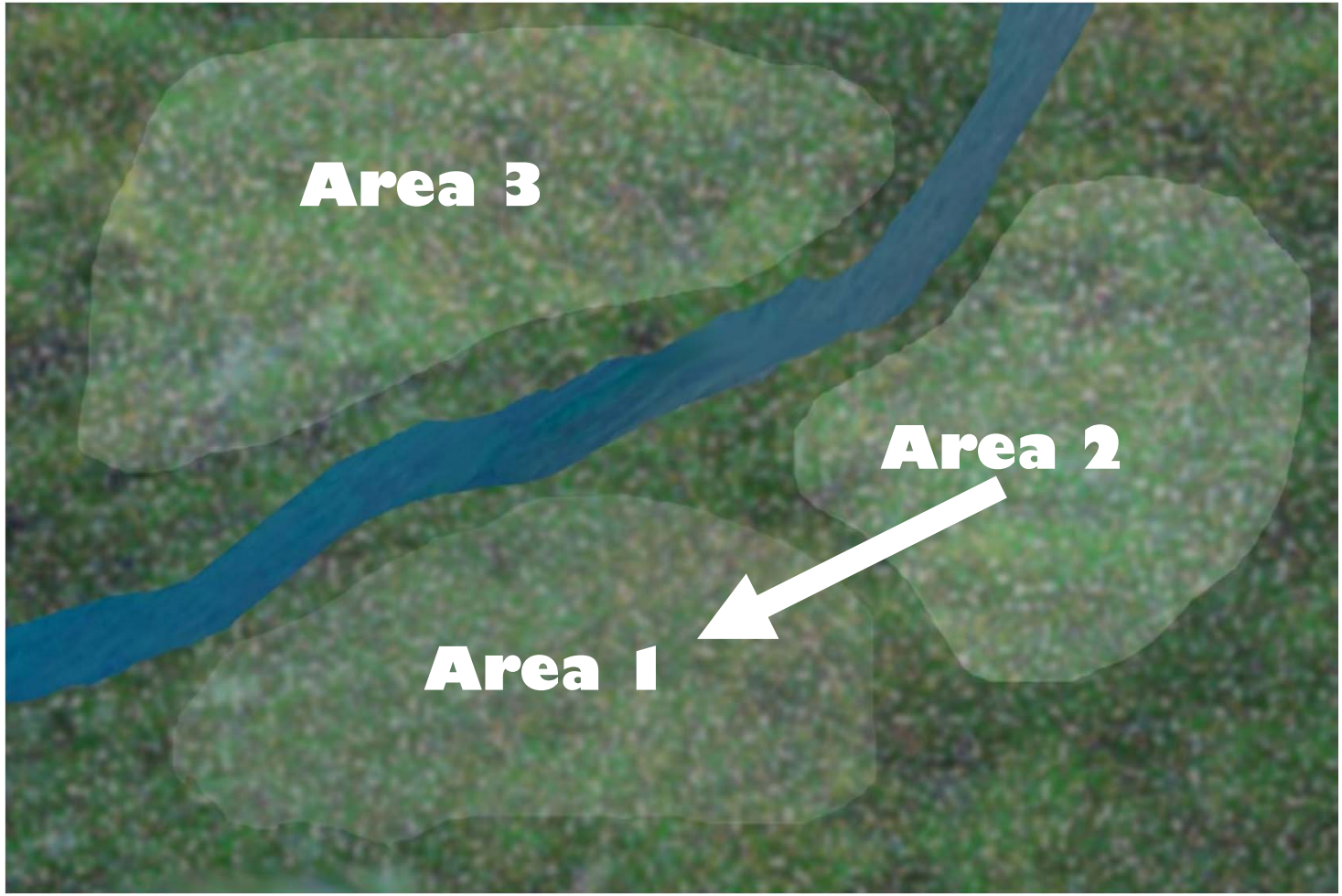
$$C_2 = .582x_1 + .359x_2 - .730x_3$$



From Laverty's presentation "Techniques for studying correlation and covariance structure"

Example 3/3

$$C_3 = .624x_1 - .733x_2 + .117x_3$$



From Laverty's presentation "Techniques for studying correlation and covariance structure"

Factor Analysis (FA)

Factor Analysis

- Data reduction tool
- Removes redundancy or duplication from a set of correlated variables
- Represents correlated variables with a smaller set of “derived” variables.
- Factors are formed that are relatively independent of one another.
- Two types of “variables”:
  - latent variables: factors
  - observed variables

Some applications of factor analysis

1. Identification of underlying factors:
   - clusters variables into homogeneous sets
   - creates new variables (i.e. factors)
   - allows us to gain insight to categories
2. Screening of variables:
   - identifies groupings to allow us to select one variable to represent many
   - useful in regression (recall collinearity)
3. Summary:
   - Allows us to describe many variables using a few factors
4. Clustering of objects:
   - Helps us to put objects (people) into categories depending on their factor scores

R code: Factor Analysis

	latitude	aspect	elevation	footprint	year	GDP	pop	slope
1	33.1	0.893	476	61	2008	333	2032	0.503
42	33.3	0.798	484	38	2007	420	3049	0.685
86	33.1	0.56	473	60	2008	256	1485	0.812
104	33.4	0.502	942	20	2006	186	488	5.002
105	33.4	0.502	942	20	2008	186	488	5.002
116	33.2	0.201	476	44	2006	169	1321	2.275

```
# Exploratory Factor Analysis (Maximum Likelihood)
# extracting 3 factors, with varimax rotation
fit <- factanal(ibis.pre, 3, rotation="varimax")
print(fit, digits=2, cutoff=.01, sort=TRUE)
```

Call:
factanal(x = ibis.pre, factors = 3, rotation = "varimax")

Uniquenesses:

y	aspect	elevation	footprint	year	GDP	pop	slope
0.20	1.00	0.20	0.64	0.90	0.00	0.00	0.90

Loadings:

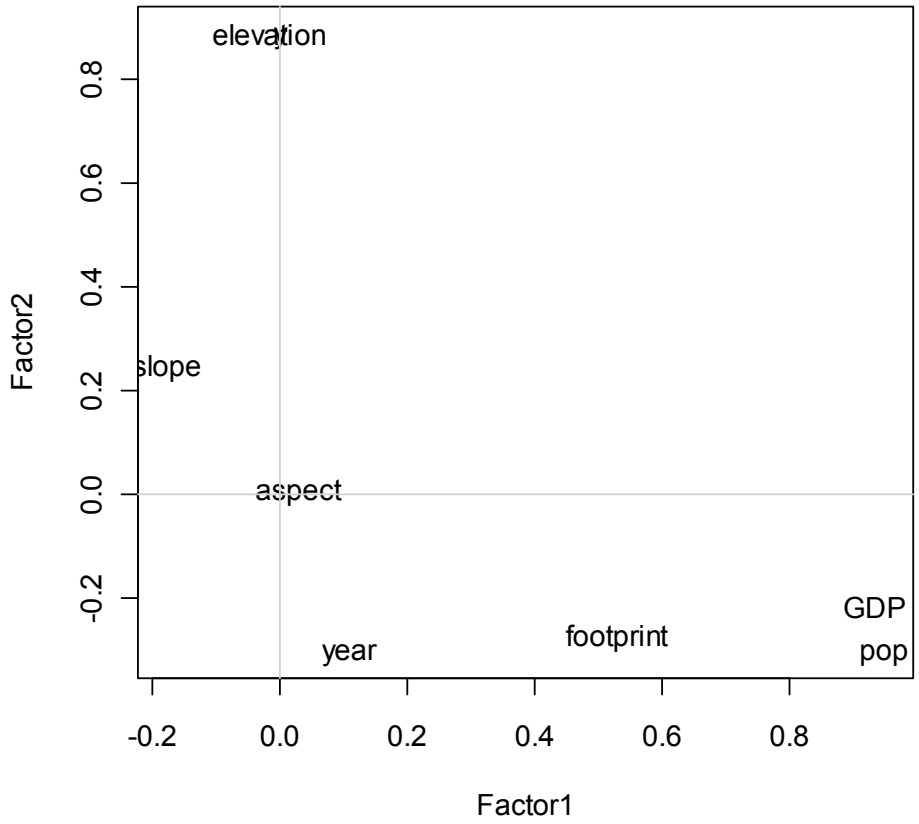
	Factor1	Factor2	Factor3
footprint	0.53	-0.27	0.07
GDP	0.93	-0.21	-0.29
pop	0.95	-0.30	0.06
y		0.89	0.12
elevation	-0.02	0.89	-0.05
aspect	0.03		
year	0.11	-0.30	0.04
slope	-0.18	0.25	0.07

	Factor1	Factor2	Factor3
SS loadings	2.09	1.95	0.11
Proportion Var	0.26	0.24	0.01
Cumulative Var	0.26	0.50	0.52

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 50.9 on 7 degrees of freedom.
The p-value is 9.78e-09

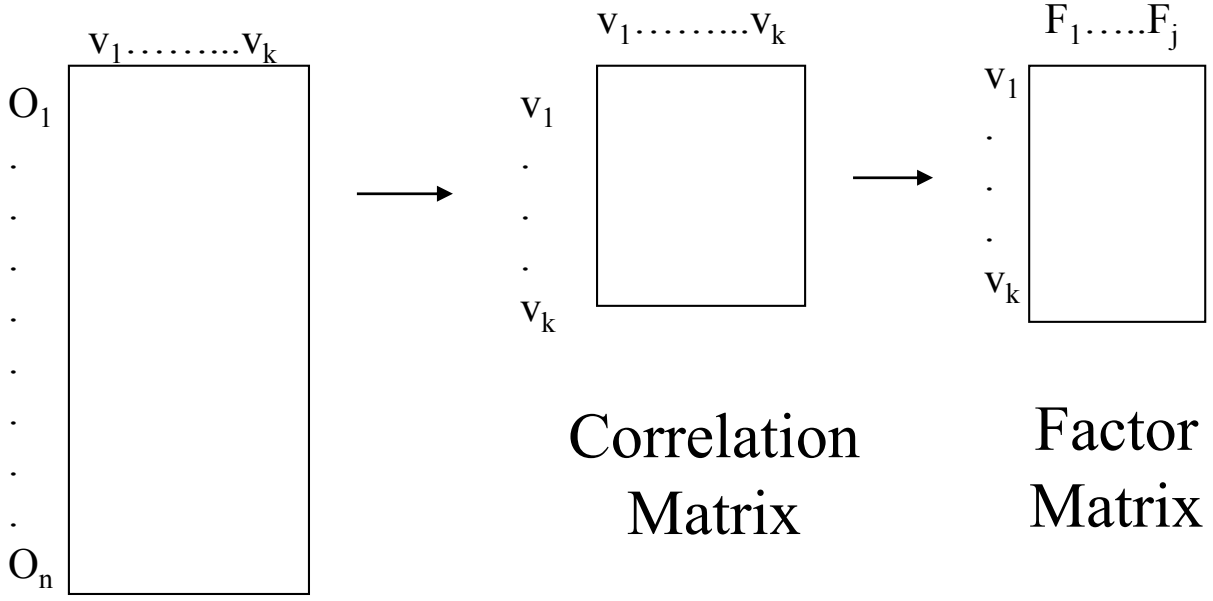
R: Result

```
# plot factor 1 by factor 2
load <- fit$loadings[,1:2]
plot(load,type="n") # set up plot
text(load,labels=names(ibis.pre),cex=1) # add variable names
abline(h = -1:1, v = -1:1, col = "lightgray", lty=1)
```



Data Matrix

- Factor analysis is **totally dependent** on correlations between variables.
- Factor analysis summarizes correlation structure



Choosing number of factors

- Intuitively: The number of uncorrelated constructs that are jointly measured by the X's.
- Only useful if number of factors is less than number of X's (recall "data reduction").

Use "principal components" to help decide

- number of factors is equivalent to number of variables
- each factor is a weighted combination of the input variables:

$$F_1 = a_{11}X_1 + a_{12}X_2 + \ldots.$$

Eigenvalues

- To select how many factors to use, consider *eigenvalues* from a principal components analysis
- Two interpretations:
  - eigenvalue ≅ equivalent number of variables which the factor represents
  - eigenvalue ≅ amount of variance in the data described by the factor.
- Rules to go by:
  - number of eigenvalues > 1
  - scree plot
  - % variance explained
  - comprehensibility
- Note: sum of eigenvalues is equal to the number of items

Steps in Factor Analysis

- Factor analysis usually proceeds in four steps:
  - 1: the correlation matrix for all variables is computed
  - 2: Factor extraction
  - 3: Factor rotation
  - 4: Make final decisions about the number of underlying factors

The Correlation Matrix

- **1: the correlation matrix**
  - Generate a correlation matrix for all variables
  - Identify variables not related to other variables
  - If the correlation between variables are small, it is unlikely that they share common factors (variables must be related to each other for the factor model to be appropriate).
  - Correlation coefficients greater than 0.3 in absolute value are indicative of acceptable correlations.
  - Examine visually the appropriateness of the factor model.

The Correlation Matrix

- **Bartlett Test of Sphericity:**
  - used to test the hypothesis the correlation matrix is an **identity matrix** (all diagonal terms are 1 and all off-diagonal terms are 0).
  - If the **value** of the test statistic for **sphericity is large** and the associated **significance level is small**, it is **unlikely** that the population correlation matrix is an identity.
- If the hypothesis that the population correlation matrix is an identity can be rejected because the observed significance level is large, the use of the factor model should be reconsidered.

Factor Extraction

- **2nd Step: Factor extraction**
  - The primary objective of this stage is to determine the factors.
  - Initial decisions can be made here about the number of factors underlying a set of measured variables.
  - Estimates of initial factors are obtained using **Principal components analysis**.
  - The principal components analysis is the most commonly used extraction method . Other factor extraction methods include:
    - Maximum likelihood method
    - Principal axis factoring
    - Alpha method
    - Unweighted lease squares method
    - Generalized least square method
    - Image factoring.

Factor Extraction

- In principal components analysis, **linear combinations** of the observed variables are formed.
- The 1st principal component is the combination that accounts for the **largest amount of variance** in the sample (1st extracted factor).
- The 2nd principle component accounts for the next largest amount of variance and **is uncorrelated with the first** (2nd extracted factor).
- Successive components explain progressively smaller portions of the total sample variance, and all are uncorrelated with each other.

Factor Extraction

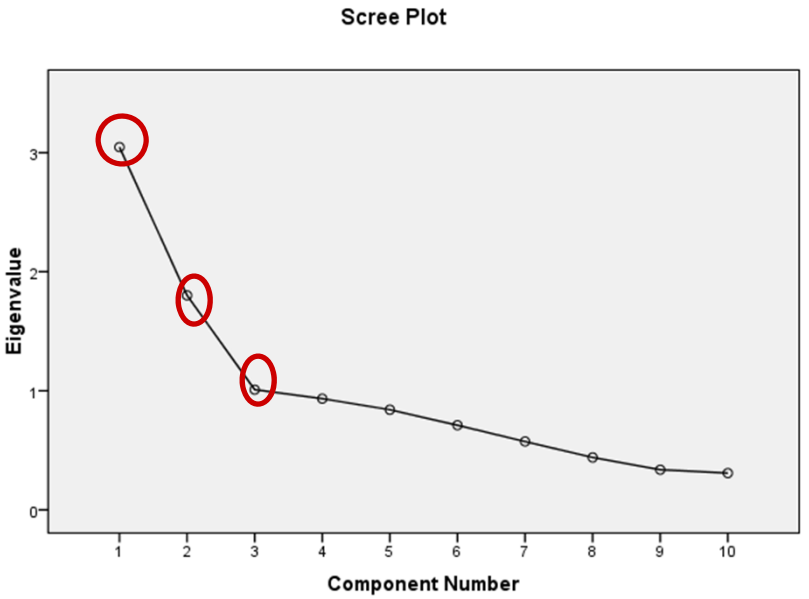
- To decide on how many factors we need to represent the data, we use 2 statistical criteria:
  - **Eigen Values,** and
  - The **Scree Plot**.

- The determination of the number of factors is usually done by considering only factors with Eigen values greater than 1.

- Factors with a variance less than 1 are no better than a single variable, since each variable is expected to have a variance of 1.

Factor Extraction

- The examination of the **Scree plot** provides a visual of the total variance associated with each factor.
- The steep slope shows the large factors.
- The gradual trailing off (scree) shows the rest of the factors usually lower than an Eigen value of 1.
- In choosing the number of factors, in addition to the statistical criteria, one should make initial decisions based on conceptual and theoretical grounds.
- At this stage, the decision about the number of factors is not final.



Factor Rotation

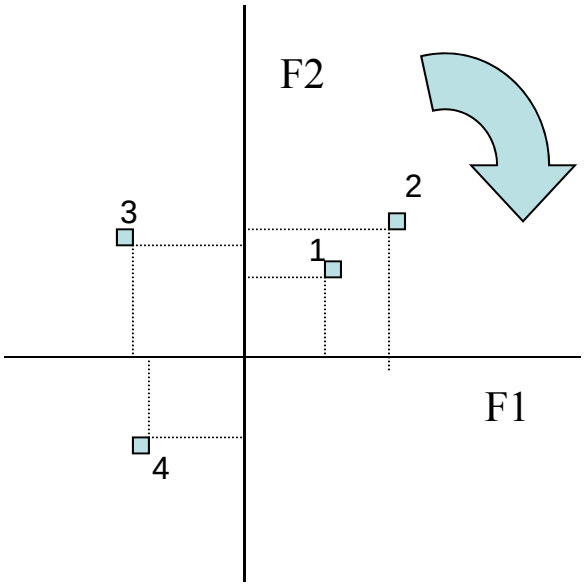
- **3rd Step: Factor rotation.**
- In this step, factors are rotated.

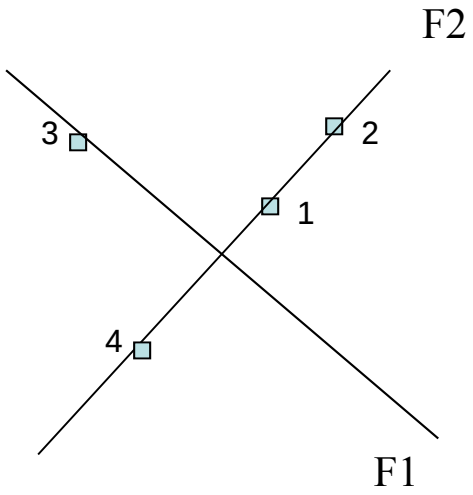
- Un-rotated factors are typically not very interpretable (most factors are correlated with may variables).

- Factors are rotated to make them more meaningful and easier to interpret (each variable is associated with a minimal number of factors).

- Different rotation methods may result in the identification of somewhat different factors.

Rotating Factors (Intuitively)





	Factor 1	Factor 2
x1	0.5	0.5
x2	0.8	0.8
x3	-0.7	0.7
x4	-0.5	-0.5

	Factor 1	Factor 2
x1	0	0.6
x2	0	0.9
x3	-0.9	0
x4	0	-0.9

Factor Rotation

- Quartimax
  - Simplify rows – variable loads high on one factor, low on others
- Varimax
  - Simplify columns – clearer separation of factors – each factor has variables that either load high or load very low
- Equimax
  - Compromise between the two – rarely used

- The most popular rotational method is **Varimax rotations.**
- Varimax use orthogonal rotations yielding uncorrelated factors/components.
- Varimax attempts to minimize the number of variables that have high loadings on a factor. This enhances the interpretability of the factors.

Making Final Decisions

- 4th Step: Making final decisions
  - The final decision about the number of factors to choose is the number of factors for the rotated solution that is **most interpretable**.
  - To identify factors, group variables that have large loadings for the same factor.
  - Plots of loadings provide a visual for variable clusters.
  - Interpret factors according to the meaning of the variables

- This decision should be guided by:
  - A priori conceptual beliefs about the number of factors from past research or theory
  - Eigen values computed in step 2.
  - The relative interpretability of rotated solutions computed in step 3.

Assumptions

- Assumption underlying factor analysis include.
  - The measured variables are linearly related to the factors + errors.

    This assumption is likely to be violated if items limited response scales (two-point response scale like True/False, Right/Wrong items).
  - The data should have a bi-variate normal distribution for each pair of variables.
  - Observations are independent.
  - The factor analysis model assumes that variables are determined by common factors and unique factors. All unique factors are assumed to be **uncorrelated** with each other.

Factor Analysis (FA) vs. PCA

- PCA analyzes variance
- FA analyzes covariance (communality)
- PCA – the goal is to extract as much variance with the least amount of factors
- FA – the goal is to explain as much of the correlations with a minimum number of factors
- PCA gives a unique solution. If all components retained, all variance explained
- FA can give multiple solutions depending on the method and the estimates of communality

Factor Analysis vs. Cluster Analysis

- Factor Analysis and Cluster Analysis are both data reduction techniques.
- Goal of Factor Analysis is to reduce original set of variables to smaller set of factors.
- Goal of Cluster Analysis is to form groups from the people or objects, thus reducing original number of elements to fewer groups.
- Factor Analysis can be seen as a clustering technique than is focused on the columns of data matrix, rather than the rows.

Correspondence Analysis (CA)

Correspondence Analysis

- Known also as Reciprocal Averaging (Hill 1973)
- Weighted averaging of site scores to yield species scores and vice versa
- Simultaneous ordination of both rows and columns of a matrix
- Used to examine relationship of species assemblages to site characteristics
- Sites typically span an environmental gradient

Correspondence Analysis

- Based on traditional Chi-Square approach to measure correspondence between rows and columns.

Observed

site	sp1	sp2	sp3	sp4	Row Total
sam1	1	1	2	2	6
sam2	2	2	4	4	12
sam3	3	3	6	6	18
sam4	4	4	8	8	24
sam5	5	5	10	10	30
Col Total	15	15	30	30	90

$\frac{15*12}{90}$

Expected

Expected	sp1	sp2	sp3	sp4
sam1	1	1	2	2
sam2	2	2	4	4
sam3	3	3	6	6
sam4	4	4	8	8
sam5	5	5	10	10

$\frac{(2-2)^2}{2}$

Chi-Sqr

Chi-Sqr				
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

Chi Sqr	Inertia
0	0

Inertia = Chi Sqr/Grand Sum

In this simple dataset, rows and columns are not independent. Thus, the contents of each cell are predictable based on row and column totals and the grand total.

Correspondence Analysis

- As rows and column deviate (more independent), Chi Sqr values (and inertia) grows.

Observed

site	sp1	sp2	sp3	sp4	Row Total
sam1	4	2	3	2	11
sam2	4	3	7	4	18
sam3	25	10	12	4	51
sam4	18	24	33	13	88
sam5	10	6	7	2	25
Col Total	61	45	62	25	193

$$\frac{18*45}{193}$$

Expected

Expected	sp1	sp2	sp3	sp4
sam1	3.476684	2.564767	3.533679	1.42487
sam2	5.689119	4.196891	5.782383	2.331606
sam3	16.11917	11.89119	16.38342	6.606218
sam4	27.81347	20.51813	28.26943	11.39896
sam5	7.901554	5.829016	8.031088	3.238342

$$\frac{(3-4.1968)^2}{4.1968}$$

Chi-Sqr

Chi-Sqr				
	0.07877	0.124363	0.0806	0.232143
	0.501505	0.341336	0.256398	1.193828
	4.892877	0.300778	1.172794	1.028178
	3.462503	0.590862	0.791607	0.224873
	0.557292	0.005016	0.132378	0.473542

Chi Sqr	Inertia
16.44164	0.08519

Inertia = Chi Sqr/Grand Sum 16.44164/193 = 0.0851

Correspondence Analysis

- As rows and column deviate (more independent), chi sqr values (and inertia) grows.

Observed

site	sp1	sp2	sp3	sp4	Row Total
sam1	4	2	3	2	11
sam2	4	3	7	4	18
sam3	25	10	12	4	51
sam4	18	24	33	13	88
sam5	10	6	7	2	25
Col Total	61	45	62	25	193

Expected

Expected	sp1	sp2	sp3	sp4
sam1	3.476684	2.564767	3.533679	1.42487
sam2	5.689119	4.196891	5.782383	2.331606
sam3	16.11917	11.89119	16.38342	6.606218
sam4	27.81347	20.51813	28.26943	11.39896
sam5	7.901554	5.829016	8.031088	3.238342

Chi-Sqr

Chi-Sqr				
	0.07877	0.124363	0.0806	0.232143
	0.501505	0.341336	0.256398	1.193828
	4.892877	0.300778	1.172794	1.028178
	3.462503	0.590862	0.791607	0.224873
	0.557292	0.005016	0.132378	0.473542

Chi Sqr	Inertia
16.44164	0.08519

This matrix describes all the variability in the dataset not explainable by row or column profiles (totals).

Total variance the analysis will attempt to explain.

Correspondence Analysis

Chi-Sqr	sp1	sp2	sp3	sp4	Row totals
sam1	0.07877	0.124363	0.0806	0.232143	*0.515876*
sam2	0.501505	0.341336	0.256398	1.193828	*2.293067*
sam3	4.892877	0.300778	1.172794	1.028178	*7.394627*
sam4	3.462503	0.590862	0.791607	0.224873	*5.069845*
sam5	0.557292	0.005016	0.132378	0.473542	*1.168228*
Col totals	*9.492948*	*1.362354*	*2.433777*	*3.152565*	

Chi Sqr	Inertia
16.44164	0.08519

Look at where the variability is in the Chi Sqr matrix…

```
Partitioning of mean squared contingency coefficient:
              Inertia Proportion
Total         0.08519          1
Unconstrained 0.08519          1
…
Importance of components:
                          CA1    CA2      CA3
Eigenvalue             0.0748 0.0100 0.000414
Proportion Explained   0.8776 0.1176 0.004850
Cumulative Proportion  0.8776 0.9951 1.000000
…
Species scores
          CA1       CA2        CA3
sp1 -0.39331 -0.030492 -0.0008905
sp2  0.09946  0.141064  0.0219980
sp3  0.19632  0.007359 -0.0256591
sp4  0.29378 -0.197766  0.0262108

Site scores (weighted averages of species scores)
         CA1     CA2     CA3
sam1 -0.2405 -1.9357  3.4903
sam2  0.9471 -2.4310 -1.6574
sam3 -1.3920 -0.1065 -0.2535
sam4  0.8520  0.5769  0.1625
sam5 -0.7355  0.7884 -0.3974
```

CA does an eigenvalue decomposition to summarize this variability in fewer axes (components).

Species and sites that contribute most to the inertia have the largest magnitude CA1 scores.

Scores are centered and scaled to be directly comparable.

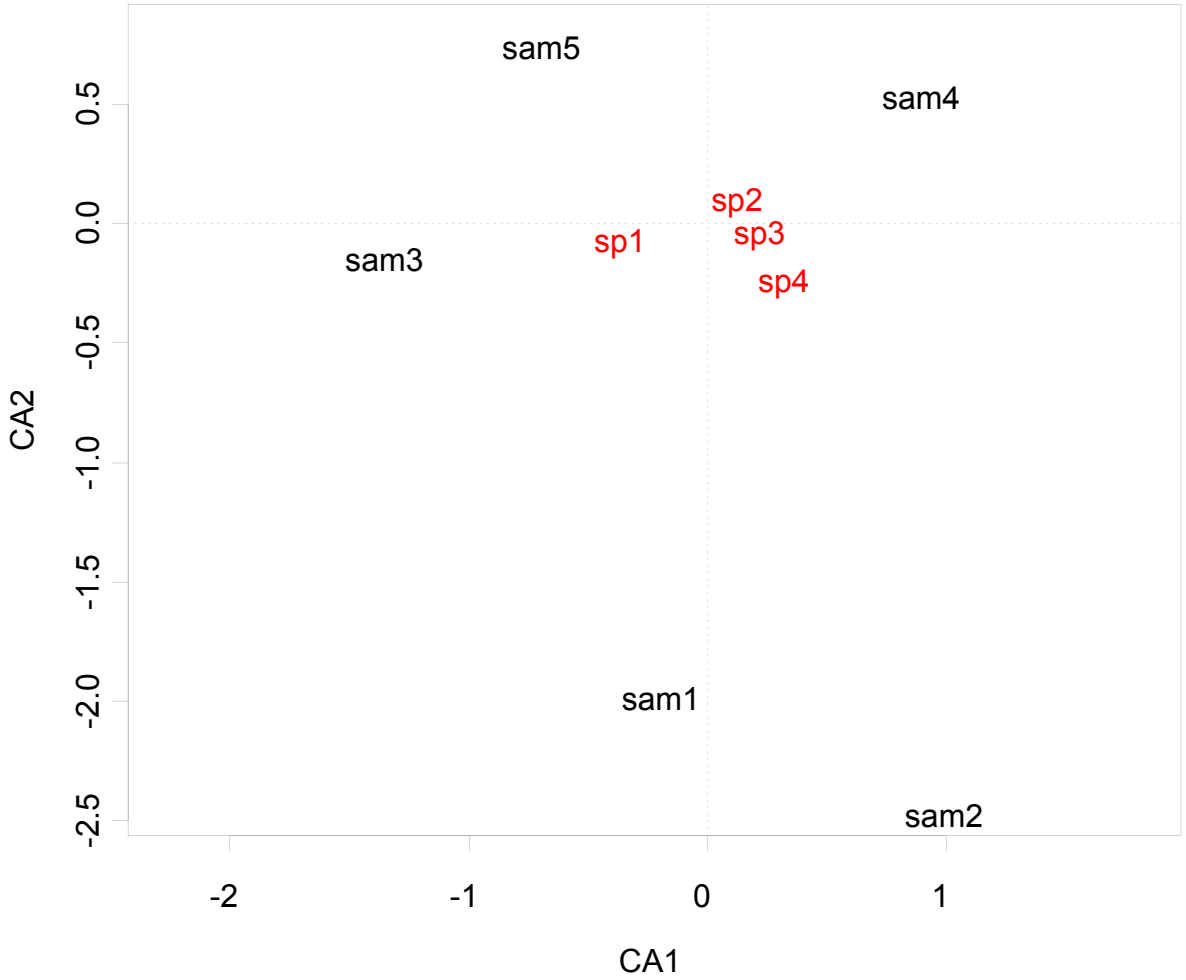
Correspondence Analysis

- Output
  - Row and column sums, total Chi Square
  - Species and sample scores that can be plotted in the same space. Interpretation is similar to sample scores and species weighted averages in NMDS.
  - # axes = n-1 for whichever dimension of the data matrix is lower (samples or species).
  - Eigenvalues – relative importance of each axis, interpreted as the percentage of total **inertia** explained.

```
Partitioning of mean squared contingency coefficient:
              Inertia Proportion
Total           1.780          1
Unconstrained   1.780          1

Eigenvalues, and their contribution to the mean squared contingency coefficient

Importance of components:
                          CA1   CA2   CA3    CA4    CA5     CA6     CA7     CA8     CA9
Eigenvalue              0.850 0.530 0.252 0.0953 0.0303 0.01097 0.00638 0.00307 0.00179
Proportion Explained    0.478 0.298 0.142 0.0536 0.0170 0.00616 0.00358 0.00173 0.00101
Cumulative Proportion 0.478 0.775 0.917 0.9705 0.9875 0.99368 0.99727 0.99899 1.00000
```

0.85/1.780=0.478

Correspondence Analysis



Distances between species are two-dimensional approximations of their Chi-square distances. Distances between samples are also two-dimensional approximations of Chi-square distances. Distances between species and sites cannot be interpreted.

How does CA work?

- Site-species matrix
- Eigenanalysis

  similar to PCA but differs in some details

  axes rotated through species and sample space with the goal of maximizing their correspondence
- Reciprocal averaging

  calculate species scores as weighted averages of the sites in which they occur

  calculate new site scores by weighted averaging of the species scores

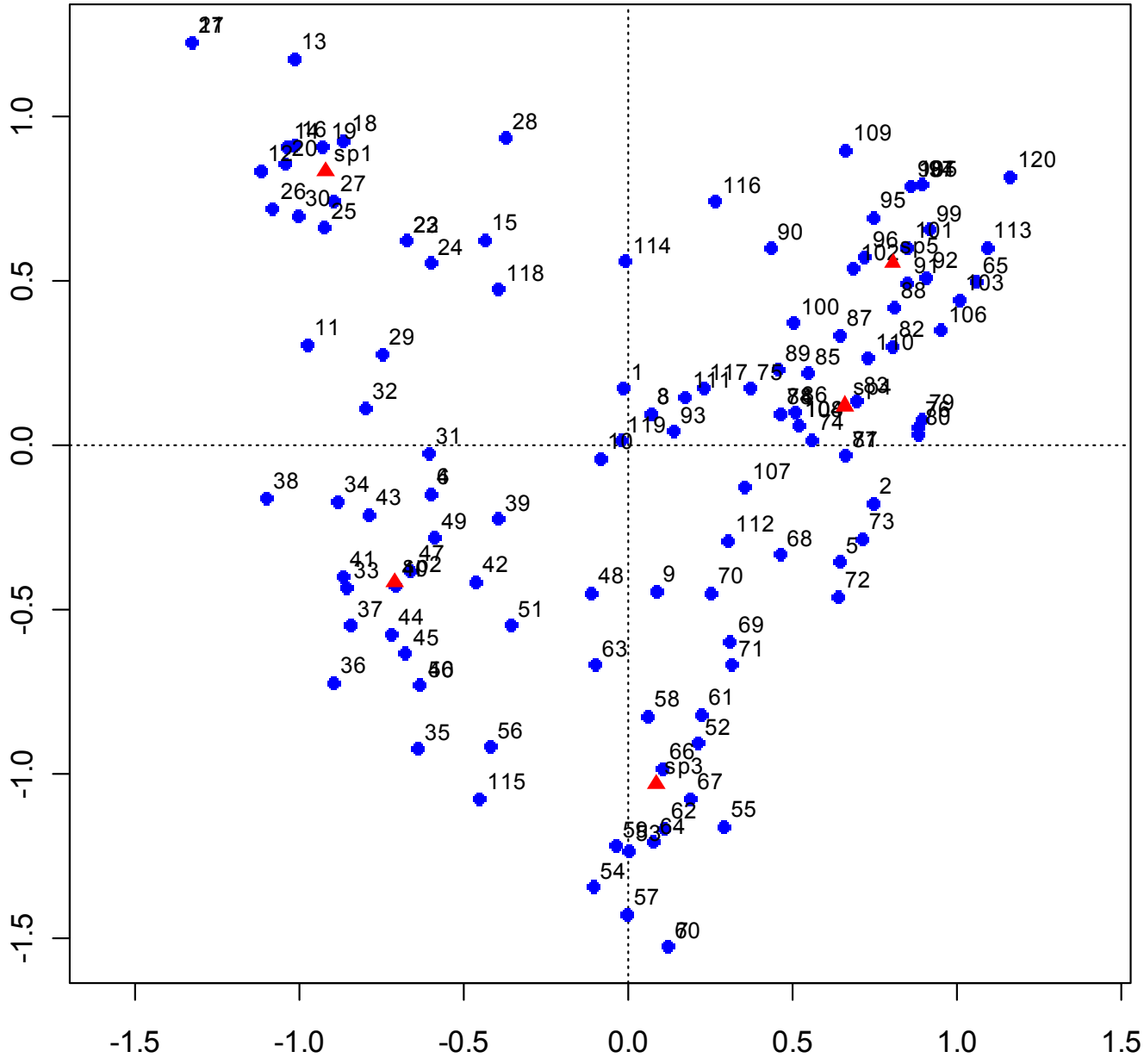
Output of CA

- Yields principal axes and scores
  - scores for rows (sites) and columns (species)
  - 1st axis has the largest eigenvalue (and accounts for largest variance); maximizes association between rows and columns
  - subsequent axes account for residual variation and have smaller eigenvalues
  - rarely use more than 2-3 axes in CA

R code: Correspondence analysis

```
library(ca) # Package for Correspondence analysis
data(author); ca(author); plot(ca(author)) # One example
ibis.ca = read.csv('D:/database/nestsite_parameters.csv', header=T)
names.ibis = ibis.ca[,1]
ibis.ca = ibis.ca[,-1]; head(ibis.ca)
rownames(ibis.ca) = names.ibis
ca(ibis.ca); plot(ca(ibis.ca))

# Generate 5 species density data at 120 sites
sp1 = round(rnorm(120, 20, 7))
sp2 = round(rnorm(120, 40, 7))
sp3 = round(rnorm(120, 60, 7))
sp4 = round(rnorm(120, 80, 7))
sp5 = round(rnorm(120, 100,7))
species = data.frame(sp1, sp2, sp3, sp4, sp5)
species = species * 0
sp.1=table(sp1)
sp.2=table(sp2)
sp.3=table(sp3)
sp.4=table(sp4)
sp.5=table(sp5)
species$sp1[c(as.numeric(names(sp.1)))] = sp.1
species$sp2[c(as.numeric(names(sp.2)))] = sp.2
species$sp3[c(as.numeric(names(sp.3)))] = sp.3
species$sp4[c(as.numeric(names(sp.4)))] = sp.4
species$sp5[c(as.numeric(names(sp.5)))] = sp.5 ; sp.5 = sp.5[1:(nrow(sp.5)-2)]
random = matrix(sample(c(0,1),600, rep=T),nrow=120, ncol=5) #Generate a noise
species = species + random
rownames(species) = c(1:120) #define row names
ca(species) #Correspondence analysis
plot(ca(species))
```

Five species at 120 sites



Correspondence Analysis

- First axis always most informative
- Number of axes produced is set by the dimensionality of the data, not a user option
- Not distance based, data transformations typically more important
- Ordinates both samples and species directly

CA vs. PCA

- Both are eigenvector methods
- PCA uses Euclidean distance, while CA uses Chi-square distance
- Underlying assumption – species abundance distributions are Gaussian (normal and unimodal)
- CA orders the points correctly on the first axis
- Curvature and “tucking in” of the ends of the gradient in PCA results in failure to order points correctly (horseshoe effect)

Advantage of CA

- Weaknesses of PCA:
  - Assumes species are linearly related with each other and/or gradients
  - **Samples are ordinated in species space**
  - Results in “horseshoe effect” where ends of ordination axes are distorted
- Correspondence analysis allows for non-linear unimodal relationships
  - Both samples and species handled similarly, axes do not explicitly represent species-space

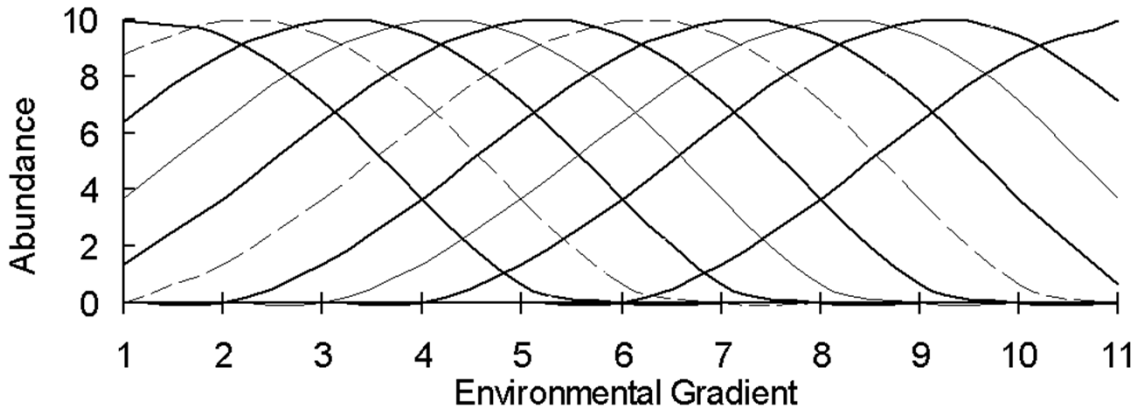


Figure 19.1. A synthetic data set of eleven species with noiseless hump-shaped responses to an environmental gradient. The gradient was sampled at eleven points (sample units), numbered 1-11.

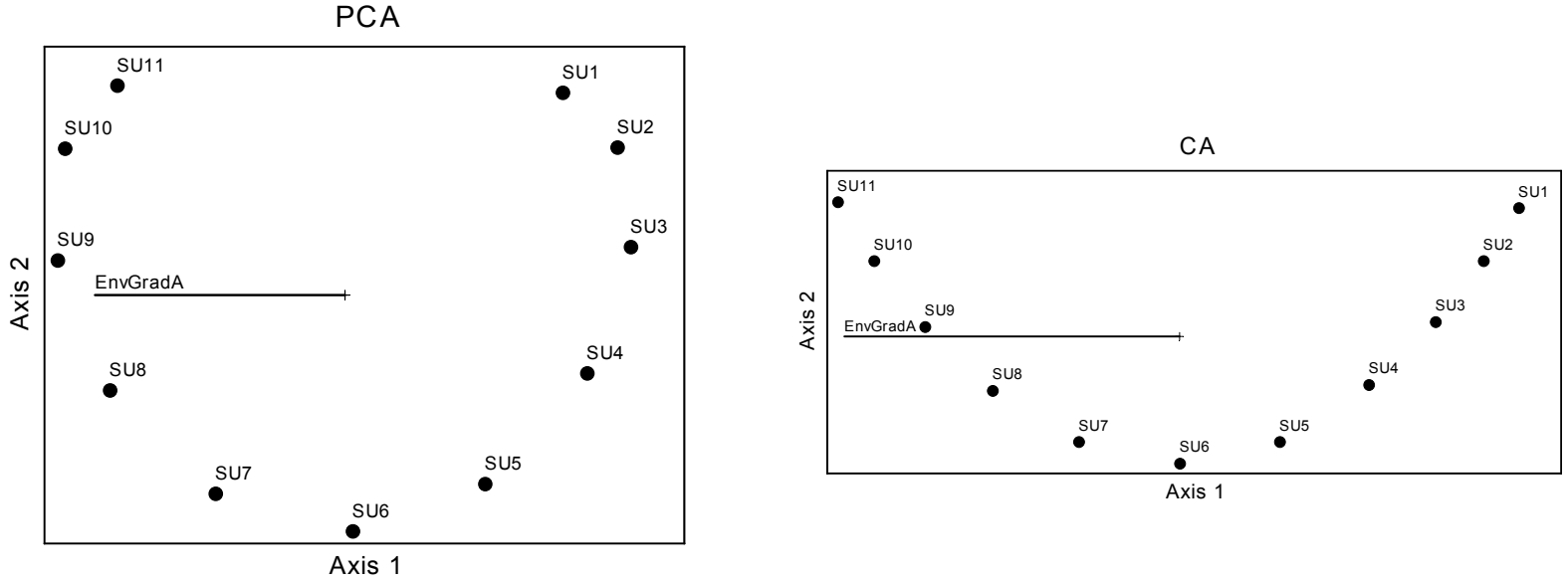
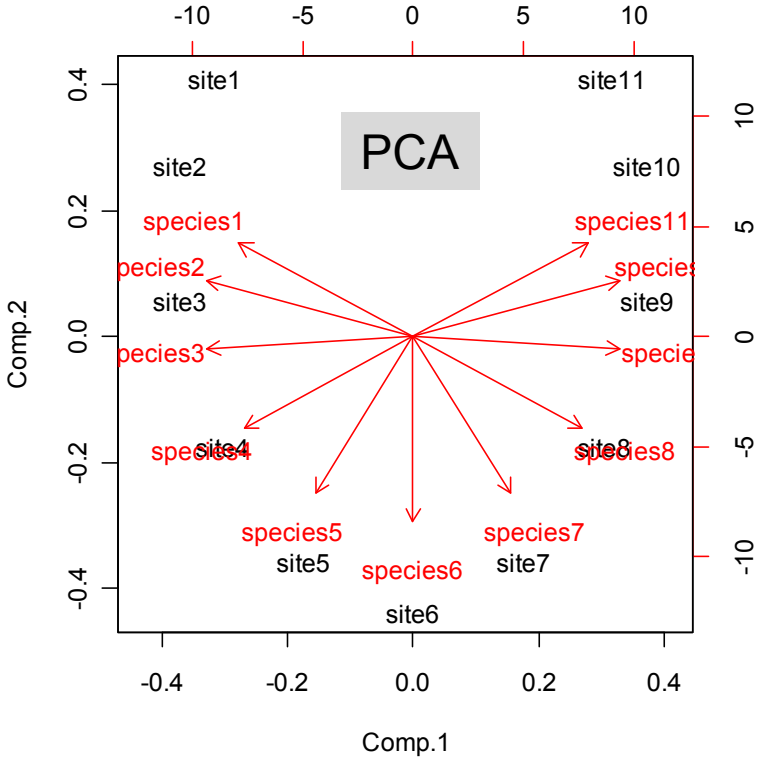
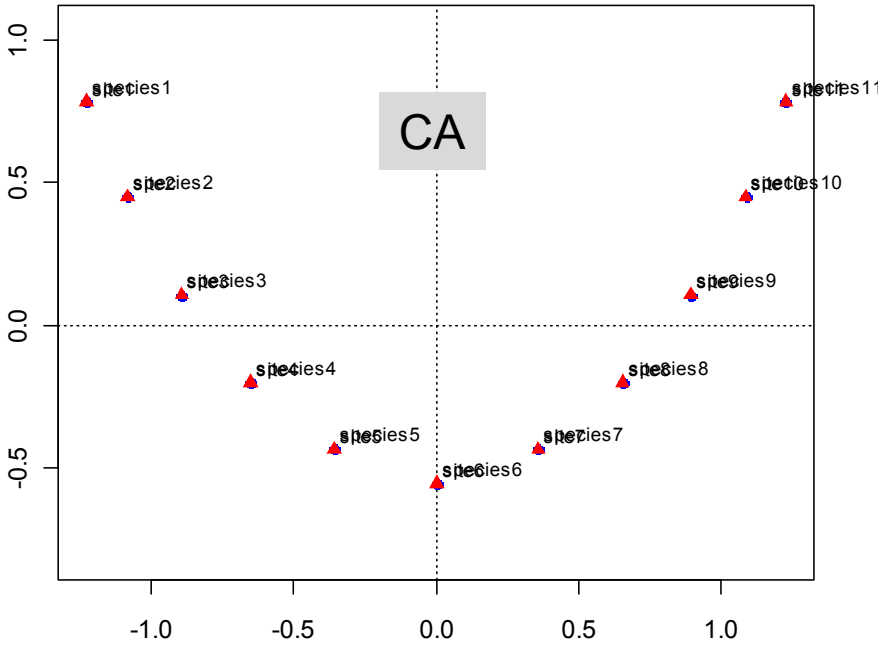


Figure 19.2. Comparison of PCA and CA of the data set shown in Figure 19.1. PCA curves the ends of the gradient in, while CA does not. The vectors indicate the correlations of the environmental gradient with the axis scores.

From: McCune, B. & J. B. Grace. 2002. *Analysis of Ecological Communities*. MjM Software Design, Gleneden Beach, Oregon http://www.pcord.com

Try by yourself






```
# horseshoe effect
species <- read.csv('D:/PCA species environment gradient.csv', header = T) # see following table for data
rownames(species) <- species$site
species <- species[,-1] #remove the first column (row names)
pca <- princomp(species)
biplot(pca)

library(ca) # Package for CA
ca(species) #Correspondence analysis
plot(ca(species))
```

site	species1	species2	species3	species4	species5	species6	species7	species8	species9	species10	species11
site1	10	8	6	4	2	0	0	0	0	0	0
site2	8	10	8	6	4	2	0	0	0	0	0
site3	6	8	10	8	6	4	2	0	0	0	0
site4	4	6	8	10	8	6	4	2	0	0	0
site5	2	4	6	8	10	8	6	4	2	0	0
site6	0	2	4	6	8	10	8	6	4	2	0
site7	0	0	2	4	6	8	10	8	6	4	2
site8	0	0	0	2	4	6	8	10	8	6	4
site9	0	0	0	0	2	4	6	8	10	8	6
site10	0	0	0	0	0	2	4	6	8	10	8
site11	0	0	0	0	0	0	2	4	6	8	10

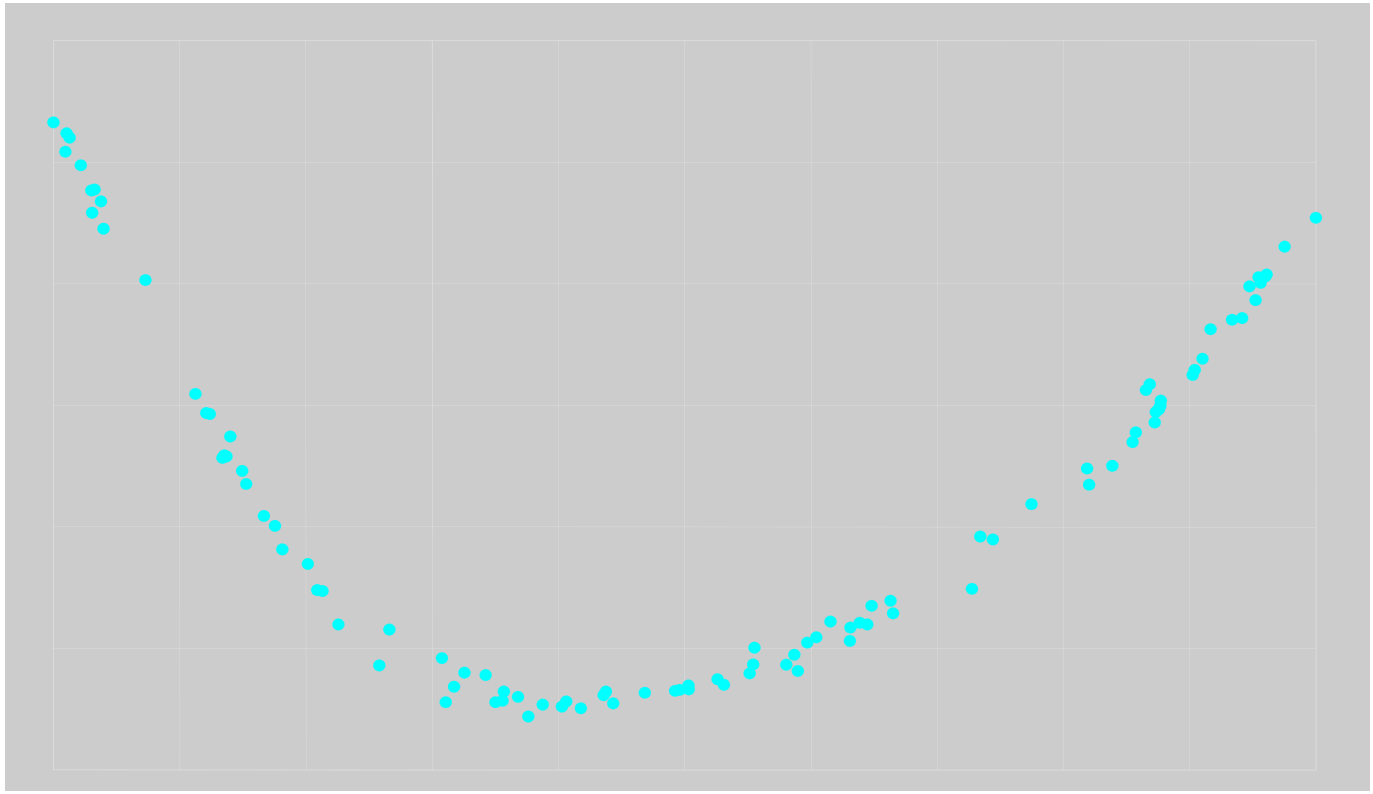
Problems with CA

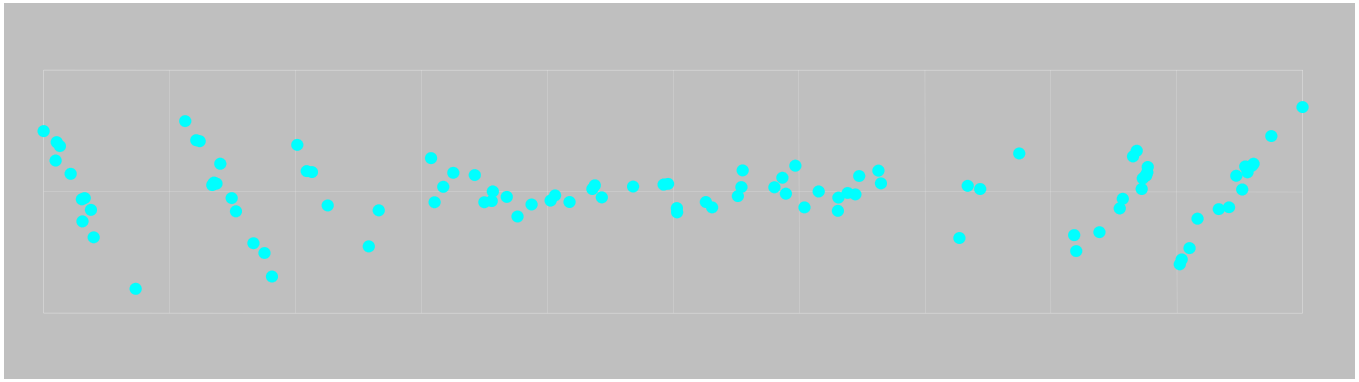
- The 1st CA axis is reliable, but 2nd and later axes are quadratic distortions of the first – produces the “arch effect”
- Distances compressed toward the ends of the axes and stretched in the middle
- Chi-square distance gives high weight to species with low abundance, which exaggerates distinctiveness of samples containing several rare species (Faith et al. 1987, Minchin 1987)

Detrended correspondence analysis (DCA)

- The “arch effect” of CA is unwanted; the ends of the axes in CA are also compressed
- Detrending (detrended correspondence analysis, DCA) deals with the arch by:
  - 5 segment smoothing of 1st axis. Divide into segments (weights of 1,2,3,2,1), center each at 0.
  - Rescaling of axis into “standard deviation” units of species turnover.
- Only first 4 axes are adjusted, the rest are discarded
- Assumptions
  - Same as for CA
  - DCA is not really an analysis. It is a post hoc modification of a CA

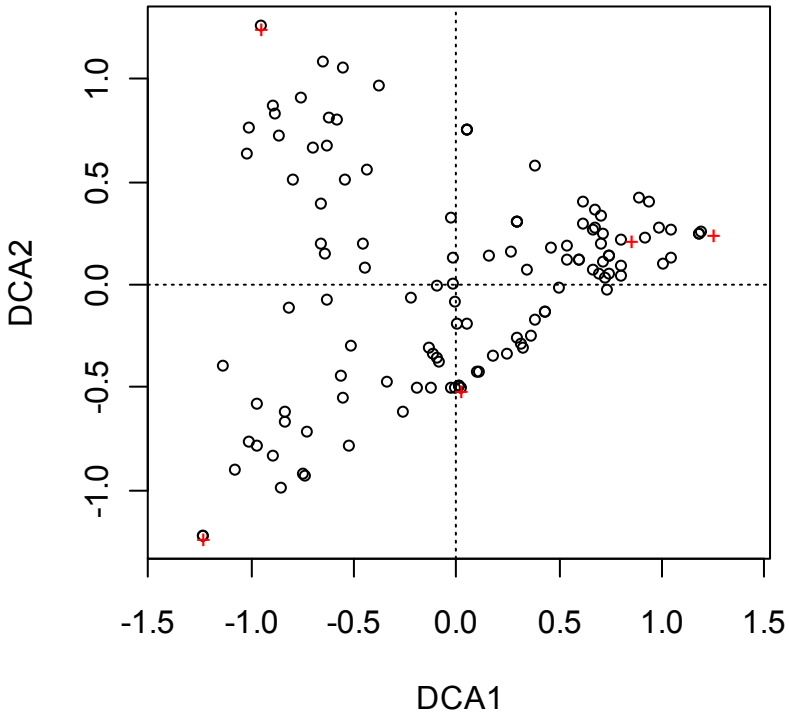
Detrended correspondence analysis (DCA)





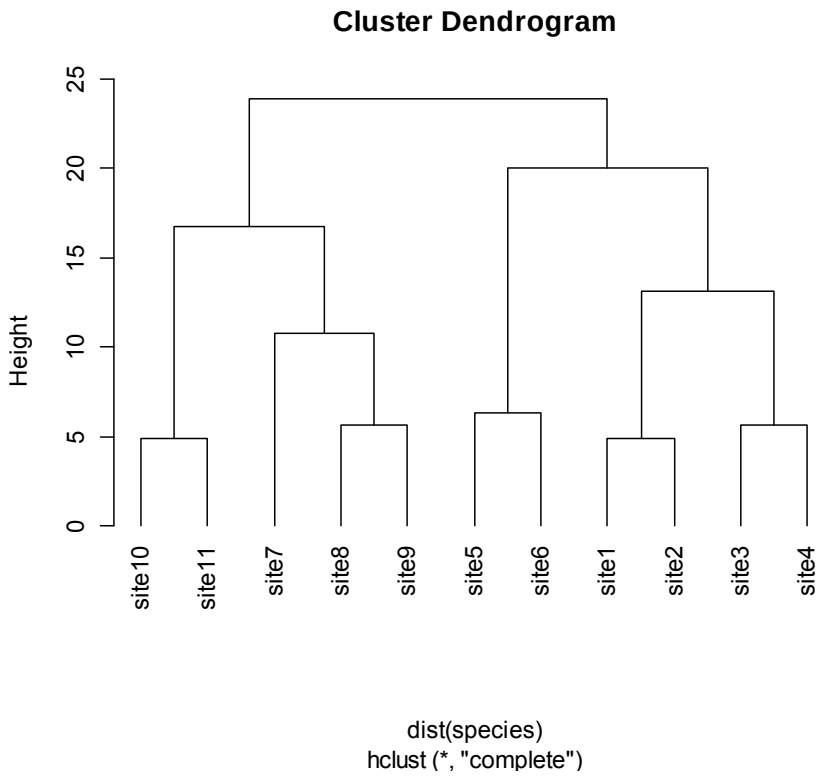
R code: Detrended correspondence analysis (DCA)

```
library(vegan)
decorana(species)
plot(decorana(species))
```

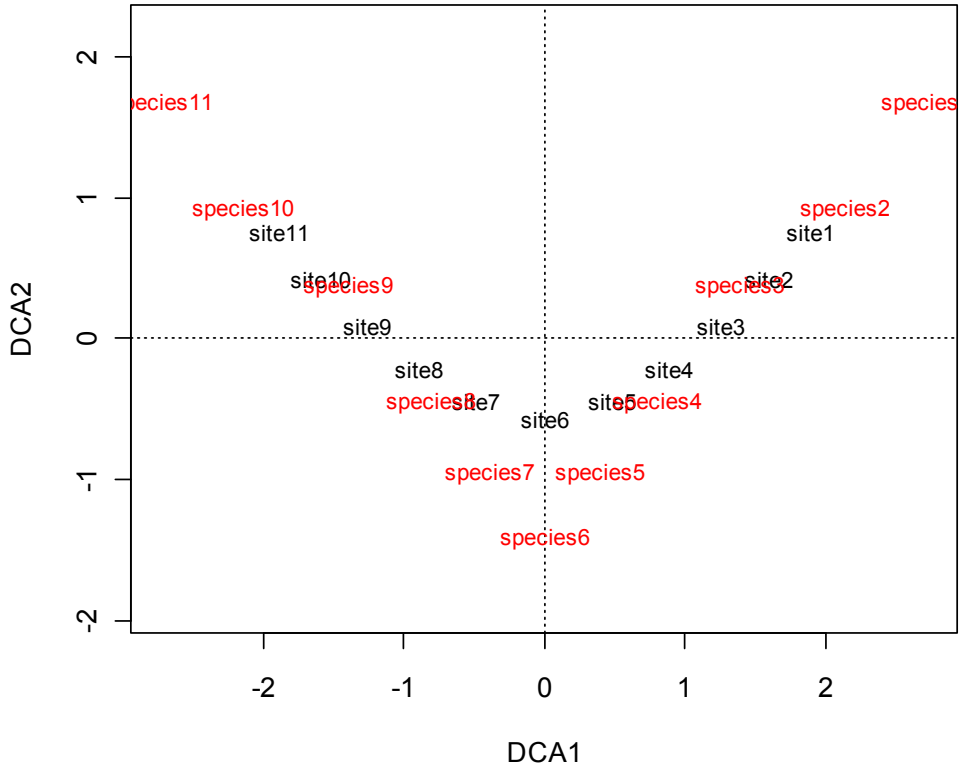


Try by yourself

Cluster analysis



DCA



Redundancy analysis (RA)

Redundancy analysis (RDA)

Examines how much of the variation in one set of variables explains the variation in another set of variables

Based on similar principles as principal components analysis and thus makes similar assumptions about the data.

Be appropriate when the expected relationship between dependent and independent variables is linear.

If the expected relationship between dependent and independent variables is Gaussian (e.g. climate and species abundance), then canonical correspondence analysis is more appropriate.

Redundancy analysis (RDA)

We have explanatory variables in X, and response variables in Y

Find those components of Y which are linear combinations of X and (among those) represent as much variance of Y as possible.

Assumption: There is a linear dependence of the response variables in Y on the explanatory variables in X.

The idea behind redundancy analysis is to apply linear regression in order to represent Y as linear function of X and then to use PCA in order to visualize the result.

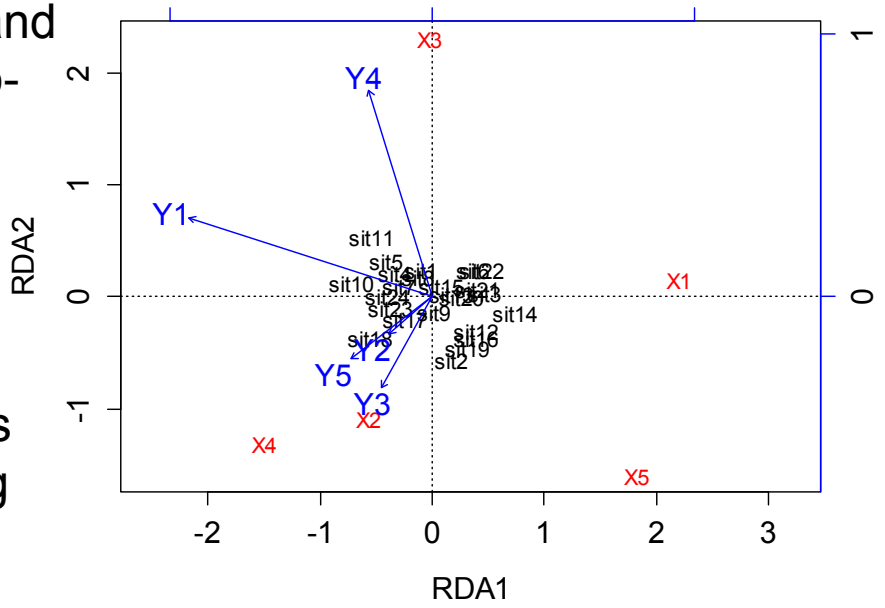
Among those components of Y which can be linearly explained with X (multivariate linear regression) take those components which represent most of the variance.

R code

```
library(vegan) # rda() is in this library
X <- matrix(rnorm(120),ncol=5)
Y <- matrix(rnorm(120),ncol=5)
colnames(X) = c('X1','X2','X3','X4','X5')
X=data.frame(X, X6 = rep(1:3, each = 8))
colnames(Y) = c('Y1','Y2','Y3','Y4','Y5')
rda.results <- rda(X,Y)
plot(rda(X,Y), scaling =1) # Distance triplot
plot(rda(X,Y), scaling =2) # Correlation triplot

# ibis data
ibis = read.csv('d:/ibis watersheds.csv',header=T)
species = ibis[1:24,2:11]
env.var = ibis[1:24,12:24]
rda.results <- rda(env.var,species)
plot(rda(env.var, species), scaling =1, main="Scaling1") # Distance triplot
plot(rda(env.var, species), scaling =2, main="Scaling2") # Correlation triplot
plot(rda(env.var, species))
```

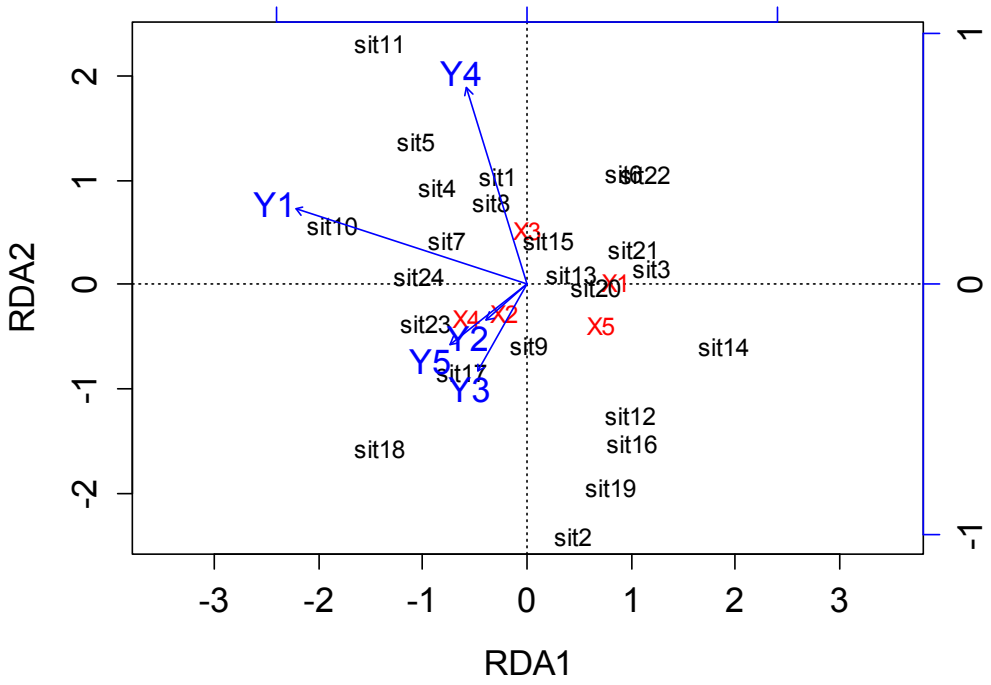
Distance triplot (scaling=1)

- Distances between points (observations) approximate distances of the observations (or the centroid of the nominal explanatory variable).
- Angles between lines of response variables and lines of explanatory variables represent a two-dimensional approximation of correlations.
- Other angles between lines are meaningless.
- The projection of a point onto the line of a response variable at right angle approximates the position of the corresponding object along the corresponding variable.
- Squares/triangles cannot be compared with lines of qualitatively explanatory variables.

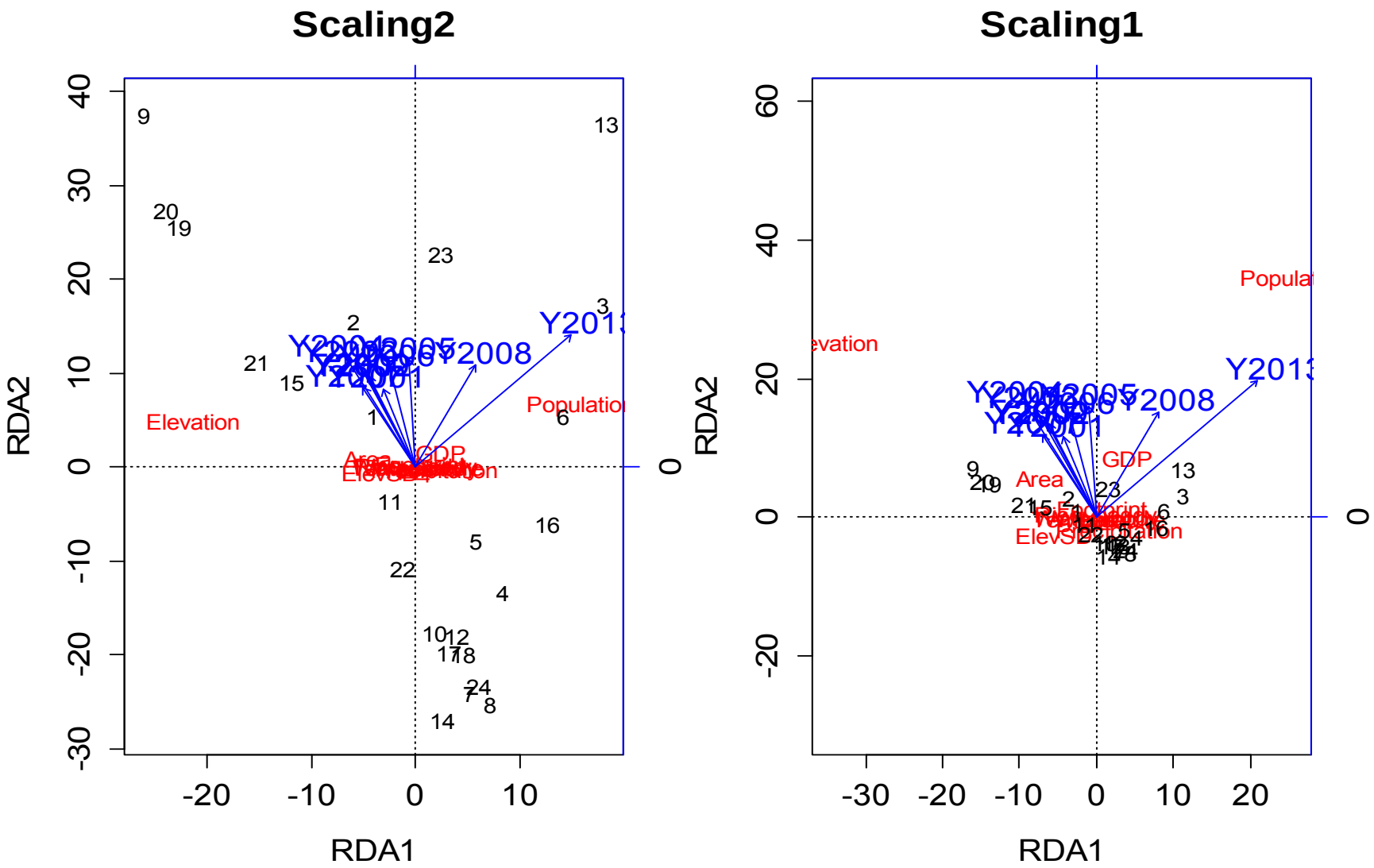


Correlation triplot (scaling=2)

- The cosine of the angle between lines (of response variable or of explanatory variable) is approximately equal to the correlation between the corresponding variables.
- Distances are meaningless.
- The projection of a point onto a line (response variable or explanatory variable) at right angle approximates the value of the corresponding variable of this observation.
- The length of lines are not important.



Triplot for the ibis data



Canonical correspondence analysis (CCA)

Canonical correspondence analysis (CCA)

CCA is a multivariate constrained ordination technique that extracts major gradients among combinations of explanatory variables in a dataset.

CCA is realized by a correspondence analysis in which weighted multiple regression is used to represent the axes as linear combination of the explanatory variables.

So **CCA** is a **CA** with the axes being linear combinations of the explanatory variables.

The requirements of a **CCA** are that the samples are random and independent and that the independent variables are consistent within the sample site and error-free.

Data of CCA

Given: Data frames/matrices Y and X
Y[j, i] are the count of species i at site j.
X[j, k] are the explanatory variable k at site j.

Goal: Find associations of species abundance and sites with each environmental condition on a site being a linear combination of the environmental variables of X.

Assumption: There is a niche dependence of the species on environmental factors

Calculation steps

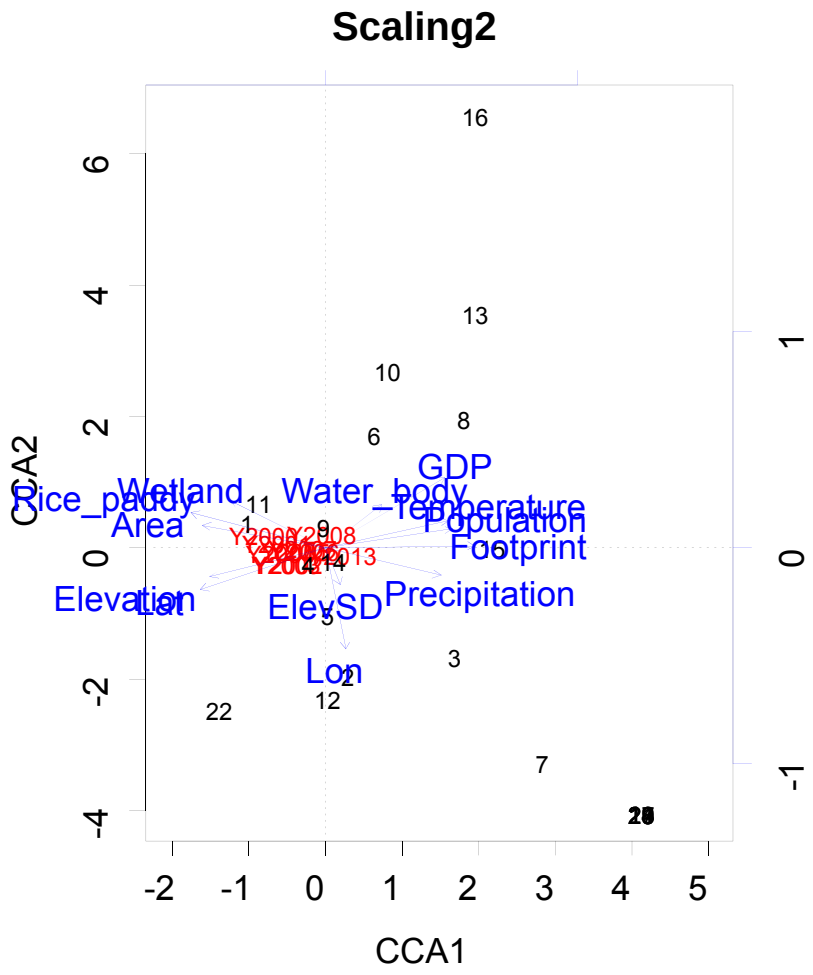
1. Start with a Chi-square species matrix [(actual - predicted)/sqrt(predicted)],
2. Regress the differences from expectation on environmental variables to get fitted values, using a weighted regression where total abundance by plots is used as the weights, and
3. Calculate the Euclidean distance of the fitted species matrix and project by eigen-analysis. The importance of specific environmental variables is then assessed by their correlation to the projected scatter diagram.

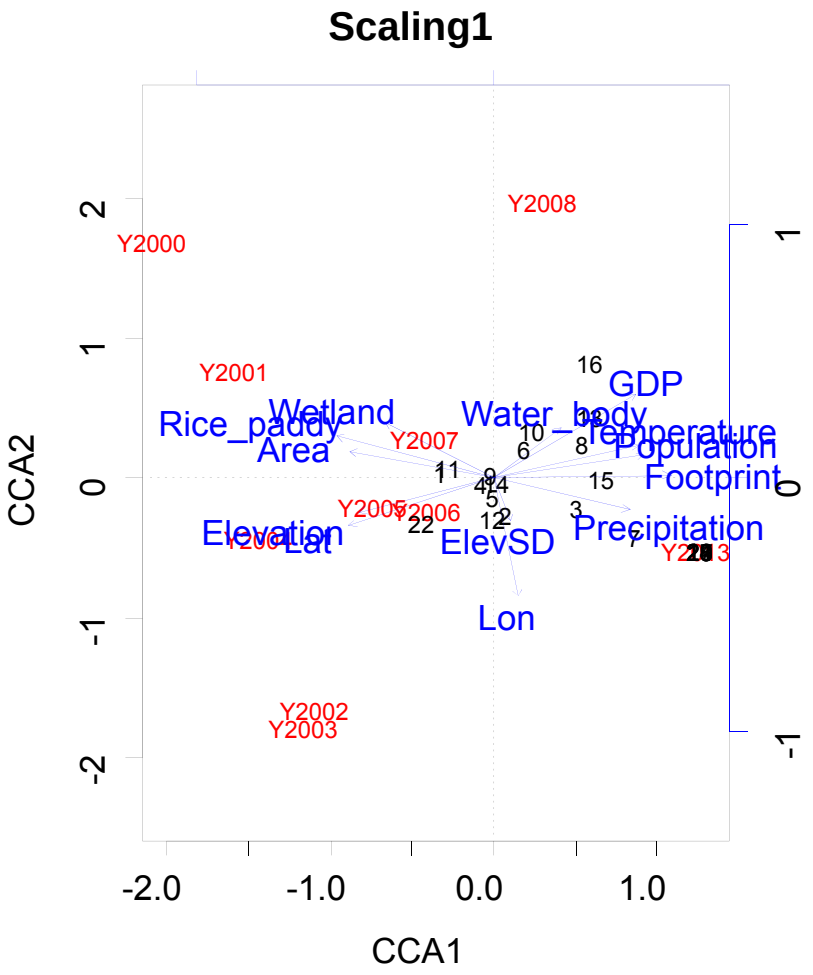
R code

```
library(vegan)
ibis = read.csv('d:/ibis watersheds.csv', header=T)
species = ibis[1:24,2:11]
env.var = ibis[1:24,12:24]
cca.ibis <- cca(species, env.var)

# the total variation (inertia) in the data is: 0.2
round(cca.ibis$tot.chi, 2)

# the sum of all canonical eigenvalues (Constrained inertia): 0.14
round(cca.ibis$CCA$tot.chi, 2)

# all explanatory variables explain 68% of the total variation in the data
cat(round(cca.ibis$CCA$tot.chi
        /cca.ibis$tot.chi*100), "% of data", "\n")

# the first two (canonical) eigenvalues are: 0.09, 0.02
round(cca.ibis$CCA$eig[1:2], 2)

# so the first two canonical axes explain 79% of the variation with the used environmental variables
cat(round(sum(cca.ibis$CCA$eig[1:2])
        /cca.ibis$CCA$tot.chi * 100), "%", "\n")

# but this is (the first two canonical axes explain) 54% of the total variation in the data
cat(round(sum(cca.ibis$CCA$eig[1:2])
        /cca.ibis$tot.chi*100), "%", "\n")
```

Triplot

```
plot(cca.ibis, scaling = 2, main="Scaling2") # species scores are scaled by eigenvalues
plot(cca.ibis, scaling = 1, main="Scaling1") # site scores are scaled by eigenvalues
```





Triplot

The species scores, the site scores and the environmental scores are plotted in a figure called a triplot (confer with triplots in RDA). These triplots are the biplots from CA with additionally the explanatory variables plotted as lines.

Again the position of a species represents the optimum value in terms of the Gausian response model (niche) along the first and second axes. For this reason, species scores are represented as labels or points.

In addition: Species can be projected perpendicular (=orthogonally) on the lines showing the species optima of the respective explanatory variables (in the respective scaling). Projecting sites perpendicular on the lines results in the values of the respective environmental variable at those sites.

The angle between lines does NOT represent correlation between the variables. Instead if the tip of a line is projected on another line or an axis then the resulting value represents a weighted correlation.

When to use PCA, CA, RDA or CCA

1. PCA should be used to analyse species data if the relations along the gradients are linear.
2. RDA should be used to analyse linear relationships between species and environmental variables.
3. CA analyses species data and unimodal relations along the gradients.
4. CCA can be used to analyse unimodal relationships between species and environmental variables.
5. PCA or RDA should be used if the beta diversity is small, or if the range of the samples covers only a small part of the gradient.
6. A long gradient has high beta diversity, and this indicates that CA or CCA should be used.

	Pure ordination	Cause-effect relation
Linear model	PCA	RDA
Unimodal model	CA	CCA

Zuur AF, N. Ieno E, and Smith GM (2007). *Analysing ecological data*. Springer. (page 242)

Principal coordinate analysis (PCoA)

Principal coordinate analysis (PCoA)

Like PCA, PCoA is an eigen-analysis of a distance or dissimilarity matrix.

In contrast to PCA, PCoA can employ a broader range of distances or dissimilarity coefficients, including ones which ignore joint absences.

E.g. various species have a patchy distribution, which makes the correlation, covariance and Chi-square functions less appropriate tools to define association.

R provides a function for calculating distances, the dist() function, which provides a fairly narrow range of distances (euclidean, manhattan, binary, canberra, and maximum).

However, the **vegan** library provides the vegdist() function, and the **LabDSV** library provides the dsvdis() function as alternatives that provide many more indices, including those commonly used in vegetation ecology.

Distance, dissimilarity, or index functions used in various programs and libraries

http://ecology.msu.montana.edu/labdsv/R/labs/lab8/lab8.html

Distance, Dissimilarity, or Index	**dist**	**vegan**	**LabDSV**
method	**method**	**index**	
euclidean	X	X	
manhattan	X	X	
binary	X		steinhaus[1]
sorensen			X[1]
canberra		X	X
bray-curtis		bray	bray/curtis
gower		X	
kulczynski		X	
ochiai			X
ruzicka			X
roberts			X
Chi-Square			chisq
morisita		X	
mountford		X	
horn		X	
minkowski	X		
footnote			
[1] = converts to presence/absence			

vegdist {vegan}

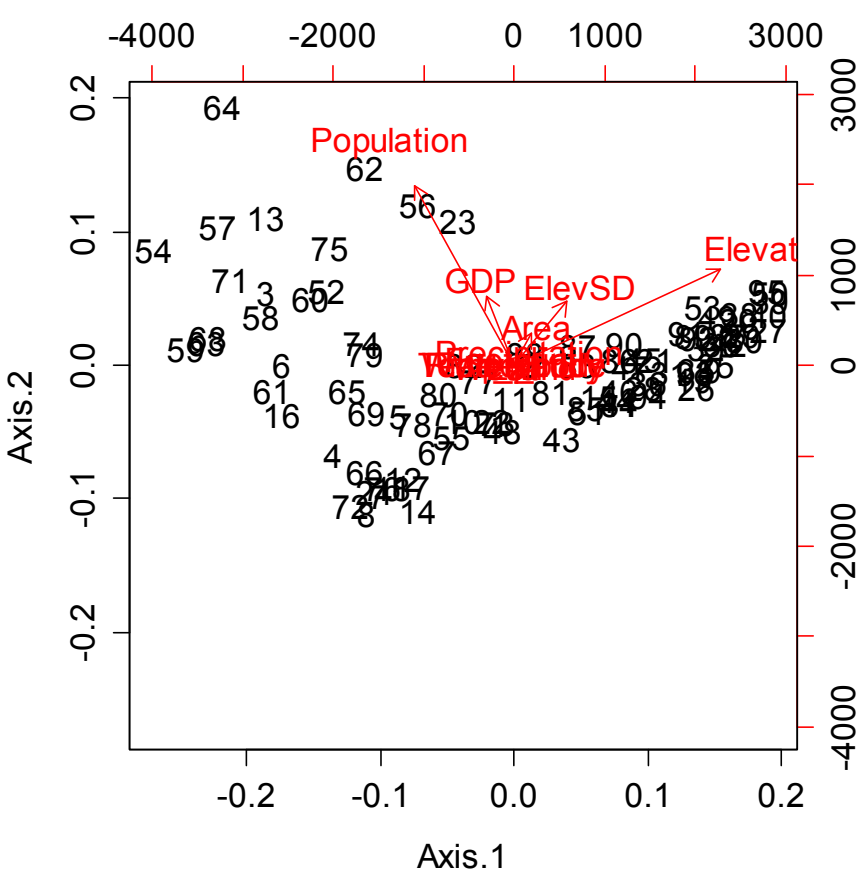
euclidean	*d[jk] = sqrt(sum(x[ij]-x[ik])^2)* binary: *sqrt(A+B-2*J)*
manhattan	*d[jk] = sum(abs(x[ij] - x[ik]))* binary: *A+B-2*J*
gower	*d[jk] = (1/M) sum(abs(x[ij]-x[ik])/(max(x[i])-min(x[i])))* binary: *(A+B-2*J)/M*, where *M* is the number of columns (excluding missing values)
altGower	*d[jk] = (1/NZ) sum(abs(x[ij] - x[ik]))* where *NZ* is the number of non-zero columns excluding double-zeros (Anderson et al. 2006). binary: *(A+B-2*J)/(A+B-J)*
canberra	*d[jk] = (1/NZ) sum ((x[ij]-x[ik])/(x[ij]+x[ik]))* where *NZ* is the number of non-zero entries. binary: *(A+B-2*J)/(A+B-J)*
bray	*d[jk] = (sum abs(x[ij]-x[ik]))/(sum (x[ij]+x[ik]))* binary: *(A+B-2*J)/(A+B)*
kulczynski	*d[jk] 1 - 0.5*((sum min(x[ij],x[ik])/(sum x[ij]) + (sum min(x[ij],x[ik])/(sum x[ik]))* binary: *1-(J/A + J/B)/2*
morisita	*d[jk] = 1 - 2*sum(x[ij]*x[ik])/((lambda[j]+lambda[k]) * sum(x[ij])*sum(x[ik]))*, where *lambda[j] = sum(x[ij]*(x[ij]-1))/sum(x[ij])*sum(x[ij]-1)* binary: cannot be calculated
horn	Like morisita, but *lambda[j] = sum(x[ij]^2)/(sum(x[ij])^2)* binary: *(A+B-2*J)/(A+B)*
binomial	*d[jk] = sum(x[ij]*log(x[ij]/n[i]) + x[ik]*log(x[ik]/n[i]) - n[i]*log(1/2))/n[i]*, where *n[i] = x[ij] + x[ik]* binary: *log(2)*(A+B-2*J)*
cao	*d[jk] = (1/S) * sum(log(n[i]/2) - (x[ij]*log(x[ik]) + x[ik]*log(x[ij]))/n[i])*, where *S* is the number of species in compared sites and *n[i] = x[ij] + x[ik]*

R code

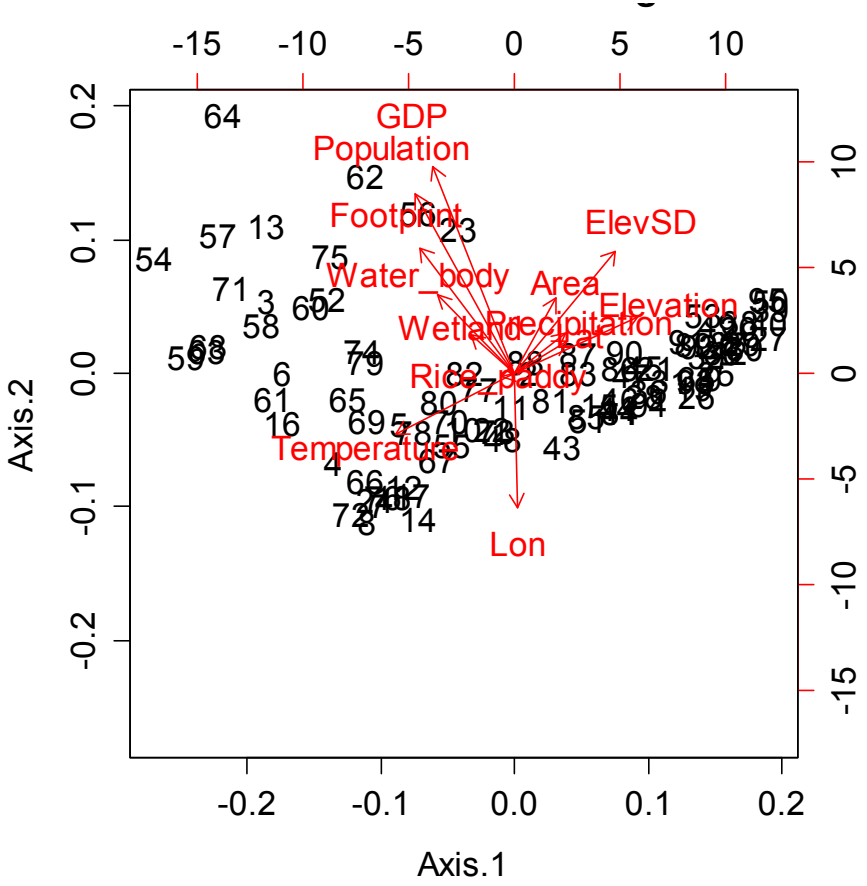
```
library(vegan)
library(ape)


# data standarization
ibis.stand = apply(ibis, 2, scale, center=TRUE, scale=TRUE)

dis <- vegdist(ibis, "bray")
dis <- vegdist(ibis, "euclidean")
dis <- vegdist(ibis, "manhattan")
dis <- vegdist(ibis, "jaccard")

res <- pcoa(dis) # library(ape)
biplot(res, ibis.stand)
```

PCoA plot

Not standardized

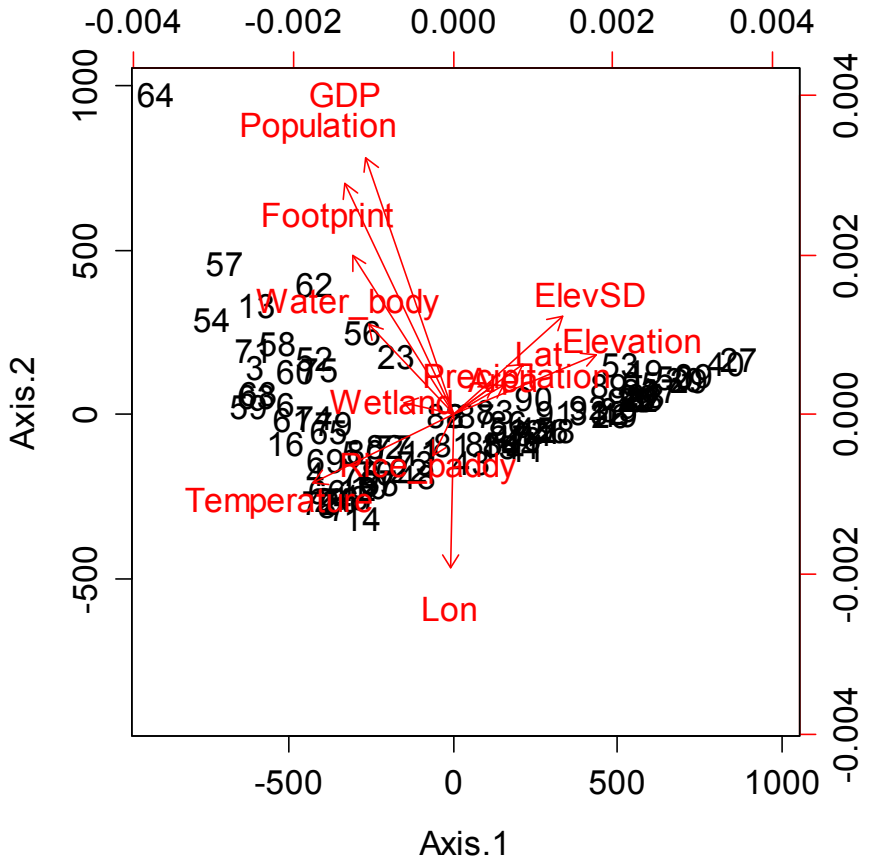


Standardized

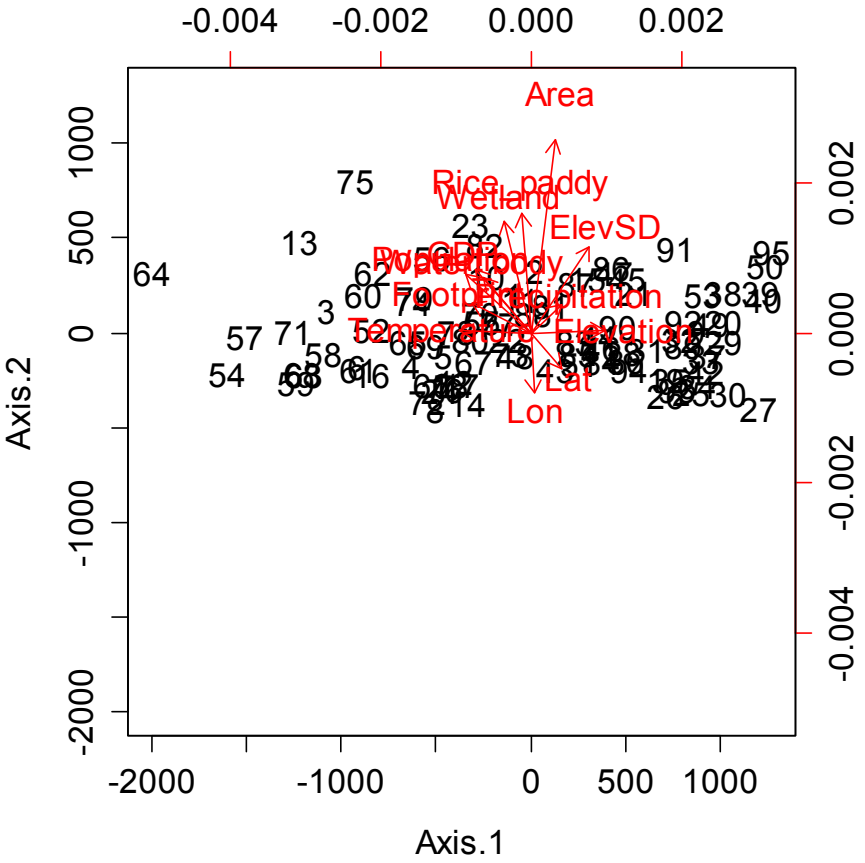


PCoA plot

Using Euclidean distance



Using Manhattan distance



Non-metric multidimensional scaling (NMDS)

Non-metric multidimensional scaling (NMDS)

In contrast to metric MDS, non-metric MDS is based on the ranked similarities/dissimilarities between pairs of samples.

NMDS can also use any measure of association, like PCoA.

It is better in preserving the high-dimensional structure with a few axes.

Its disadvantage is that it is not based on an eigenvalue solution but on numerical optimization methods and for larger datasets the calculations tend to become time consuming.

Bray-Curtis similarity

Sample	1	2	3	4
Species				
A	0	0	0	1.7
B	1.3	0	0	2.1
C	1.8	0	2.5	1.7
D	1.7	3.5	1.9	0
E	1.2	4.3	3.4	0
F	0	0	0	0

$$S'_{il} = 100 \left\{ 1 - \frac{\sum_{j=1}^{n} | y_{ij} - y_{lj} |}{\sum_{j=1}^{n} | y_{ij} + y_{lj} |} \right\}$$

Sample	1	2	3	4
1	—			
2	42.0	—		
3	68.1	67.9	—	
4	52.2	0.0	25.6	—

Sample	1	2	3	4
1	—			
2	4	—		
3	1	2	—	
4	3	6	5	—

Modified from Clarke & Warwick, 1994

Redrawn from Clarke & Warwick, 1994.
Change in marine communities: an approach to statistical analysis and interpretation.
Plymouth: Plymouth Marine Laboratory.

Non-metric multidimensional scaling

1. Choose a measure of association and calculate the distance matrix D.
2. Specify *m,* the number of axes.
3. Construct a starting configuration E. This can be done with PCoA.
4. Regress the configuration on D: $D_{ij} = a + \beta E_{ij} + \varepsilon_{ij}$.
5. Measure the relationship between the m dimensional configuration and the real distances by fitting a non-parametric (monotonic) regression curve in the Shepard diagram. A monotonic regression is constrained to increase. If a parametric regression line is used, we obtain PCoA.
6. The discrepancy from the fitted curve is called STRESS.
7. Using non-linear optimisation routines, obtain a new estimation of E and go to step 4 until convergence.

Goodness-of-fit and STRESS

- The STRESS measure (STandardized REsiduals Sum of Squares) is a function of the original and derived distances to evaluate the goodness-of-fit of a MDS solution:

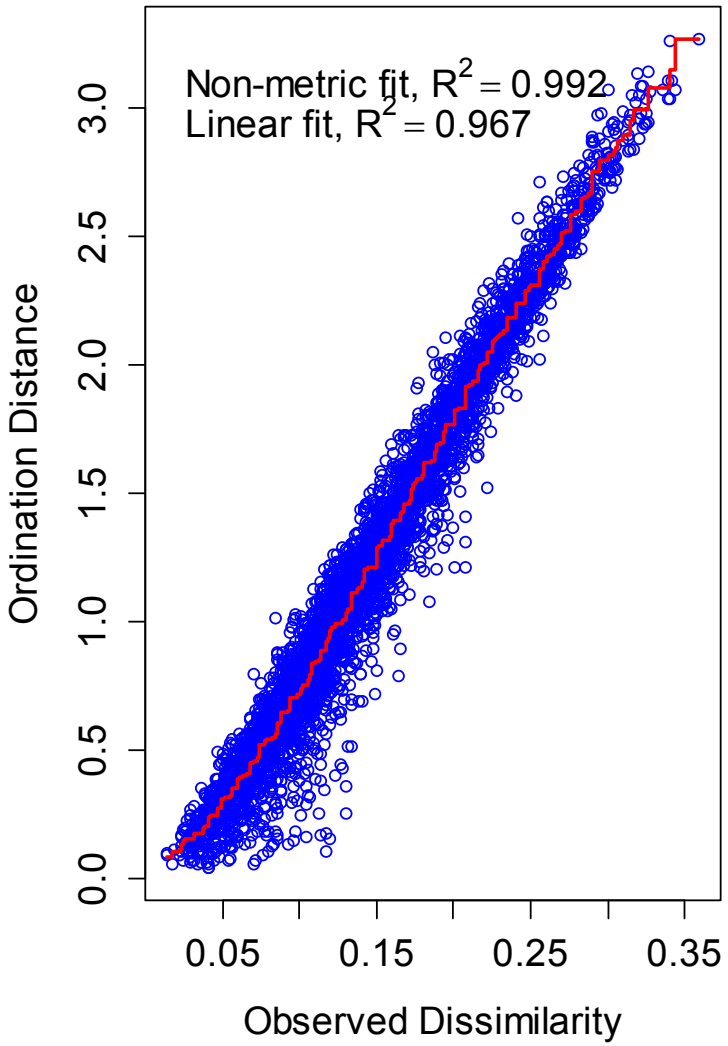
$$STRESS = \sqrt{\frac{\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}(d_{ij}-\hat{d}_{ij})^2}{\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}d_{ij}^2}}$$

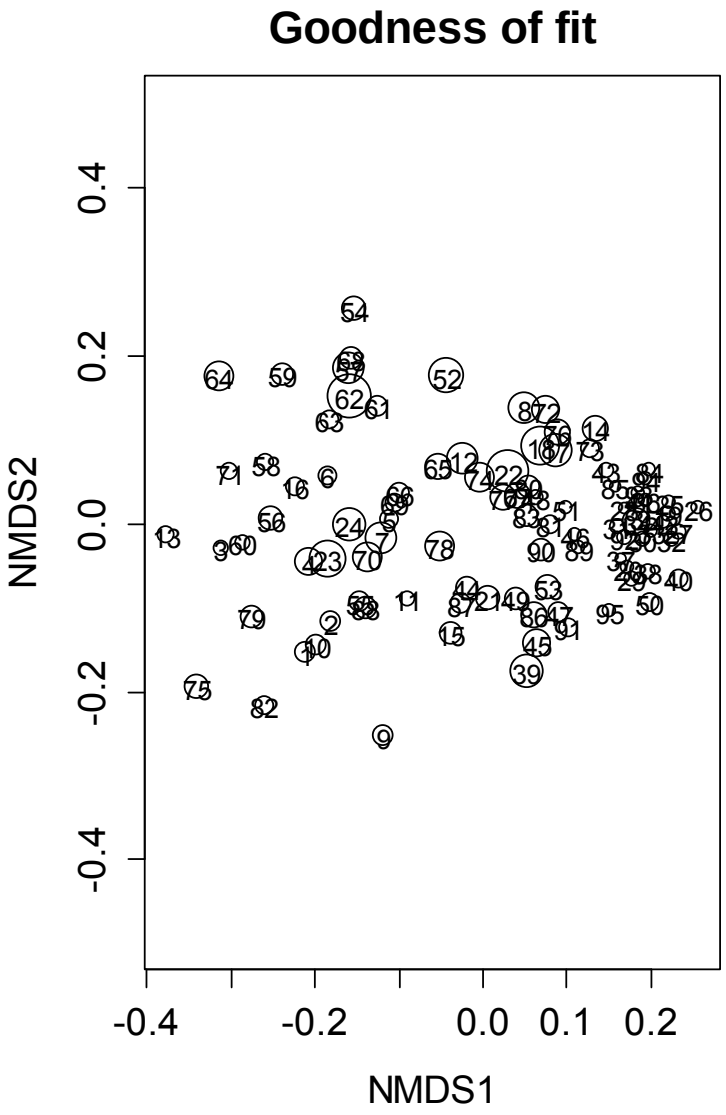
- The smaller the stress function, the closer are the derived distances to the original ones.

R code

```
NMDS <- metaMDS (ibis) #vegan

par (mfrow = c(1,2),mar=c(5,4,3,2))
stressplot (NMDS)
plot (NMDS, display = 'sites', type = 't', main = 'Goodness of fit')
points (NMDS, display = 'sites', cex = goodness (NMDS)*200)
```

Plots

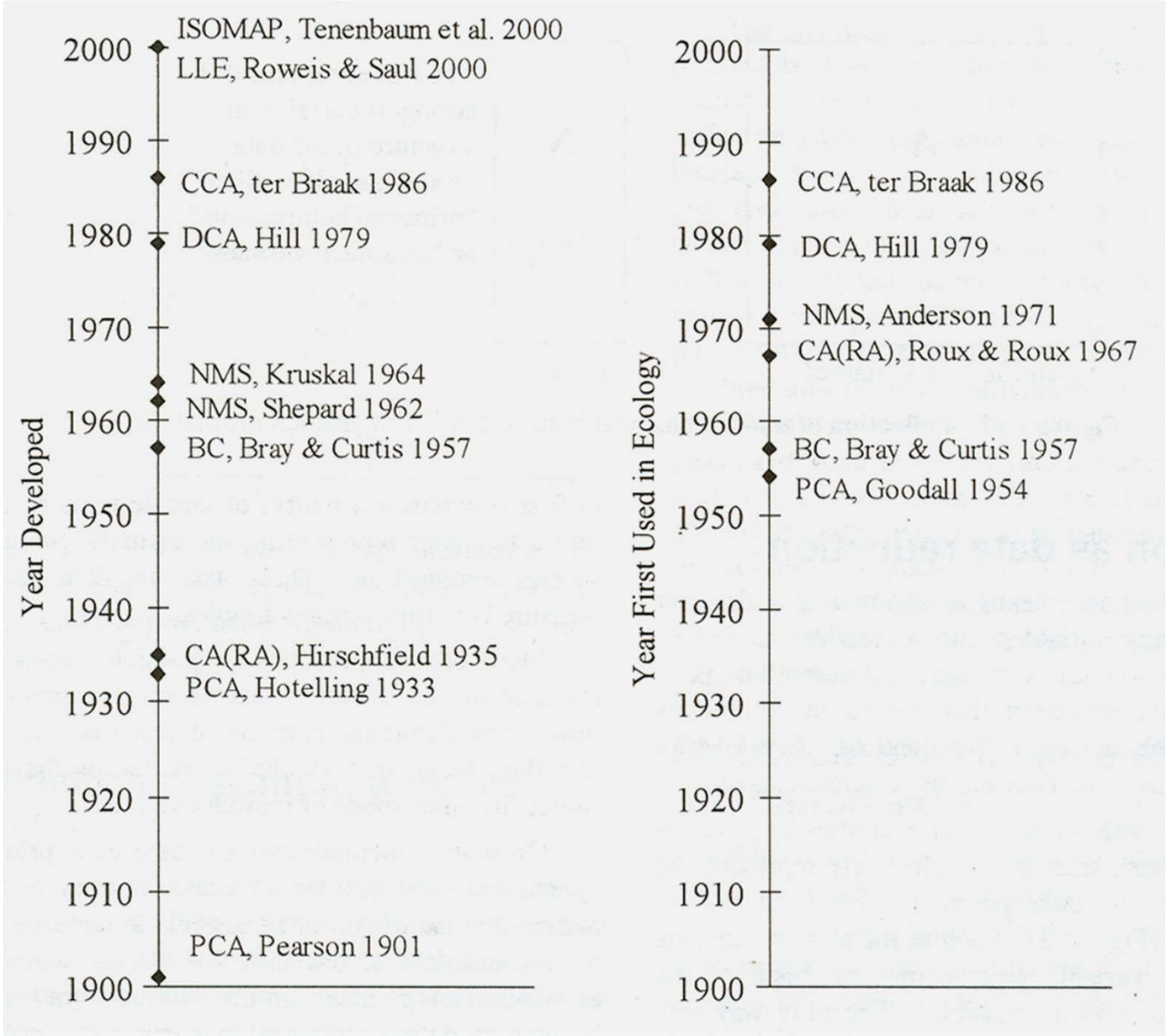

Non-metric fit, $R^2 = 0.992$
Linear fit, $R^2 = 0.967$
Ordination Distance
Observed Dissimilarity
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.05
0.15
0.25
0.35


Goodness of fit
NMDS2
NMDS1
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.1
0.2

STRESS and number of dimensions

- The STRESS value decreases as the number of dimensions increases
- The number of dimensions can be evaluated through a *scree diagram* of STRESS against the number of dimensions (as for FA, PCA or cluster analysis) where the optimal number corresponds to an elbow
- The preferred number of dimensions is usually two or three which allows for graphical examination
- The search usually goes from one to five dimensions
- Identification of the optimal number within the metric and non-metric iterative algorithm
  - An additional step evaluates the STRESS function
  - The algorithm stops when the addition of a further dimension does not reduce the STRESS value to a perceptible extent
- With two dimensions a STRESS value below 0.05 is generally considered to be satisfactory.

History of Ordinations in Ecology



From: McCune, B. and J. B. Grace. 2002. Analysis of Ecological Communities. MJM Software Design.

Assignment 9

Due June 5, 2014. Deliver to Jiajie.

General objectives: learn about Principal Components Analysis.

- Develop a dataset to perform:
  - Principal Components Analysis Y-X1, X2, X3, etc.
- Describe the dataset, list the Eigenvalues of the Covariance Matrix, describe the Cumulative Proportion of the variance explained), describe the loadings of original variables on components