

Multivariate Ordination

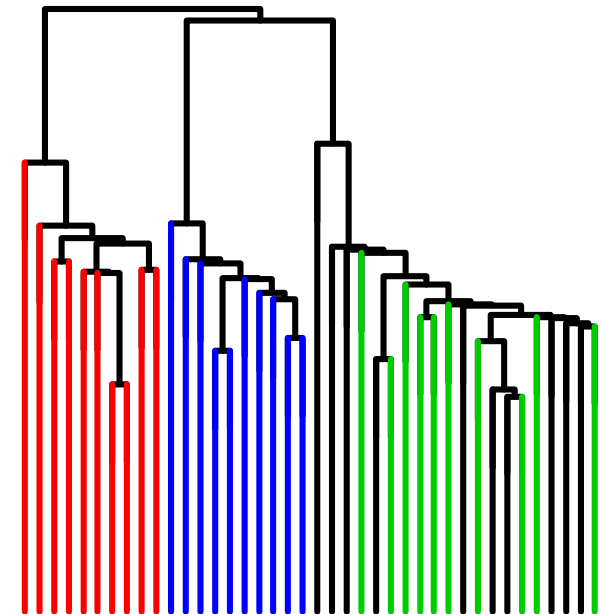
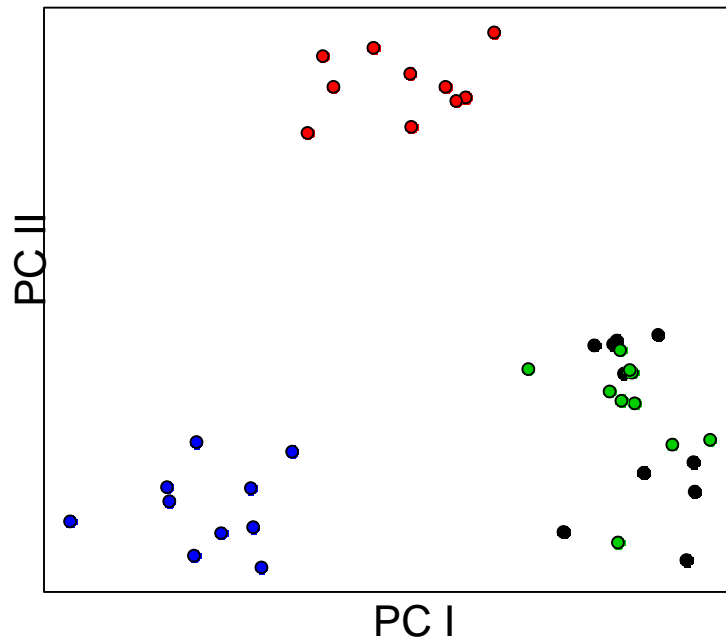
Advanced Biostatistics

Dean Adams

Lecture 8

EEOB 590C

- Data are points in multivariate space
- Human perception does not work very well in >3 dimensions
- Look for patterns in high-dimensional spaces
- Generate summary plots of dataspace (ordination)
- Look for relationships of points (clustering)



Obtain summarized visualization of points in high-dimensional data space

Describe variation with a new set of variables (typically orthogonal)

These are produced under the criterion of finding what varies the most

They are ordered to represent more -> less variation

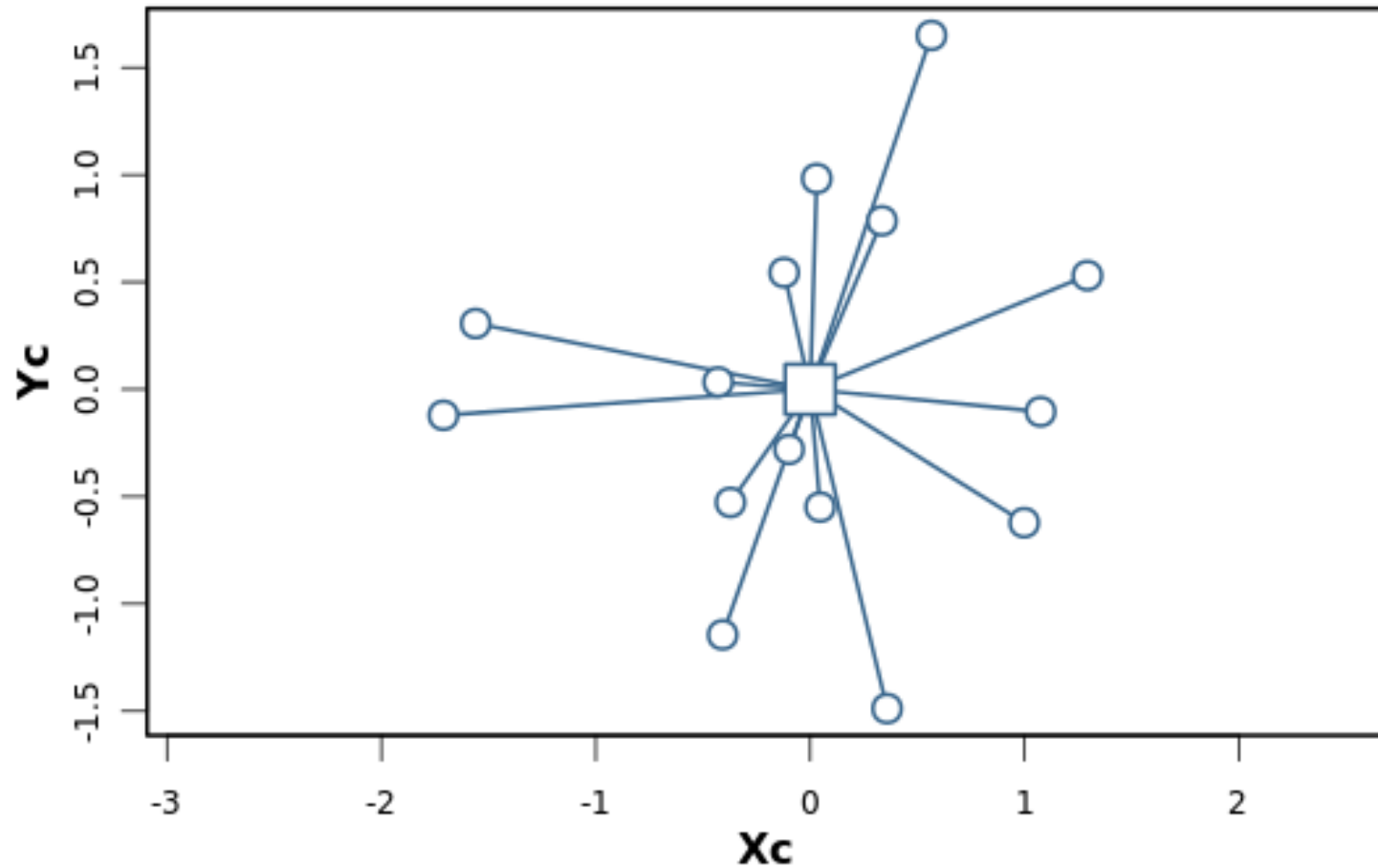
Plot first dimensions to summarize data

Look at association with original variables to know what varies

- Underlying matrices: covariance vs. correlation (vs. distances)
- Rigid Rotation: Principal Components Analysis (PCA)
- Decomposing a distance matrix: Principal Coordinates Analysis (PCoA)
- Iterative optimization: Non-Metric MultiDimensional Scaling (NMDS)
- Non-continuous data: Correspondence Analysis (CA)

Ordination	Main Matrices	Slide 5
Create an $n \times 1$ vector of 1s	$\mathbf{1}^t = [1 \quad 1 \quad \dots \quad 1]$	
Find the variable means	$\bar{\mathbf{y}}^t = (\mathbf{1}^t \mathbf{1})^{-1} \mathbf{1}^t \mathbf{Y}$	
Make a matrix of variable means	$\bar{\mathbf{Y}} = \mathbf{1} (\mathbf{1}^t \mathbf{1})^{-1} \mathbf{1}^t \mathbf{Y}$	
Mean-center the data matrix	$\mathbf{Y}_c = \mathbf{Y} - \bar{\mathbf{Y}}$	
Calculate the sums of squares and cross-products (SSCP)	$\mathbf{S} = \mathbf{Y}_c^t \mathbf{Y}_c$	
Make the SSCP matrix into a covariance matrix	$\mathbf{C} = \frac{1}{(n - 1)} \mathbf{S}$	
Make a diagonal matrix of just the variances	$\mathbf{V} = \text{diag}(\mathbf{C})$	
Calculate a correlation matrix	$\mathbf{R} = \mathbf{V}^{-1/2} \mathbf{C} \mathbf{V}^{-1/2}$	

Ordination	Variances and Distances	Slide 6
Mean-center the data matrix	$\mathbf{Y}_c = \mathbf{Y} - \bar{\mathbf{Y}}$	



A **rigid rotation** of the multivariate data space

It preserves Euclidean Distances between objects (no distortion)

The rotation is done to find the direction of most variation

This is done using **Singular Value Decomposition (SVD)***

It produces a set of **new axes**, rank-ordered by % variation

These axes are linear combinations of initial variables

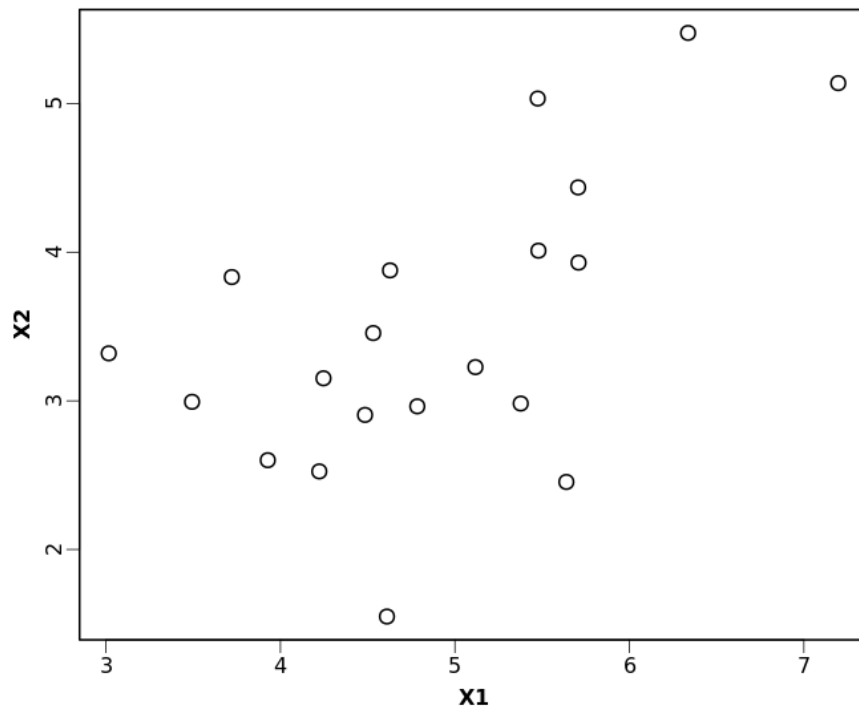
They are **orthogonal**, i.e. uncorrelated (mathematically!)

The data are **projected** on these new axes for visualization

*Or eigenanalysis, which is a special case of SVD (see below)

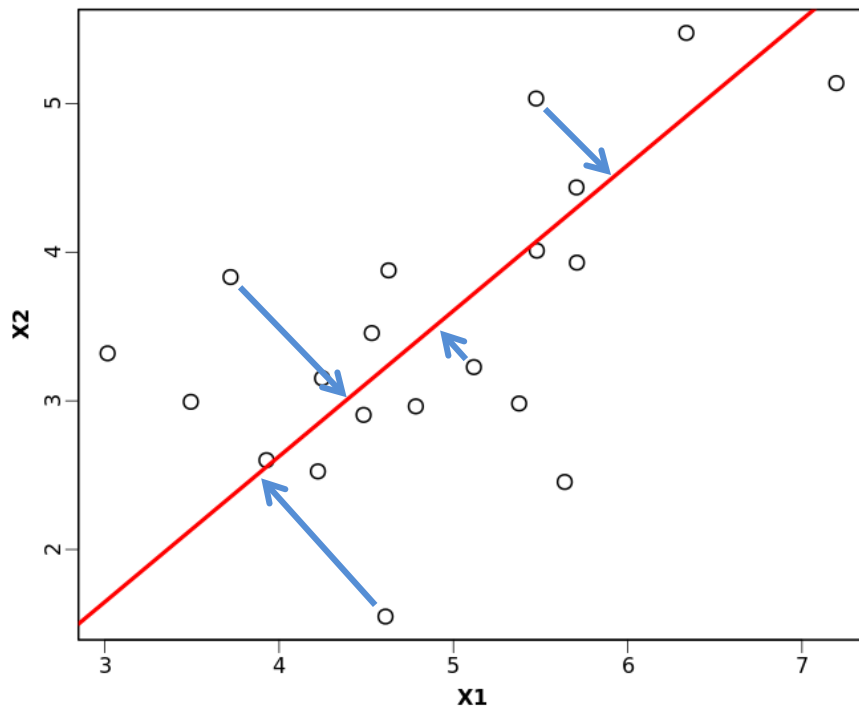
Consider the simple case where $k=2$

X_1, X_2 : two continuous variables that co-vary



Consider the simple case where $k=2$

X_1, X_2 : two continuous variables that co-vary



There is a line for which the distances of the points to it are minimized

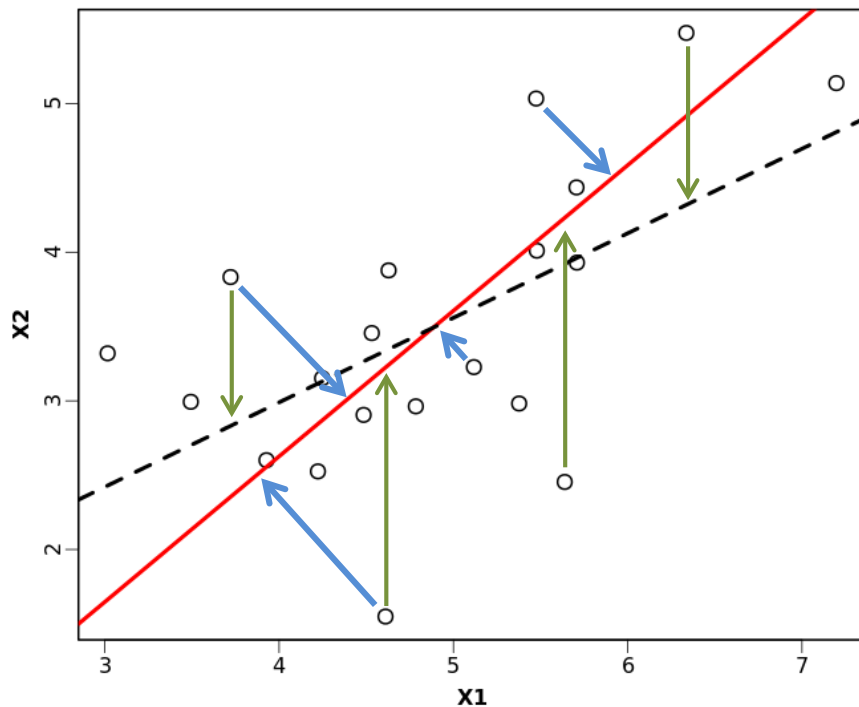
This line summarizes the maximum covariation between X_1 and X_2

This is known as the major axis regression of X_2 on X_1

This is NOT the least squares regression line

Consider the simple case where $k=2$

X_1, X_2 : two continuous variables that co-vary



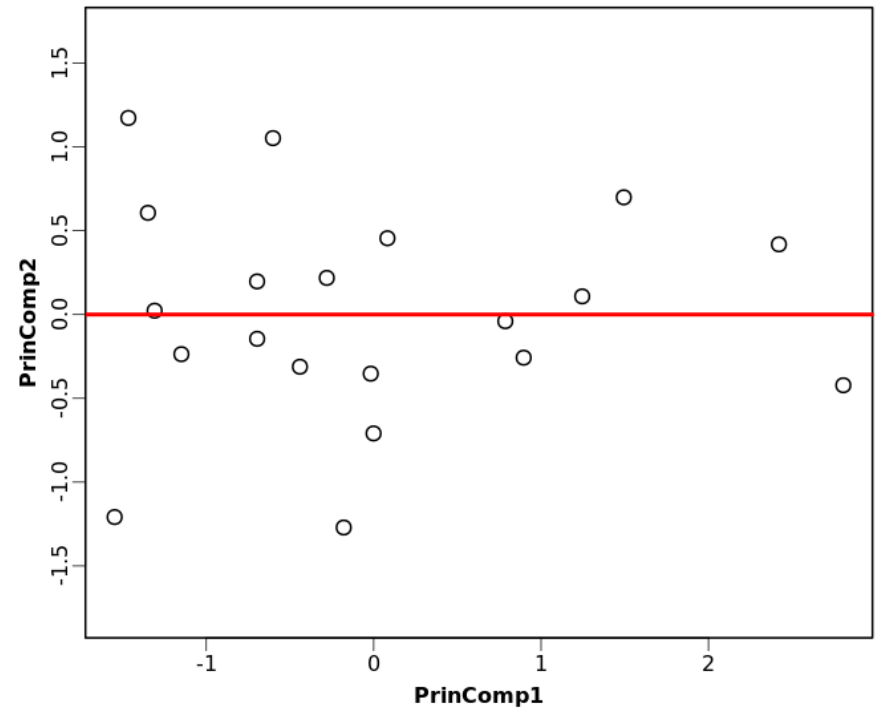
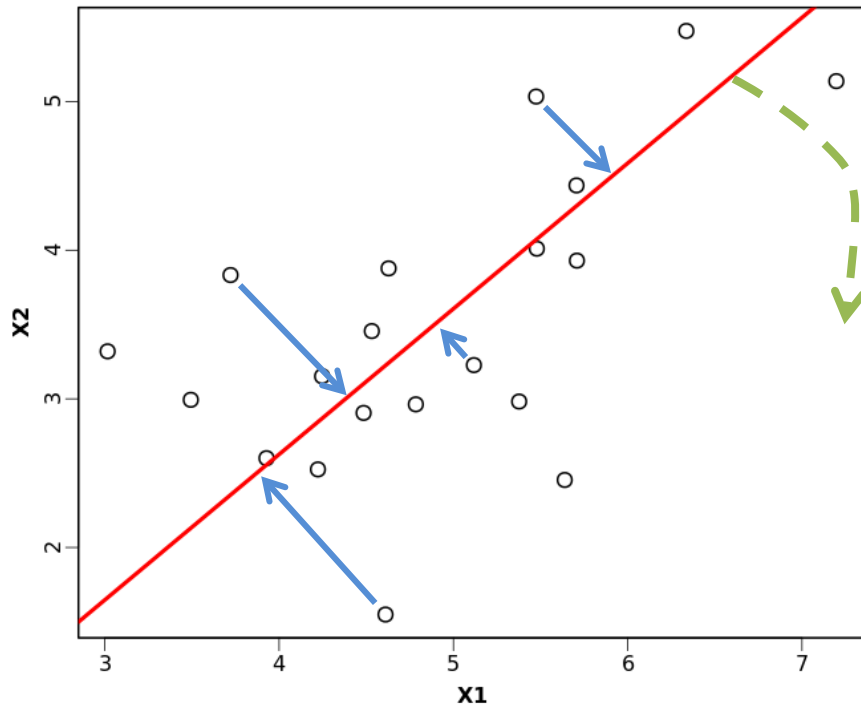
There is a line for which the distances of the points to it are minimized

This line summarizes the maximum covariation between X_1 and X_2

This is known as the major axis regression of X_2 on X_1

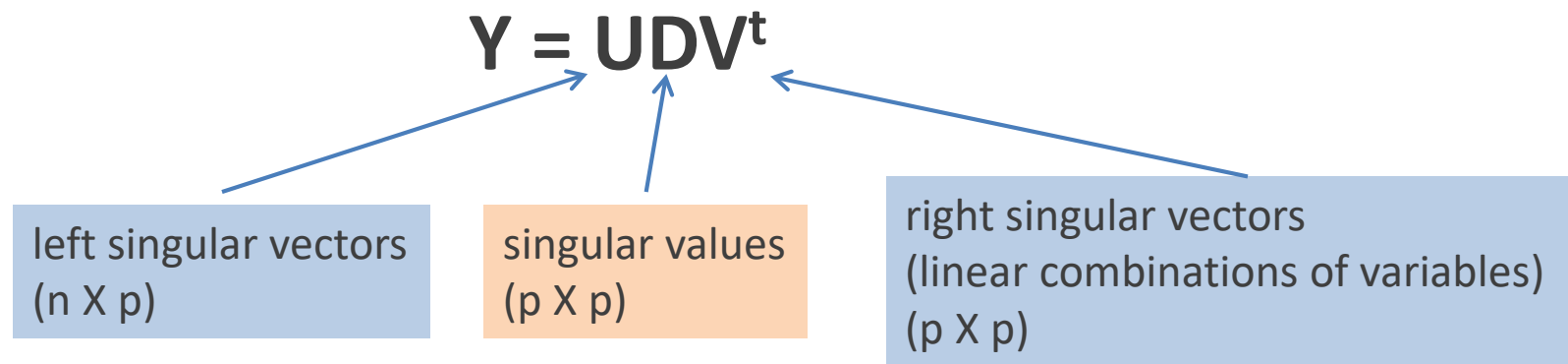
This is NOT the least squares regression line

The orthogonal best fit line can be rotated to visualize the simplest covariation pattern



Principal components (the direction of maximal variation) can be calculated for any number of dimensions

1. Start with the data matrix Y
2. Decompose this into singular vectors and singular values as



D^2 : expresses the % variance explained by each PC axis

UD : PC scores for each observation

NOTE: U also found from eigen-analysis of YY^t , and V found from eigen-analysis of Y^tY .

The 'classic' implementation of PCA is using eigen-analysis: $S = E\Lambda E^t$, where E contains the eigenvectors and Λ the eigenvalues. PCA scores are found as: $P=YE$

Usually we think of PCA using eigenanalysis

Eigenanalysis: a specific case of SVD for square symmetric matrices

1. Start with covariance (or correlation) matrix of Y
2. Decompose using eigenanalysis as

$$\mathbf{C} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^t$$

The diagram illustrates the eigenanalysis decomposition of the covariance matrix \mathbf{C} . The equation $\mathbf{C} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^t$ is shown at the top. Below it, two boxes are positioned: a blue box labeled 'eigenvectors' on the left and an orange box labeled 'eigenvalues' on the right. Two blue arrows point from these boxes towards the equation: one arrow points from the 'eigenvectors' box to the \mathbf{E} term, and the other arrow points from the 'eigenvalues' box to the $\mathbf{\Lambda}$ term.

$\mathbf{\Lambda}$: % variance explained

PC scores are found by projection of Y on the eigenvectors, as $\mathbf{P} = \mathbf{Y}\mathbf{E}$

PCA is a rigid rotation where new axes explain % variation

- Distances among objects preserved (so long as all dimensions containing variation are used)!

PC axes are loadings of each variable on PC axis (variables with values closer to -1 & +1 are more influential in that direction)

How well does a particular PC plot represent relationships among objects?

Assess by:

- % variation explained by PC axes

- Mantel correlation and Shepard diagram (plot of distances in reduced PC space vs. distances in full data space)

To interpret a PCA we look at:

- % of variation explained
- structure of PCs (loadings of eigenvectors, or correlations with variables)
- ordination of observations in PC plot

-PCA does **nothing to the data**, it is just a rigid rotation

-It does **NOT find a particular dimension** (group differences, allometry, altitudinal gradient etc): it only finds the direction of highest variability in the data

-PC axes are orthogonal, so caution is needed in interpreting their biological meaning

-Some criteria exist for how many PC axes to interpret (broken stick model, log-ratio method, etc.: see Legendre & Legendre Num. Ecol.; also Mardia et al. 1979 Mult. Anal.; Jackson 1993, Ecol.)

In some cases, some dimensions may have 0% variation ($\lambda < 0$)

When $N > p$, there are $N-1$ PC axes with variation*

When $p > N$, there are p PC axes

Fewer PCs with variation may also occur when there is redundancy in the data

- compositional data: $A+B+C=1$,

- linear dependencies: $Y_1 = Y_2 + Y_3$

*This property holds for metric distances

If a variable varies much **more** than others, it dominates the PCA

(“more” is relative: 0 - 100cms is numerically “more” than 0 - 1m)

Particularly relevant when variables are measured in different units

Standardizing data often alleviates this problem

a) center and scale Y (subtract the mean and divide by standard deviation)

b) use the correlation matrix

a) and b) are equivalent, i.e. $\text{cov}(Y.\text{std}) = \text{cor}(Y)$

```
> pca.bumpus <- prcomp(Y)
```

```
> summary(pca.bumpus)
```

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	0.06288	0.03574	0.02074	0.01577	0.01497	0.01485
Proportion of Variance	0.62225	0.20098	0.06766	0.03915	0.03527	0.03470
Cumulative Proportion	0.62225	0.82322	0.89088	0.93003	0.96530	1.00000

```
> pca.bumpus$rotation
```

	PC1	PC2	PC3	PC4	PC5	PC6
AE	0.2671219	0.02110608	-0.03036462	0.92618034	-0.16953793	-0.2018051
BHL	0.2591896	-0.08769478	-0.26991755	-0.17087231	-0.85357143	0.3073977
FL	0.4531669	-0.39808021	0.10060855	0.10052754	0.37487601	0.6895013
TTL	0.4603097	-0.51621150	0.35219851	-0.24850014	-0.07573124	-0.5745514
SW	0.2464018	-0.10723566	-0.87884180	-0.09611748	0.29570037	-0.2423767
SKL	0.6192799	0.74526673	0.14033437	-0.17859440	0.09469297	-0.0226606

```
> pca.bumpus <- prcomp(Y, scale.=T)
```

```
> summary(pca.bumpus)
```

Importance of components:

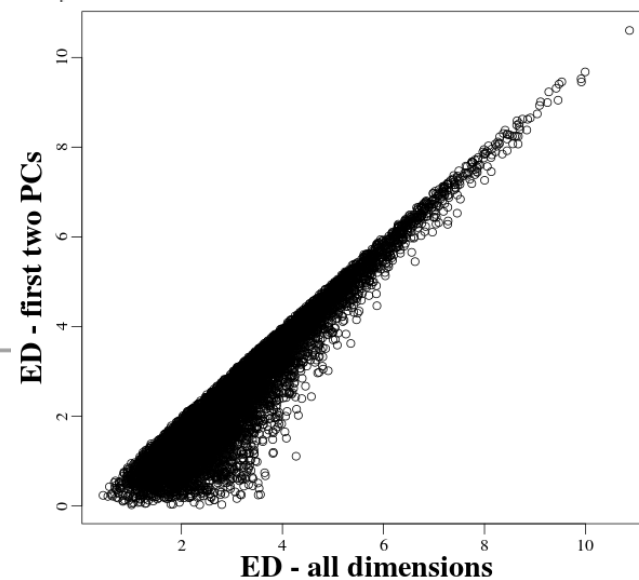
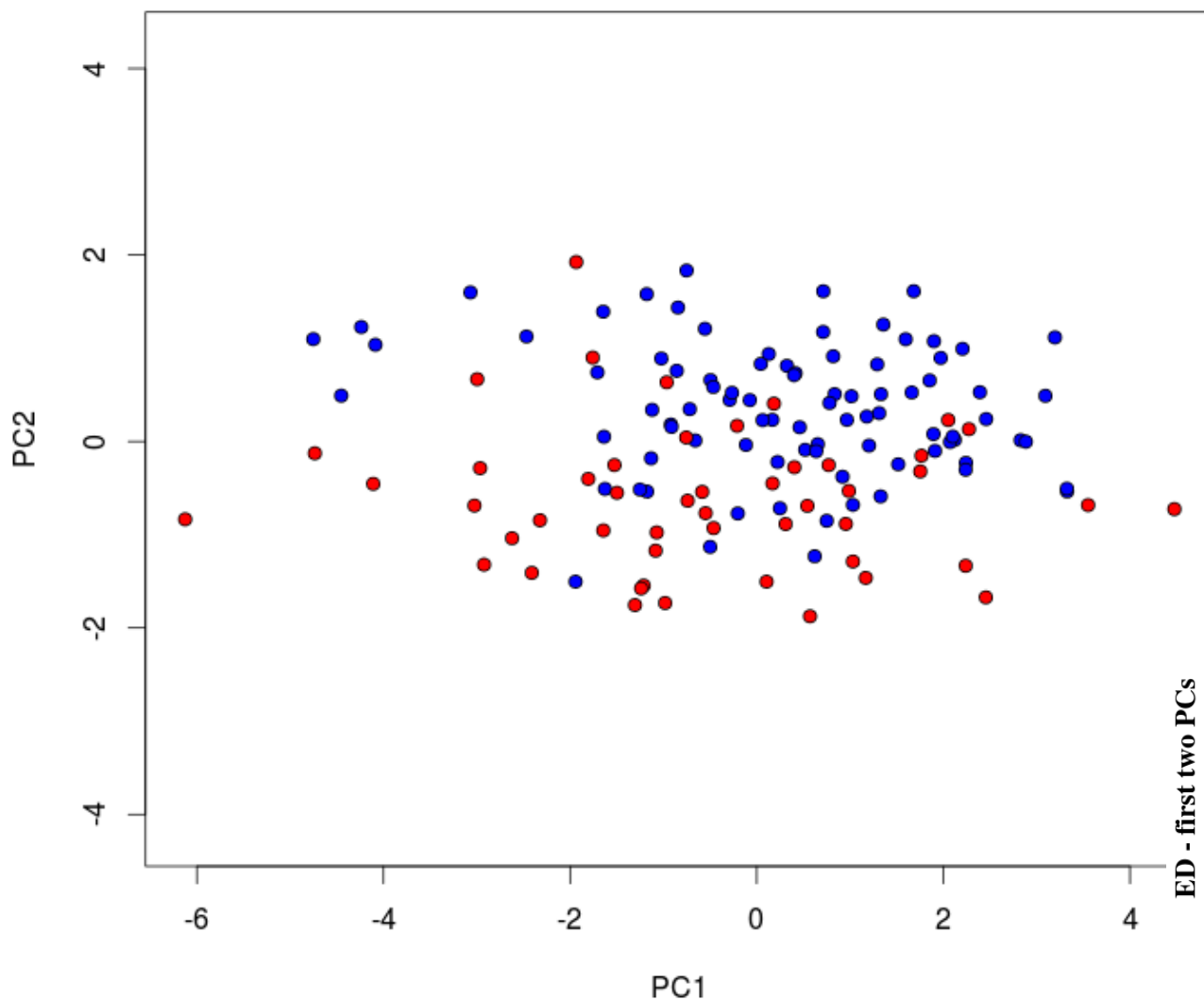
	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	1.9110	0.8633	0.7838	0.66811	0.59933	0.4278
Proportion of Variance	0.6086	0.1242	0.1024	0.07439	0.05987	0.0305
Cumulative Proportion	0.6086	0.7329	0.8352	0.90963	0.96950	1.0000

```
> pca.bumpus$rotation
```

	PC1	PC2	PC3	PC4	PC5	PC6
AE	0.4061953	0.38791502	-0.25664018	-0.51015126	-0.5984397	-0.016694516
BHL	0.4196692	-0.07341605	0.21416001	0.73531917	-0.4815613	0.005262311
FL	0.4531108	-0.33702800	-0.26338163	-0.06552501	0.2371850	0.742416731
TTL	0.4303778	-0.43806789	-0.36689593	-0.03673575	0.2153367	-0.663731472
SW	0.3690596	-0.13025008	0.82639874	-0.37602393	0.1340963	-0.067225280
SKL	0.3635158	0.72227113	-0.03679566	0.22806658	0.5379220	-0.058756221

PC1: loadings similar and positive (size)

PC2: relative shape: SKL vs. TTL & FL



```
> pca.bumpus$rotation[,2]
```

AE

BHL

FL

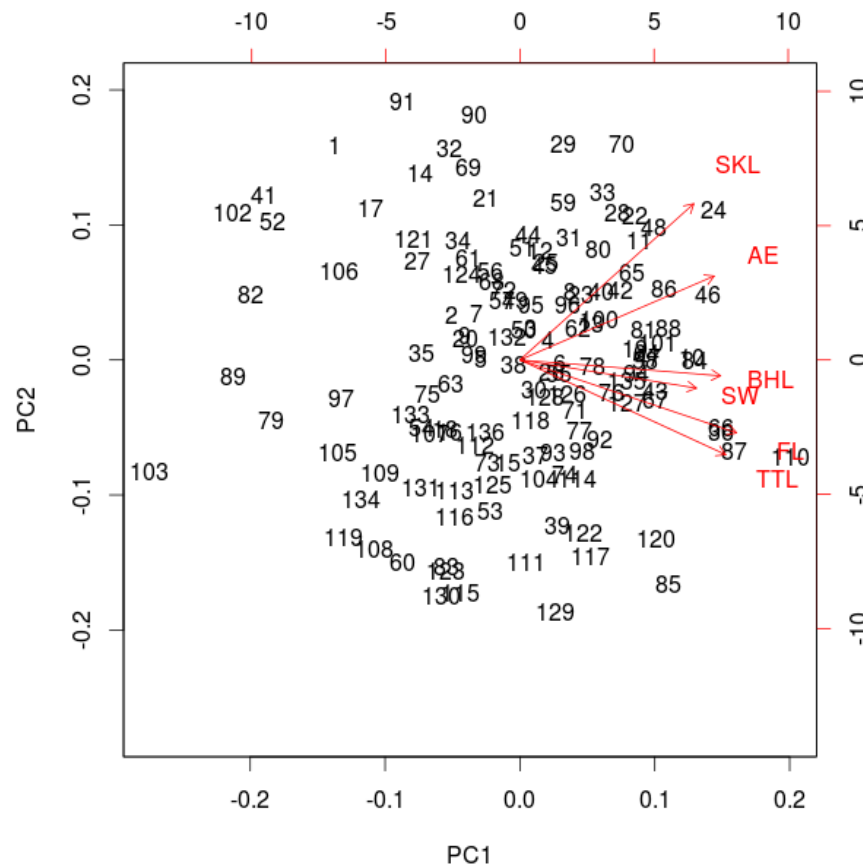
TTL

SW

SKL

0.38791502 -0.07341605 -0.33702800 -0.43806789 -0.13025008 0.72227113

- Ordination plot of objects (rows) and variables (columns)
- Look for sets of vectors with small angles, and clusters of points
- Can use to identify variables with high association with objects



PCA is helpful for exploring multivariate space and it may **suggest** patterns

It is exploratory! It is nothing more than a rigid rotation of the multivariate data space

No hypotheses are tested

Using the correlation vs. covariance matrix has important implications

It preserves the Euclidean distances among objects in the original multivariate data space

Very useful for summarizing the results of hypothesis-testing approaches applied to multivariate data(e.g. MANOVA)

Ordination from distance matrix among objects (Q-mode)

-Preserves object distances from *any* distance measure (D_E , Jaccard, etc.)

Projects data into Euclidean space

If D_{euclid} used, PCoA identical to PCA (Gower, 1966)

Sometimes called metric multidimensional scaling (MDS) because it preserves relationships among objects

* *aka* Classical Multidimensional Scaling

Start with \mathbf{D} , the matrix of pairwise distances among objects (d_{ij}^2)

Transform as $\mathbf{A} = -\frac{1}{2}\mathbf{D}^2$

Double-center $\mathbf{G} = \left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^t\right)\mathbf{A}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^t\right)$

$\mathbf{I} = n \times n$ identity matrix
 $\mathbf{1} = n \times 1$ matrix of 1s
 \mathbf{G} = double-centered matrix
(named after Gower, 1966)

This positions the centroid of the scatter at the origin by subtracting row and column means from each element, and adding the grand mean

Decompose using eigenanalysis $\mathbf{G} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^t$

Eigenvectors are the *COORDINATES* of objects in Euclidean space

PCoA embeds a set of objects into a Euclidean space

-Identical to PCA for continuous multivariate data

Strength: Enables ordination for *any* data types expressed as distances (genetic distance, Hamming distance, geographic distance, etc.)

Can yield negative eigenvalues (if distances are semi-metric or nonmetric)

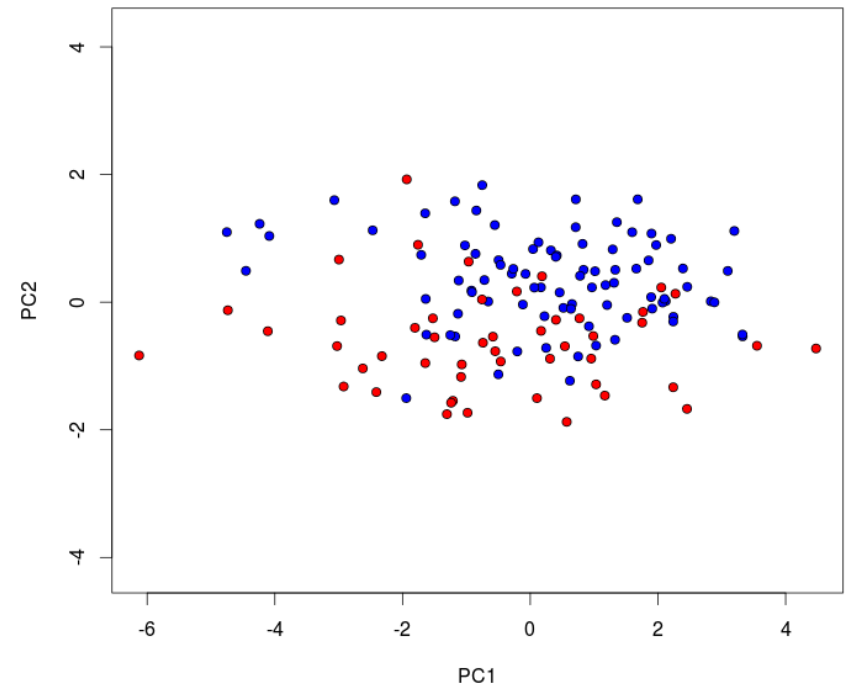
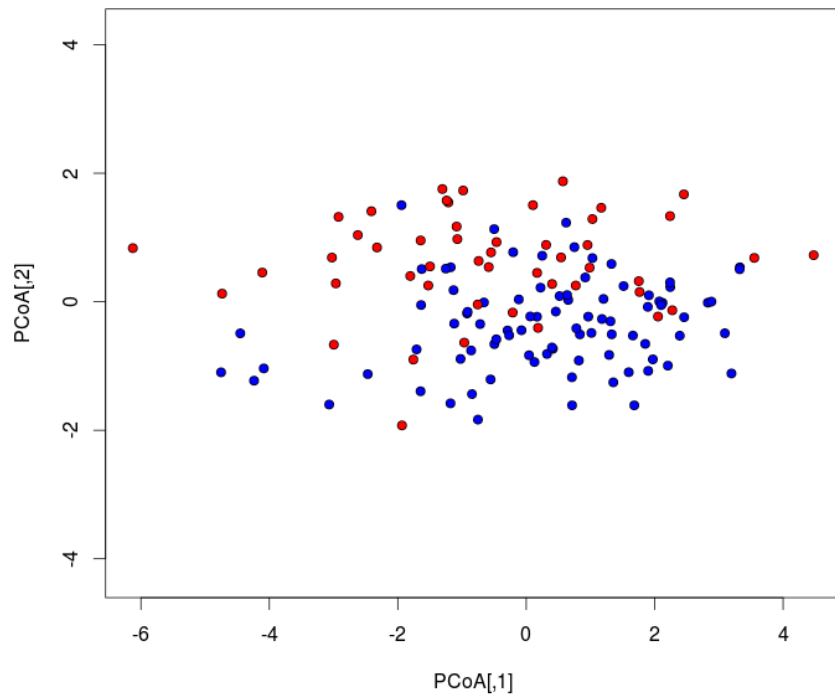
Properties of Distances

Metric (Euclidean): (1) $d_{11} = 0$, (2) $d_{12} = d_{21}$, (3) triangle inequality

Semimetric (pseudometrics): no triangle inequality

Non-Metric: d_{12} may be negative

```
> Y <- scale(Y, center = T, scale = T)
> bumpus.dist <- dist(Y)
> PCoA <- cmdscale(bumpus.dist)
```



PCA and PCoA preserve distances among objects

Non-metric MDS generates ordination where similar objects are close together and dissimilar objects are far apart

Preserves the relative **order** of the objects, but not the distances themselves

Objective is to find a low dimensional representation of the data space

- Start with distance matrix
- Specify number of dimensions for MDS ordination *a priori*
- Construct initial configuration of objects (a ‘guess’).

This step is **crucial**, as no distances are optimized, but rather the fit is tested and iterated until stable (the result from PCoA is often used)

- Calculate fitted distances in NMDS space and compare to true distances
- Calculate predicted distances, and goodness of fit (stress)

$$Stress1 = \sqrt{\sum (d_{fitted} - \hat{d}_{fitted})^2 / \sum d_{fitted}^2}$$

- Move objects in NMDS plot and repeat (GOAL: Monotonic fit of D_{obs} vs. D_{fit})
- Iterate until Δ stress is below threshold (i.e., convergence)

Note: other Stress equations exist

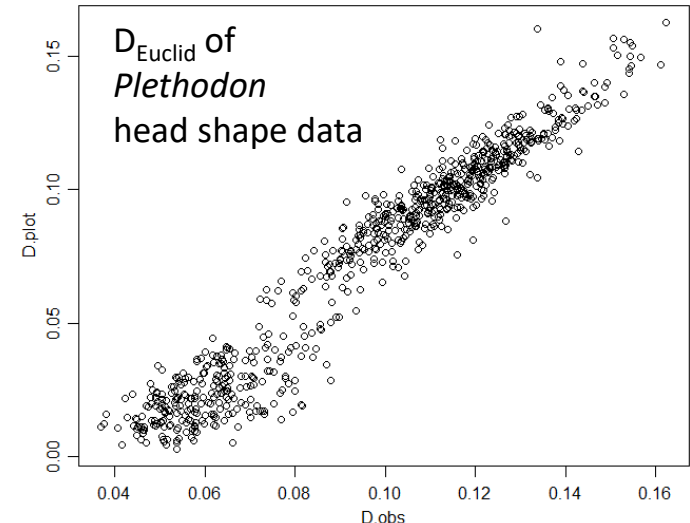
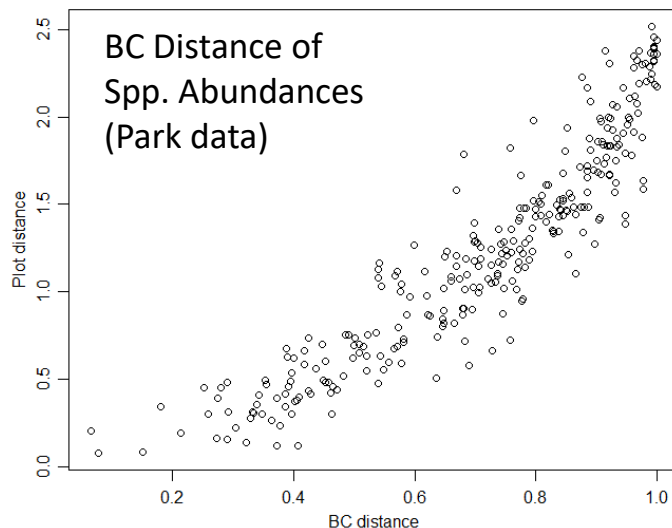
-NMDS seems arbitrary, but works rather well

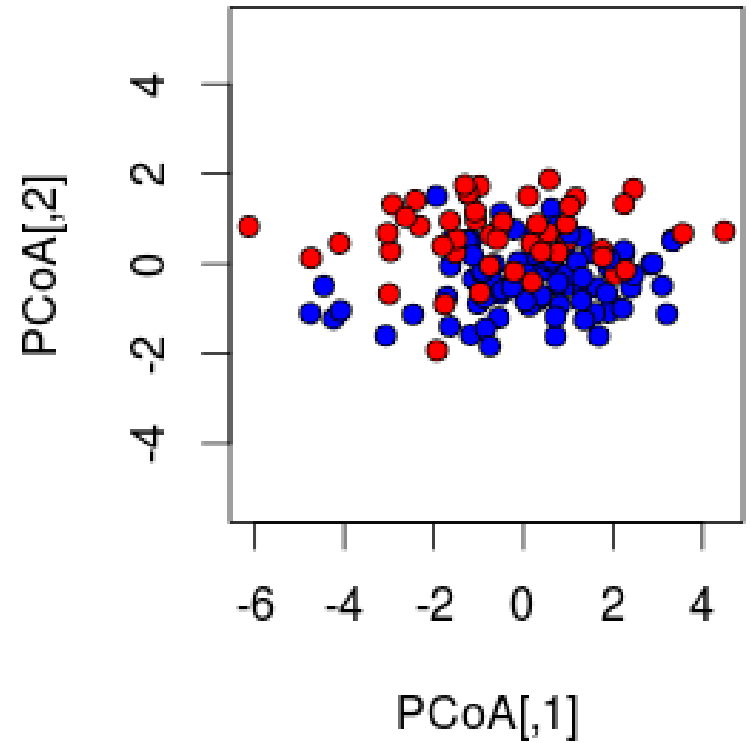
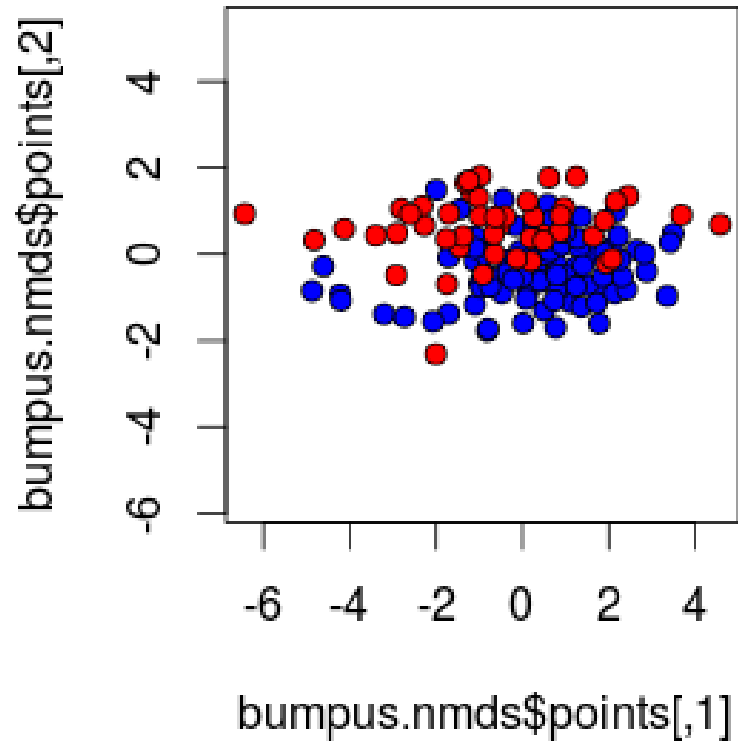
Positives:

- Generally yields fewer dimensions than PCA, PCoA
- Does not require full distance matrix (missing values ok)

Negatives:

- Arbitrary optimization
- Results dependent on starting configuration ('guess')
- Does not preserve distances among objects (though that is not the objective)
 - Minimizes monotonic relationship of D_{fit} vs. D_{obs}





Ordination for count and frequency data

Very frequently used in ecology (community data, species presence etc)

Preserves the χ^2 distance among objects (a weighted Euclidean distance of conditional probabilities)

$$D_{Euclid} = \sqrt{\sum_k (y_{ki} - y_{kj})^2}$$

$$D_{X^2} = \sqrt{\sum_k \frac{1}{y_k} \left(\frac{y_{ki}}{y_i} - \frac{y_{kj}}{y_j} \right)^2}$$

$$\text{where } y_k = \sum y_{+k} / \sum y_{++}$$

$$\text{and } y_i = \sum y_{i+}$$

Provides a test of independence of rows and columns

1. Calculate matrix of relative frequencies (Q) from contingency table

Elements of Q : $p_{ij} = f_{ij}/f_{total}$

2. Standardize Q by centering by rows and columns*

Elements of \bar{Q} : $q_{ij} = \left[\frac{p_{ij} - p_{i+}p_{j+}}{\sqrt{p_{i+}p_{j+}}} \right]$

3. Decompose using SVD $SVD(\bar{Q}) = \hat{U}WU'$

- \hat{U} : eigenvectors for rows
- U' : eigenvectors for columns
- W : singular values
- Obtain ordination plot from scaled and projected row and column factors

* Hence CA is aka *reciprocal averaging*

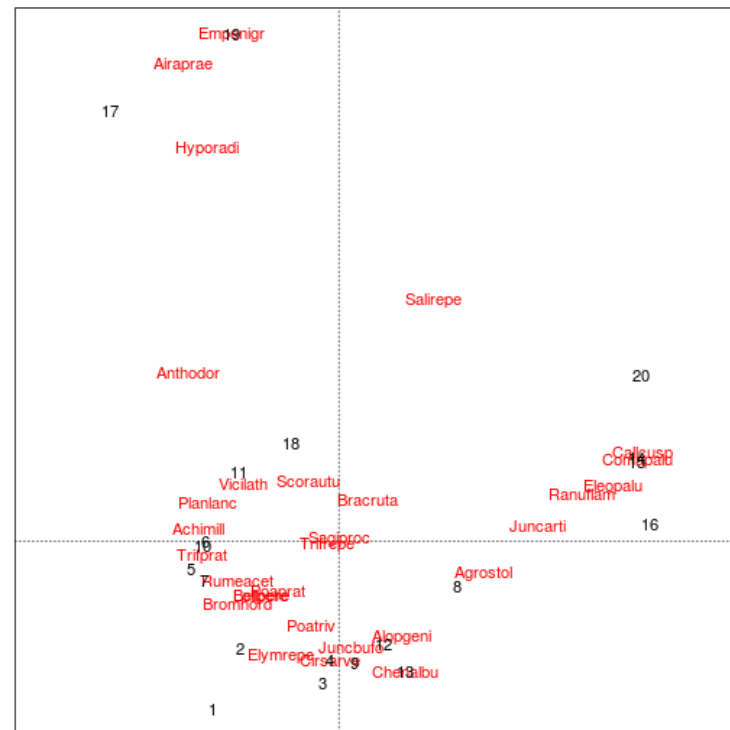
- Similarity of objects from frequency data can be viewed using ordination
- Test of independence between objects and variables (significance implies some objects have higher frequencies on particular variables)
- Ordination can be as a biplot (simultaneous plot of rows and columns: objects and variables)
- Interesting Note: Eigenanalysis of $\overline{Q}Q'$ yields \hat{U} , and
eigenanalysis of $\overline{Q}'Q$ yields U'
- Advantage of SVD is that it decomposes rows AND columns simultaneously

data(dune) from vegan: dune meadow vegetation data from the Dutch island of Terschelling

rows: 20 2x2m randomly selected plots sampled in the island in 1982 for a dune conservation project

columns: plant species

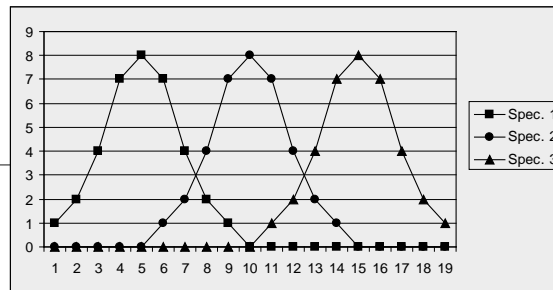
values: cover class (standardized estimate of density)



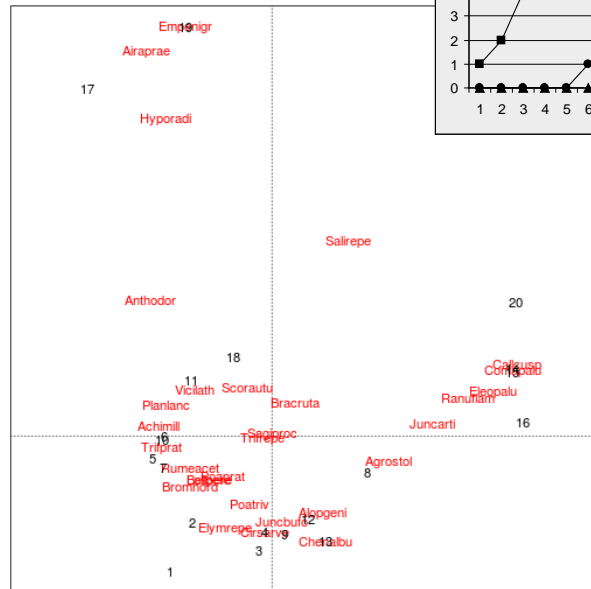
Although most ordination methods construct **mathematically** orthogonal axes, real life is not constrained to be orthogonal!

This may create problems, e.g. the “arch effect” (*aka* the “horseshoe” effect)

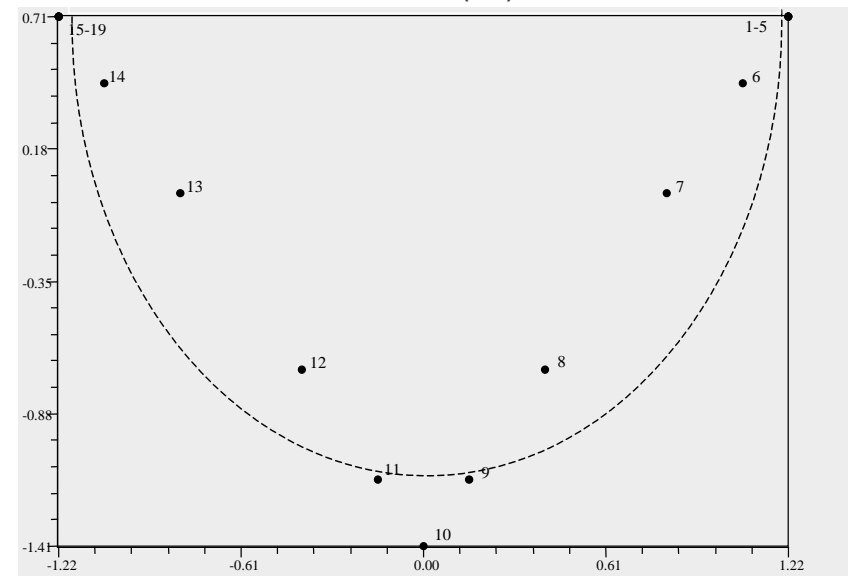
The 2nd axis is an arched function of the 1st axis (common in ecological data, e.g. species counts in sites along an environmental gradient)



Hypothetical abundance data of 3 species for sites along an ecological gradient



Ordination (CA) of sites from above data



‘Correct’ for arch effect in CA plots to represent underlying gradient in ordination

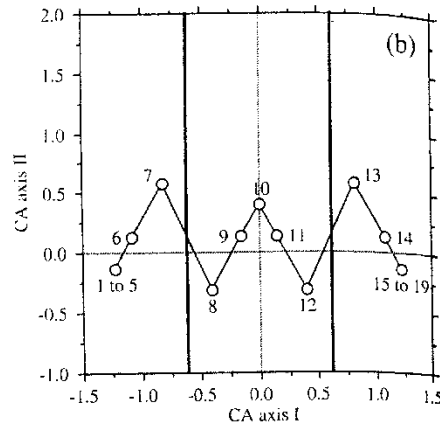
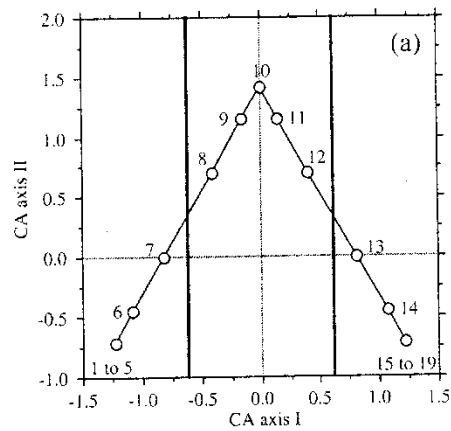
Two common approaches:

DCA by segments

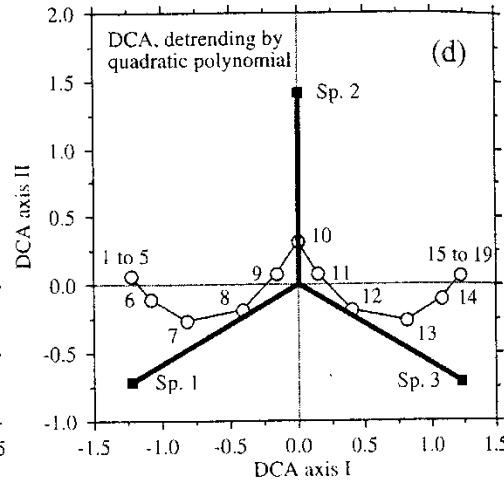
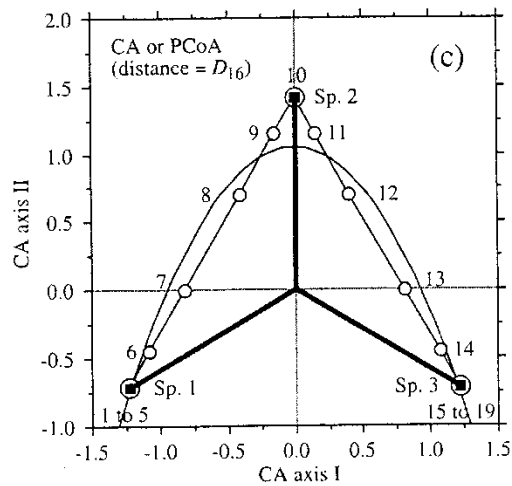
- Calculate CA ordination and divide DC1 into segments
- Calculate DC2 (for each segment, calculate mean for DC2, and subtract from DC2 scores)

DCA by polynomials

- Additional constraint imposed in CA algorithm, such that DC2 is orthogonal to DC1 (linear), and also polynomials of DC1 (square, cubic, quartic, etc.)



DCA by segments



DCA by polynomials

Approaches are arbitrary (how to choose segments? What order polynomial to use?)

DC2 now completely **meaningless** (proximities of objects on DC2 cannot be interpreted)

DCA plot **distorted**, no longer represents distances among objects

Detrending eliminates (hides) arch effect, that may in itself be useful information (i.e., the arch *IS* the pattern in the data!)

Detrending should absolutely be avoided!

For detailed critique see Wartenberg, Ferson, and Rohlf, 1987. *Am. Nat.* 129:434-448.