Multivariate statistics

AUTHORS
Camila Pacheco
Katrín Björnsdóttir

Today

- Learn the basics of multivariate analysis to reveal patterns
- Use `R` to perform Unconstrained ordinations
- Learn the following methods:
  - Clustering analysis
  - Detrended correspondence analysis (DCA)
  - Principal Component Analysis (PCA)
  - Non-metric Multidimensional Scaling (NMDS)
- Break ☻
- Practice

Required Material

https://github.com/lacapary/BIO503/

Required Material

You are required to have downloaded and installed

```
        install.packages(c("vegan",
                           "ape",
                           "factoextra",
                           "dendextend"))
```

Required Material

Do not hesitate to ask questions!

Recap: Linear models

- We learned some models to study at ecological data.
- These models allowed us to ask questions such as:
  - What are the effects of precipitation and temperature on species richness? or

  - How does the abundance of species change between habitats?

Multivariate statistics

Sometimes, we want to figure out things from ecological data that has more than one main outcome or dependent variable.

our research question might be:

- How does the plants composition change along an elevation gradient?
- What is the composition dissimilarity of plants communities?
- How closely-related are local vegetation communities in terms of their composition ?

In all these questions, the outcome is composed of several variables, e.g. usually a list of samples and the types of species in them, or a list of samples and the environment they’re in.

Multivariate statistics

Matrix Species

Site	**Species 1**	**Species 2**	**…**	**Species n**
1	abundance 1	abundance 2	…	abundance n
2	abundance 1	abundance 2	…	abundance n
m	…	…	…	…

Matrix locations

Site	**Temperature**	**Precipitation**	**…**	**Driver n**
1	Temperature 1	Precipitation 2	…	Driver n
2	Temperature 1	Precipitation 2	…	Driver n
m	…	…	…	…

Multivariate statistics

Matrix algebra

Multivariate statistics

Matrix algebra

$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

Multivariate statistics

Association matrices

- Q-mode : analysis for objects or sites
- R-mode : analysis for descriptors or species

What is ordination?

> Ordination is a collective term for multivariate techniques which summarize a multidimensional dataset in such a way that when it is projected onto a low dimensional space, any intrinsic pattern the data may possess becomes apparent upon visual inspection (Pielou, 1984).

- In ecological terms, ordination helps us understand community data.
- Like how many species are in different locations. It does this by creating a simple space where similar species and samples are near each other, and different ones are far apart.
- Ideally, this space shows important environmental differences clearly.

The data for this session?

The data originates from the research of Batterink & Wijffels (1983), published as a report in Dutch.

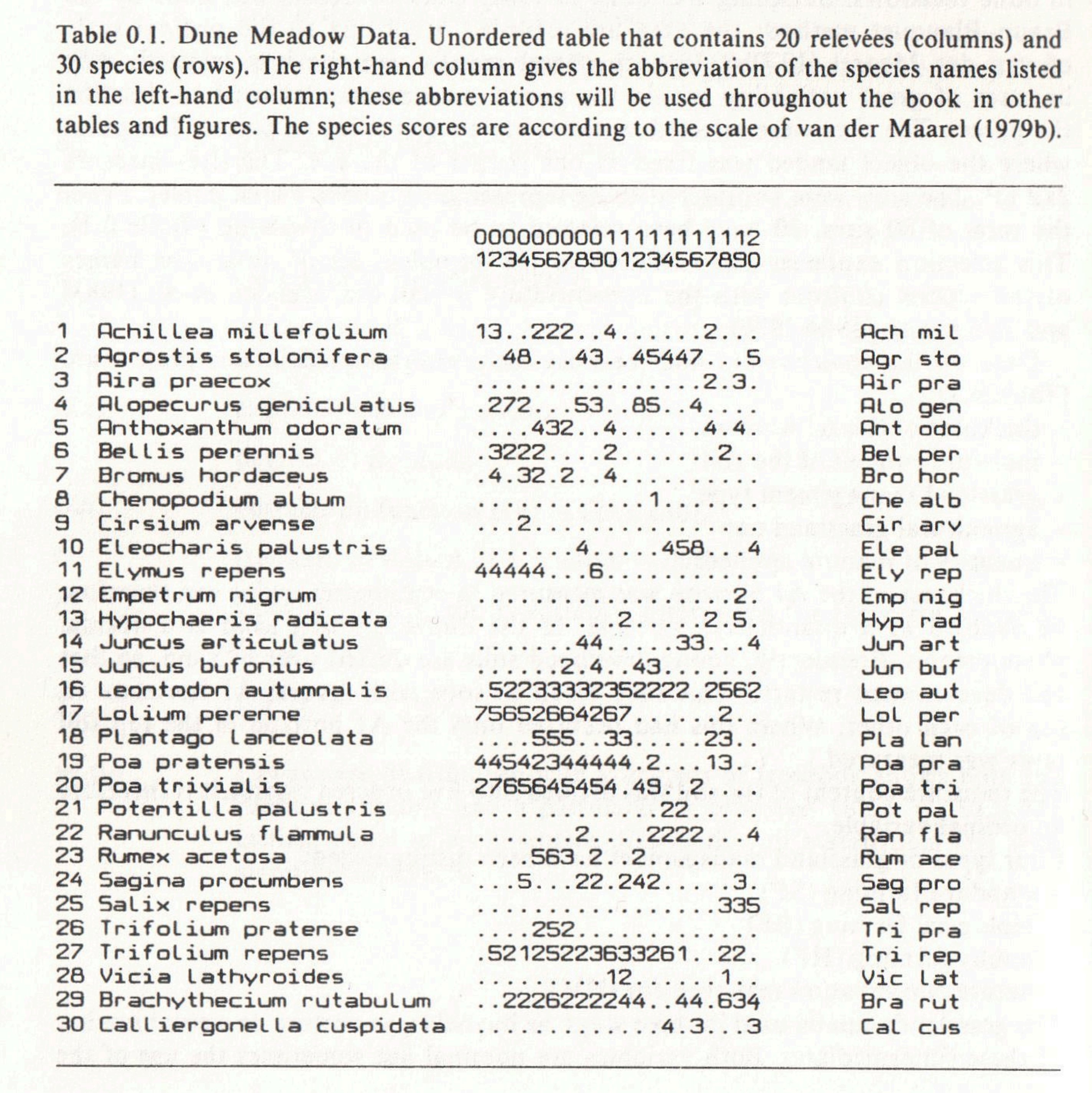


Table 0.1. Dune Meadow Data. Unordered table that contains 20 relevées (columns) and 30 species (rows). The right-hand column gives the abbreviation of the species names listed in the left-hand column; these abbreviations will be used throughout the book in other tables and figures. The species scores are according to the scale of van der Maarel (1979b).

		00000000011111111112	
		12345678901234567890	
1	Achillea millefolium	13..222..4......2...	Ach mil
2	Agrostis stolonifera	..48...43..45447...5	Agr sto
3	Aira praecox	................2.3.	Air pra
4	Alopecurus geniculatus	.272...53..85..4....	Alo gen
5	Anthoxanthum odoratum	....432..4......4.4.	Ant odo
6	Bellis perennis	.3222....2.......2..	Bel per
7	Bromus hordaceus	.4.32.2..4..........	Bro hor
8	Chenopodium album	............1.......	Che alb
9	Cirsium arvense	...2................	Cir arv
10	Eleocharis palustris	.......4.....458...4	Ele pal
11	Elymus repens	44444...6...........	Ely rep
12	Empetrum nigrum	..................2.	Emp nig
13	Hypochaeris radicata	..........2.....2.5.	Hyp rad
14	Juncus articulatus	.......44.....33...4	Jun art
15	Juncus bufonius	......2.4..43.......	Jun buf
16	Leontodon autumnalis	.52233332352222.2562	Leo aut
17	Lolium perenne	75652664267......2..	Lol per
18	Plantago lanceolata	....555..33.....23..	Pla lan
19	Poa pratensis	44542344444.2...13..	Poa pra
20	Poa trivialis	2765645454.49..2....	Poa tri
21	Potentilla palustris	.............22.....	Pot pal
22	Ranunculus flammula	.......2....2222...4	Ran fla
23	Rumex acetosa	....563.2..2........	Rum ace
24	Sagina procumbens	...5...22.242.....3.	Sag pro
25	Salix repens	335	Sal rep
26	Trifolium pratense	....252.............	Tri pra
27	Trifolium repens	.52125223633261..22.	Tri rep
28	Vicia lathyroides	.........12......1..	Vic lat
29	Brachythecium rutabulum	..2226222244..44.634	Bra rut
30	Calliergonella cuspidata	.............4.3...3	Cal cus

The data for this session?

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.0     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2     
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(vegan)
```

```
Loading required package: permute
Loading required package: lattice
This is vegan 2.6-4
```

```
# Load the community dataset which we`ll use in the examples today

dune2_spe <- read_csv("Data/dune2_spe.csv")
```

```
Rows: 20 Columns: 28
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
dbl (28): Achimill, Agrostol, Airaprae, Alopgeni, Anthodor, Bellpere, Bromho...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dune2_env <- read_csv("Data/dune2_env.csv")
```

```
Rows: 20 Columns: 5
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (2): Management, Use
dbl (3): A1, Moisture, Manure

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

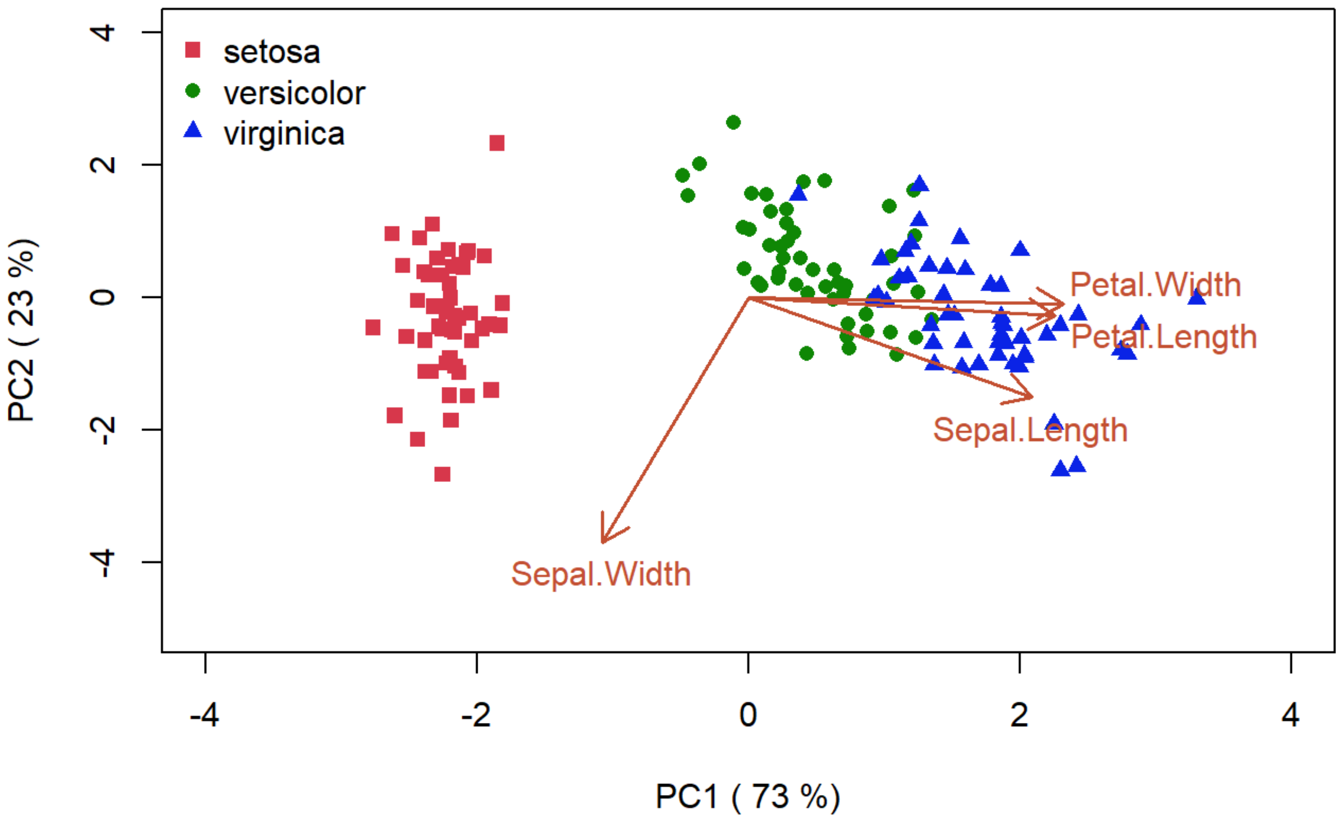
```
# Open the dataset and look if you can find any patterns
head(dune2_spe)
```

```
# A tibble: 6 × 28
  Achimill Agrostol Airaprae Alopgeni Anthodor Bellpere Bromhord Chenalbu
     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1        1        0        0        0        0        0        0        0
2        3        0        0        2        0        3        4        0
3        0        4        0        7        0        2        0        0
4        0        8        0        2        0        2        3        0
5        2        0        0        0        4        2        2        0
6        2        0        0        0        3        0        0        0
# ℹ 20 more variables: Cirsarve <dbl>, Comapalu <dbl>, Eleopalu <dbl>,
#   Elymrepe <dbl>, Empenigr <dbl>, Hyporadi <dbl>, Juncarti <dbl>,
#   Juncbufo <dbl>, Lolipere <dbl>, Planlanc <dbl>, Poaprat <dbl>,
#   Poatriv <dbl>, Ranuflam <dbl>, Rumeacet <dbl>, Sagiproc <dbl>,
#   Salirepe <dbl>, Scorautu <dbl>, Trifprat <dbl>, Trifrepe <dbl>,
#   Vicilath <dbl>
```

```
head(dune2_env)
```

```
# A tibble: 6 × 5
     A1 Moisture Management Use      Manure
  <dbl>    <dbl> <chr>      <chr>     <dbl>
1   2.8        1 SF         Haypastu      4
2   3.5        1 BF         Haypastu      2
3   4.3        2 SF         Haypastu      4
4   4.2        2 SF         Haypastu      4
5   6.3        1 HF         Hayfield      2
6   4.3        1 HF         Haypastu      2
```

What is ordination?



Type of ordinations?

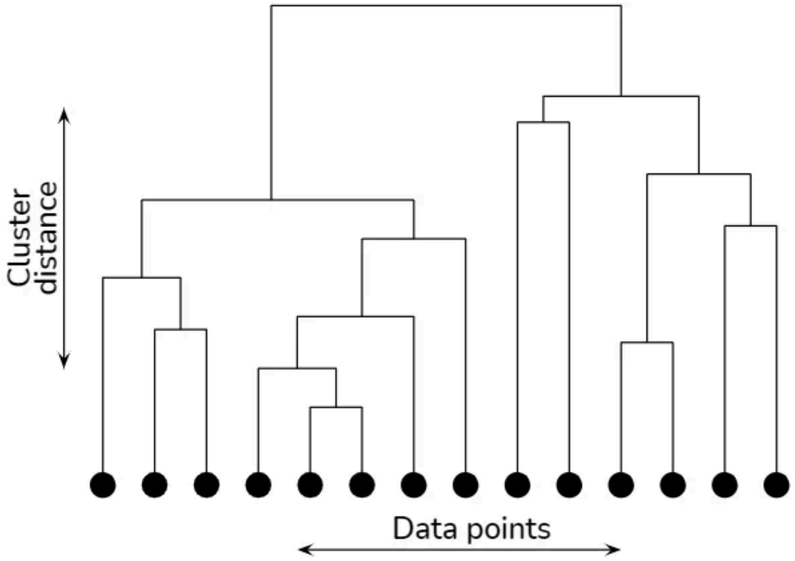
- Unconstrained Ordination: we're basically letting the data *speak for itself.* We don't impose any specific relationships or constraints between the variables.
- Constrained Ordination: we impose some restrictions or *constraints* on the analysis based on what we already know or suspect about the data.
- In simple terms, unconstrained ordination lets the data tell its story without interference, while constrained ordination guides the analysis based on what we already know or suspect.
- We are going to focus in Unconstrained Ordinations

Ordination vs. Clustering

- Ordination and clustering are the two main classes of multivariate methods that community ecologists employ.
- To some degree, these two approaches are complementary.
- Hierarchical data clustering allows you to explore your data and look for discontinuities (e.g. gaps in your data), gradients and meaningful ecological units (e.g. groups or subgroups of species).
- Given the continuous nature of communities, ordination can be considered a more natural approach. Ordination aims at arranging samples or species continuously along gradients.

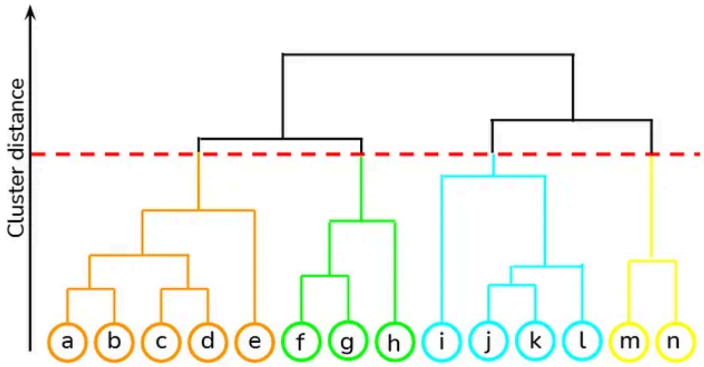
Clustering

- Hierarchical clustering offers insight into how your biodiversity data are organized and can help you to disentangle different patterns and the scales at which they can be observed.
- Its results can be represented as dendrograms (tree-like diagrams), which describe how closely observations are.

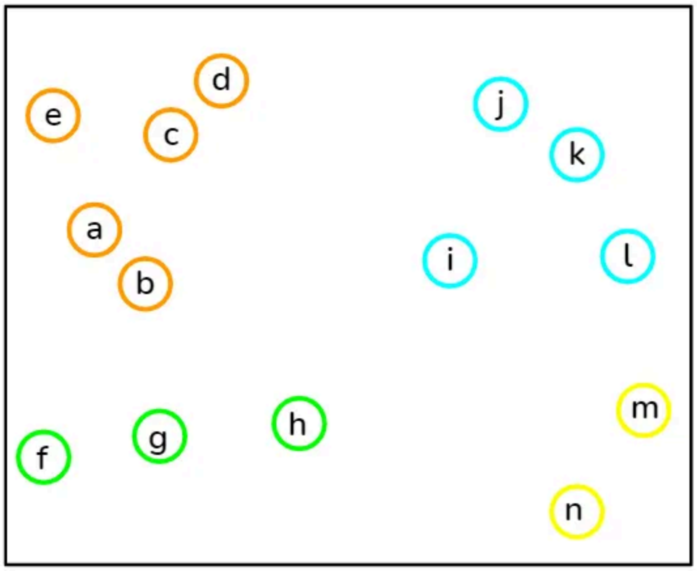


source: Prasad Pai

Clustering



source: Prasad Pai



source: Prasad Pai

Clustering

```
library(dendextend)
```

```
Registered S3 method overwritten by 'dendextend':
  method     from 
  rev.hclust vegan


---------------------
Welcome to dendextend version 1.17.1
Type citation('dendextend') for how to cite the package.

Type browseVignettes(package = 'dendextend') for the package vignette.
The github page is: https://github.com/talgalili/dendextend/

Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
You may ask questions at stackoverflow, use the r and dendextend tags: 
	 https://stackoverflow.com/questions/tagged/dendextend
```

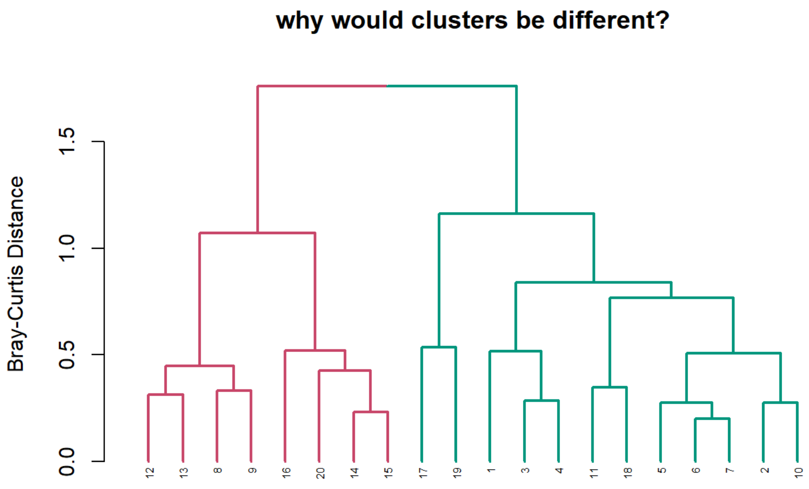
```
    To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
---------------------

Attaching package: 'dendextend'

The following object is masked from 'package:permute':

    shuffle

The following object is masked from 'package:stats':

    cutree
```

```
dis_data<-dune2_spe %>%
  vegdist(method = "bray",upper=FALSE)

dend <- dis_data %>%
  hclust(method="ward.D2") %>%
  as.dendrogram()

dend
```

```
'dendrogram' with 2 branches and 20 members total, at height 1.757535
```

Clustering

```
dend.plot <-  dend %>%
  set("branches_lwd", 2) %>% # Branches line width
  set("branches_k_color",  k = 2) %>% # Color branches by groups
  set("labels_cex", 0.5) # Change label size

plot(dend.plot, ylab = "Bray-Curtis Distance", main = "why would clusters be different?")
```



Clustering

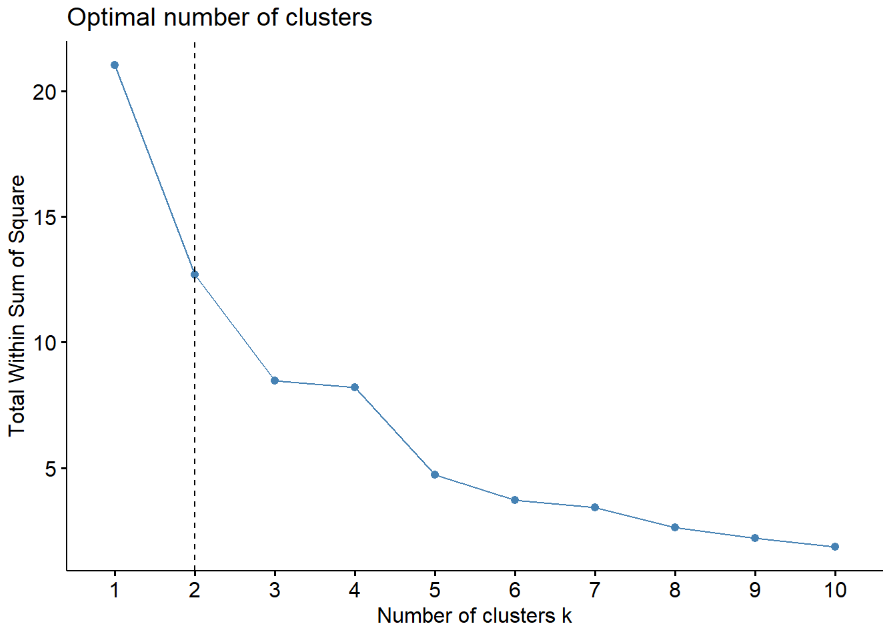
What is the ideal number of clusters?

```
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

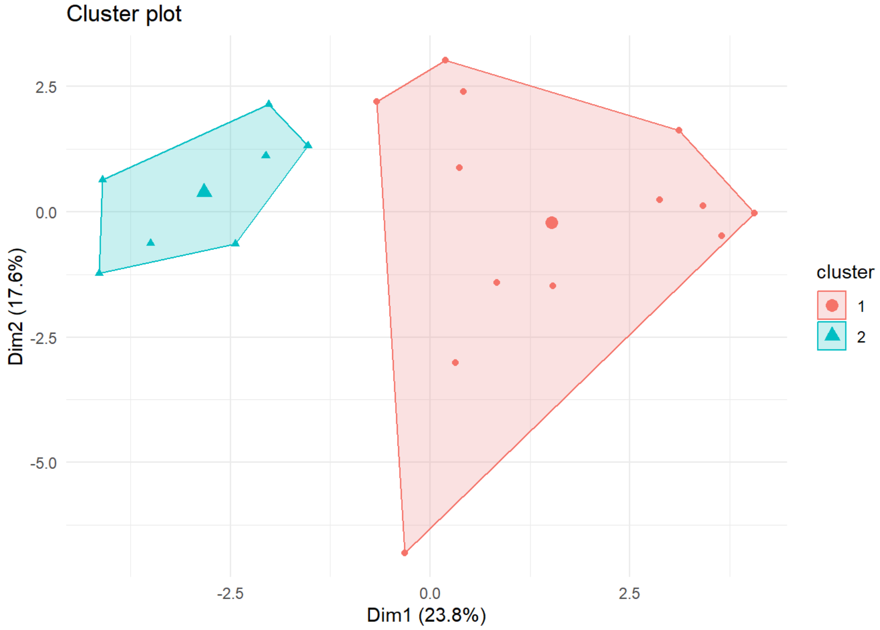
```
varespec_m<-dis_data |> as.matrix()
```

```
fviz_nbclust(varespec_m, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)
```



Clustering

```
fviz_cluster(kmeans(varespec_m, centers = 2), geom = "point", data = dune2_spe)+ theme_minimal()
```



Ordinations

- Assess relationships within a set of variables (species or environmental variables)
- Find key components of variation among samples, sites, species
- Reduce the number of dimensions in multivariate data while limiting substantial loss of information
- Create new variables for use in subsequent analyses

Doing an ordination

This ordination goes in two steps:

- First, we will perform an ordination on a species abundance matrix.
- Then we will use environmental data (samples by environmental variables) to interpret the gradients that were uncovered by the ordination.

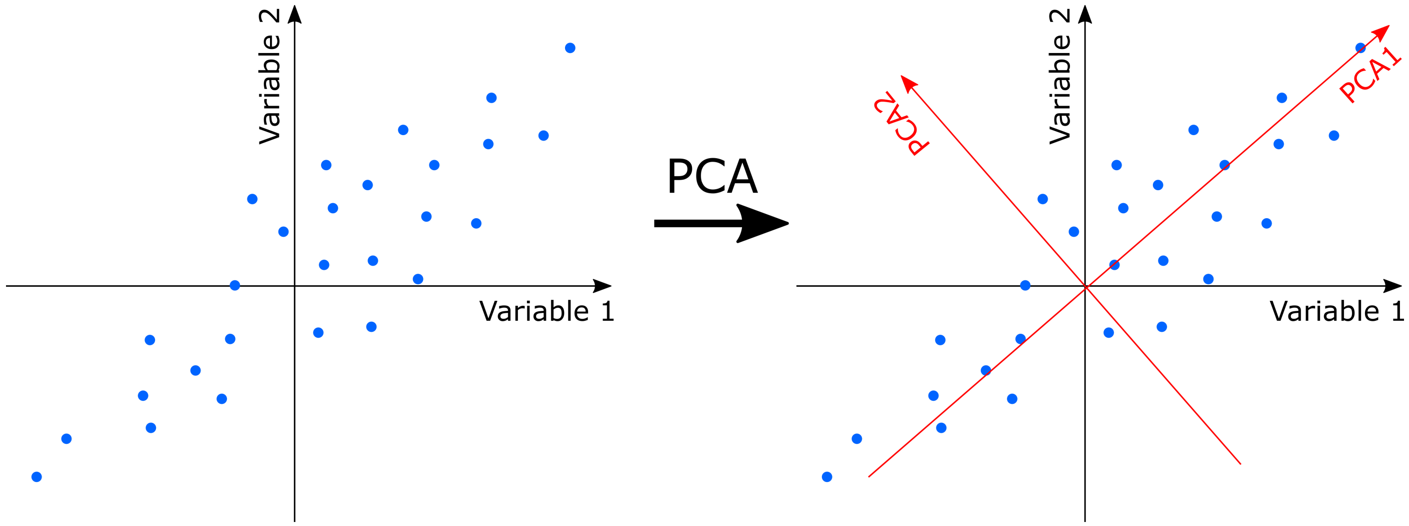
Different ordination techniques

- *P*rincipal *C*omponent *A*nalysis (PCA)
- *D*etrended *C*orrespondence *A*nalysis (DCA)
- *N*on-metric *M*ulti*di*mensional *S*caling (NMDS)
- And *MORE*.....

*P*rincipal *C*omponent *A*nalysis (PCA)

- It is a linear dimensionality-reduction technique, i.e. it reduces strongly correlated data.
- In a nutshell, the PCA linearly transforms the feature from the original space to a new feature space, containing principal components that explain most of the variance in the dataset

Principal Component Analysis (PCA)



source:Coding club

Principal Component Analysis (PCA)

Euclidean distances among samples

- The axes (also called principal components or PC) are orthogonal to each other (and thus independent).
- Each PC is associated with an eigenvalue.
- The sum of the eigenvalues will equal the sum of the variance of all variables in the data set.
- The eigenvalues represent the variance extracted by each PC, and are often expressed as a percentage of the sum of all eigenvalues (i.e. total variance).

Principal Component Analysis (PCA)

- The relative eigenvalues thus tell how much variation that a PC is able to 'explain'.
- Axes are ranked by their eigenvalues:
  - the first axis has the highest eigenvalue and thus explains the most variance
  - the second axis has the second highest eigenvalue, etc.

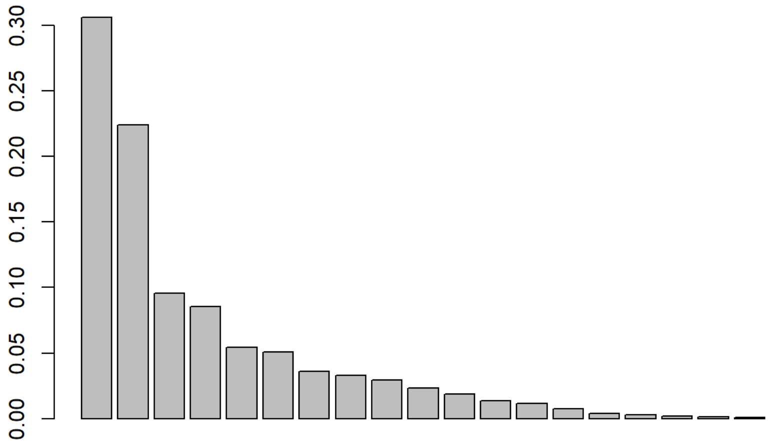
Principal Component Analysis (PCA)

```
PCA <- rda(dune2_spe, scale = FALSE)# Use scale = TRUE if your variables are on different scales (e.g. for abiotic variables).
# Here, all species are measured on the same scale
# So use scale = FALSE
PCA
```

```
Call: rda(X = dune2_spe, scale = FALSE)

              Inertia Rank
Total           78.97     
Unconstrained   78.97   19
Inertia is variance 

Eigenvalues for unconstrained axes:
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8 
24.181 17.678  7.557  6.760  4.274  4.009  2.835  2.584 
(Showing 8 of 19 unconstrained eigenvalues)
```

Principal Component Analysis (PCA)

```
# Now plot a bar plot of relative eigenvalues. This is the percentage variance explained by each axis
barplot(as.vector(PCA$CA$eig)/sum(PCA$CA$eig))
```

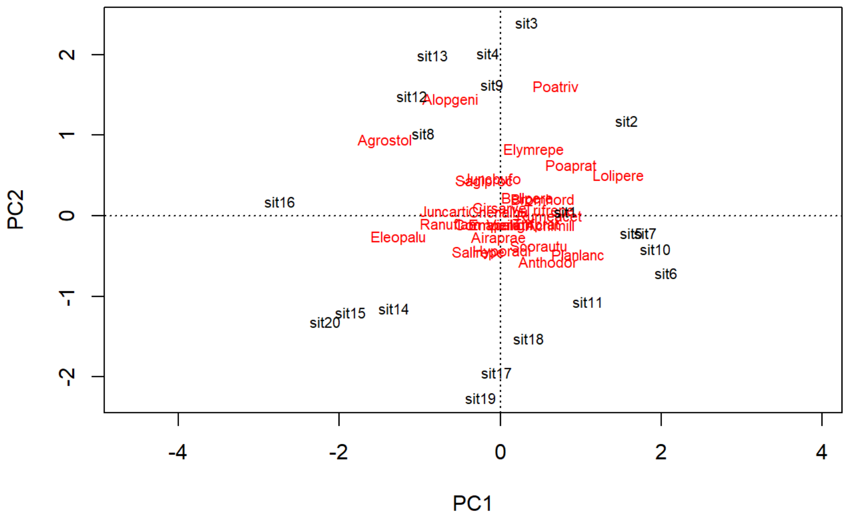


```
# Calculate the percent of variance explained by first two axes
sum((as.vector(PCA$CA$eig)/sum(PCA$CA$eig))[1:2]) # 53%, this is ok.
```

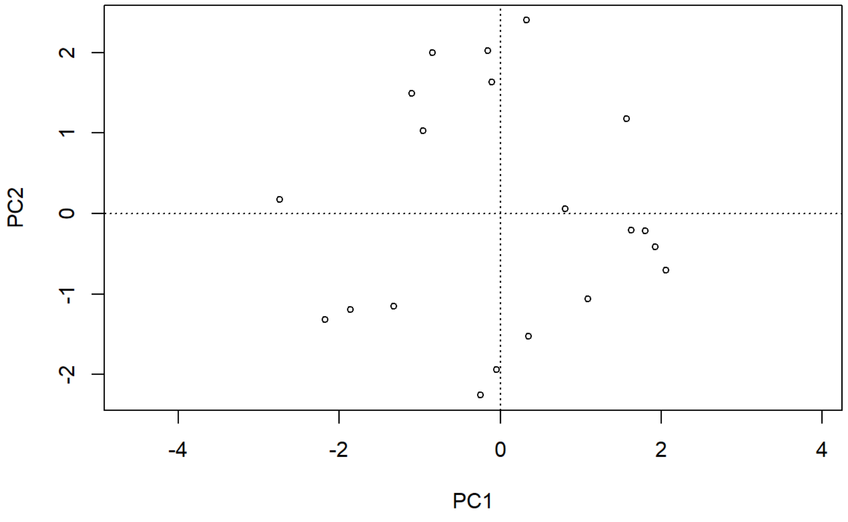
```
[1] 0.5300765
```

Principal Component Analysis (PCA)

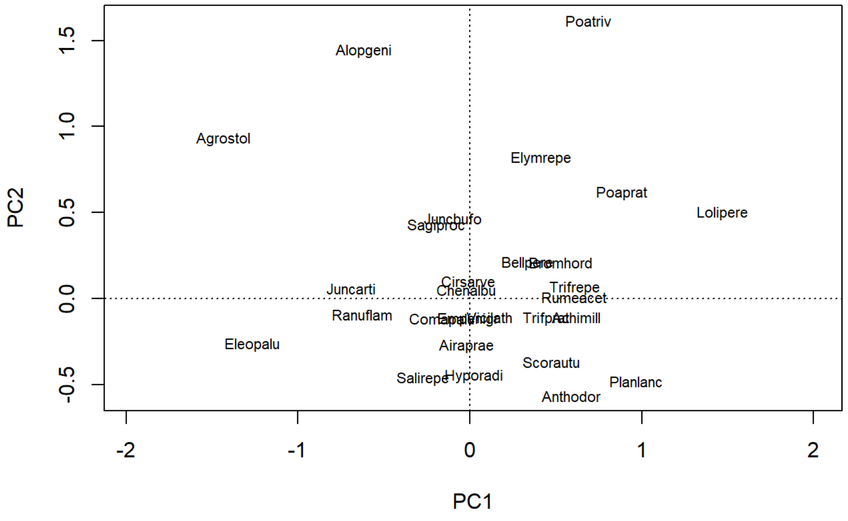
```
plot(PCA)
```



```
plot(PCA, display = "sites", type = "points")
```

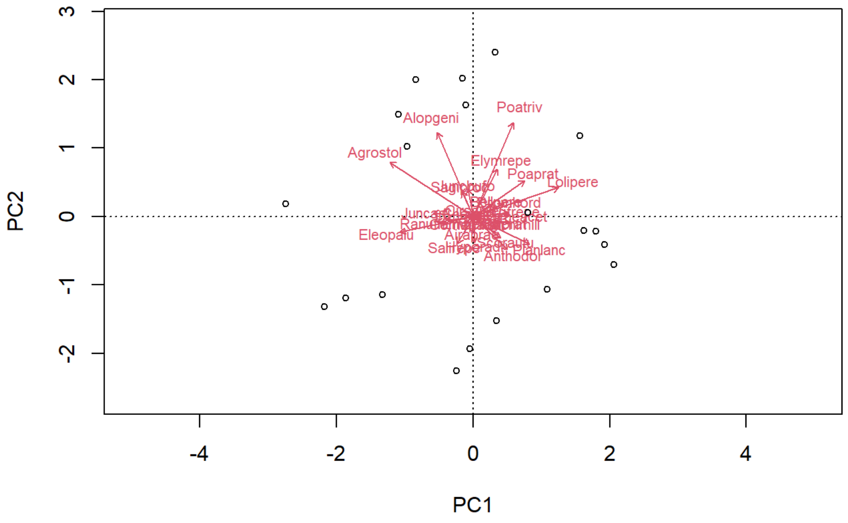


```
plot(PCA, display = "species", type = "text")
```



Principal Component Analysis (PCA)

```
# In a biplot of a PCA, species' scores are drawn as arrows
# that point in the direction of increasing values for that variable
biplot(PCA, choices = c(1,2), type = c("text", "points"), xlim = c(-5,5)) # biplot of axis 1 vs 2
```



Principal Component Analysis (PCA)

- This implies that the abundance of the species is continuously increasing in the direction of the arrow, and decreasing in the opposite direction.
- Thus PCA is a linear method.
- PCA is extremely useful when we expect species to be linearly (or even monotonically) related to each other.
- Unfortunately, we rarely encounter such a situation in nature.

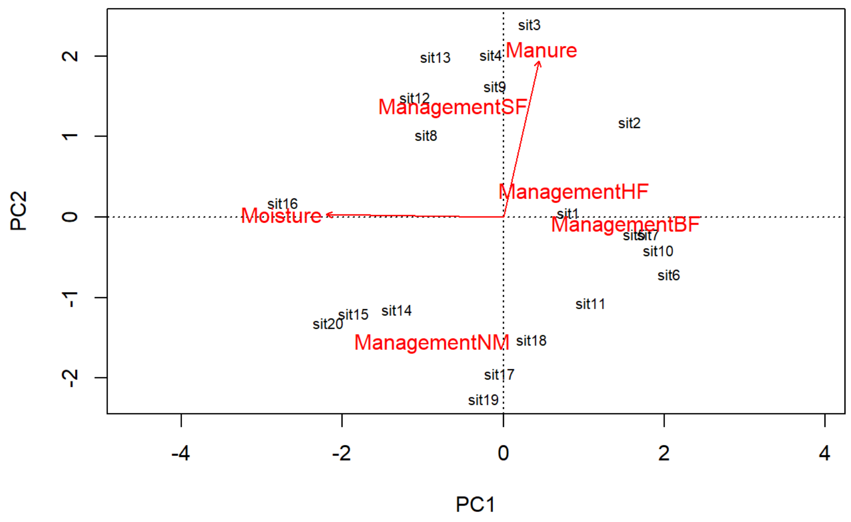
Environmetal Variables and Triplot

```
fit <- envfit(PCA, dune2_env, perm = 999)
```

```
scores(fit, "vectors")
```

```
                PC1         PC2
A1       -0.4981858 -0.11968978
Moisture -0.8609679  0.01183937
Manure    0.1718683  0.75960897
```

```
plot(PCA,dis="site")
plot(fit, p.max = 0.05, col = "red")
```

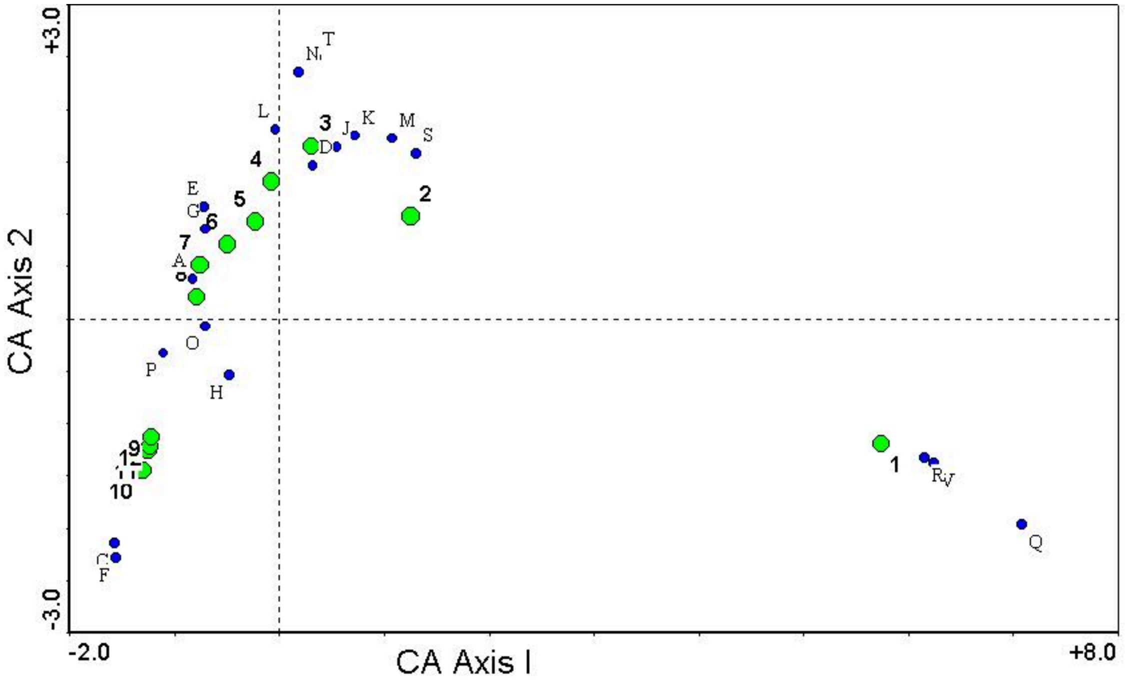


*D*etrended *c*orrespondence *a*nalysis (DCA)

chi-square distance metric among samples

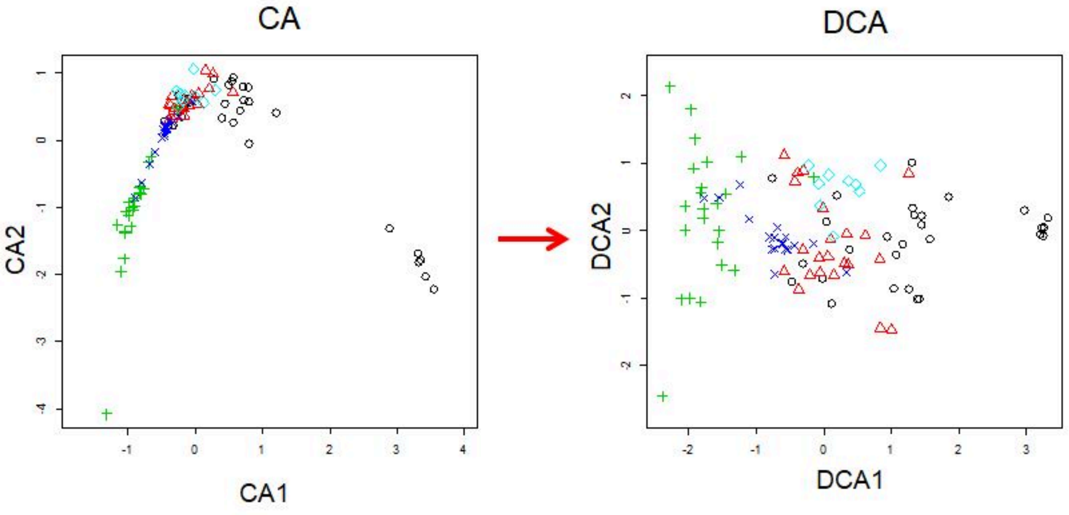
Correspondence analysis(*CA*) is an ordination method.

- It can calculate and display correspondence between samples and species in the same ordination space.
- It has a problem :suffers from creating often strong arch artefact in ordination diagrams. Which is caused by a non-linear correlation between first and higher axes



Detrended correspondence analysis (DCA)

Arch can be removed by detrending(*smooths out the data to make it easier to see the main patterns*), which is the base of the detrended correspondence analysis (DCA).



source: davidzeleny

Detrended correspondence analysis (DCA)

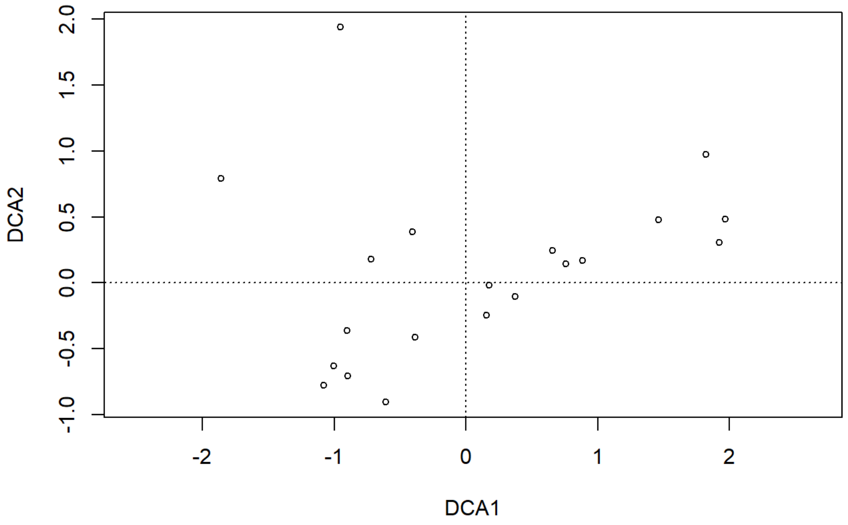
```
DCA<-decorana(dune2_spe)
DCA
```

```
Call:
decorana(veg = dune2_spe)

Detrended correspondence analysis with 26 segments.
Rescaling of axes with 4 iterations.
Total inertia (scaled Chi-square): 2.1866

                  DCA1   DCA2    DCA3    DCA4
```

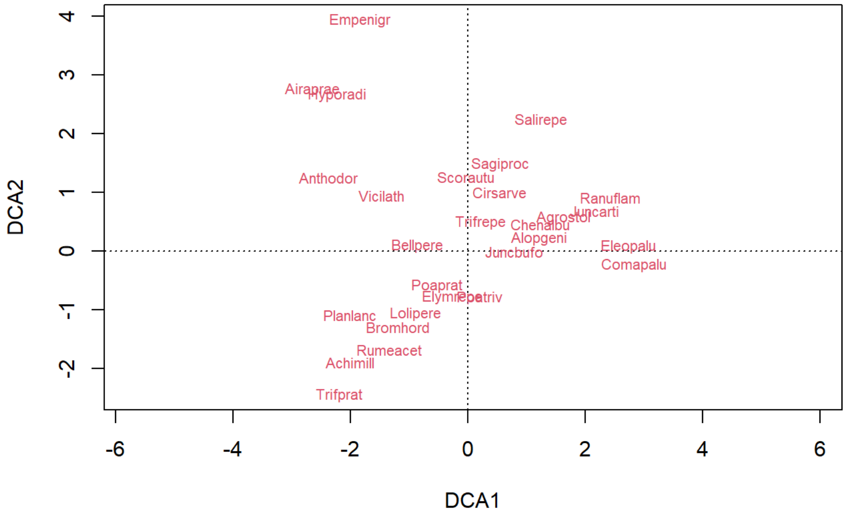
```
Eigenvalues          0.5392 0.3257 0.16889 0.19567
Additive Eigenvalues 0.5392 0.3175 0.15318 0.18878
Decorana values      0.5636 0.3194 0.07921 0.04138
Axis lengths         3.8264 2.8444 2.03949 2.17577
```

Detrended correspondence analysis (DCA)

```
ordiplot (DCA, display = 'sites', type = 'p')
```



```
ordiplot (DCA, display = 'species', type = 't')
```



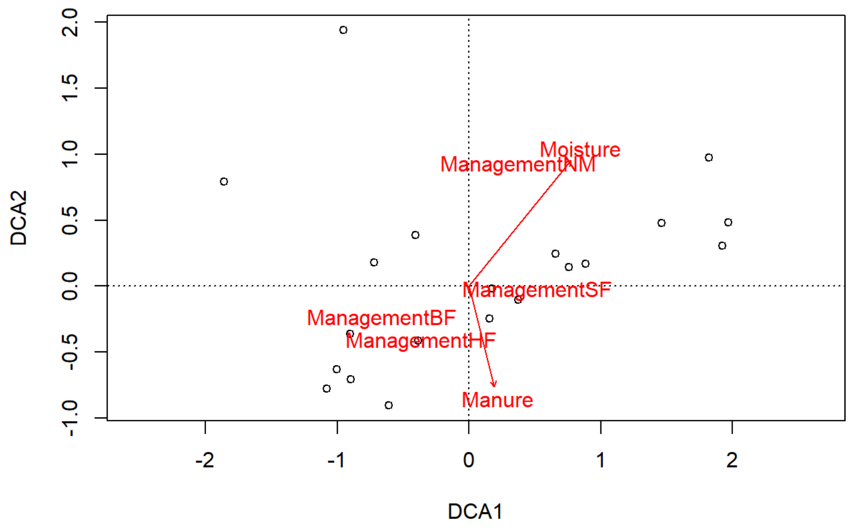
Triplot

```
fit <- envfit(DCA, dune2_env, perm = 999)
scores(fit, "vectors")
```

```
          DCA1       DCA2
A1   0.5149060 0.01196127
```

```
Moisture 0.5571229  0.68949173
Manure   0.1388042 -0.55260255
```

```
        plot(DCA,dis="site")
        plot(fit, p.max = 0.05, col = "red")
```



*N*on-metric *M*ulti*d*imensional *S*caling (NMDS)

- It uses an iterative optimization algorithm to find the best representation of distances in reduced space.
- NMDS is not an eigenanalysis. This has three important consequences:
    - There is no unique ordination result
    - The axes of the ordination are not ordered according to the variance they explain
    - The number of dimensions of the low-dimensional space must be specified before running the analysis

Non-metric Multidimensional Scaling (NMDS)

- The lower the stress value (a measure of goodness-of-fit), the better the representation of objects in the ordination-space is.
- `distance` specifies the distance metric to use
- `k` specifies the number of dimensions.

Non-metric Multidimensional Scaling (NMDS)

Methodology of NMDS:

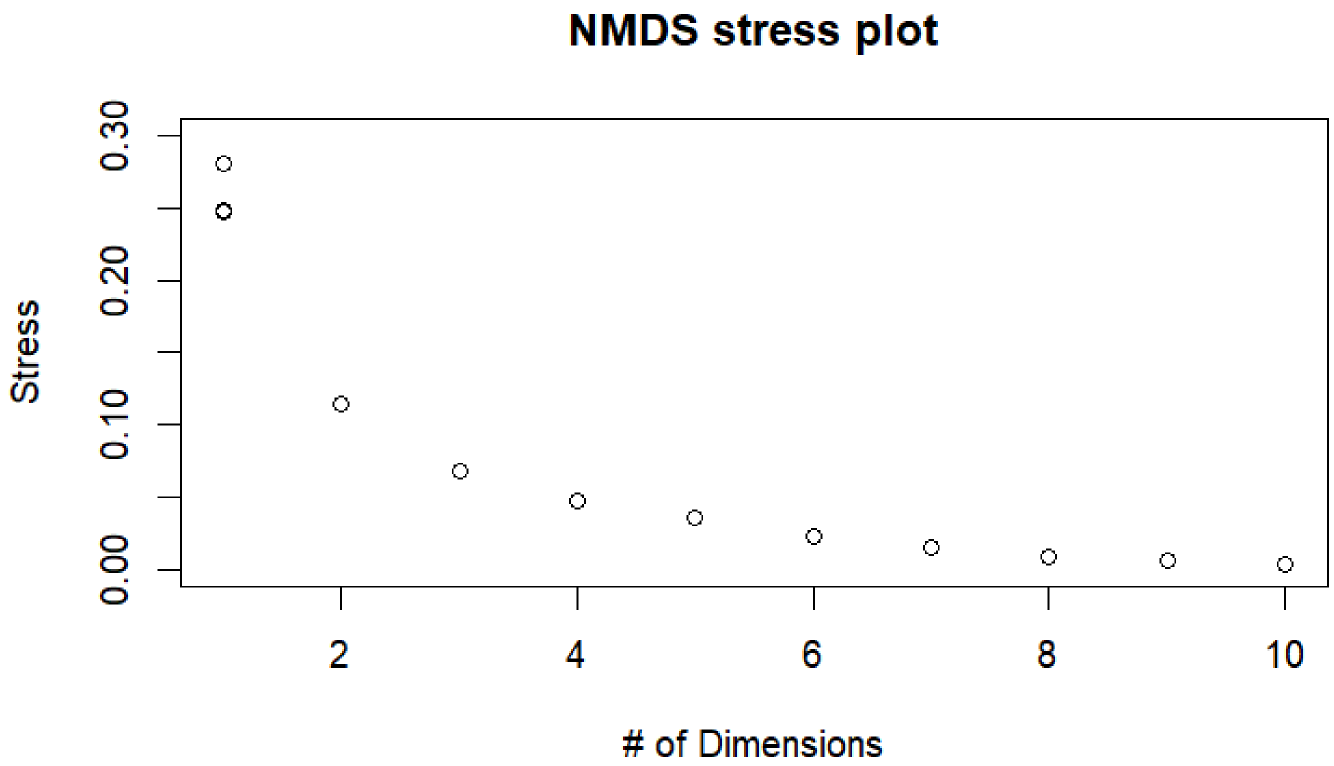
Step 1: Perform NMDS with 1 to 10 dimensions Step 2: Check the stress vs dimension plot Step 3: Choose optimal number of dimensions Step 4: Perform final NMDS with that number of dimensions Step 5: Check for convergent solution and final stress

Non-metric Multidimensional Scaling (NMDS)

```
        # First step is to calculate a distance matrix. See PCOA for more information about the distance measures
        # Here we use bray-curtis distance, which is recommended for abundance data
        dist <- vegdist(dune2_spe,  method = "bray")

        # In this part, we define a function NMDS.scree() that automatically
        # performs a NMDS for 1-10 dimensions and plots the nr of dimensions vs the stress
        NMDS.scree <- function(x) { #where x is the name of the data frame variable
          plot(rep(1, 10), replicate(10, metaMDS(x, autotransform = F, k = 1)$stress), xlim = c(1, 10),ylim = c(0, 0.30), xlab = "# of Dimensions", y
          for (i in 1:10) {
            points(rep(i + 1,10),replicate(10, metaMDS(x, autotransform = F, k = i + 1)$stress))
          }
```

```
}
NMDS.scree(dist)
```

Non-metric Multidimensional Scaling (NMDS)



Non-metric Multidimensional Scaling (NMDS)

```
# Because the final result depends on the initial
# random placement of the points
# we`ll set a seed to make the results reproducible
set.seed(2)

# Here, we perform the final analysis and check the result
NMDS1 <- metaMDS(dist, k = 3, trymax = 100, trace = F)
# Do you know what the trymax = 100 and trace = F means?
# Let's check the results
NMDS1
```

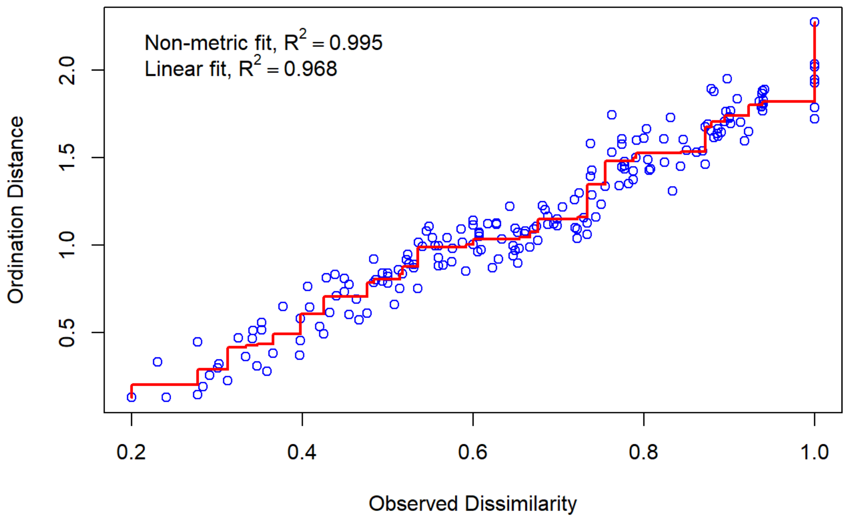
```
Call:
metaMDS(comm = dist, k = 3, trymax = 100, trace = F)

global Multidimensional Scaling using monoMDS

Data:     dist
Distance: bray

Dimensions: 3
Stress:     0.06826238
Stress type 1, weak ties
Best solution was repeated 5 times in 20 tries
The best solution was from try 9 (random start)
Scaling: centring, PC rotation, halfchange scaling
Species: scores missing
```

```
# If you don`t provide a dissimilarity matrix, metaMDS automatically applies Bray-Curtis. So in our case, the results would have to be the sa
NMDS2 <- metaMDS(dune2_spe, k = 2, trymax = 100, trace = F)
NMDS2
```

```
Call:
metaMDS(comm = dune2_spe, k = 2, trymax = 100, trace = F)

global Multidimensional Scaling using monoMDS

Data:     dune2_spe
Distance: bray

Dimensions: 2
Stress:     0.1149964
Stress type 1, weak ties
Best solution was repeated 10 times in 20 tries
The best solution was from try 10 (random start)
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on 'dune2_spe'
```

Non-metric Multidimensional Scaling (NMDS)

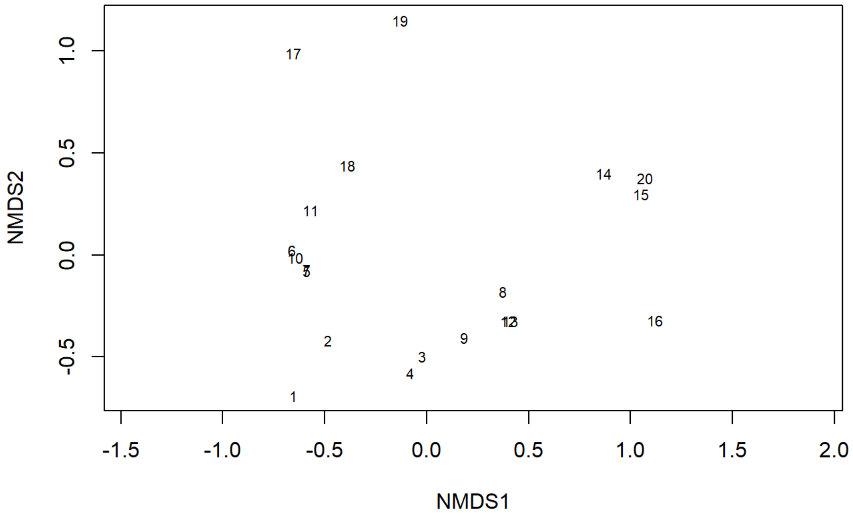
```
stressplot(NMDS1)
```



Non-metric Multidimensional Scaling (NMDS)

```
plot(NMDS1, type = "t")
```

```
species scores not available
```

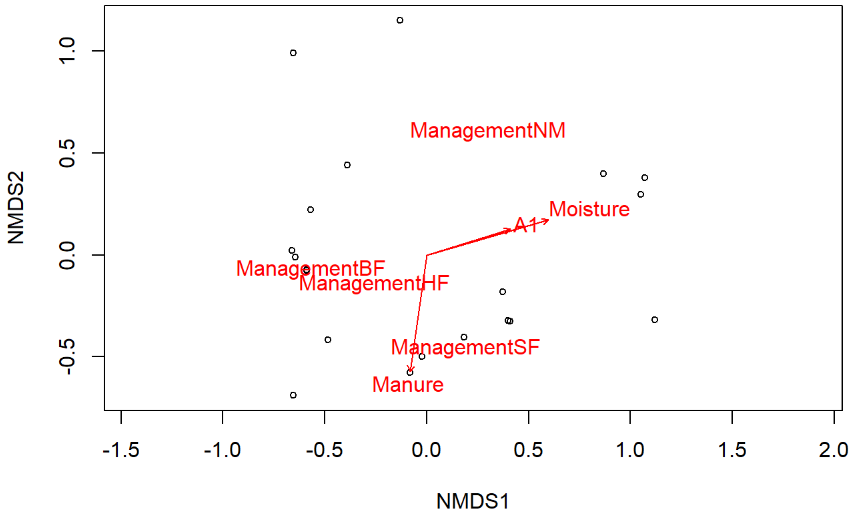


#Triplot

```
fit <- envfit(NMDS1, dune2_env, perm = 999)
scores(fit, "vectors")
```

```
              NMDS1      NMDS2
A1        0.5926574  0.1838978
Moisture  0.8639184  0.2488776
Manure   -0.1197459 -0.8307617
```

```
plot(NMDS1,dis="site")
plot(fit, p.max = 0.05, col = "red")
```



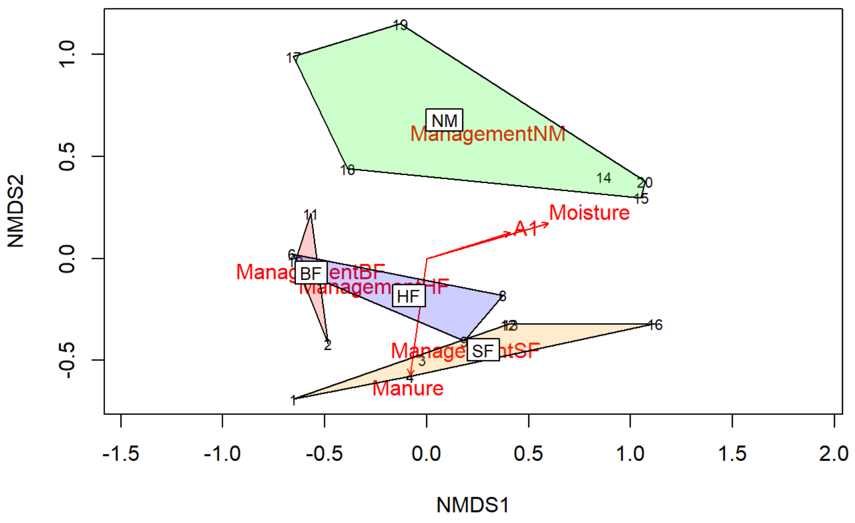
Ordihull

```
group_colors <- c("red", "blue", "green", "orange")

# Plot the NMDS1 ordination with no points plotted initially
plot(NMDS1, type="t")
plot(fit, p.max = 0.05, col = "red")
```

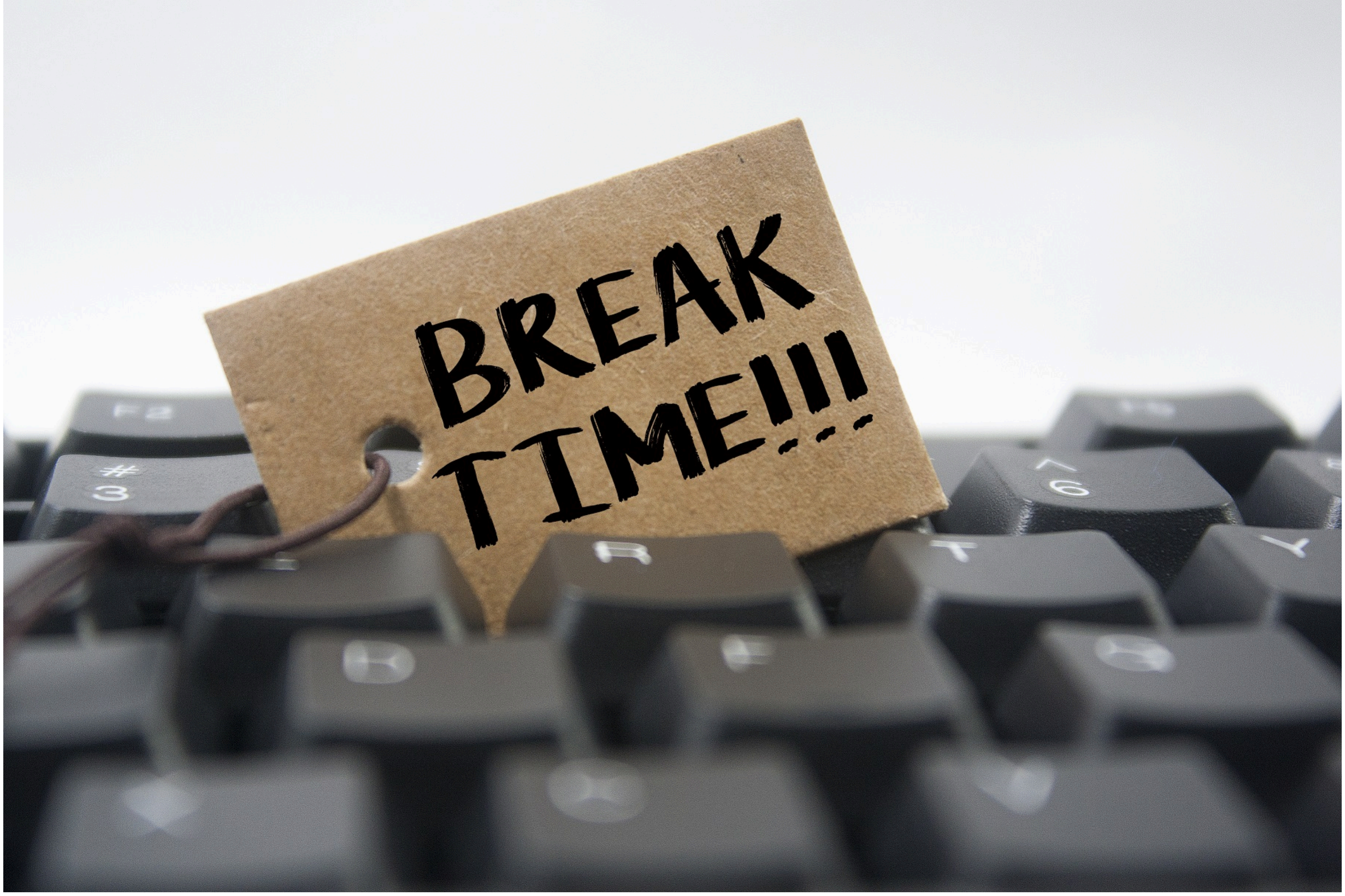
```
        # Add convex hulls around groups defined by the 'Management' variable,
        # and label the points with their corresponding group names
        with(dune2_env, ordihull(NMDS1, Management,
                                 draw = 'polygon',
                                  alpha = 50,
                                 label = TRUE,col = group_colors ))
```

Ordihull

```
species scores not available
```



BREAK


BREAK
TIME!!!