Workshop 9: Multivariate analyses

QCBS R Workshop Series

Québec Centre for Biodiversity Science

About this workshop

Required packages

- ape
- gclus
- vegan

```
install.packages(c('ape', 'gclus', 'vegan'))
```

Learning objectives

Use R to perform an unconstrained ordination

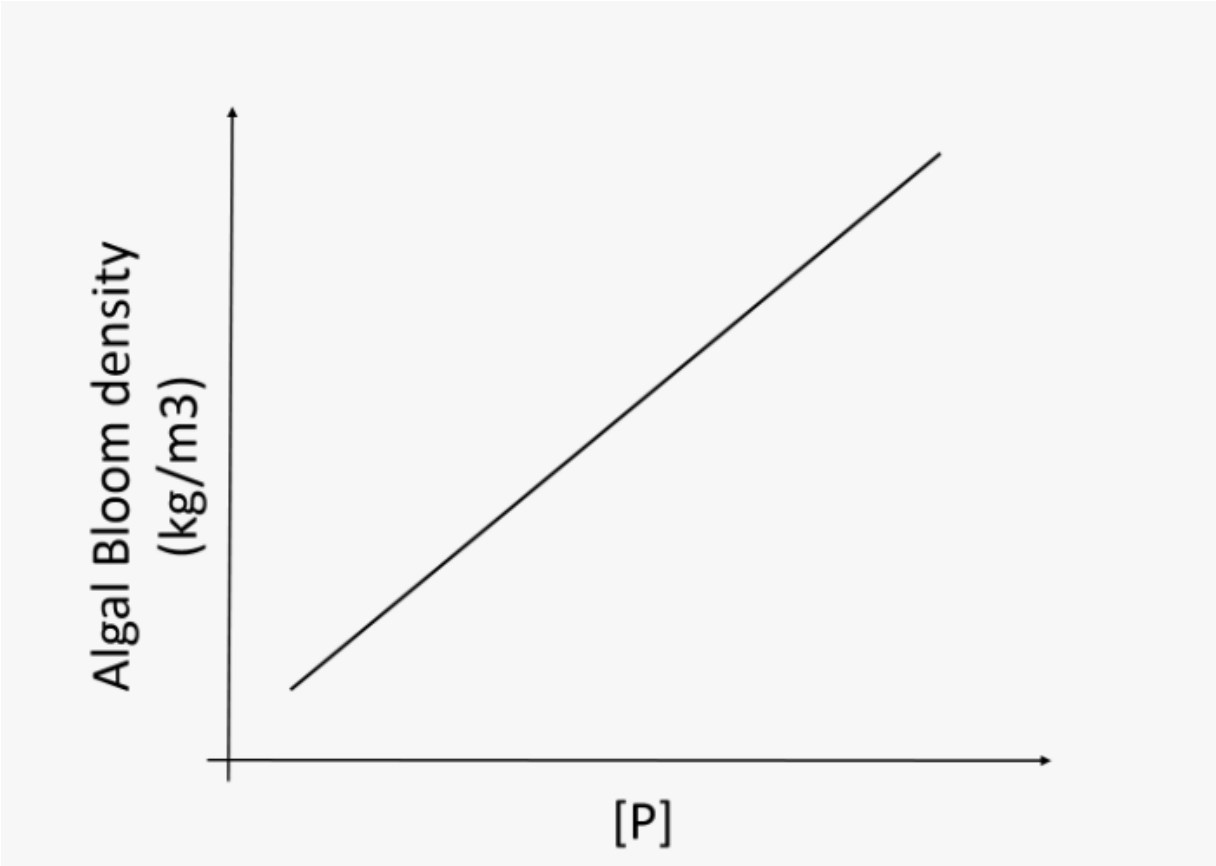
Use R to create dendrogram

1. Introduction

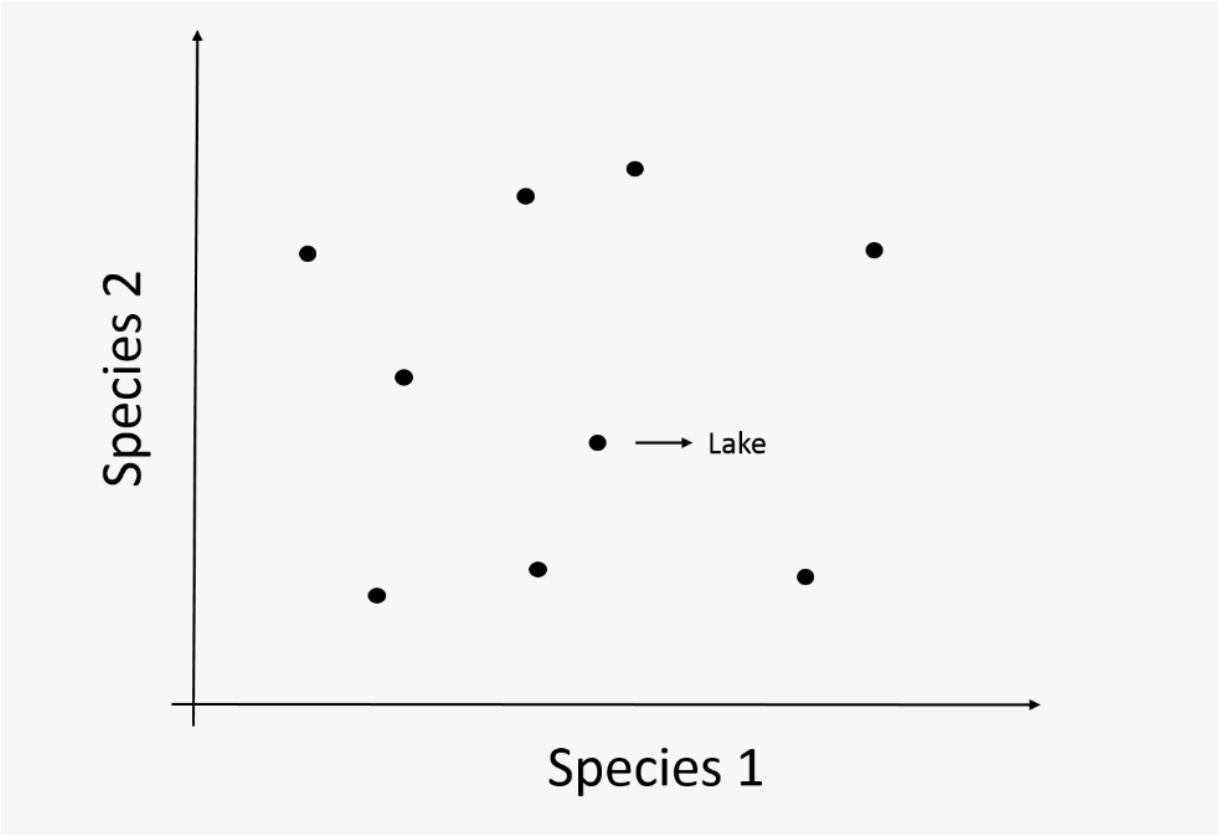
What is ordination?

One Dimension

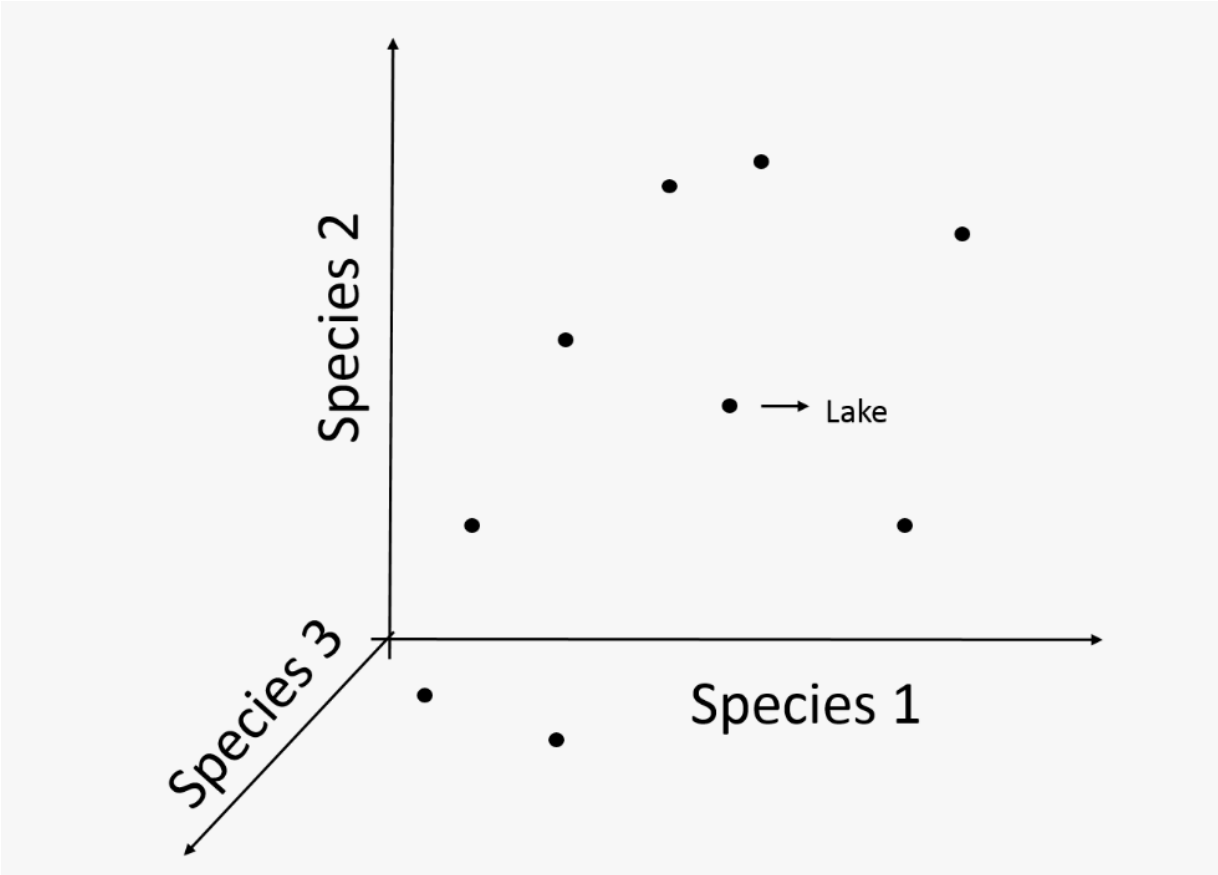
What if we are interested in this response for different species of algae involved in the algal bloom density?



Two Dimensions

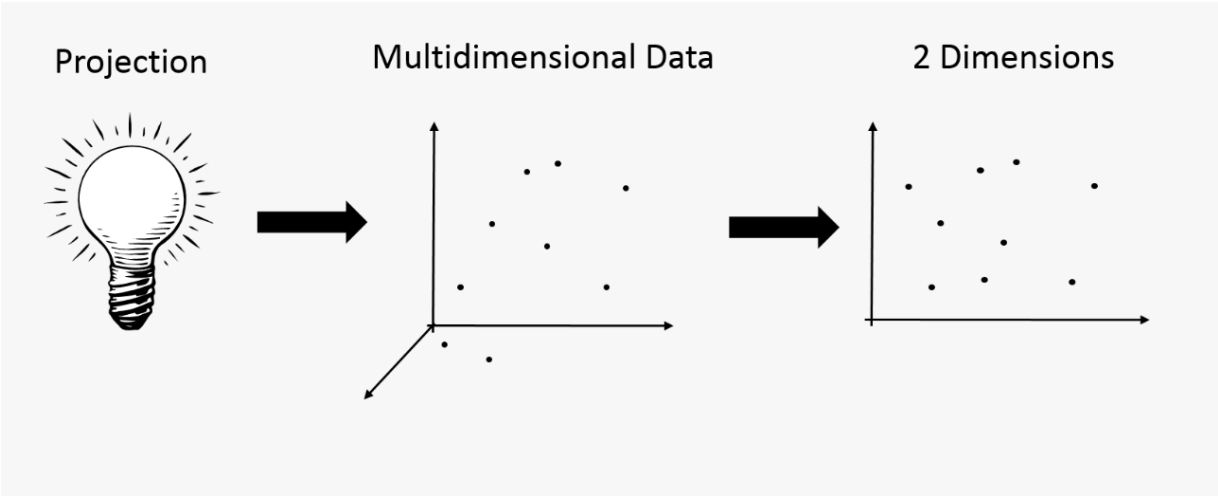


Three Dimensions

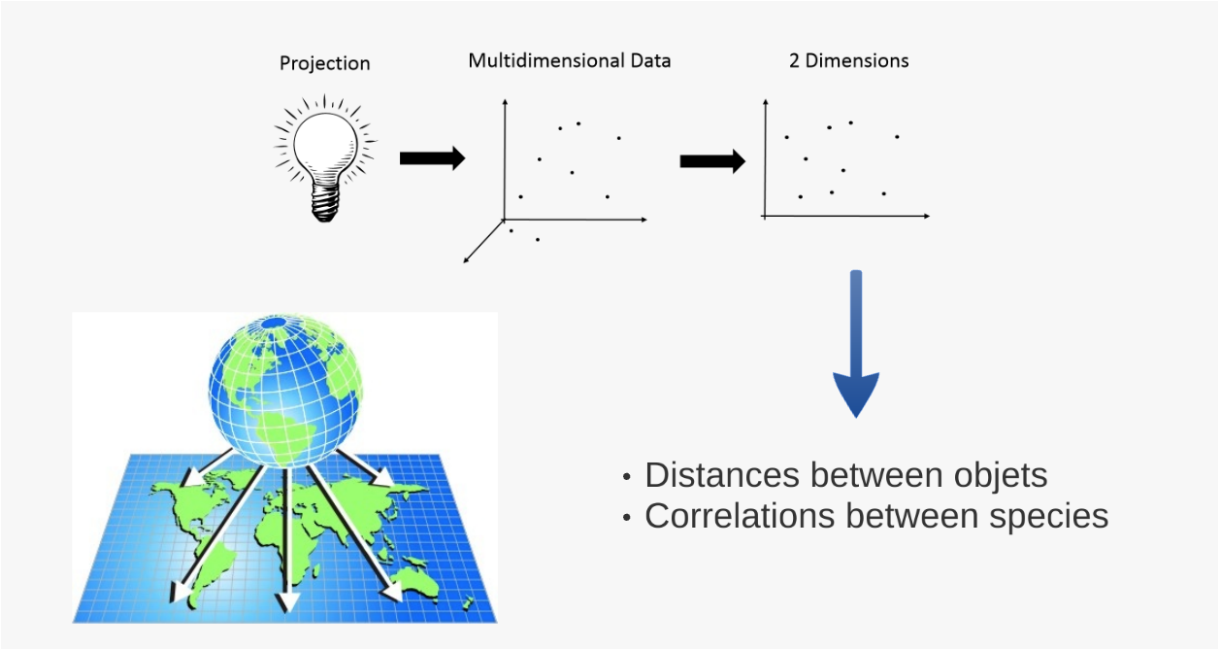


4,5,6, or more Dimensions

Ordination in reduced space



Ordination in reduced space



- Matrix algebra is complex and hard to understand
- A global understanding is enough in order to use ordination methods adequately

Methods for scientific research

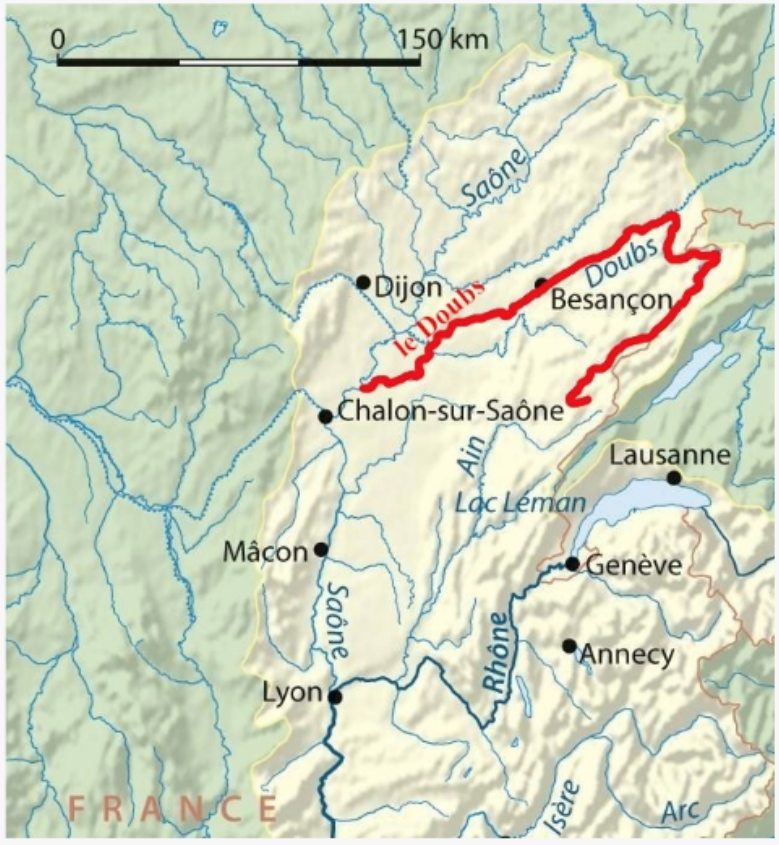
- **Questions / Hypothesis**
- **Experimental design**
- **Data Collection**
- **Transformation / Distance**
- **Analysis**
- **Redaction**
- **Communication**

2. Exploring data

Doubs River Fish Dataset

Verneaux (1973) dataset:

- characterization of fish communities
- 27 different species
- 30 different sites
- 11 environmental variables



Doubs River Fish Dataset

Load the Doubs River species data (Doubs.Spe.csv)

```
spe ← read.csv("data/doubsspe.csv", row.names = 1)
spe ←  spe[-8,] # remove site with no data
```

Load the Doubs River environmental data (Doubs.Env.csv)

```
env ← read.csv("data/doubsenv.csv", row.names = 1)
env ← env[-8,] # remove site with no data
```

Proceed with caution, only execute once

Expore Doubs Dataset

Explore the content of the fish community dataset

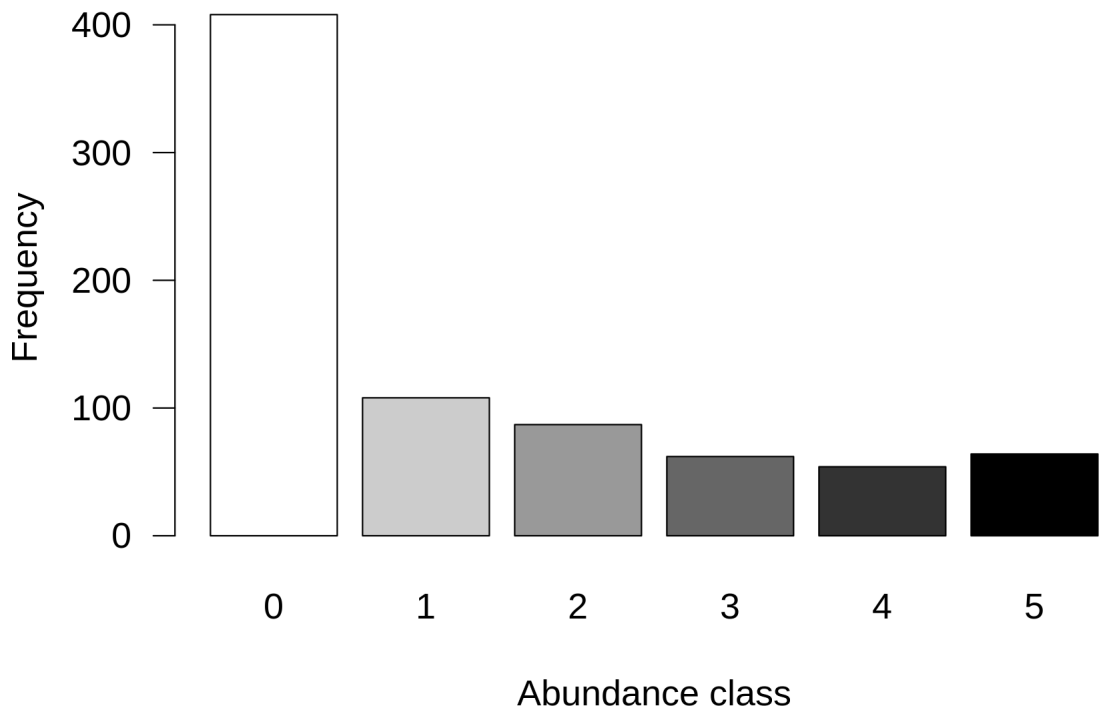
```
names(spe) # Names of objects
dim(spe) # dimensions
str(spe) # structure of objects
summary(spe) # summary statistics
head(spe) # first 6 rows
```

```
#   CHA TRU VAI LOC OMB BLA HOT TOX VAN CHE BAR SPI GOU BRO PER BOU PSO ROT CAR
# 1   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
# 2   0   5   4   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
# 3   0   5   5   5   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
# 4   0   4   5   5   0   0   0   0   0   1   0   0   1   2   2   0   0   0   0
# 5   0   2   3   2   0   0   0   0   5   2   0   0   2   4   4   0   0   2   0
# 6   0   3   4   5   0   0   0   0   1   2   0   0   1   1   1   0   0   0   0
#   TAN BCO PCH GRE GAR BBO ABL ANG
# 1   0   0   0   0   0   0   0   0
# 2   0   0   0   0   0   0   0   0
# 3   0   0   0   0   0   0   0   0
# 4   1   0   0   0   0   0   0   0
# 5   3   0   0   0   5   0   0   0
# 6   2   0   0   0   1   0   0   0
```

Species Frequencies

Take a look at the distribution of species frequencies

```
ab ← table(unlist(spe))
barplot(ab, las = 1, col = grey(5:0/5),
        xlab = "Abundance class", ylab = "Frequency")
```



Note the proportion of 0s

Species Frequencies

How many zeros?

```
sum(spe == 0)
# [1] 408
```

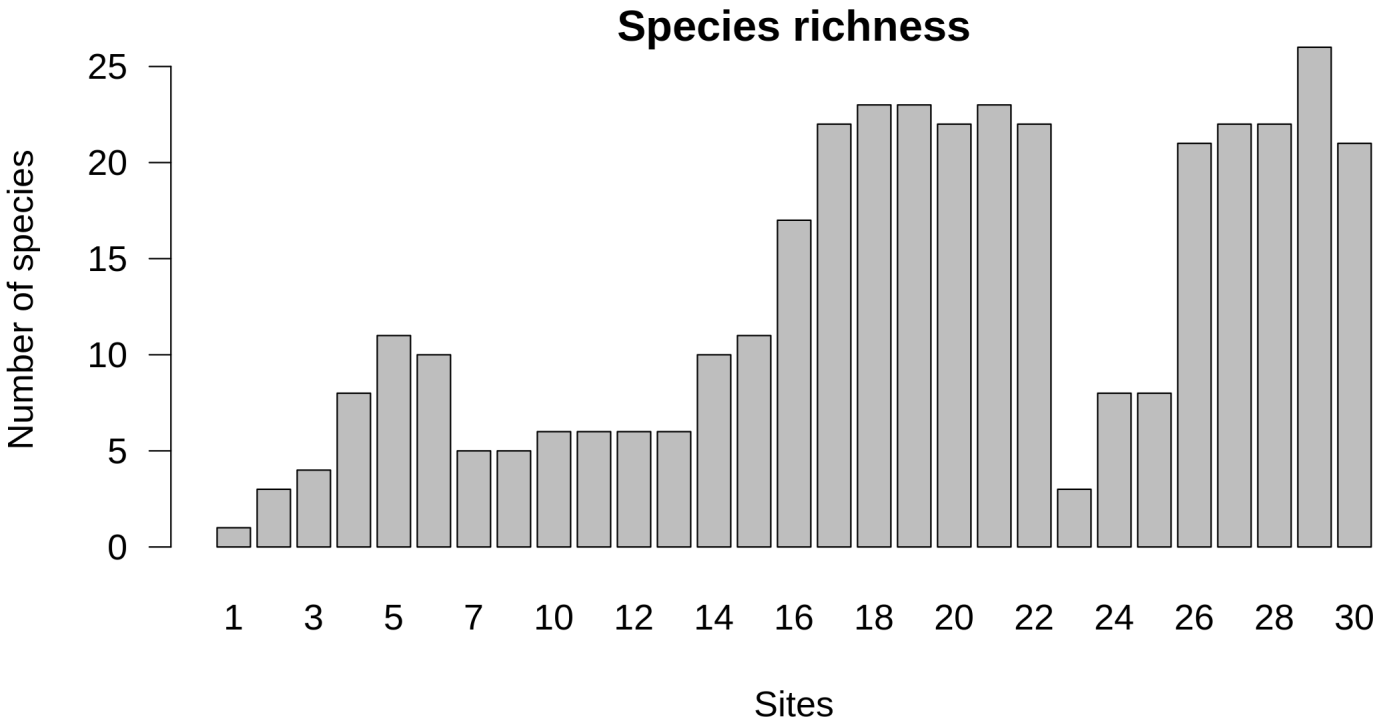
What proportion of zeros?

```
sum(spe == 0)/(nrow(spe)*ncol(spe))
# [1] 0.5210728
```

Total Species Richness

Visualize how many species are present at each site:

```
site.pre ← rowSums(spe > 0)
barplot(site.pre, main = "Species richness",
        xlab = "Sites", ylab = "Number of species",
        col = "grey ", las = 1)
```



Understand your data!

...to choose the appropiate transformation and distance

- Are there many zeros?
- What do they mean?

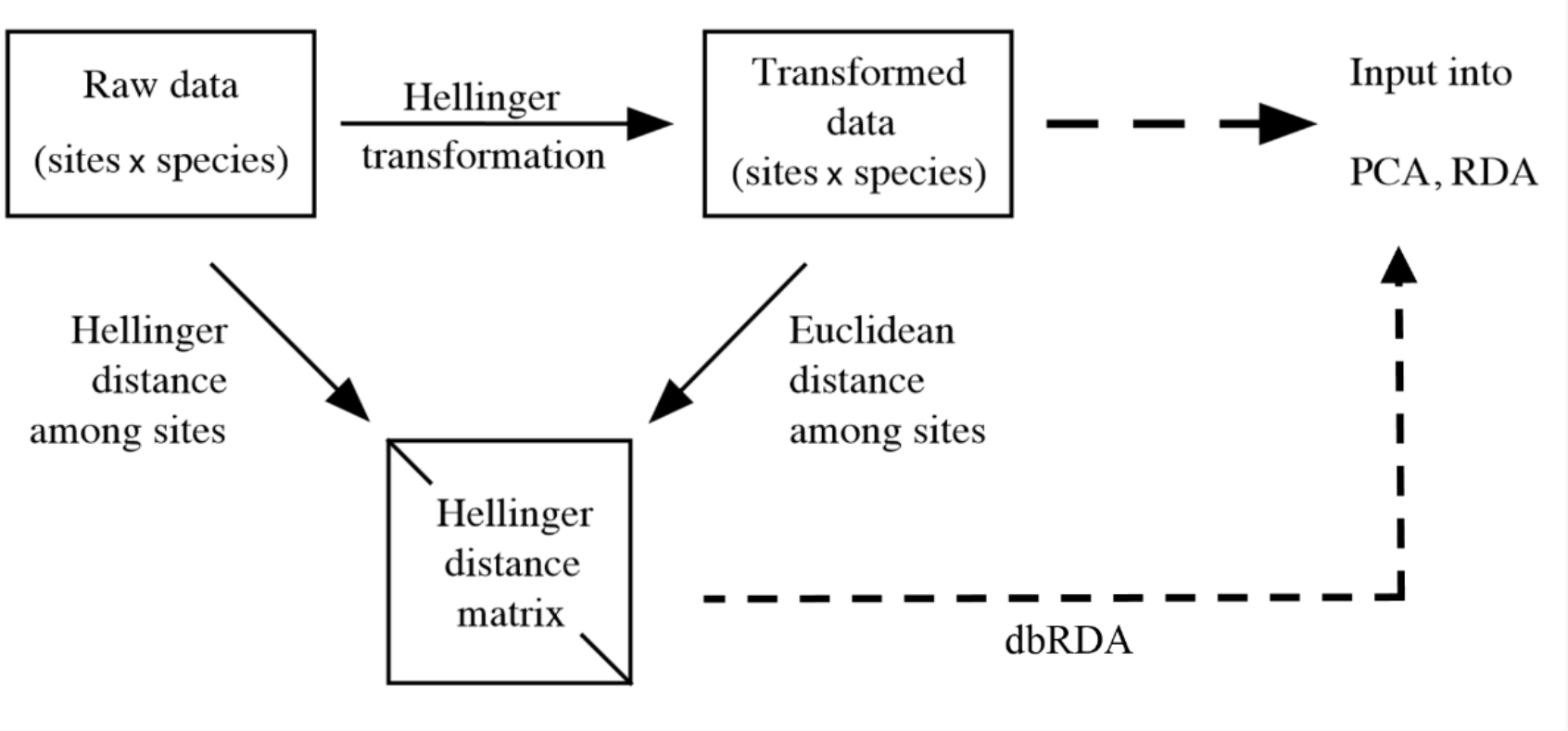
A measured 0 (e.g 0mg/L, 0°C) is not the same than a 0 representing an absence observations

Before transforming your community data...

Important considerations:

- relative abundances/counts/presence-absence?
- asymmetrical distributions?
- many rare species?
- overabundance of dominant species?
- double Zero problem?

Transforming community data



Modified from Legendre & Gallagher (2001)

Examples

Transforming counts into presence - absence

```
library(vegan)
spec.pa ← decostand(spe, method = "pa")
```

Reducing the weight of rare species

```
spec.hel ← decostand(spe, method = "hellinger")
spec.chi ← decostand(spe, method = "chi.square")
```

Reducing the weight of very abundant species

```
spe.pa ← decostand(spe,method = "log")
```

Doubs Environmental Data

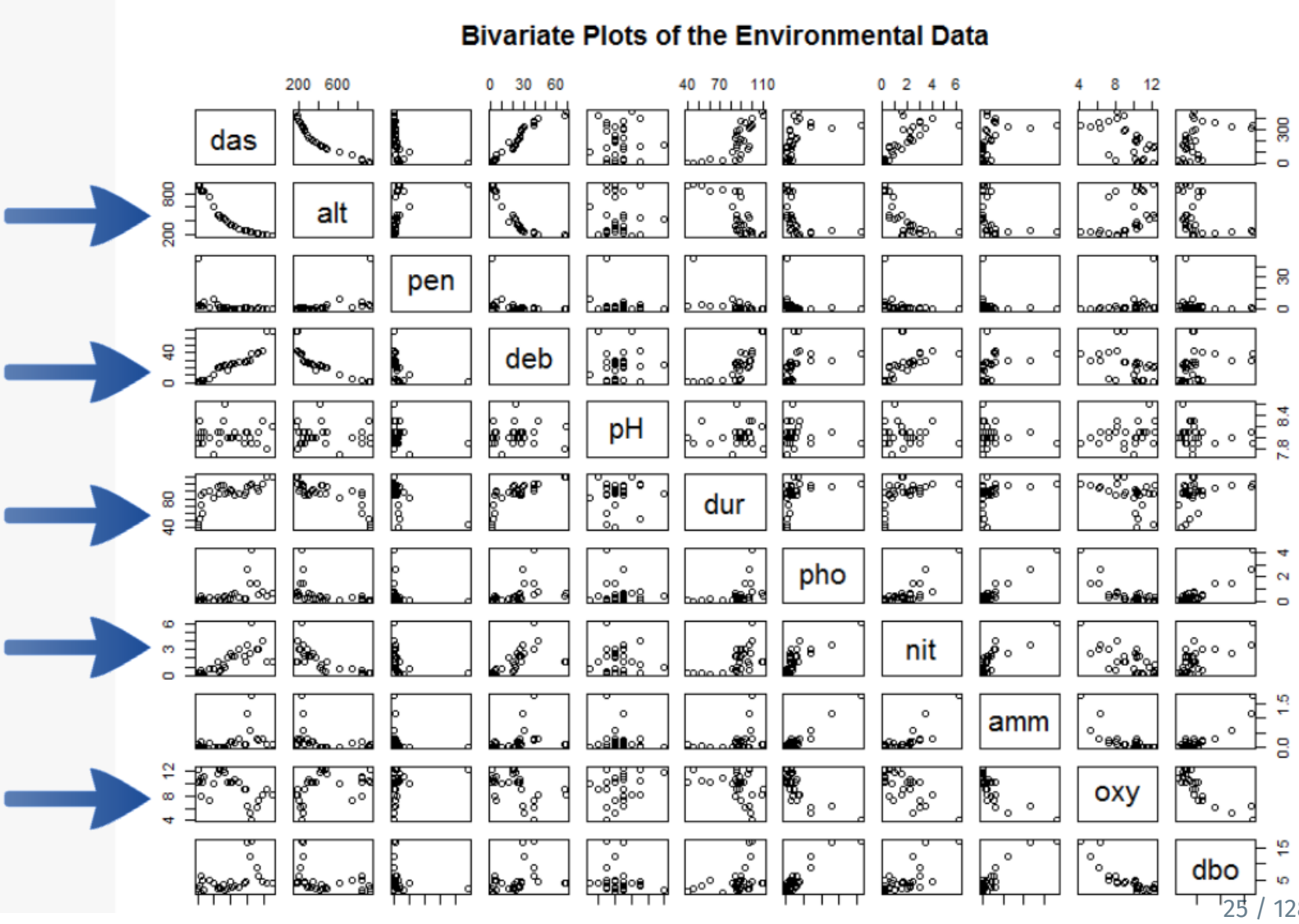
```
names(env) # Names of objects
dim(env) # dimensions
str(env) # structure of objects
summary(env) # summary statistics
head(env) # first 6 rows
```

```
head(env) # first 6 rows
#    das alt  pen  deb  pH dur  pho  nit  amm  oxy dbo
# 1  0.3 934 48.0 0.84 7.9  45 0.01 0.20 0.00 12.2 2.7
# 2  2.2 932  3.0 1.00 8.0  40 0.02 0.20 0.10 10.3 1.9
# 3 10.2 914  3.7 1.80 8.3  52 0.05 0.22 0.05 10.5 3.5
# 4 18.5 854  3.2 2.53 8.0  72 0.10 0.21 0.00 11.0 1.3
# 5 21.5 849  2.3 2.64 8.1  84 0.38 0.52 0.20  8.0 6.2
# 6 32.4 846  3.2 2.86 7.9  60 0.20 0.15 0.00 10.2 5.3
```

Explore colinearity by visualizing correlations between variables

```
pairs(env, main = "Bivariate Plots of the Environmental Data")
```

Doubs Environmental Data



Standardization

Standardizing environmental variables is crucial as you cannot compare the effects of variables with different units

```
## ?decostand
env.z ← decostand(env, method = "standardize")
```

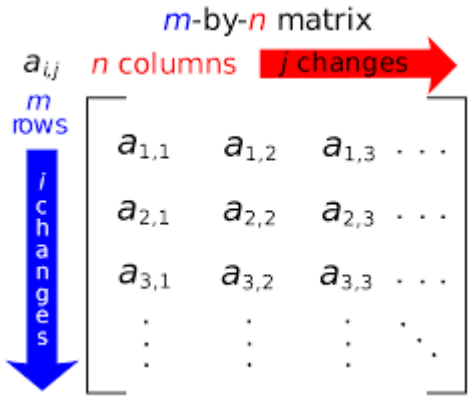
This centers and scales the variables to make your downstream analysis more appropriate

```
apply(env.z, 2, mean)
#           das           alt           pen           deb            pH
# -7.959539e-17 -4.795165e-17  2.494600e-17 -7.323225e-17 -1.730430e-15
#           dur           pho           nit           amm           oxy
# -2.028505e-16  4.445790e-17  2.875893e-17  2.754434e-17 -4.038167e-16
#           dbo
#  9.829975e-17
apply(env.z, 2, sd)
# das alt pen deb  pH dur pho nit amm oxy dbo
#   1   1   1   1   1   1   1   1   1   1   1
```

3. Similarity / Dissimilarity

Association measures

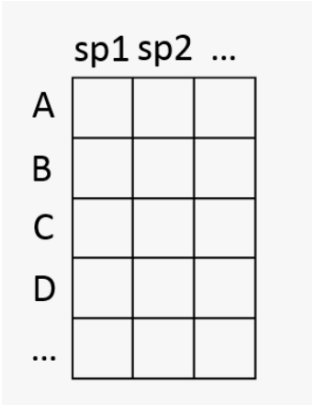
Matrix algebra is at the heart of all ordinations

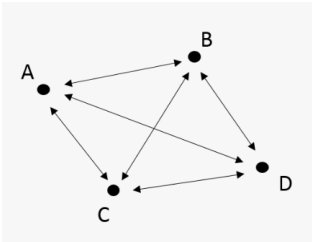


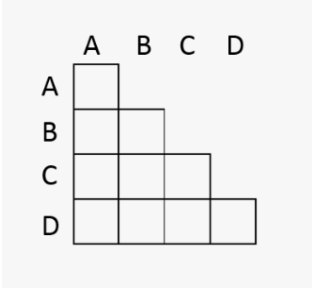
- Exploring various measures of distance between objects provides some understanding of the engine under the hood

Breaking out of 1D

- As you have seen, ecological datasets can sometimes be very large matrices
- Ordinations compute the relationships between species or between sites
- We can simplify these relationships using methods of dissimilarity



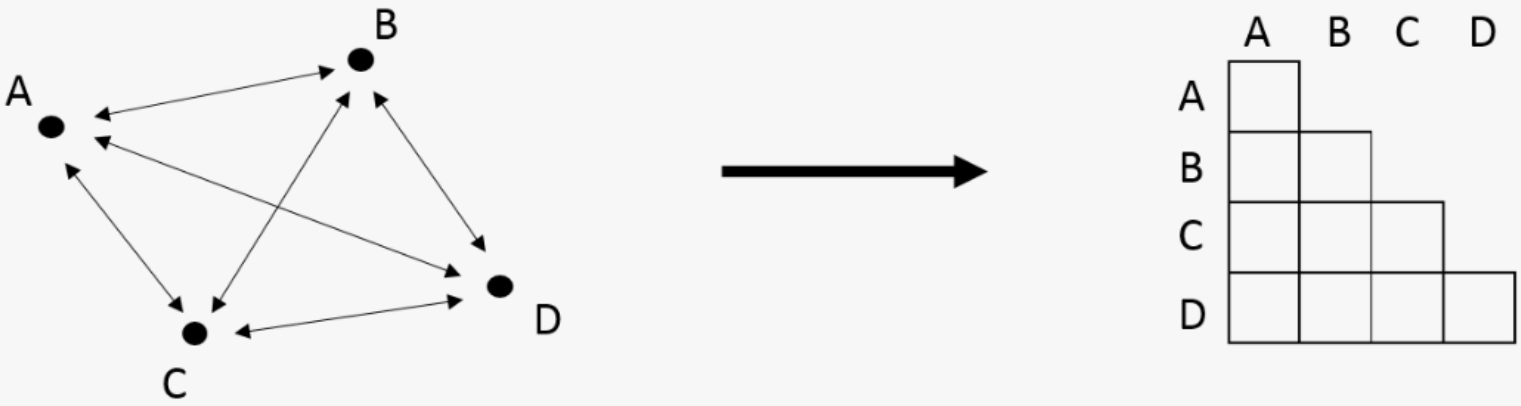




Similarity / Dissimilarity

- Useful to understand your dataset
- Appropriate measure required by some types of ordinations

Similarity: S = 1 - D Distance: D = 1-S



Community distance measures

- Euclidean
- Manhattan
- Chord
- Hellinger
- Chi-square
- Bray-Curtis

Each of these will be useful in different situations

Comparing Doubs Sites

The `vegdist()` function contains all common distances

```
?vegdist
```

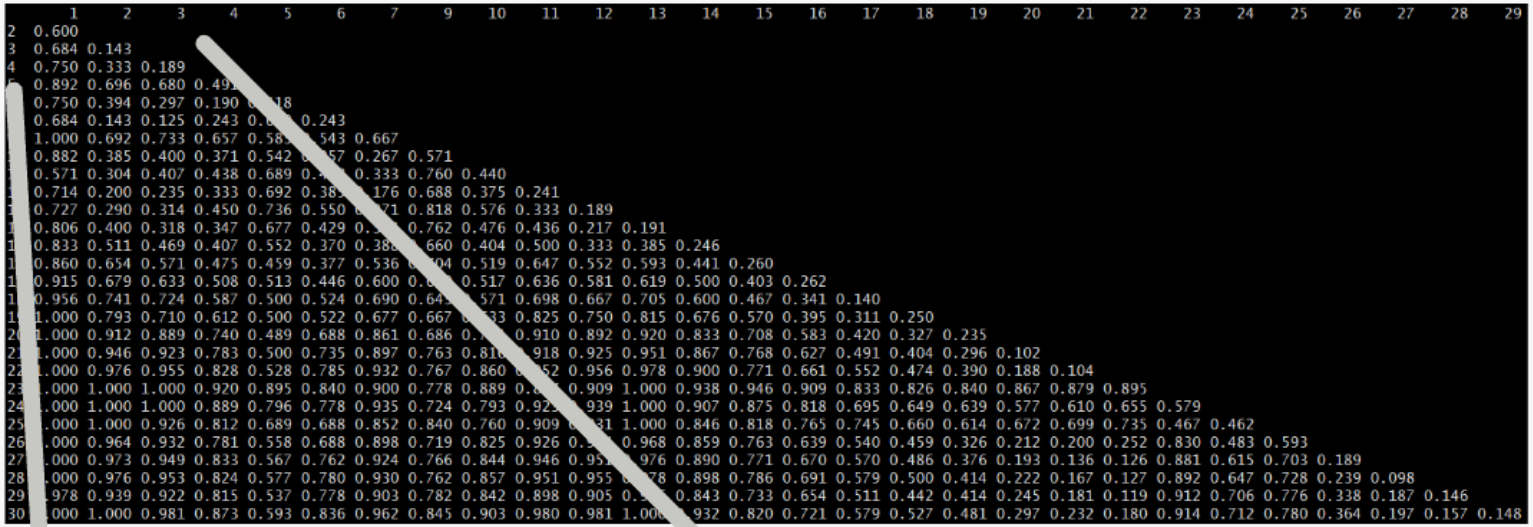
How different is the community composition across the 30 sites of the Doubs River?

```
spe.db.pa ← vegdist(spe, method = "bray")
```

Comparing Doubs Sites

```
      1     2     3     4     5     6     7     9    10    11    12    13    14    15    16    17    18    19    20    21    22    23    24    25    26    27    28    29
2  0.600
3  0.684 0.143
4  0.750 0.333 0.189
5  0.892 0.696 0.680 0.491
6  0.750 0.394 0.297 0.190 0.418
7  0.684 0.143 0.125 0.243 0.640 0.243
9  1.000 0.692 0.733 0.657 0.583 0.543 0.667
10 0.882 0.385 0.400 0.371 0.542 0.257 0.267 0.571
11 0.571 0.304 0.407 0.438 0.689 0.438 0.333 0.760 0.440
12 0.714 0.200 0.235 0.333 0.692 0.385 0.176 0.688 0.375 0.241
13 0.727 0.290 0.314 0.450 0.736 0.550 0.371 0.818 0.576 0.333 0.189
14 0.806 0.400 0.318 0.347 0.677 0.429 0.364 0.762 0.476 0.436 0.217 0.191
15 0.833 0.511 0.469 0.407 0.552 0.370 0.388 0.660 0.404 0.500 0.333 0.385 0.246
16 0.860 0.654 0.571 0.475 0.459 0.377 0.536 0.704 0.519 0.647 0.552 0.593 0.441 0.260
17 0.915 0.679 0.633 0.508 0.513 0.446 0.600 0.690 0.517 0.636 0.581 0.619 0.500 0.403 0.262
18 0.956 0.741 0.724 0.587 0.500 0.524 0.690 0.643 0.571 0.698 0.667 0.705 0.600 0.467 0.341 0.140
19 1.000 0.793 0.710 0.612 0.500 0.522 0.677 0.667 0.633 0.825 0.750 0.815 0.676 0.570 0.395 0.311 0.250
20 1.000 0.912 0.889 0.740 0.489 0.688 0.861 0.686 0.771 0.910 0.892 0.920 0.833 0.708 0.583 0.420 0.327 0.235
21 1.000 0.946 0.923 0.783 0.500 0.735 0.897 0.763 0.816 0.918 0.925 0.951 0.867 0.768 0.627 0.491 0.404 0.296 0.102
22 1.000 0.976 0.955 0.828 0.528 0.785 0.932 0.767 0.860 0.952 0.956 0.978 0.900 0.771 0.661 0.552 0.474 0.390 0.188 0.104
23 1.000 1.000 1.000 0.920 0.895 0.840 0.900 0.778 0.889 0.867 0.909 1.000 0.938 0.946 0.909 0.833 0.826 0.840 0.867 0.879 0.895
24 1.000 1.000 1.000 0.889 0.796 0.778 0.935 0.724 0.793 0.923 0.939 1.000 0.907 0.875 0.818 0.695 0.649 0.639 0.577 0.610 0.655 0.579
25 1.000 1.000 0.926 0.812 0.689 0.688 0.852 0.840 0.760 0.909 0.931 1.000 0.846 0.818 0.765 0.745 0.660 0.614 0.672 0.699 0.735 0.467 0.462
26 1.000 0.964 0.932 0.781 0.558 0.688 0.898 0.719 0.825 0.926 0.934 0.968 0.859 0.763 0.639 0.540 0.459 0.326 0.212 0.200 0.252 0.830 0.483 0.593
27 1.000 0.973 0.949 0.833 0.567 0.762 0.924 0.766 0.844 0.946 0.951 0.976 0.890 0.771 0.670 0.570 0.486 0.376 0.193 0.136 0.126 0.881 0.615 0.703 0.189
28 1.000 0.976 0.953 0.824 0.577 0.780 0.930 0.762 0.857 0.951 0.955 0.978 0.898 0.786 0.691 0.579 0.500 0.414 0.222 0.167 0.127 0.892 0.647 0.728 0.239 0.098
29 0.978 0.939 0.922 0.815 0.537 0.778 0.903 0.782 0.842 0.898 0.905 0.906 0.843 0.733 0.654 0.511 0.442 0.414 0.245 0.181 0.119 0.912 0.706 0.776 0.338 0.187 0.146
30 1.000 1.000 0.981 0.873 0.593 0.836 0.962 0.845 0.903 0.980 0.981 1.000 0.932 0.820 0.721 0.579 0.527 0.481 0.297 0.232 0.180 0.914 0.712 0.780 0.364 0.197 0.157 0.148
```

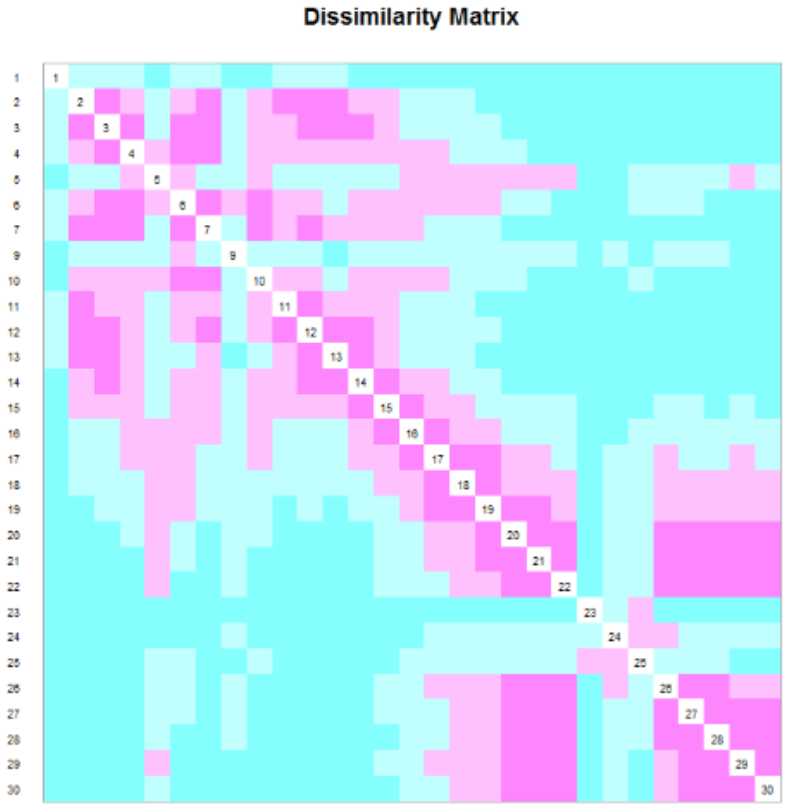
Comparing Doubs Sites

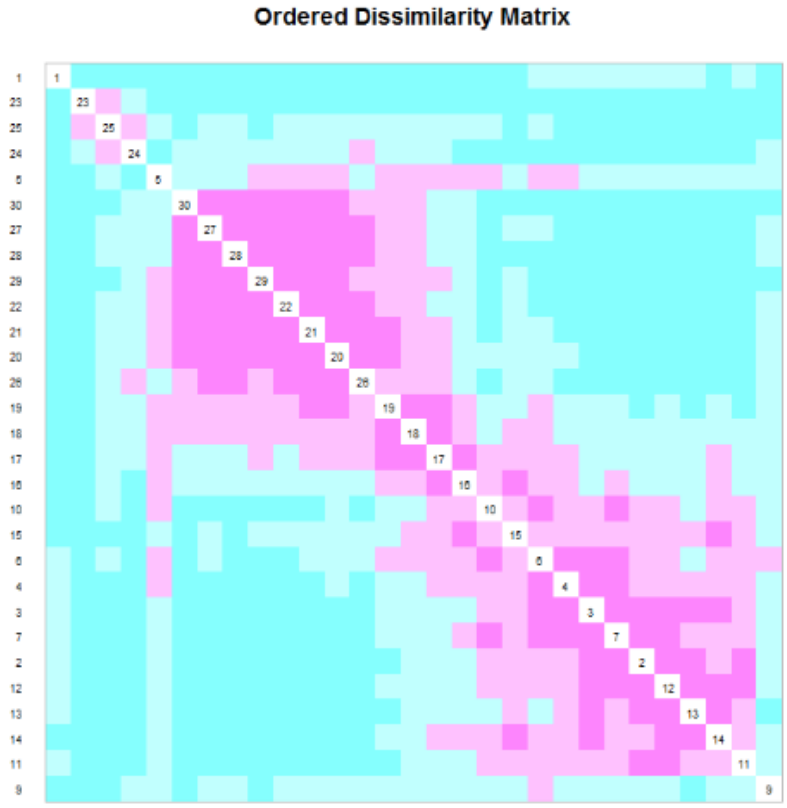


	1	2	3
1	0.000	0.600	0.684
2	0.600	0.000	0.143
3	0.684	0.143	0.000

- Diagonal is zero
- Site 2 and 3 most similar
- Site 1 and 3 most different

Visualization of distance matrices





Challenge #1

Discuss with your neighbor:

How can we tell how similar objects are when we have multivariate data?

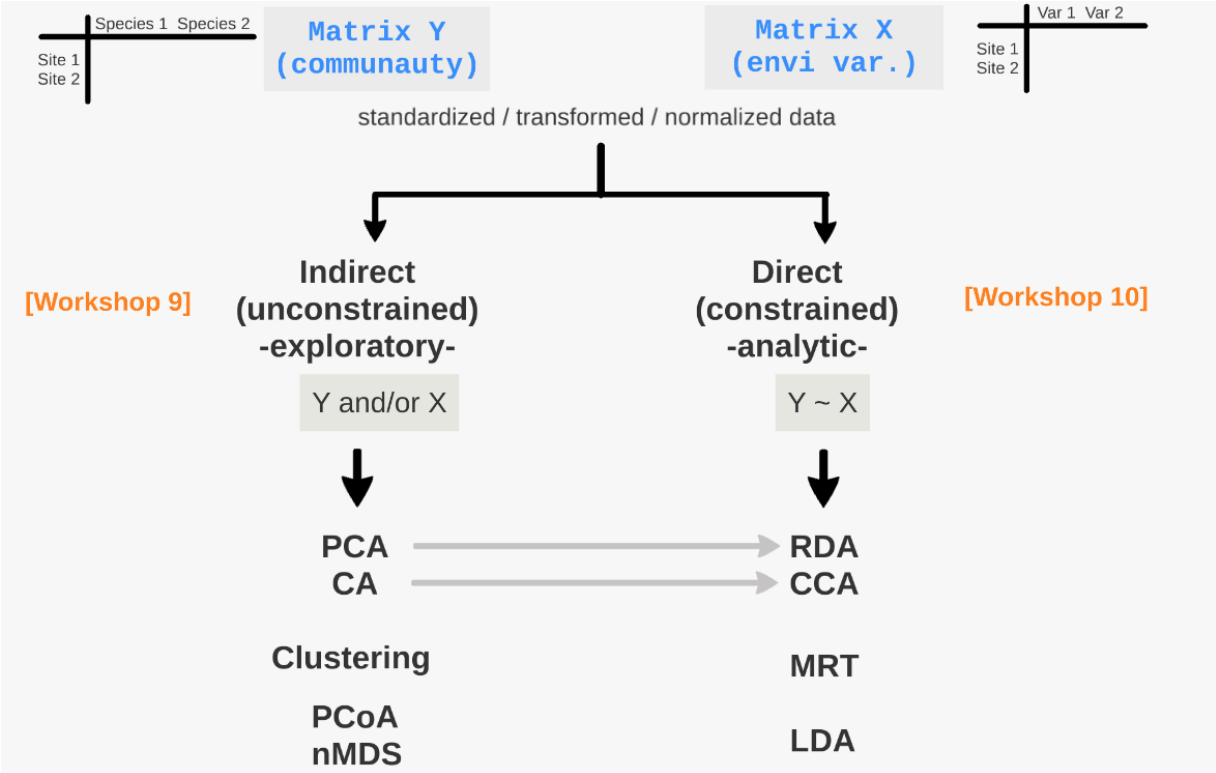
- Make a list of all your suggestions

And what about ordination?

With ordination methods, we order our objects (site) according to their similarity

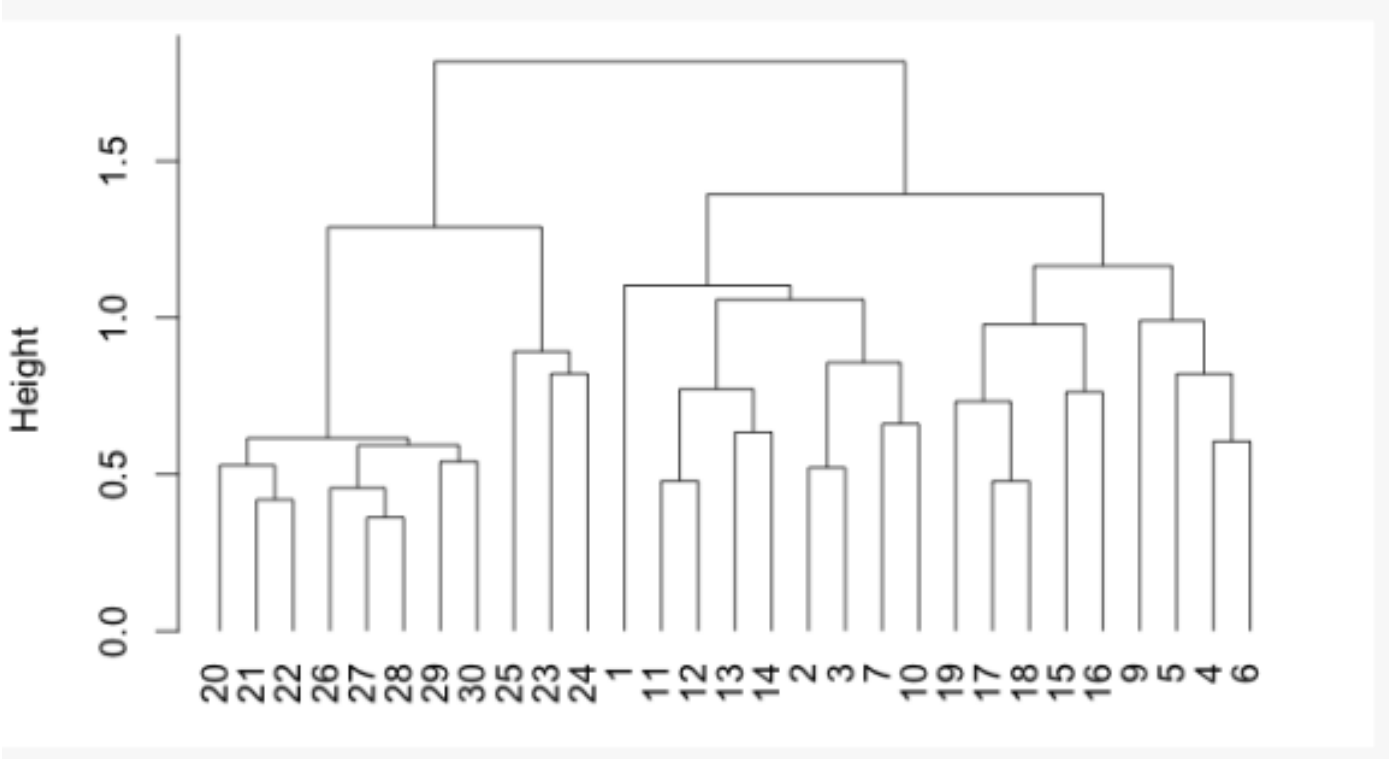
- The more the sites are similar, the closer they are in the ordination space (smaller distances)
- In Ecology, we usually calculate the similarity between sites according to their species composition or their environmental conditions.

Schematic analysis of multivariate analysis



Clustering

- To highlight structures in the data by partitioning either objects or the descriptors
- Results are represented as dendrograms (trees)
- Not a statistical method



Overview of 3 hierarchical methods

- Single linkage agglomerative clustering
- Complete linkage, agglomerative clustering
- Ward's minimum variance clustering
- Elements of lower are nested in higher ranking clusters
  - (e.g. species, genus, family, order)

Hierarchical methods

A distance matrix is first sorted in increasing distance order

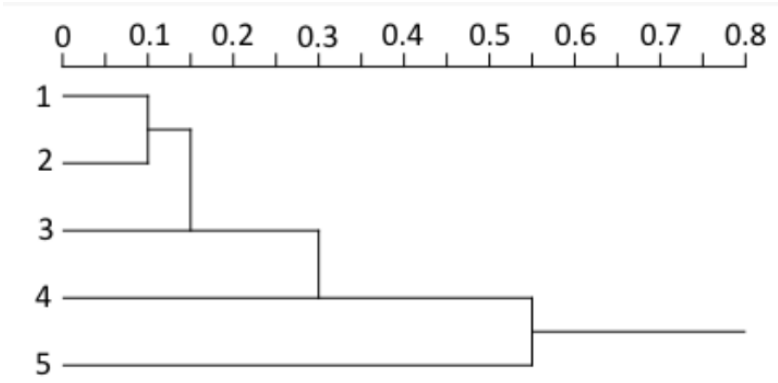
	2	3	4	5
1	0.10	0.15	0.40	0.80
2		0.60	0.35	0.70
3			0.30	0.65
4				0.75

Dist	Pair
0.10	1-2
0.15	1-3
0.30	3-4
0.35	2-4
0.40	1-4
0.60	2-3
0.65	3-5
0.70	2-5
0.75	4-5
0.80	1-5

Single linkage clustering

Dist	Pair
0.10	1-2
0.15	1-3
0.30	3-4
0.35	2-4
0.40	1-4
0.60	2-3
0.65	3-5
0.70	2-5
0.75	4-5
0.80	1-5

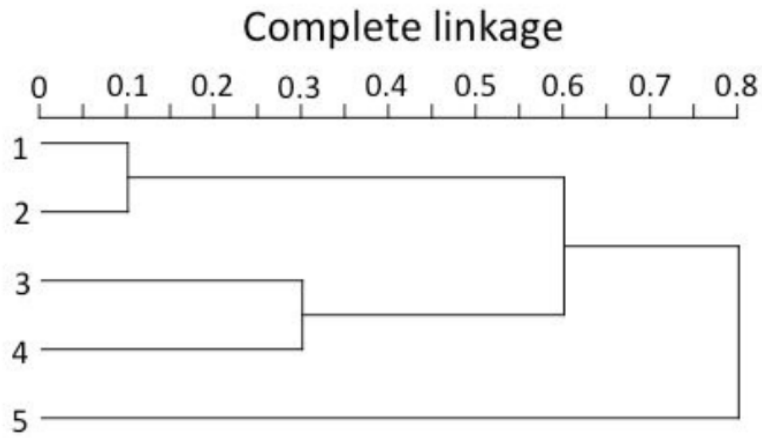
- The two closest objects merge
- The next two closest objects/clusters merge
- and so on



Complete linkage clustering

Dist	Pair
0.10	1-2
0.15	1-3
0.30	3-4
0.35	2-4
0.40	1-4
0.60	2-3
0.65	3-5
0.70	2-5
0.75	4-5
0.80	1-5

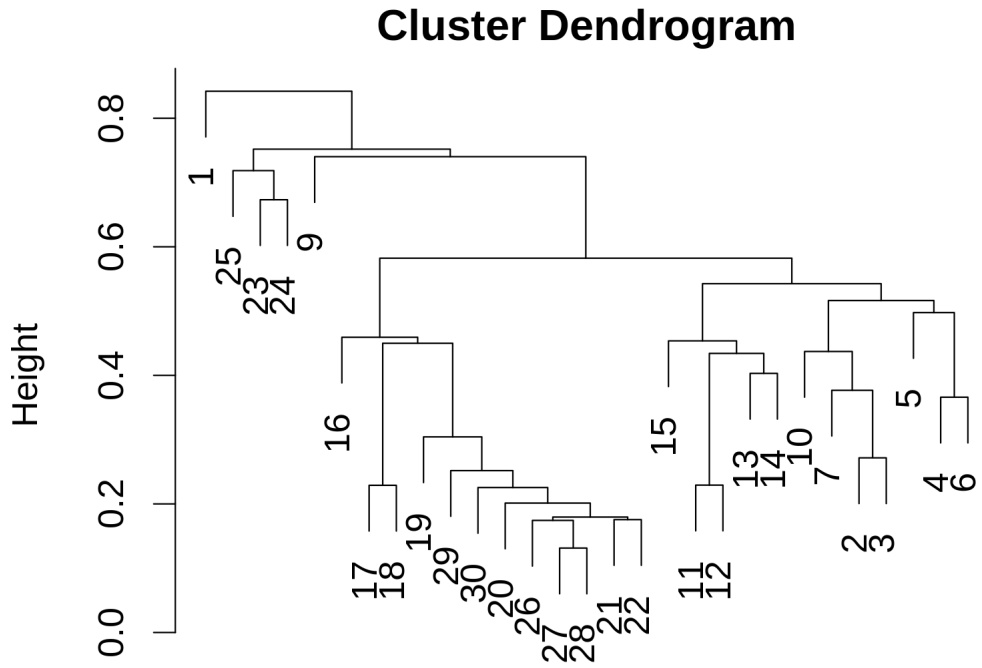
- The two closest objects merge
- The next two objects/cluster will agglomerate when linked to the furthest element of the group



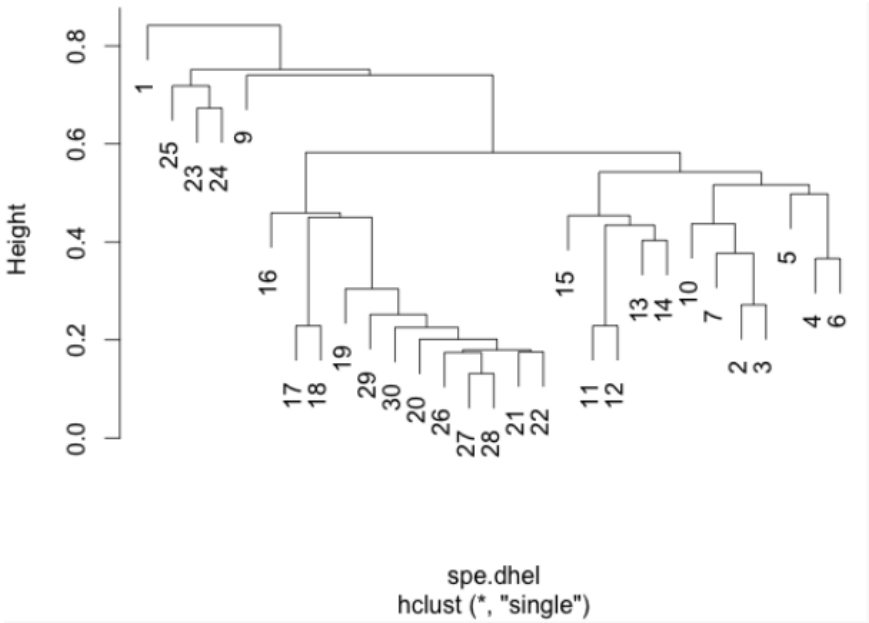
Comparison

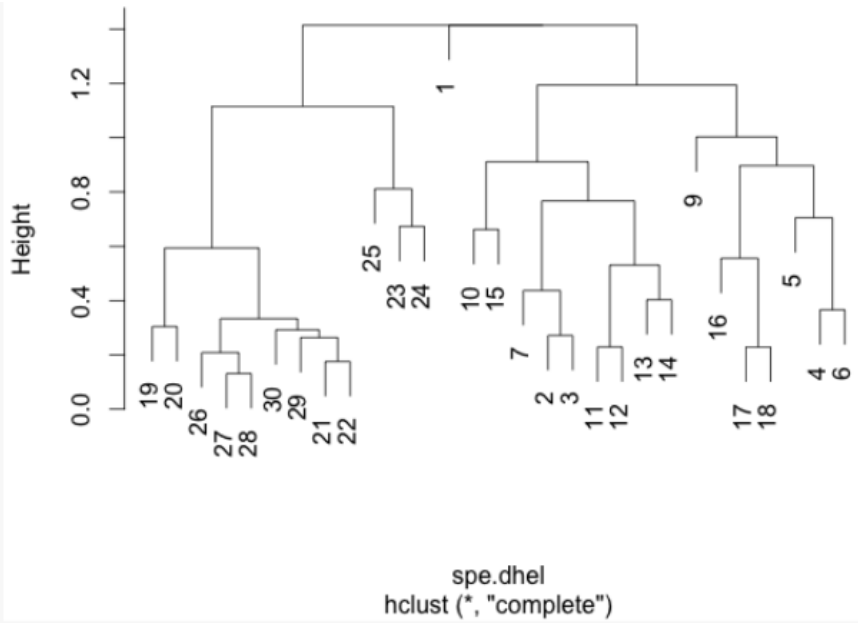
Create a distance matrix from Hellinger transformed Doubs river data and compute the single linkage clustering

```
spe.dhe1 ← vegdist(spec.hel, method = "euclidean")
spe.dhe1.single ← hclust(spe.dhe1, method = "single")
plot(spe.dhe1.single)
```



Comparison





Single linkage:

Chains of objects occur (e.g. 19,29,30,26)

Complete linkage: Contrasted groups are formed of objects occur

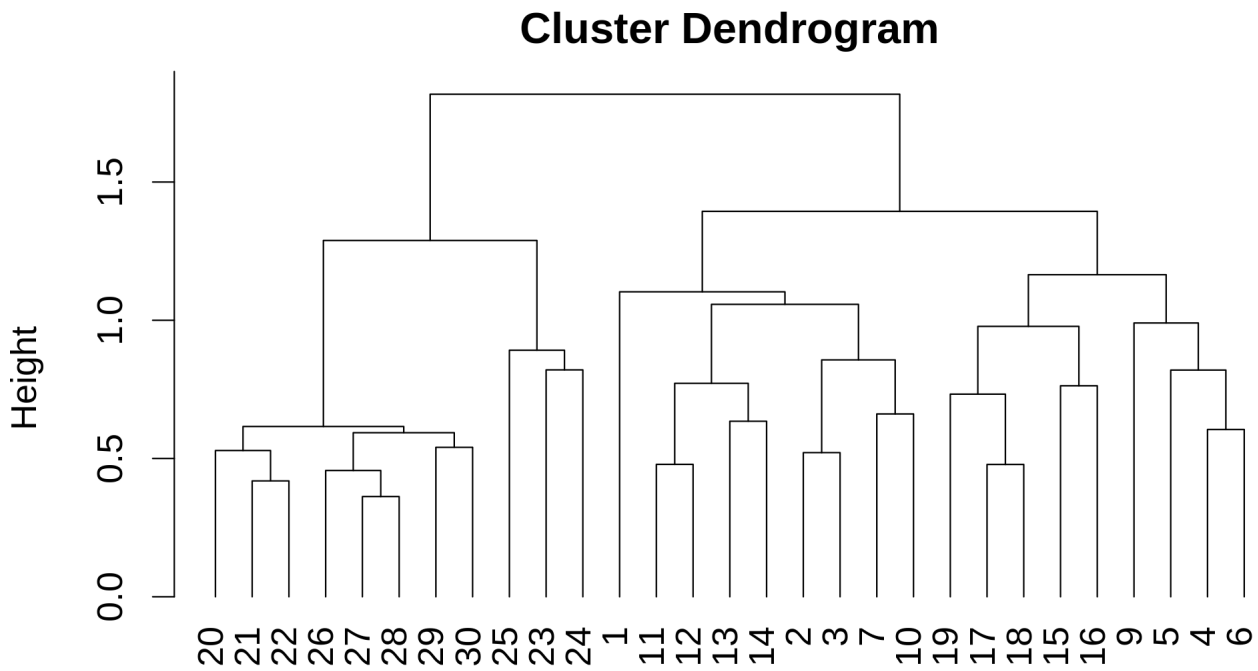
Ward's minimum variance method

- Uses the criterion of least squares to cluster objects into groups
  - At each step, the pair of clusters merging is the one leading to the minimum increase in total within-group sum of squares

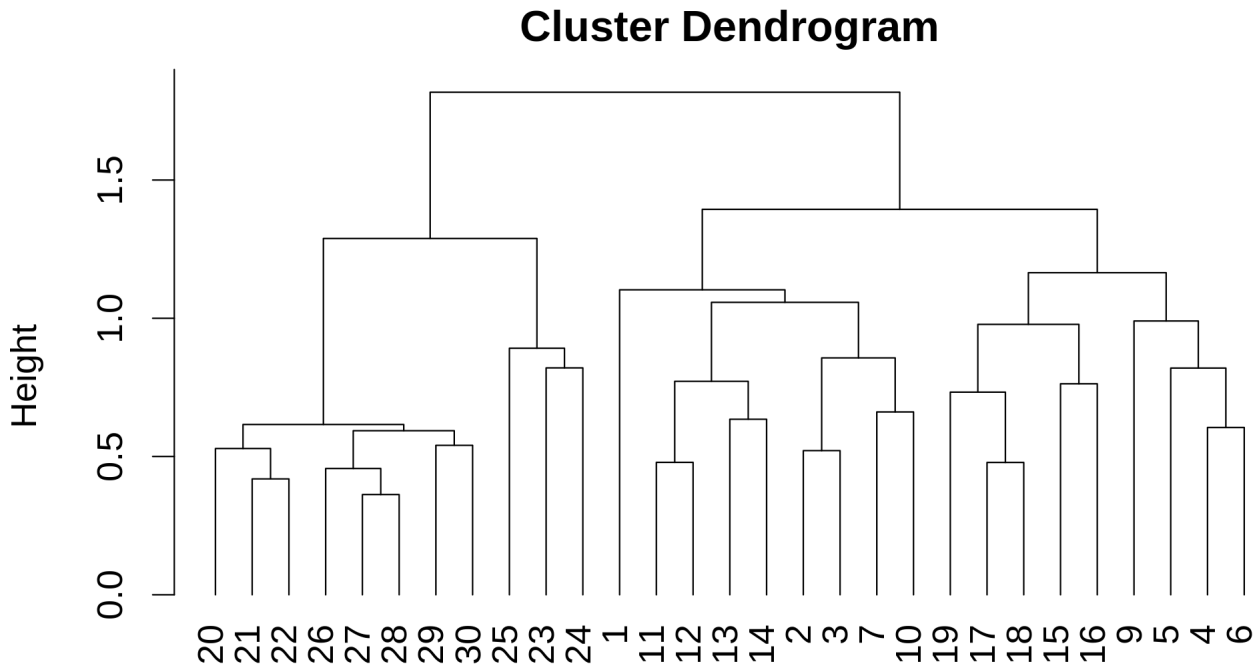
Ward's method

Compute the Ward's minimum variance clustering and plot the dendrogram by using the square root of the distances:

```
spe.dhel.ward ← hclust(spe.dhe1, method = "ward.D2")
spe.dhel.ward$height ← sqrt(spe.dhel.ward$height)
plot(spe.dhel.ward, hang = -1) # hang = -1 aligns objects at the same level
```



Ward's method



Clusters generated using this method tend to be more spherical and to contain similar number of objects

How to choose the right method?

- Depends on the objective
  - highlights gradients? contrasts?
- If more than on method seems appropriate, compare dendrograms
- Again: this is **not** an statistical method But! is possible to:
  - determine the optimal number of interpretable clusters
  - compute clustering statistics
  - combine clustering to ordination to distinguish groups of sites

4. Unconstrained ordination

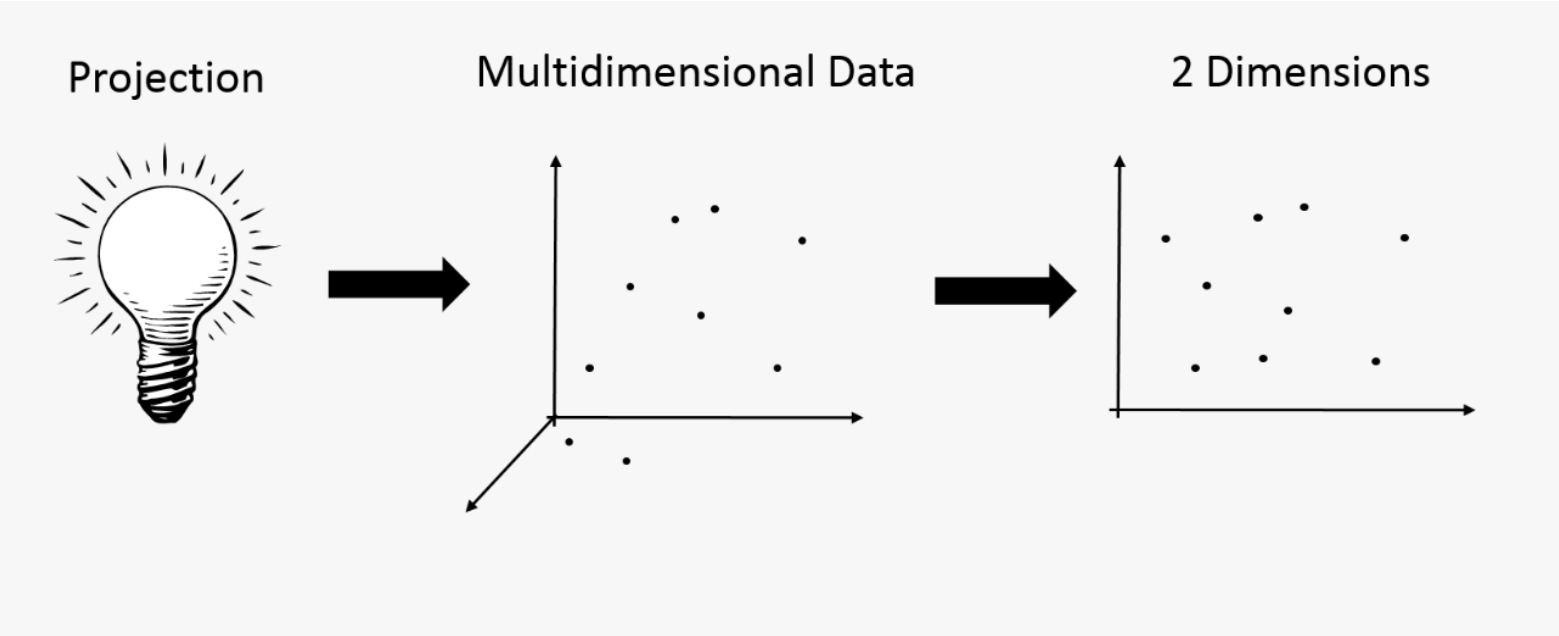
Definitions

- **Variance:** measure of a variable y_j dispersion from its mean
- **Co-variance:** measure of co-dispersion of variables y_j et y_i from their means
- **Correlation:** measure of the link strength between 2 variables: $r_{ij} = (d_{ij}/d_j x d_k)$
- **Eigenvalues:** Proportio of variance (dispersion) represented by one ordination axe.
- **Orthogonality:** right angle between 2 axis or 2 arrows which means that these 2 are independent = non correlated.
- **Score:** position of a dot on an axis. All the scores of a dot give its coordinates in the multidimensional space. They can be used as new variable for other analyses (e.g. linear combination of measured variables).
- **Dispersion** (inertia): Measure of the total variability of the scatter plot (descriptors) in the multidimensional space with regards to its center of gravity.

Unconstrained ordination

- Asses relationships **within** a set of variables (species or environmental variables, not **between** sets, i.e. constrained analysis)
- Find key components of variation between samples, sites, species, etc... ç
- Reduce the number of dimensions in multivariate data without substantial loss of information
- Create new variables for use in subsequent analysis (such as regression)

4.1. Principal Component Analysis (PCA)



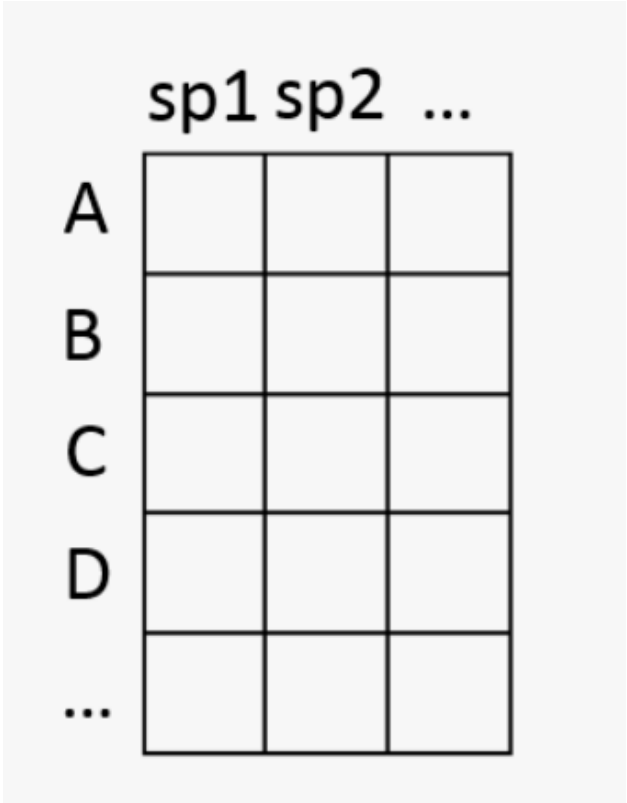
- Preserves, in 2D, the maximum amount of variation in the data
- The resulting, synthetic variables are orthogonal (and therefore uncorrelated)

PCA - What you need

- A set of variables that are response variables (e.g. community composition) OR explanatory variables (e.g. environmental variables)

NOT BOTH!

- Samples that are measured for the same set of variables
- Generally a dataset that is longer than it is wide is preferred

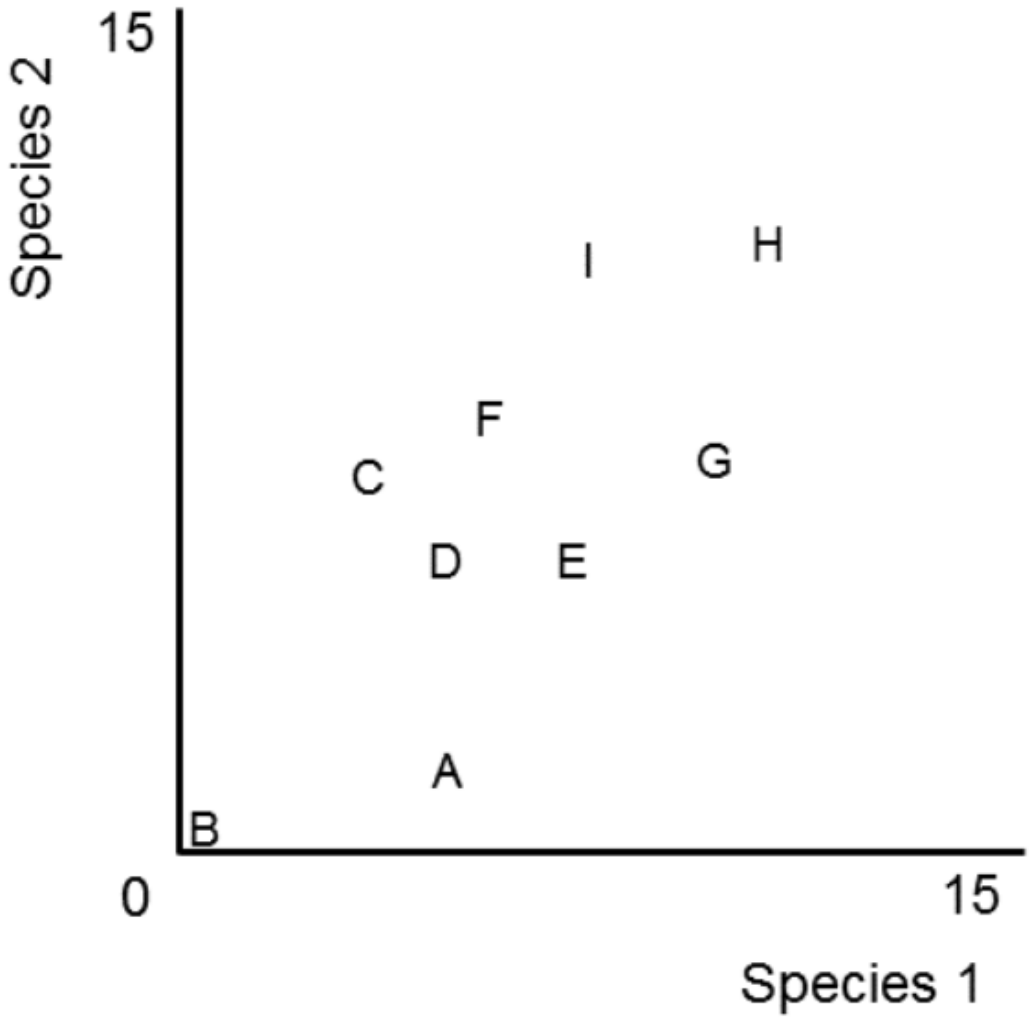


PCA - Walkthrough

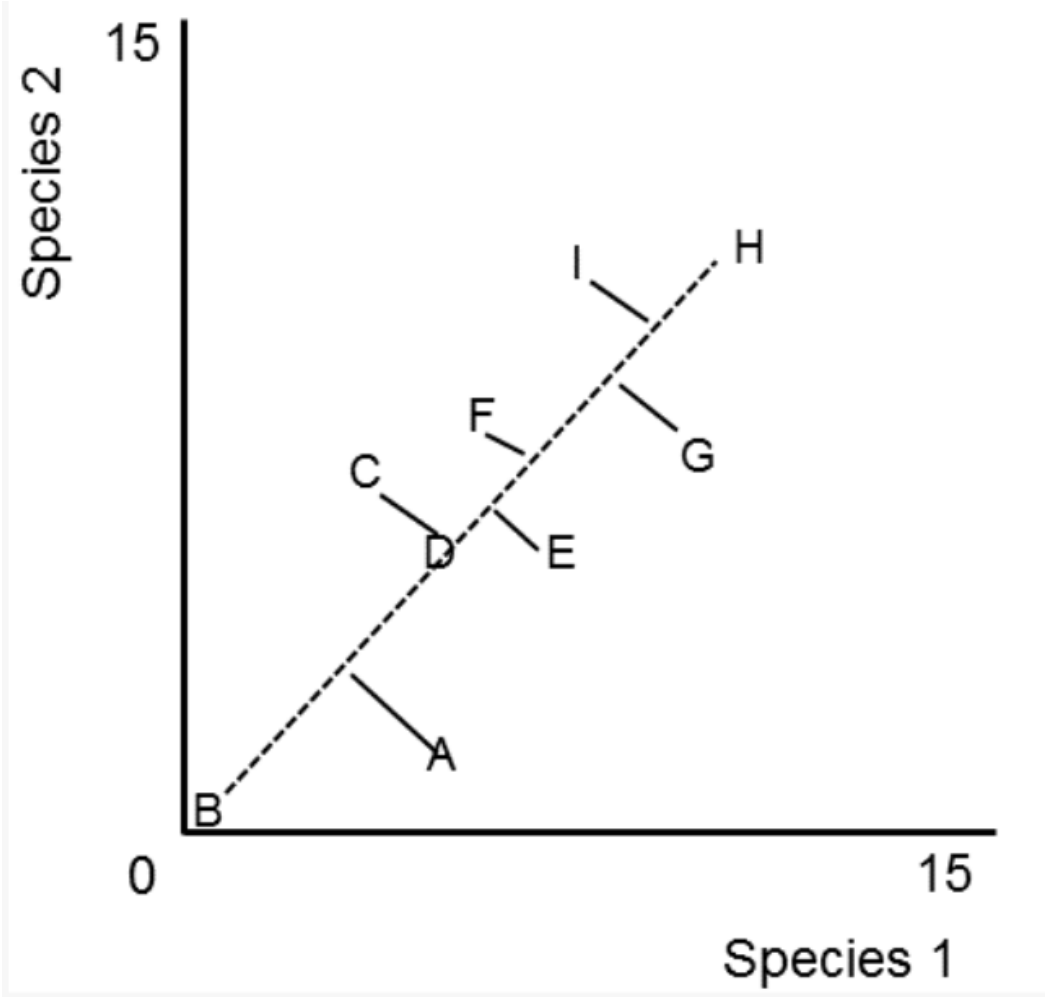
Site	Species 1	Species 2
A	7	3
B	4	3
C	12	10
D	23	11
E	13	13
F	15	16
G	18	14

A simplified example

PCA - Walkthrough

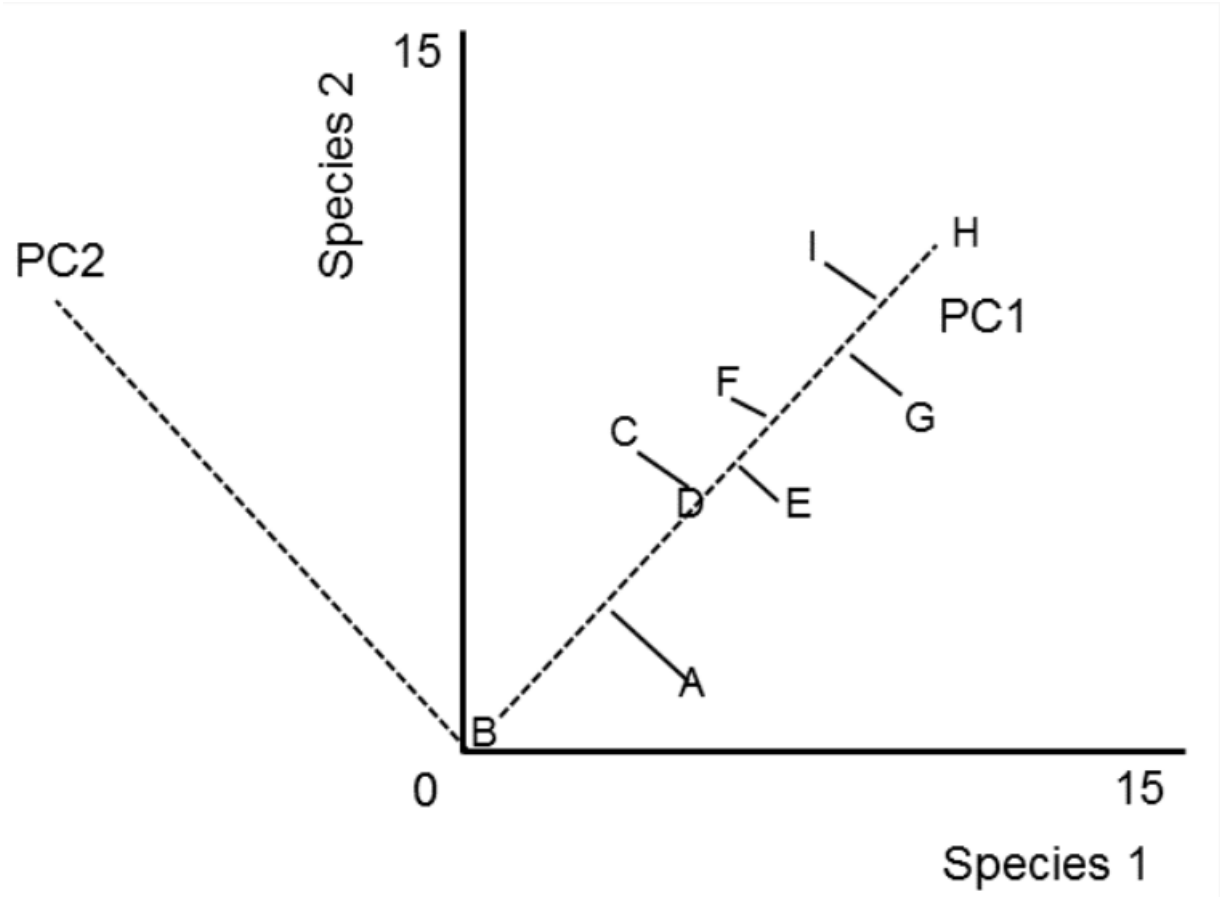


PCA - Walkthrough



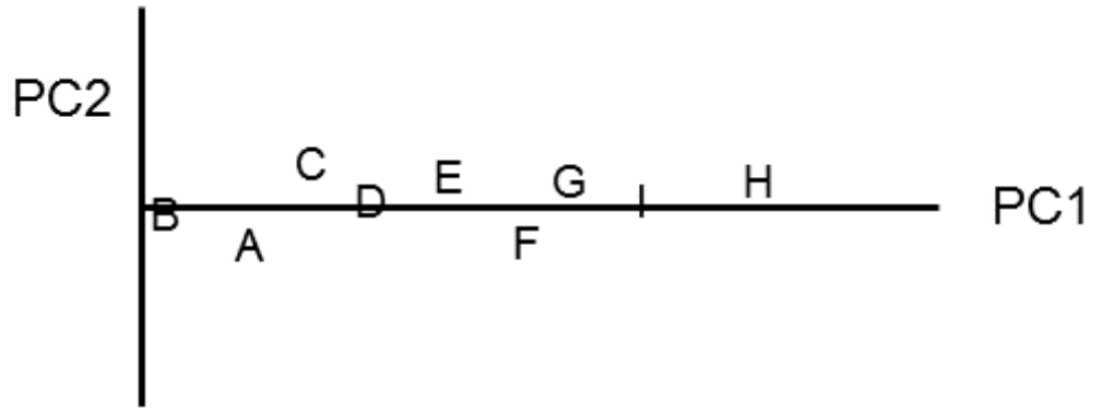
Our first component is essentially drawn trough the maximum amount of observed variation... or the best fit line through the points

PCA - Walkthrough



A second principal component is then added perpendicular (90 degrees in 2D) to the first axis

PCA - Walkthrough



The final plot then is the two PC axes rotated where the axes are now principal components as opposed to species

PCA - Multidimensional case

- **PC1** --> axis that maximizes the variance of the points that are projected perpendicularly onto the axis.
- **PC2** --> must be perpendicular to PC1, but the direction is again the one in which variance is maximized when points are perpendicularly projected
- **PC3** --> and so on: perpendicular to the first two axes

When there are more than two dimensions, PCA produces a new spaces in which all PCA axes are orthogonal (i.e. non-correlated) and where the PCA axes are ordered according to the percent of variance of the original data they explain

PCA - Let's try it on Fish Species!

- For both PCA and RDA, we will be using the `rda()` function in the vegan package
- Run a PCA on the Hellinger-transformed fish data and extract the results

```
spe.h.pca ← rda(spec.hel)

summary(spe.h.pca)
#
# Call:
# rda(X = spec.hel)
#
# Partitioning of variance:
#               Inertia Proportion
# Total          0.5025          1
# Unconstrained  0.5025          1
#
# Eigenvalues, and their contribution to the variance
#
# Importance of components:
#                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
# Eigenvalue            0.2580 0.06424 0.04632 0.03850 0.02197 0.01675 0.01472
# Proportion Explained  0.5133 0.12784 0.09218 0.07662 0.04371 0.03334 0.02930
# Cumulative Proportion 0.5133 0.64118 0.73337 0.80999 0.85370 0.88704 0.91634
```

Function `rda()`

- RDA is in 2 steps
  - multiple regressions
  - PCA on regressed values
- If we give only one table to the function `rda()` it does directly a PCA without doing regression

rda(Y~X) → RDA

rda(Y) or rda(X) → PCA

PCA - Interpretation of Output

```
Partitioning of variance:
              Inertia Proportion
Total          0.5025          1
Unconstrained  0.5025          1
```

- Total variance explained by the descriptors (here the fish species)
- In PCA, not that the "Total" and "Unconstrained" portion of the explained variance is identical

PCA - Interpretation of Output

	PC1	PC2	PC3	PC4	PC5
Eigenvalue	0.2579605	0.06424089	0.04632294	0.03850244	0.02196526
Proportion Explained	0.5133400	0.12784000	0.09218000	0.07662000	0.04371000
Cumulative Proportion	0.5133400	0.64118000	0.73337000	0.80999000	0.85370000

- List the eigenvalues associated to each Principal Component (in this output there are 27 PCs, as this is the number of dimensions in the data)

An eigenvalue is the value of the change in the length of a vector, and for our purposes is the amount of variation explained by Principal Component

0.258 + 0.064 + ... = 0.5025 Total explained variance

PCA - Interpretation of Output

Partitioning of variance:

	Inertia	Proportion
Total	0.5025	1
Unconstrained	0.5025	1

	PC1	PC2	PC3	PC4	PC5
Eigenvalue	0.2579605	0.06424089	0.04632294	0.03850244	0.02196526
Proportion Explained	0.5133400	0.12784000	0.09218000	0.07662000	0.04371000
Cumulative Proportion	0.5133400	0.64118000	0.73337000	0.80999000	0.85370000

- List of the proportion of variance explained by each Principal Component (as well as cumulative)

51.3% of 0.5025 is 0.258

PCA - Interpretation of Output

```
Scaling 2 for species and site scores
* Species are scaled proportional to eigenvalues
* Sites are unscaled: weighted dispersion equal on all dimensions
* General scaling constant of scores:  1.93676
```

- There are two ways to represent an ordination in 2D, here the output is informing us that it used the default scaling, which is type 2...

More on this later!

PCA - Interpretation of Output

Species scores

	PC1	PC2	PC3	PC4	PC5	PC6
CHA	0.17336	0.08295	-0.064963	0.2539861	-0.0285801	0.019057
TRU	0.64860	0.01162	-0.261994	-0.1606020	-0.0745819	-0.088616
VAI	0.51810	0.14773	0.165304	0.0241017	0.1012928	0.104748
LOC	0.38606	0.16615	0.242995	-0.0275216	0.1258011	0.048299
OMB	0.16893	0.06274	-0.096143	0.2426514	0.0140574	0.062117

- Species refers to your descriptors (i.e. the columns in your dataset), which here are the Fish species
- Scores refer to the position of every species in the PC. Essentially they are the coordinates of each species along the principal component

PCA - Interpretation of Output

```
Site scores (weighted sums of species scores)

         PC1      PC2      PC3      PC4      PC5      PC6
1   0.367401 -0.39935 -1.08857 -0.63304 -0.512027 -0.858378
2   0.503582 -0.05683 -0.19259 -0.43441  0.389533  0.069451
3   0.461709  0.02262 -0.06522 -0.49798  0.309425  0.270577
4   0.298336  0.15130  0.26748 -0.53196  0.003088  0.184821
5  -0.002222  0.07631  0.54769 -0.50936 -0.780261 -0.169353
6   0.212816  0.08345  0.55091 -0.42210 -0.139518 -0.104278
```

- Site refers to the rows in your dataset, which here are the different sites along the Doubs river (but it can be points in time, etc)

Accessing Parts of the Output

The output is very dense, but you can access specific information if needed. For example, you can access the eigenvalues associated contribution to variance :

```
summary(spe.h.pca, display = NULL)
#
# Call:
# rda(X = spec.hel)
#
# Partitioning of variance:
#               Inertia Proportion
# Total          0.5025          1
# Unconstrained  0.5025          1
#
# Eigenvalues, and their contribution to the variance
#
# Importance of components:
#                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
# Eigenvalue            0.2580 0.06424 0.04632 0.03850 0.02197 0.01675 0.01472
# Proportion Explained  0.5133 0.12784 0.09218 0.07662 0.04371 0.03334 0.02930
# Cumulative Proportion 0.5133 0.64118 0.73337 0.80999 0.85370 0.88704 0.91634
#                            PC8      PC9     PC10     PC11     PC12     PC13
# Eigenvalue            0.01156 0.006936 0.006019 0.004412 0.002982 0.002713
# Proportion Explained  0.02300 0.013803 0.011978 0.008781 0.005935 0.005399
# Cumulative Proportion 0.93934 0.953144 0.965123 0.973903 0.979838 0.985237
```

Accessing Parts of the Output

You can calculate the eigenvalues from scratch

```
eigen(cov(spec.hel))
# eigen() decomposition
# $values
#  [1] 2.579605e-01 6.424089e-02 4.632294e-02 3.850244e-02 2.196526e-02
#  [6] 1.675463e-02 1.472430e-02 1.155759e-02 6.936149e-03 6.019271e-03
# [11] 4.412388e-03 2.982309e-03 2.713021e-03 1.834874e-03 1.454670e-03
# [16] 1.117858e-03 8.308832e-04 5.415301e-04 4.755244e-04 3.680458e-04
# [21] 2.765106e-04 2.252760e-04 1.429425e-04 7.618319e-05 4.989831e-05
# [26] 1.525627e-05 9.117507e-06
#
# $vectors
#              [,1]        [,2]        [,3]          [,4]         [,5]
#  [1,] -0.12492725 -0.11979088 -0.11047444  0.4737644443  0.070581708
#  [2,] -0.46740781 -0.01678206 -0.44554311 -0.2995735541  0.184188349
#  [3,] -0.37336215 -0.21333150  0.28111355  0.0449572376 -0.250153773
#  [4,] -0.27821421 -0.23994030  0.41323337 -0.0513364598 -0.310679865
#  [5,] -0.12173642 -0.09059800 -0.16349912  0.4526216196 -0.034716207
#  [6,] -0.05610722 -0.21147318 -0.05340233  0.4363710457  0.254947303
#  [7,]  0.13325245 -0.07077305 -0.07670861  0.0371901204 -0.169737613
#  [8,]  0.10553143 -0.25754282 -0.01860002  0.1212372044  0.001538404
#  [9,]  0.08240964 -0.22633305  0.24186207 -0.0237391101  0.491305413
# [10,]  0.06977391  0.22309480  0.41314626  0.2097008258 -0.057746857
```

Accessing Parts of the Output

You may wish to extract the scores (either from species or sites) for use in subsequent analysis or for plotting

- Access the species scores along the 1st and 2nd PC:

```
spe.scores ← scores(spe.h.pca,
                    display = "species",
                    choices = c(1,2))
```

- Access the site scores along the 1st and 2nd PC:

```
site.scores ← scores(spe.h.pca,
                     display = "sites",
                     choices = c(1,2))
```

Selecting Significant PCs

- The strength of PCA is that we can condense the variance contained in a huge dataset into a set of synthetic variables that is manageable
- In our case, there are still 27 Principal Components, but only the first few account for any significant amount of variance, while the rest can simply be discarded as noise...

How do we manage this?

Kaiser - Guttman criterion

Selects principal components which capture more variance than the average of all PCs

- Extract the eigenvalues associated to the PCs

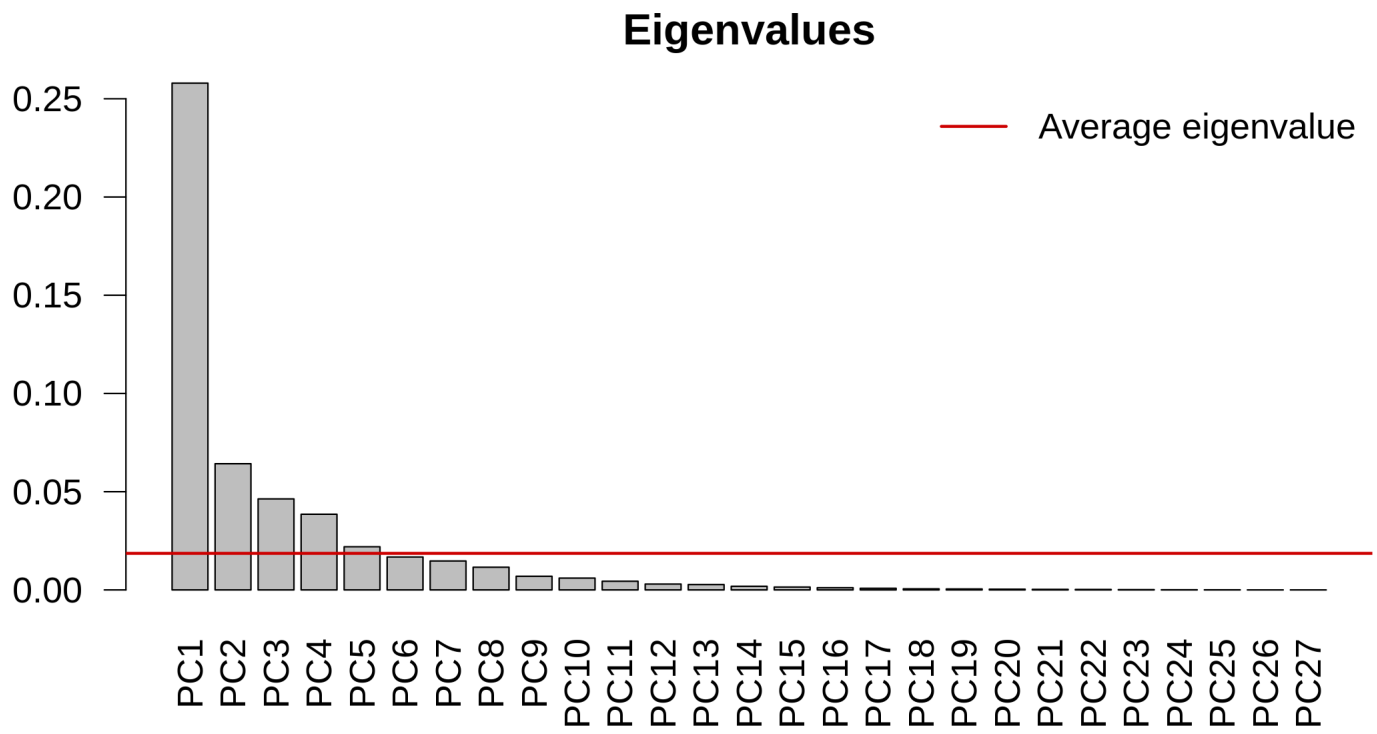
```
ev ← spe.h.pca$CA$eig
```

- Select all eigenvalues above average

```
ev[ev>mean(ev)]
#        PC1        PC2        PC3        PC4        PC5
# 0.25796049 0.06424089 0.04632294 0.03850244 0.02196526
```

Kaiser - Guttman criterion (visualization)

```
n ← length(ev)
barplot(ev, main = "Eigenvalues", col = "grey", las = 2)
abline(h = mean(ev), col = "red3", lwd = 2)
legend("topright", "Average eigenvalue",
       lwd = 2, col = "red3" , bty = "n")
```



PCA - environmental variables

We can also run PCAs on standardized environmental variables, to compare sites for example, or how variables are correlated...

- Run a PCA on the standardized environmental variables and extract the results

```
env.pca ← rda(env.z)
summary(env.pca, scaling  = 2) # default
#
# Call:
# rda(X = env.z)
#
# Partitioning of variance:
#               Inertia Proportion
# Total              11          1
# Unconstrained      11          1
#
# Eigenvalues, and their contribution to the variance
#
# Importance of components:
#                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
# Eigenvalue            6.0980 2.1671 1.03760 0.70351 0.35185 0.31913 0.16455
# Proportion Explained  0.5544 0.1970 0.09433 0.06396 0.03199 0.02901 0.01496
# Cumulative Proportion 0.5544 0.7514 0.84570 0.90966 0.94164 0.97066 0.98561
```

PCA - environmental variables

- Extract the eigenvalues associated to the PCs:

```
ev ← env.pca$CA$eig
```

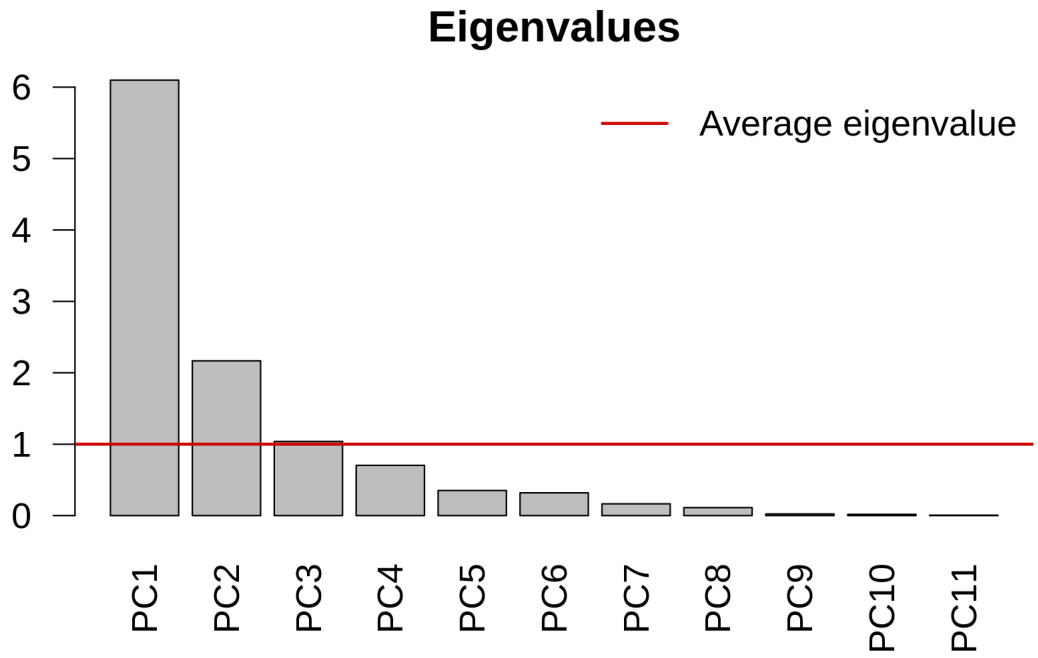
- Select all the eigenvalues above average

```
ev[ev>mean(ev)]
#      PC1      PC2      PC3
# 6.097995 2.167126 1.037603
```

PCA - environmental variables

- Plot the eigenvalues above average

```
n ← length(ev)
barplot(ev, main = "Eigenvalues", col = "grey", las = 2)
abline(h = mean(ev), col = "red3", lwd = 2)
legend("topright", "Average eigenvalue",
       lwd = 2, col = "red3" , bty = "n")
```

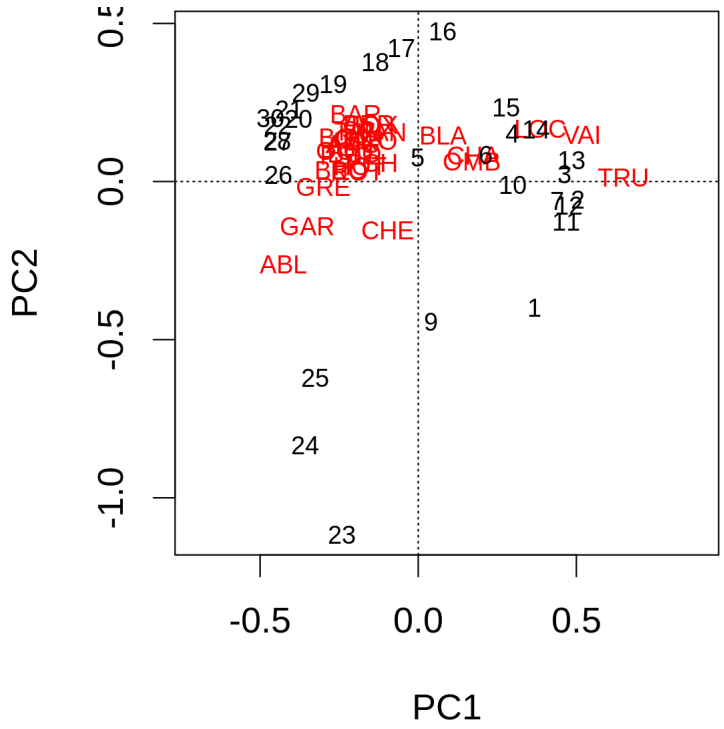


PCA - Visualization

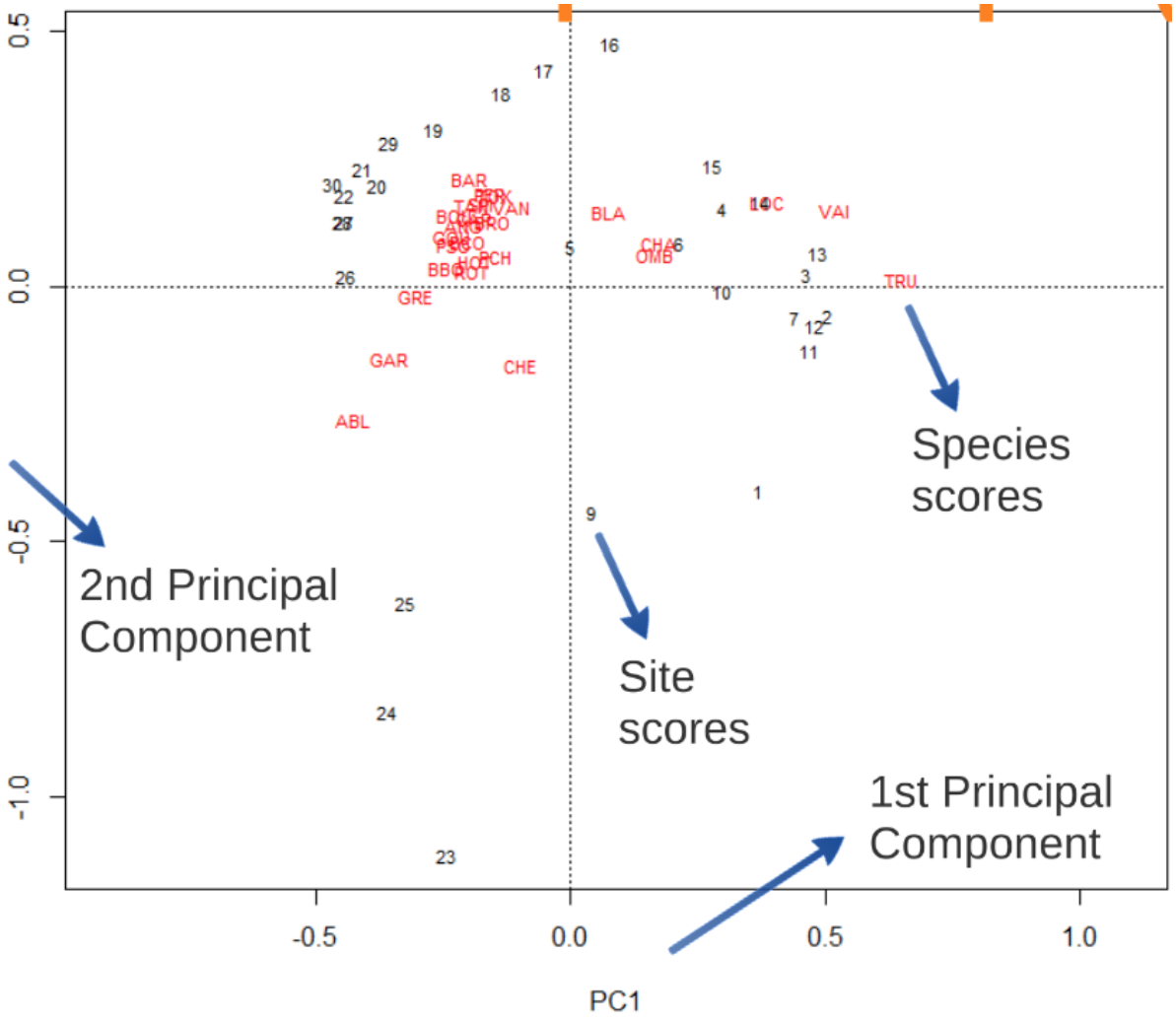
The abundance of information produced by PCA is easier to understand and interpret using biplots to visualize patterns

- We can produce a quick biplot of the PCA using the function `plot()` in base R

```
plot(spe.h.pca)
```



PCA basic biplot with plot()

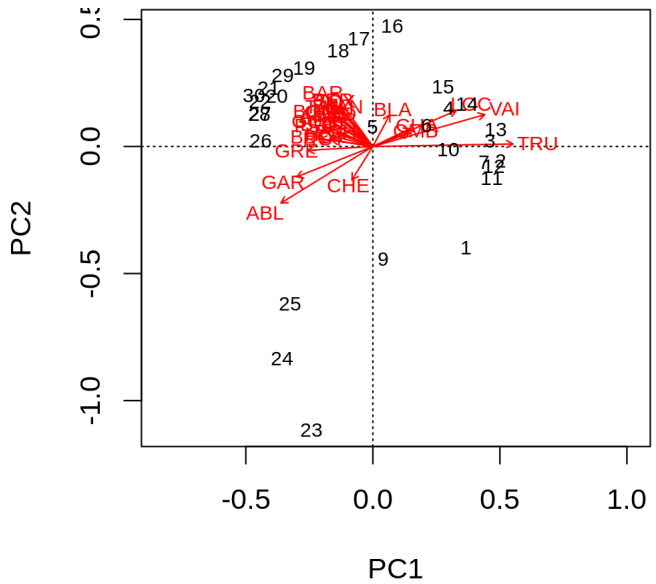


`plot()` *is quick but its hard to interpret the angles between species*

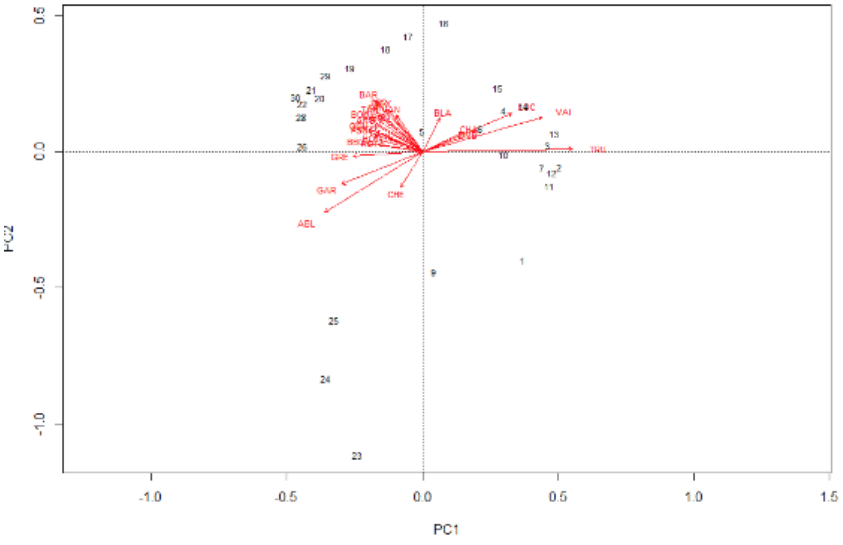
PCA basic `biplot()`

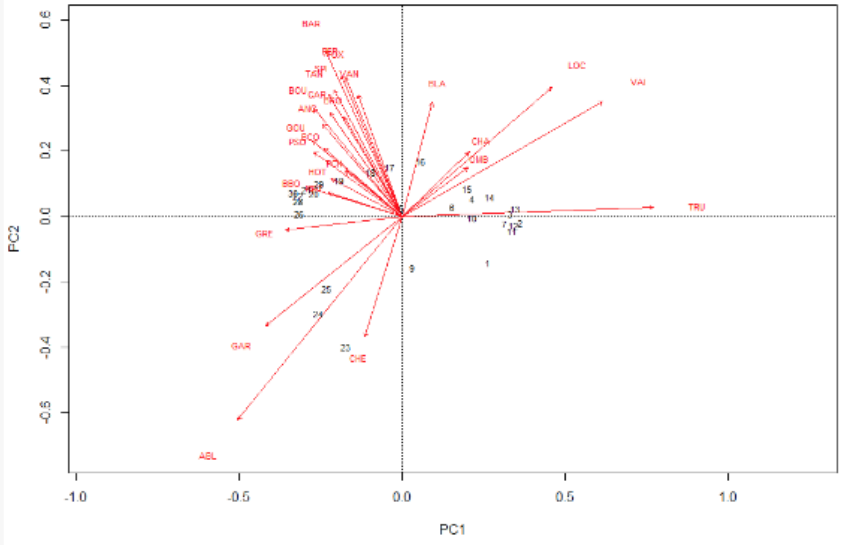
- Using the function `biplot()` from base R, arrows are plotted to show the directionality and angle of the descriptors in the ordination
- **Descriptors at 180 degrees of each other are negatively correlated**
- **Descriptors at 90 degrees of each other have zero correlation**
- **Descriptors at 0 degrees of each other are positively correlated**

```
biplot(spe.h.pca)
```



PCA scaling types





Type 2 scaling (DEFAULT): distances among objects are not approximations of Euclidean distances; angles between descriptor (species) vectors reflect their correlations.

Best for interpreting relationships among descriptors (species)!

Type 1 scaling: attempts to preserve the Euclidean distance (in multidimensional space) among objects (sites): the angles among descriptor (species) vector are meaningless.

Best for interpreting relationships among objects (sites)!

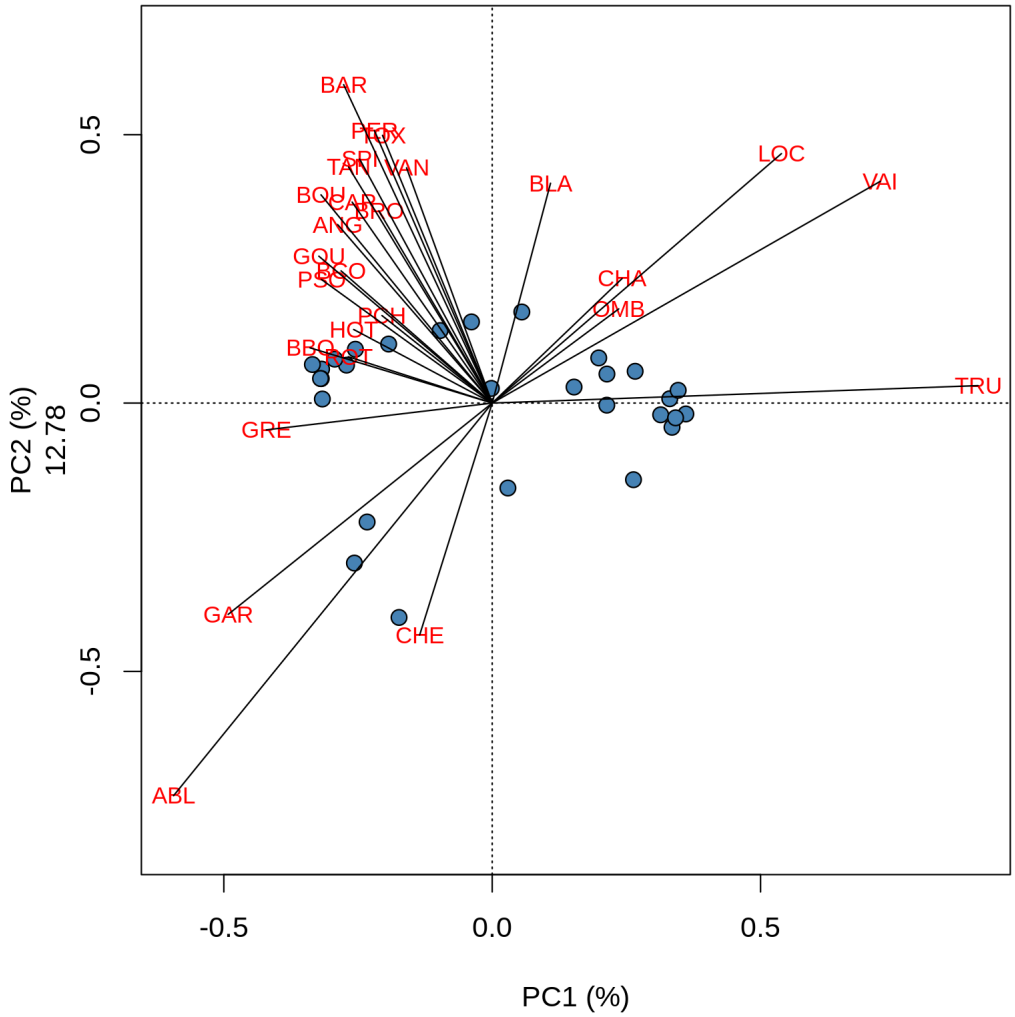
Advanced "biplotting"

- By extracting specific parts of the PCA output, we can build more detailed and aesthetic plots:

```
plot(spe.h.pca, scaling  = 1, type = "none",
     xlab = c("PC1 (%)", round(spe.h.pca$CA$eig[1]/sum(spe.h.pca$CAeig)*100,2)),
     ylab = c("PC2 (%)", round(spe.h.pca$CA$eig[2]/sum(spe.h.pca$CA$eig)*100,2)))
points(scores(spe.h.pca, display = "sites", choices = c(1,2), scaling = 1),
       pch=21, col = "black", bg = "steelblue" , cex  = 1.2)
text(scores(spe.h.pca, display = "species", choices = 1, scaling = 1),
     scores(spe.h.pca, display = "species", choices = 2, scaling = 1),
     labels = rownames(scores(spe.h.pca, display = "species", scaling = 1)),
     col = "red", cex = 0.8)
spe.cs ← scores(spe.h.pca, choices = 1:2, scaling = 1 , display = "sp")
arrows(0, 0, spe.cs[,1], spe.cs[,2], length = 0)
```

use the `arrows()` function in `graphics` to add vectors

Advanced "biplotting"



EASTER EGG: ggvegan

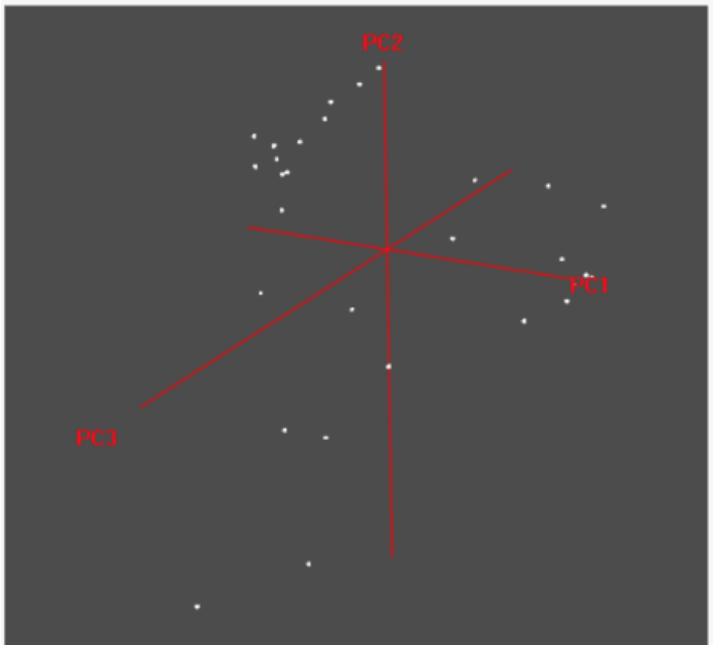
- A set of tools for producing biplots using ggplot2

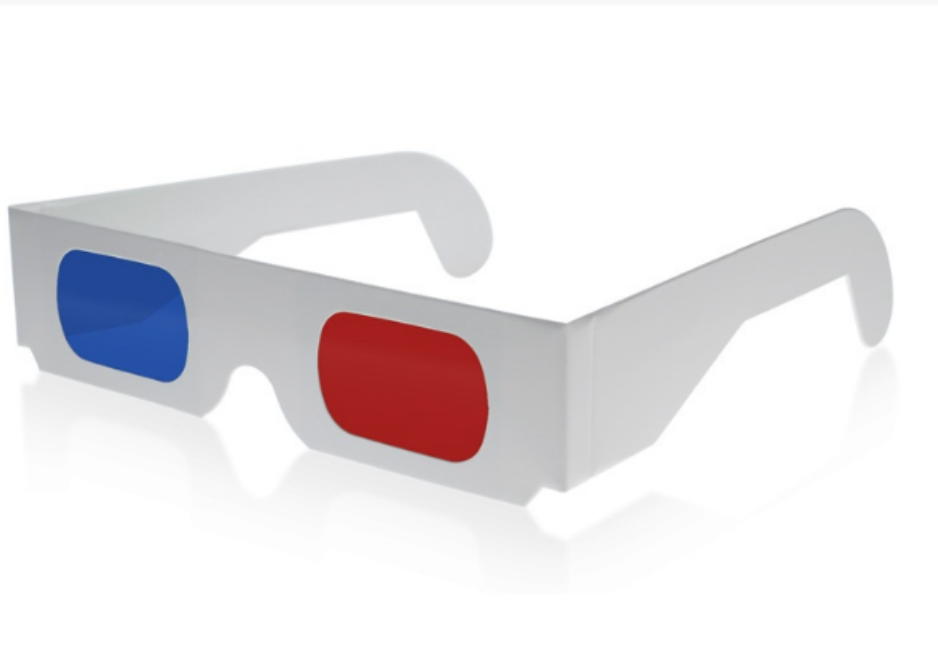
```
install.packages("devtools")
require("devtools")
install_github("ggvegan", "gavinsimpson")
require("ggvegan")
autoplot()
```

EASTER EGG: rgl and vegan 3d

Interactive 3D biplots using rgl

```
require(rgl)
require(vegan3d)
ordirgl(spe.h.pca)
```





Challenge # 3

Using everything you have learned to execute a PCA on the mite species abundance data

```
data(mite)
```

- What are the significant axes?
- Which groups of sites can you identify?
- Which groups of species are related to these groups of sites?

Solution #3

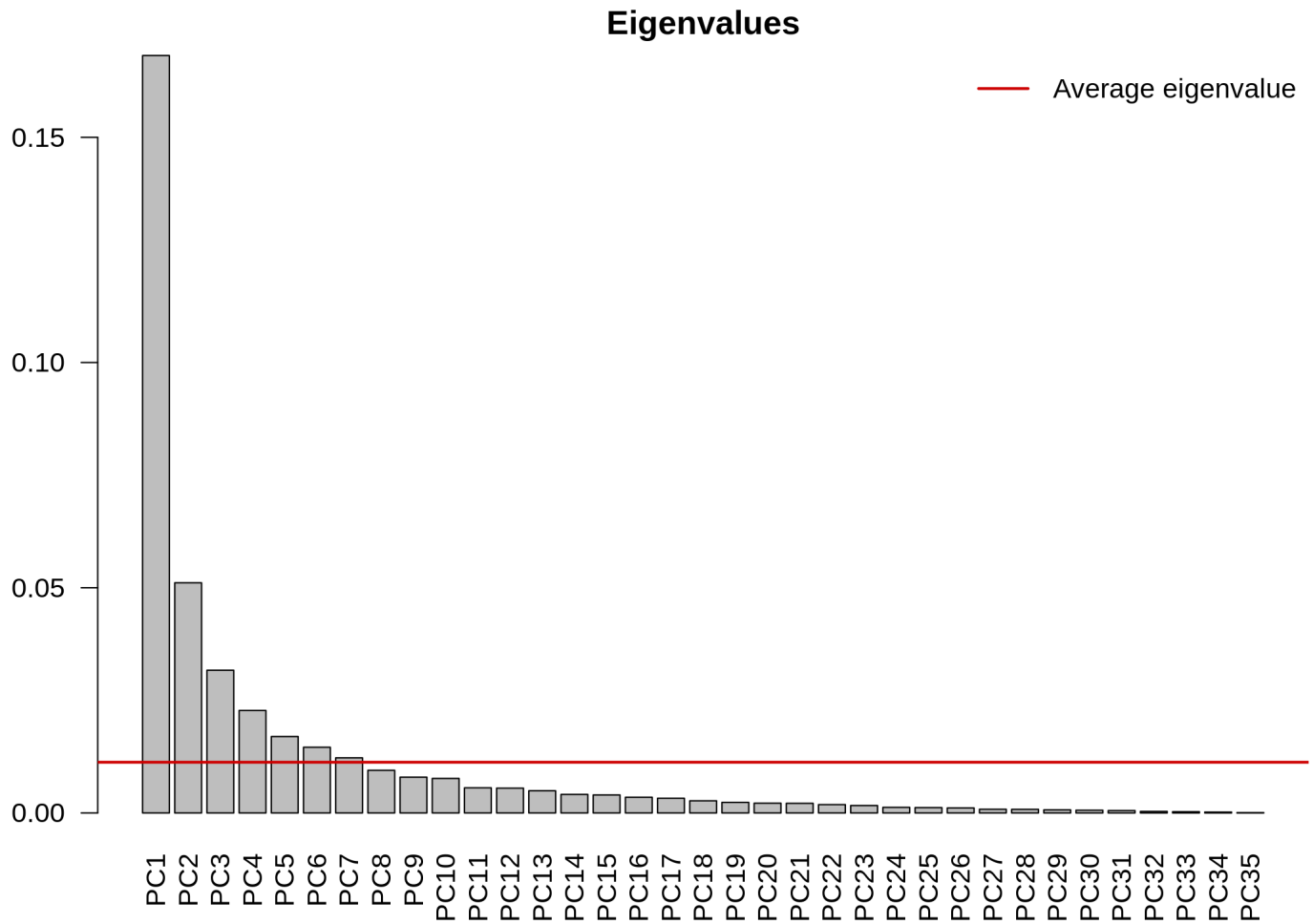
- Compute PCA on the Hellinger-transformed species data

```
mite.spe.hel ← decostand(mite, method = "hellinger")

mite.spe.h.pca ← rda(mite.spe.hel)
```

- Check significant axes using the Guttman-Kaiser criterion

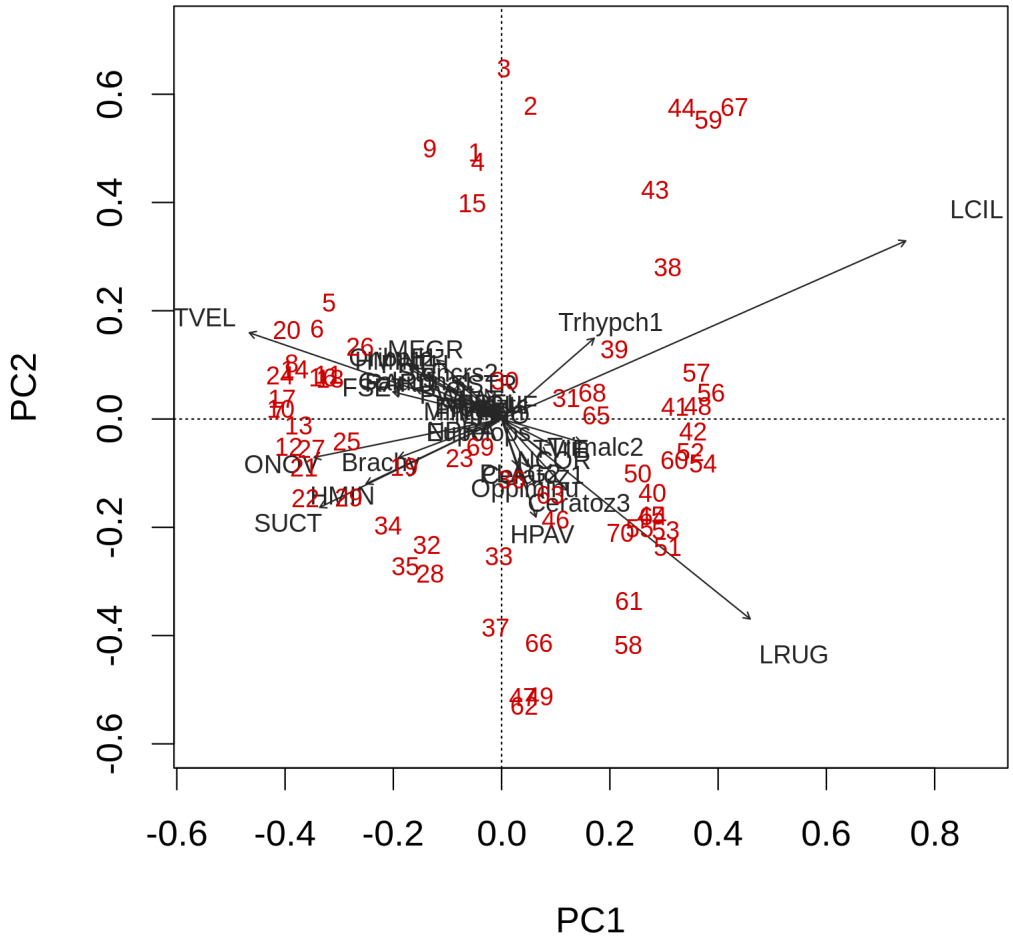
```
ev ← mite.spe.h.pca$CA$eig
ev[ev>mean(ev)]
n ← length(ev)
barplot(ev, main = "Eigenvalues", col = "grey", las = 2)
abline(h = mean(ev), col = "red3", lwd = 2)
legend("topright", "Average eigenvalue", lwd = 2, col = "red3", bty = "n")
```

Solution #3


Eigenvalues
Average eigenvalue
0.15
0.10
0.05
0.00
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
PC28
PC29
PC30
PC31
PC32
PC33
PC34
PC35

Solution #3

```
biplot(mite.spe.h.pca, col = c("red3", "grey15"))
```



Warnings

- PCA is a linear method and thus relies on a few assumptions
  - multinormal distribution of the data (only if you wish to make inferences)
  - not too many zeros
  - the gradient of interest is causing the majority of the variance in the dataset

Violation of these can cause a horseshoe shape in your biplots, where opposite ends of the horseshoe are close together but in reality represent opposite ends of a gradient

Warnings

- We can avoid some of these problems in PCA by choosing appropriate transformations for your community composition data or environmental data
- In some cases, such as studies that cover very large environmental gradients, it may be appropriate to use other types of unconstrained ordinations

Euclidean vs Chi^2 distances

- PCA preserves **euclidean distances** between objects, and thus postulates **linear relationships** between species, and between species and environmental gradients.
- ... but in **some cases, species instead present unimodal responses** to environmental gradients

Principles of CA

- In such cases, CA should be preferred compared to PCA as it preserves **Chi2 distances between sites**... and thus better represents uni modal relationships

How to run a CA?

- CA is implemented in the `vegan` package using the function `cca()`:

```
spe.ca ← cca(spe[-8,])
# only take columns which rowsums are > than 0.
```

- CA on fish species abundances

CA: R output

- CA results are presented in the same way as PCA results and can be called using:

```
summary(spe.ca)
#
# Call:
# cca(X = spe[-8, ])
#
# Partitioning of scaled Chi-square:
#               Inertia Proportion
# Total           1.128          1
# Unconstrained   1.128          1
#
# Eigenvalues, and their contribution to the scaled Chi-square
#
# Importance of components:
#                            CA1    CA2     CA3     CA4     CA5     CA6     CA7
# Eigenvalue            0.6062 0.1423 0.10251 0.07319 0.04912 0.03909 0.03341
# Proportion Explained  0.5374 0.1262 0.09087 0.06488 0.04354 0.03465 0.02962
# Cumulative Proportion 0.5374 0.6635 0.75437 0.81925 0.86279 0.89745 0.92706
#                            CA8     CA9     CA10     CA11     CA12     CA13
# Eigenvalue            0.01709 0.01302 0.010765 0.008141 0.007533 0.005820
# Proportion Explained  0.01515 0.01154 0.009543 0.007217 0.006678 0.005159
# Cumulative Proportion 0.94221 0.95375 0.963294 0.970511 0.977188 0.982347
```

CA: Interpretation of results

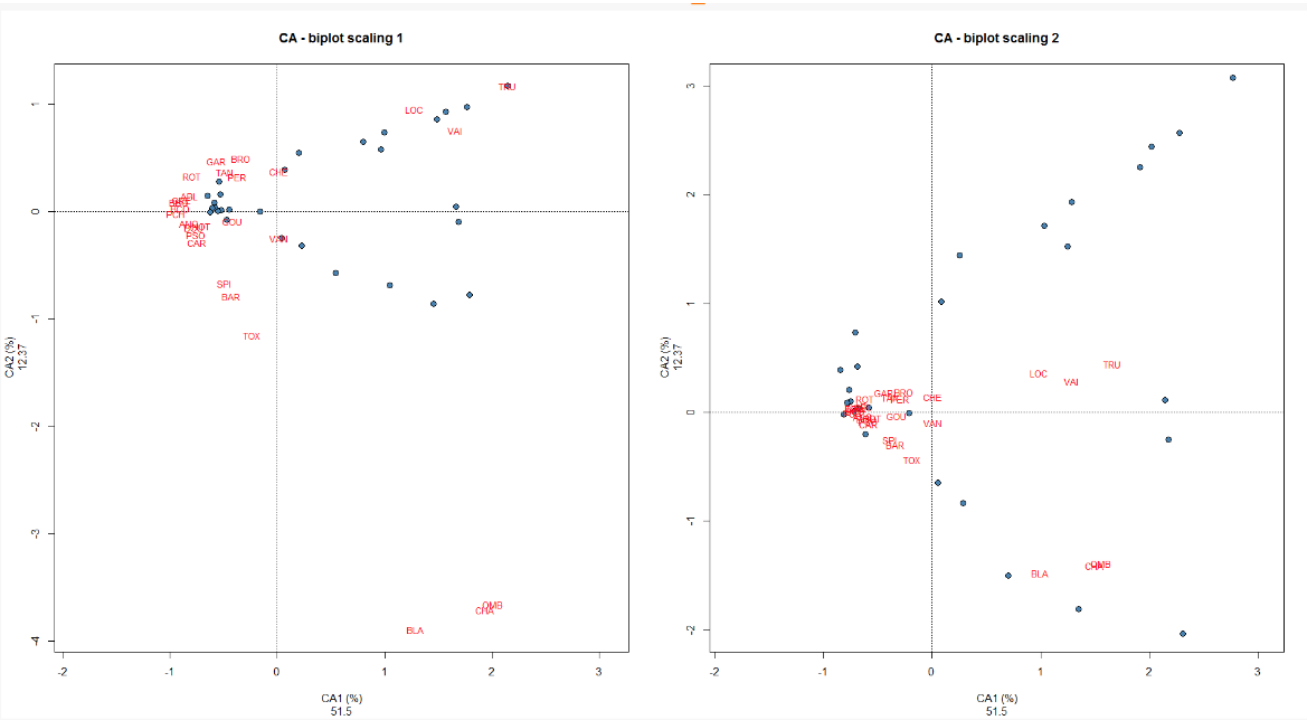
```
Call:
cca(X = spe)

Partitioning of mean squared contingency coefficient:
              Inertia Proportion
Total           1.167          1
Unconstrained   1.167          1

Eigenvalues, and their contribution to the mean squared contingency coefficient

Importance of components:
                          CA1    CA2     CA3     CA4     CA5     CA6     CA7     CA8
Eigenvalue            0.601 0.1444 0.10729 0.08337 0.05158 0.04185 0.03389 0.02883
Proportion Explained  0.515 0.1237 0.09195 0.07145 0.04420 0.03586 0.02904 0.02470
Cumulative Proportion 0.515 0.6388 0.73069 0.80214 0.84634 0.88220 0.91124 0.93594
                          CA9    CA10    CA11     CA12     CA13     CA14     CA15
Eigenvalue            0.01684 0.01083 0.01014 0.007886 0.006123 0.004867 0.004606
Proportion Explained  0.01443 0.00928 0.00869 0.006760 0.005250 0.004170 0.003950
Cumulative Proportion 0.95038 0.95965 0.96835 0.975100 0.980350 0.984520 0.988470
                          CA16     CA17     CA18     CA19     CA20      CA21
Eigenvalue            0.003844 0.003067 0.001823 0.001642 0.001295 0.0008775
Proportion Explained  0.003290 0.002630 0.001560 0.001410 0.001110 0.0007500
Cumulative Proportion 0.991760 0.994390 0.995950 0.997360 0.998470 0.9992200
                          CA22      CA23      CA24      CA25      CA26
Eigenvalue            0.0004217 0.0002149 0.0001528 8.949e-05 2.695e-05
Proportion Explained  0.0003600 0.0001800 0.0001300 8.000e-05 2.000e-05
Cumulative Proportion 0.9995900 0.9997700 0.9999000 1.000e+00 1.000e+00

Scaling 2 for species and site scores
* Species are scaled proportional to eigenvalues
* Sites are unscaled: weighted dispersion equal on all dimensions
```

26 CA axes identified

% CA1 = 51.50%

% CA2 = 12.37%

CA: biplots



The group of sites on the left is characterized by the species *GAR*, *TAN*, *PER*, *ROT*, *PSO*, and *CAR*

The group of sites in the upper right corner is characterized by the species *LOC*, *VAI* and *TRU* The group of sites in the lower right corner is characterized by the species *BLA*, *CHA*, and *OMB*

Challenge #4

Using everything you have learned to execute a CA on the mite species abundance data:

```
mite.spe ← mite
```

- What are the significant axes?
- Which groups of sites can you identify?
- Which groups of species are related to these groups of sites?

Solution #4

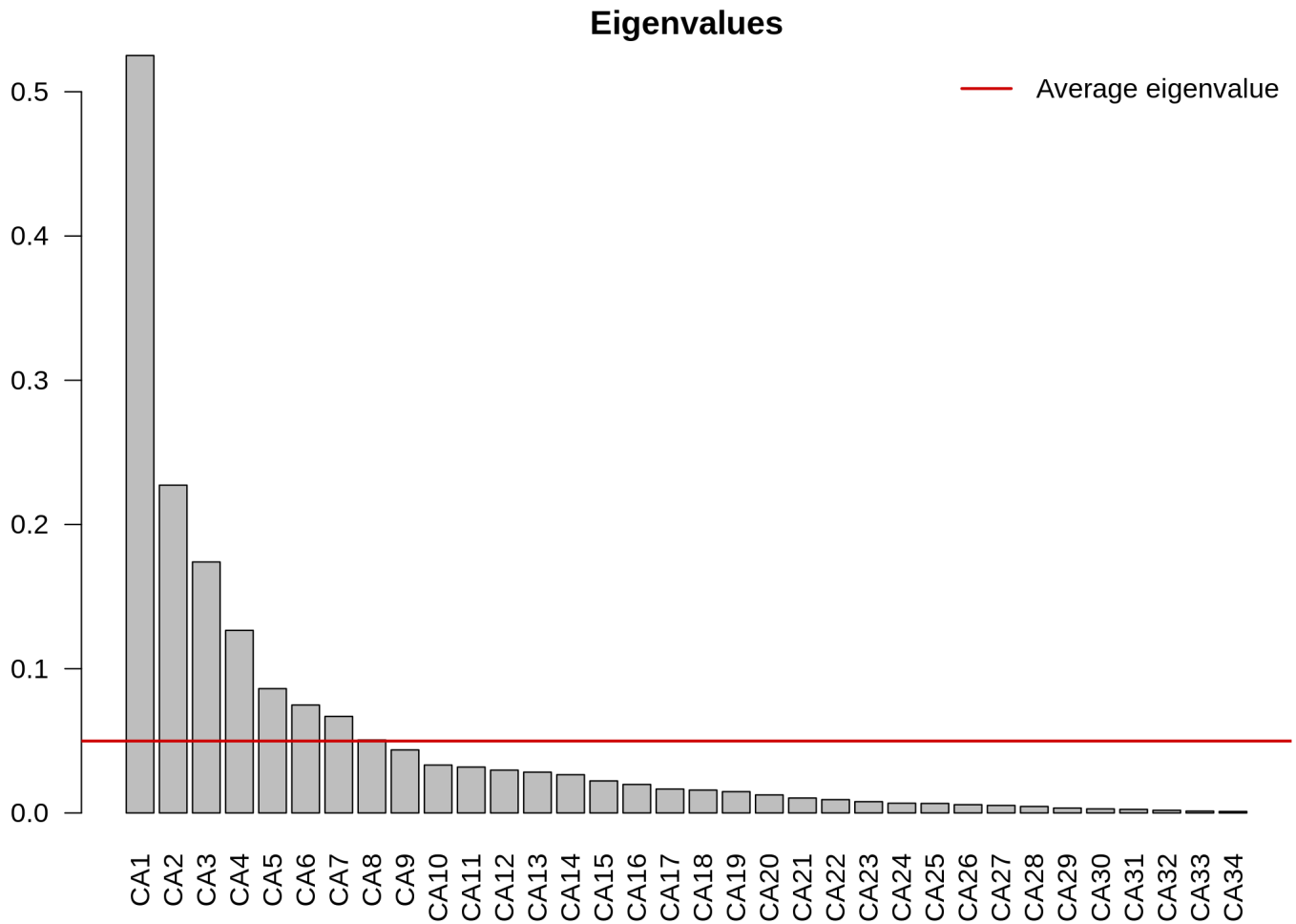
- Compute CA:

```
mite.spe.ca ← cca(mite.spe)
```

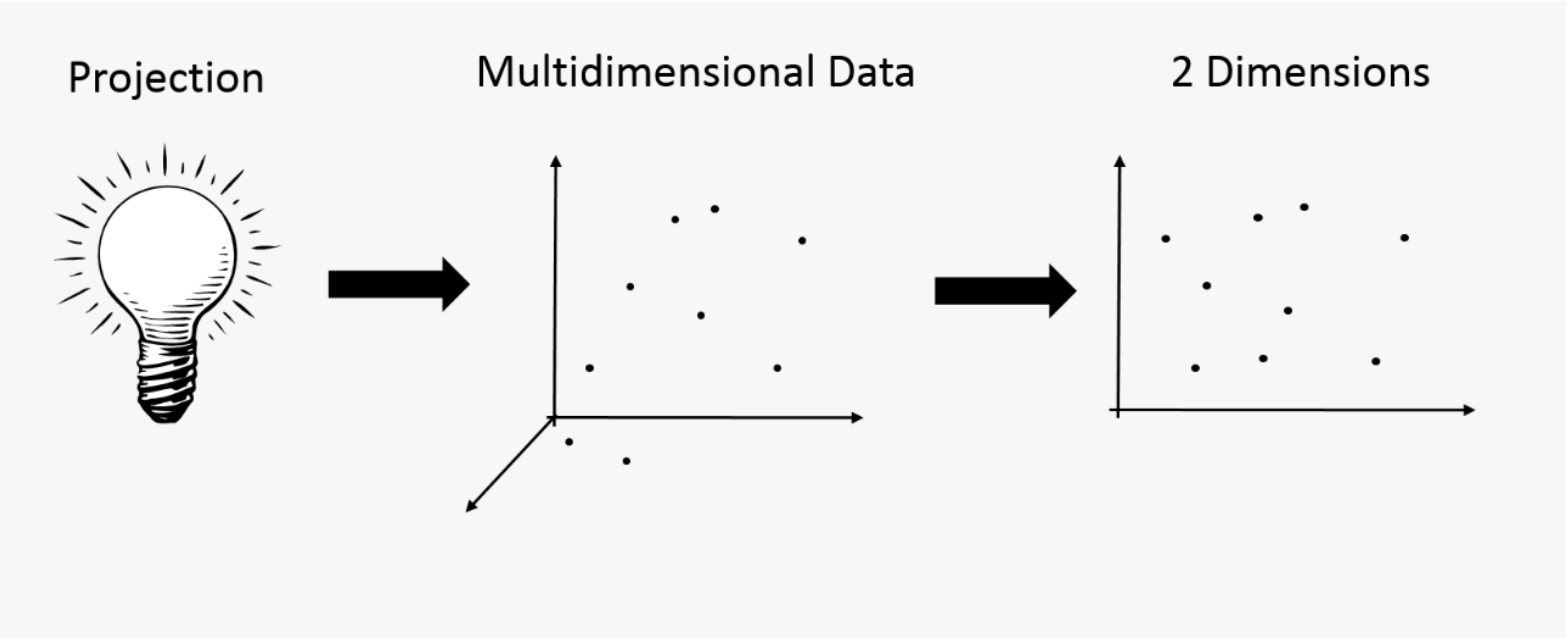
- Check significant axes using the Guttman-Kaiser criterion

```
ev ← mite.spe.ca$CA$eig
ev[ev > mean(ev)]
n ← length(ev)
barplot(ev, main = "Eigenvalues", col = "grey", las = 2)
abline(h = mean(ev), col = "red3", lwd = 2)
legend("topright", "Average eigenvalue", lwd = 2, col = red3, bty = "n")
```

Solution #4


Eigenvalues
Average eigenvalue
0.5
0.4
0.3
0.2
0.1
0.0
CA1
CA2
CA3
CA4
CA5
CA6
CA7
CA8
CA9
CA10
CA11
CA12
CA13
CA14
CA15
CA16
CA17
CA18
CA19
CA20
CA21
CA22
CA23
CA24
CA25
CA26
CA27
CA28
CA29
CA30
CA31
CA32
CA33
CA34

4.3. Principal Coordinates Analysis



- In PCA, we preserve the maximum amount of variation in the data
- In PCoA, we preserve as best we can in 2D (Euclidean) distances between each object in multidimensional space

PCoA can be especially useful when the dataset is wider than it is long (Typical problem in Genetics)

PCoA - Let's try it on Fish species!

- For computing PCoA, we can use the `cmdscale()` or the `pcoa()`functions from the **stats** and **ape** packages:

```
?cmdscale
library(ape)
?pcoa
```

- Run a PCoA on the Hellinger-distances of the fish dataset and extract the results

```
spe.h.pcoa ← pcoa(dist(spec.hel))
summary(spe.h.pcoa)
#            Length Class      Mode
# correction   2    -none-     character
# note         1    -none-     character
# values       5    data.frame list
# vectors    783    -none-     numeric
# trace        1    -none-     numeric
```

PCoA - Interpretation of Output

```
$values
    Eigenvalues Relative_eig Broken_stick Cumul_eig Cumul_br_stick
1  7.2228938501 5.133437e-01  0.144128028 0.5133437      0.1441280
2  1.7987448715 1.278400e-01  0.107090991 0.6411837      0.2512190
3  1.2970422885 9.218307e-02  0.088572472 0.7333668      0.3397915
4  1.0780684157 7.662021e-02  0.076226793 0.8099870      0.4160183
5  0.6150272794 4.371107e-02  0.066967534 0.8536980      0.4829858
```

- Eigenvalues
- Relative eigenvalues
- Broken stick model: to evaluate which axes are significant
- Cumulative eigenvalues: cumulative value of the relative eigenvalues
- Cumulative broken stick: cumulative value of the broken stick model

PCoA - Interpretation of Output

	Axis.1	Axis.2	Axis.3	Axis.4	Axis.5	Axis.6
1	-0.509824403	-0.276543720	0.64011383	-0.339373399	0.207330880	0.303563377
2	-0.698794880	-0.039355856	0.11324989	-0.232885899	-0.157730682	-0.024561130
3	-0.640690642	0.015667069	0.03835044	-0.266970577	-0.125293094	-0.095688892
4	-0.413985947	0.104770836	-0.15728486	-0.285182806	-0.001250382	-0.065361378
5	0.003083242	0.052843104	-0.32206098	-0.273069271	0.315944703	0.059891284
6	-0.295314224	0.057788054	-0.32395301	-0.226290236	0.056493824	0.036877698

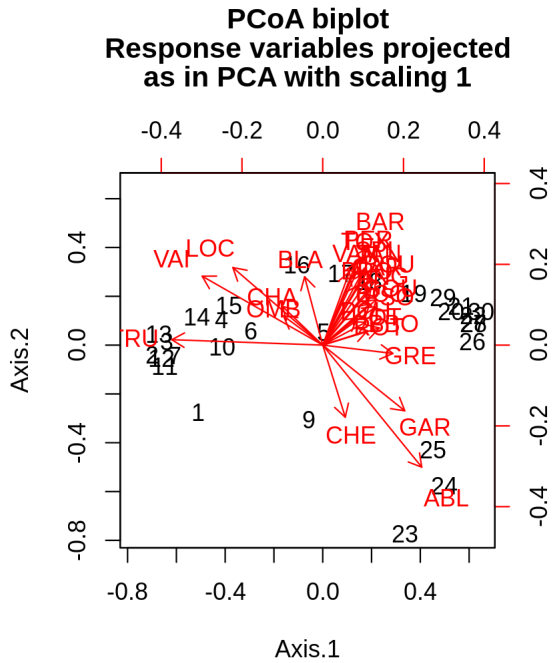
- Vectors: the eigenvectors associated to each eigenvalue contain the coordinates, in Euclidean space, of each site.

These are the most useful for subsequent analysis as they capture the distance among objects very well

PCoA biplot with `biplot.pcoa()`

We can display, in 2D, the distances between sites using the `biplot.pcoa()` function, as well as the species associated to each site

```
biplot.pcoa(spe.h.pcoa, spec.hel)
```



PCoA and non-metric distances

- PCoA can also be used to capture information contained in non-metric distances, such as the popular Bray-Curtis distance. Let's give it a try:

```
spe.bray.pcoa ← pcoa(spe.db.pa)
# spe.bray.pcoa
```

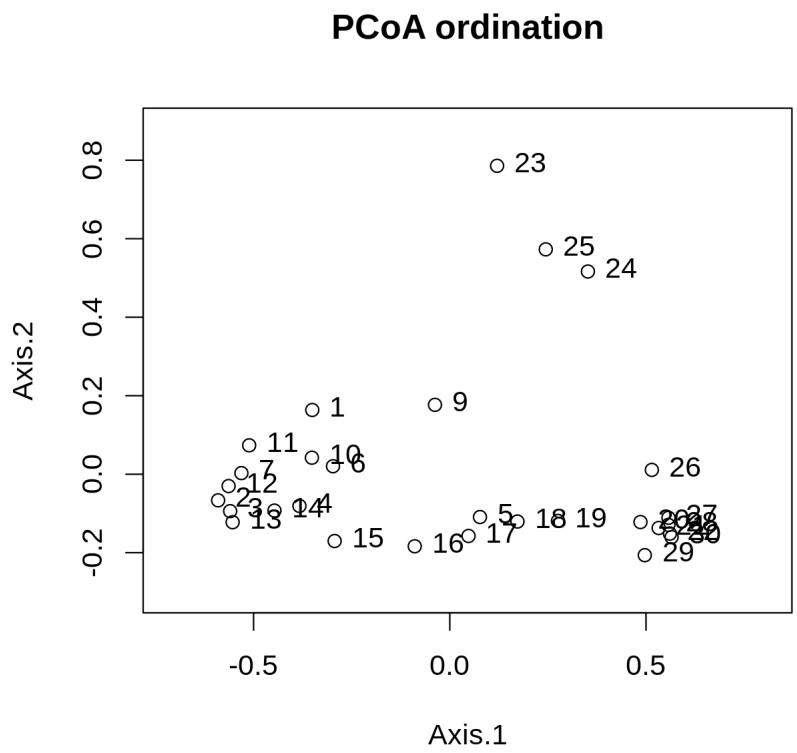
- Examine the output and notice the negative eigenvalues. This is because non-metric distances cannot be represented in Euclidean space without corrections (*see* Legendre & Legendre 2012 for more details on this):

```
spe.bray.pcoa ← pcoa(spe.db.pa, correction = "cailliez")
# spe.bray.pcoa
```

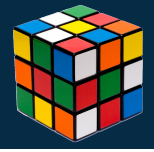
PCoA and non-metric distances

- Now let's visualize this using a biplot without the species (more common approach for PCoA)

```
biplot.pcoa(spe.bray.pcoa)
```



Challenge #5



Execute a PCoA on the Hellinger-transformed mite species abundance data

- What are the significant axes?
- Which groups of sites can you identify?
- Which groups of species are related to these groups of sites
- How do the PCoA results compare with the PCA results?

Solution #5

- Hellinger transform the species data

```
mite.spe.hel ← decostand(mite.spe, method = "hellinger")
```

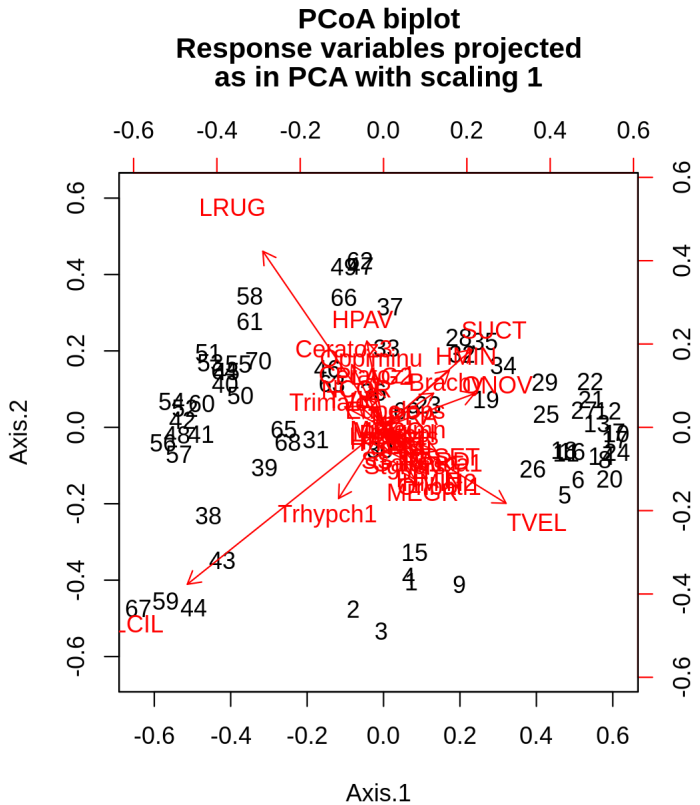
- Compute PCoA

```
mite.spe.h.pcoa ← pcoa(dist(mite.spe.hel))
```

Solution #5

- Build a biplot to visualize the data:

```
biplot.pcoa(mite.spe.h.pcoa, mite.spe.hel)
```



Non-metric Multidimensional scaling (NMDS)

- In PCA, CA and PCoA, objects are ordinated in a few number of dimensions (i.e. axis) generally > 2
- Consequently, 2D-biplots can fail to represent all the variation of the dataset
- In some cases, the objective is however to represent the data in a specified small number of dimensions
- Then, how do you plot the ordination space to represent all the variation in the data?

Principles of NMDS

- NMDS
  - is the non-metric counterpart of PCoA
  - uses an iterative optimization algorithm to find the best representation of distances in reduced space
  - increasingly popular
- In NMDS, users can thus specify;
  - the number of dimensions
  - the distance measure

How to run a NMDS

- nMDS is implemented in the `vegan` package using function `metaMDS()` where
  - *distance* specifies the distance measure to use
  - *k* specifies the number of dimensions

```
spe.nmds ← metaMDS(spe, distance = 'bray', k = 2)
# Run 0 stress 0.07477822
# Run 1 stress 0.1120183
# Run 2 stress 0.09234818
# Run 3 stress 0.07506752
#  ... Procrustes: rmse 0.0147091  max resid 0.06367012
# Run 4 stress 0.1226549
# Run 5 stress 0.1247914
# Run 6 stress 0.1225934
# Run 7 stress 0.08841674
# Run 8 stress 0.07429516
#  ... New best solution
#  ... Procrustes: rmse 0.02409357  max resid 0.09225307
# Run 9 stress 0.07383678
#  ... New best solution
#  ... Procrustes: rmse 0.01387956  max resid 0.0638735
# Run 10 stress 0.08844686
# Run 11 stress 0.1121913
# Run 12 stress 0.0884532
```

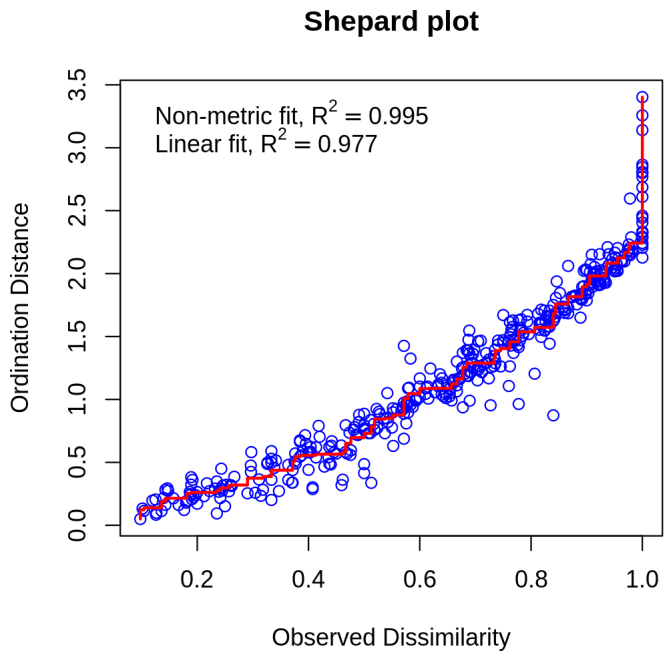
NMDS: goodness-of-fit

- NMDS applies an iterative procedure that tries to position the objects in the requested number of dimensions in such a way as to minimize a stress function (scaled from 0 to 1) which measure the goodness-of-fit of the distance adjustment in the reduced-space configuration.
- Consequently, the lower the stress value, the better the representation of objects in the ordination-space is.

NMDS: goodness-of-fit

- The Shepard diagram and stress values can be obtained from:

```
spe.nmds$stress
# [1] 0.07376222
stressplot(spe.nmds, main = "Shepard plot")
```

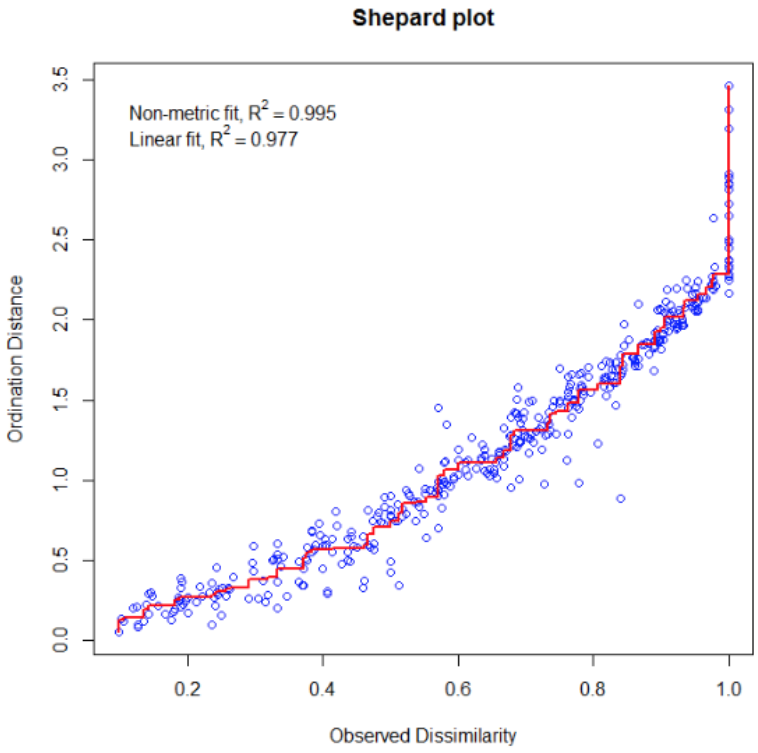


NMDS on fish abundances

- Run the NMDS and assess the goodness of fit

```
spe.nmds ← metaMDS(spe, distance = 'bray', k = 2)
spe.nmds$stress
stressplot(spe.nmds, main = "Shepard plot")
```

NMDS on fish abundances



- The Shepard plot identifies a strong correlation between observed dissimilarity and ordination distance (R2 > 0.95) highlighting a high goodness of fit of the NMDS

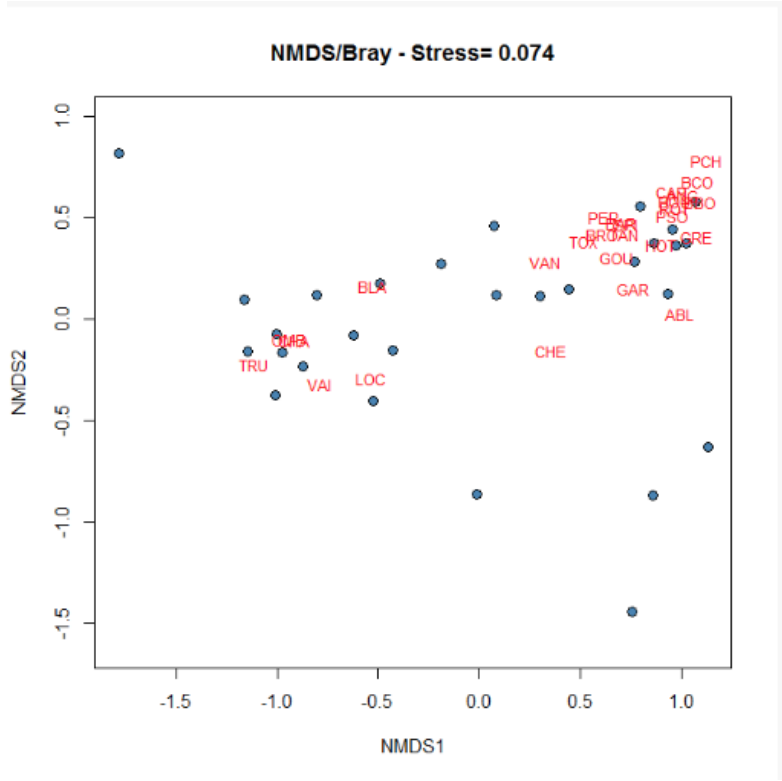
NMDS on fish abundances

- Construct the biplot

```
plot(spe.nmds, type = "none",
     main = paste("NMDS/Bray - Stress =",
                  round(spe.nmds$stress, 3)),
     xlab = c("NMDS1"), ylab = "NMDS2")

points(scores(spe.nmds, display = "sites",
              choiches = c(1,2),
              pch = 21,
              col = "black",
              g = "steelblue",
              cex = 1.2))
text(scores(spe.nmds, display = "species", choices = c(1)),
            scores(spe.nmds, display = "species", choices = c(2)),
            labels = rownames(scores(spe.nmds, display = "species")),
            col = "red", cex = 0.8)
```

NMDS on fish abundances

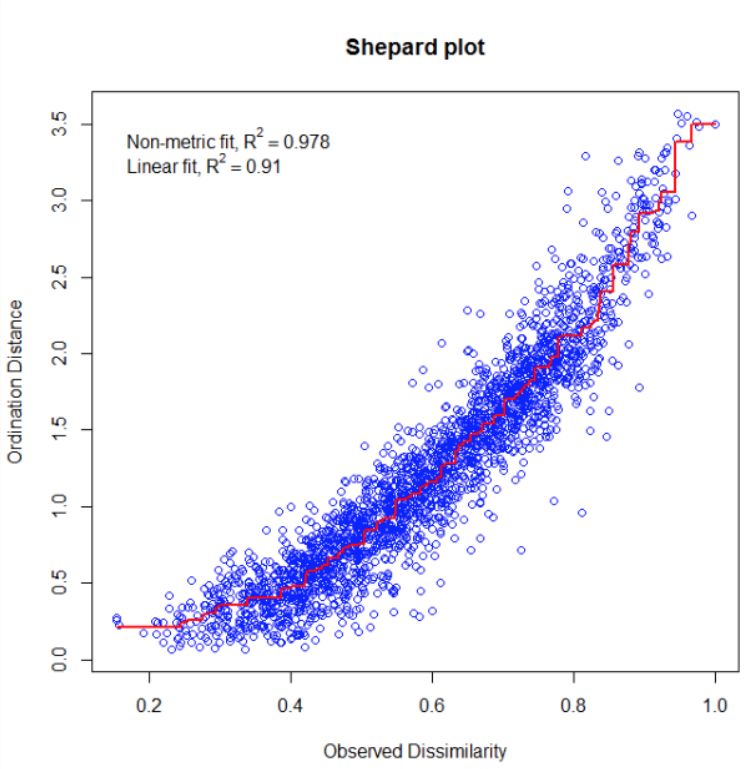
The biplot of the NMDS shows a group of closed sites characterized by the species BLA, TRU, VAI, LOC, CHA and OMB, while the other species form a cluster of sites in the upper right part of the graph. Four sites in the lower part of the graph are strongly different from the others



Challenge #6

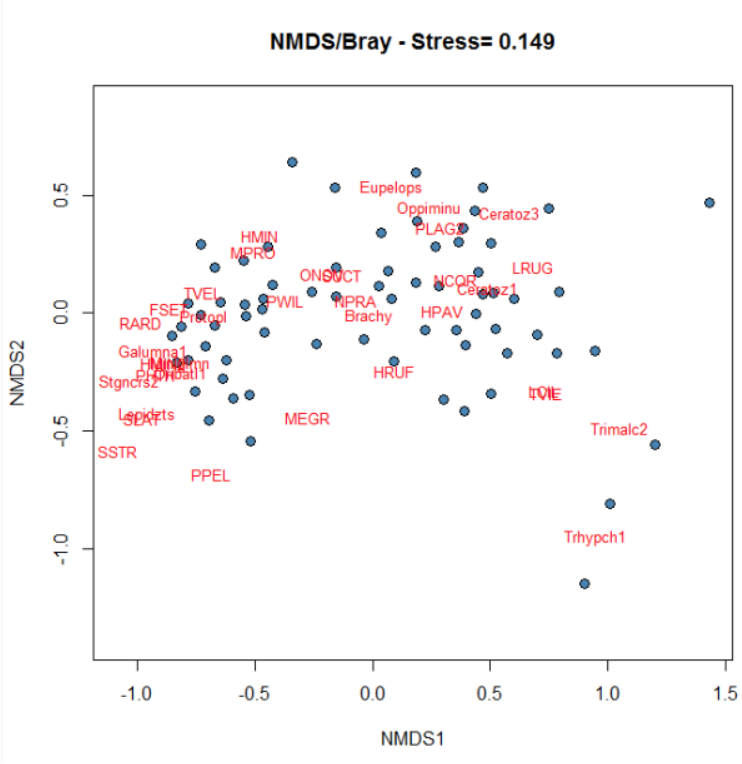
- Run the NMDS of the mite species abundances in 2 dimensions based on a Bray-Curtis distance.
- Assess the goodness-of-fit of the ordination and interpret the biplot

Solution #6



The correlation between observed dissimilarity and ordination distance (R2 > 0.91) and the stress value relatively low, showing together a good accuracy of the NMDS ordination

Solution #6



No cluster of sites can be precisely defined from the NMDS biplot showing that most of the species occurred in most of the sites, i.e. a few sites shelter specific communities

Conclusion

Many ordination techniques exist, but their specificity should guide your choices on which methods to use

	Distance preserved	Variables	Maximum number of axis
PCA	Euclidean	Quantitative data, linear relationships	p
CA	Chi2	Non-negative, quantitative homogeneous data, binary data	p-1
PCoA	User defined	Quantitative, semi-quantitative, mixed data	p-1
NMDS	User defined	Quantitative, semi-quantitative, mixed data	User defined

Prime time 4 quiz time

What does PCA stand for?

Principal Component Analysis

Which one is the best way to visualize the *distances* between the community composition of many sites?

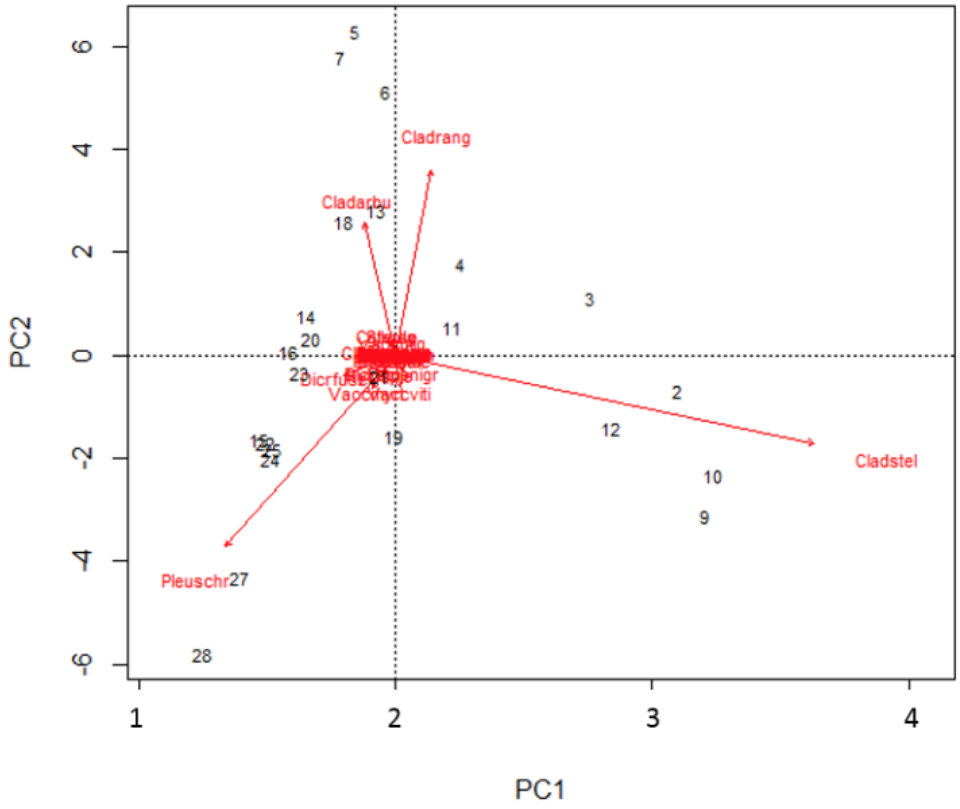
Principal Coordinate Analysis (PCoA)

What does an eigenvalue represent in PCA?

The proportion of variance explained by a principal component

Prime time 4 quiz time

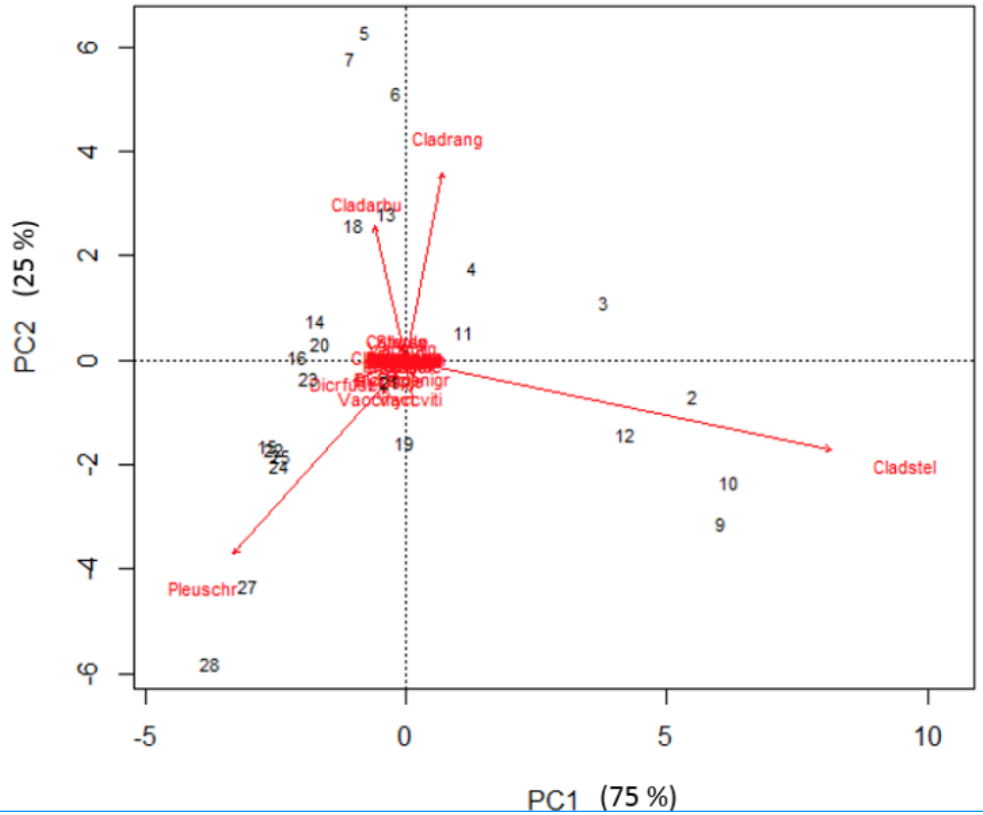
Spot what is sketchy



- **Data non centered, Yikes!**

Prime time 4 quiz time

Spot what is sketchy



- **2 first PCs explain 100% of the variation!**

Live Long and Ordinate

Thank you for attending this workshop!