Ordination

- Principal component analysis (PCA)
- Factor analysis (FA)
- Correspondence analysis (CA)
- **Principal coordinate analysis (PCoA) or multidimensional scaling (MDS)**
- **Non-metric multidimensional scaling (NMDS)**
- **Redundancy analysis (RDA)**
- **Canonical correspondence analysis (CCA)**
- **Generalized Joint Attribute Modeling (GJAM)**

Principal coordinate analysis (PCoA)

Principal coordinate analysis (PCoA)

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

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

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

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

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

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

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

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

vegdist {vegan}

euclidean *d[jk] = sqrt(sum(x[ij]-x[ik])^2)*

binary: *sqrt(A+B-2*J)*

manhattan *d[jk] = sum(abs(x[ij] - x[ik]))*

binary: *A+B-2*J*

gower *d[jk] = (1/M) sum(abs(x[ij]-x[ik])/(max(x[i])-min(x[i])))*

binary: *(A+B-2*J)/M*,

where *M* is the number of columns (excluding missing values)

altGower *d[jk] = (1/NZ) sum(abs(x[ij] - x[ik]))*

where *NZ* is the number of non-zero columns excluding double-zeros (Anderson et al. 2006).

binary: *(A+B-2*J)/(A+B-J)*

canberra *d[jk] = (1/NZ) sum ((x[ij]-x[ik])/(x[ij]+x[ik]))*

where *NZ* is the number of non-zero entries.

binary: *(A+B-2*J)/(A+B-J)*

bray *d[jk] = (sum abs(x[ij]-x[ik]))/(sum (x[ij]+x[ik]))*

binary: *(A+B-2*J)/(A+B)*

kulczynski *d[jk] 1 - 0.5*((sum min(x[ij],x[ik])/(sum x[ij]) + (sum min(x[ij],x[ik])/(sum x[ik]))*

binary: *1-(J/A + J/B)/2*

morisita *d[jk] = 1 - 2*sum(x[ij]*x[ik])/((lambda[j]+lambda[k]) * sum(x[ij])*sum(x[ik]))*, where

lambda[j] = sum(x[ij](x[ij]-1))/sum(x[ij])*sum(x[ij]-1)*

binary: cannot be calculated

horn Like morisita, but *lambda[j] = sum(x[ij]^2)/(sum(x[ij])^2)*

binary: *(A+B-2*J)/(A+B)*

binomial *d[jk] = sum(x[ij]*log(x[ij]/n[i]) + x[ik]*log(x[ik]/n[i]) - n[i]*log(1/2))/n[i]*,

where *n[i] = x[ij] + x[ik]*

binary: *log(2)*(A+B-2*J)*

cao *d[jk] = (1/S) * sum(log(n[i]/2) - (x[ij]*log(x[ik]) + x[ik]*log(x[ij]))/n[i])*,

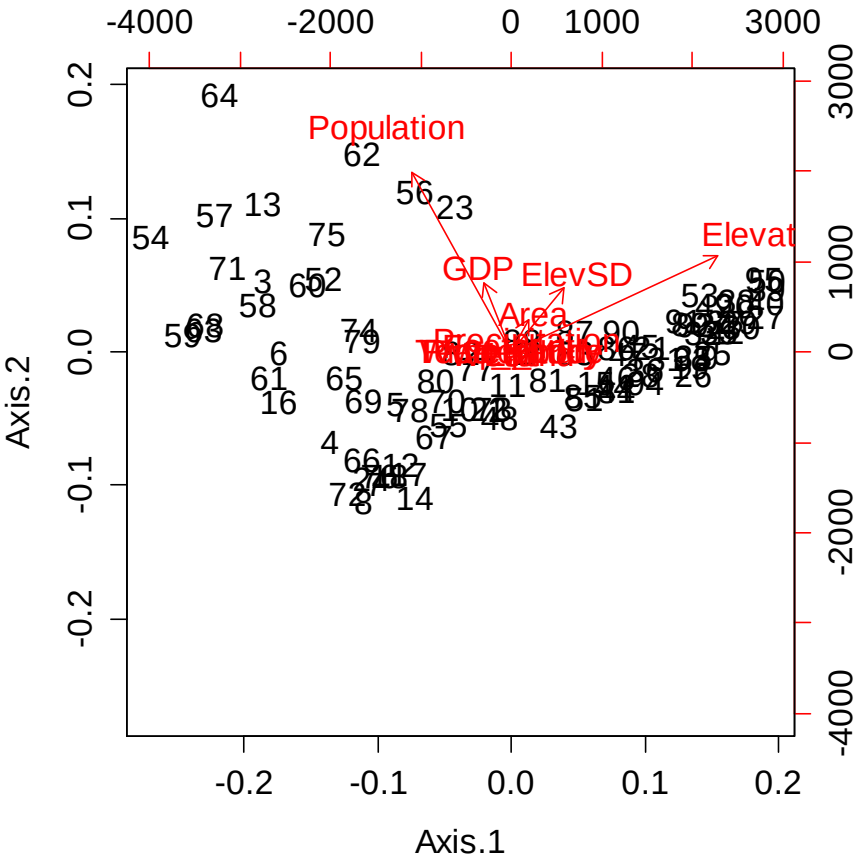
where *S* is the number of species in compared sites and *n[i] = x[ij] + x[ik]*

R code

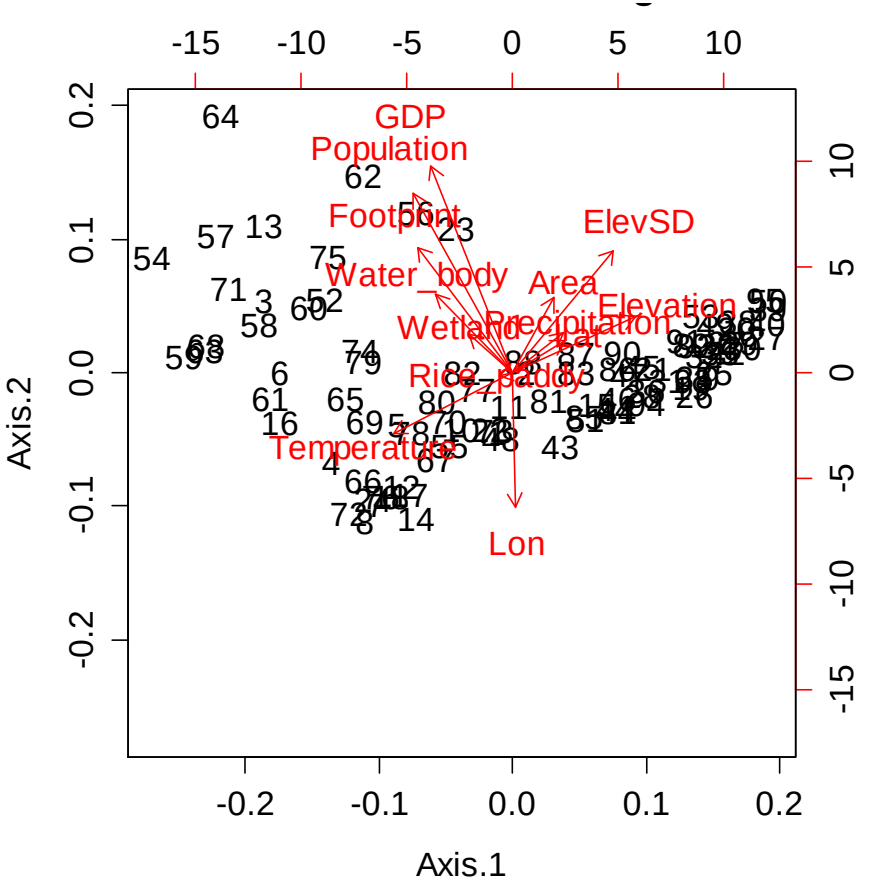
```
library(vegan)
library(ape)


# data standarization
cars = apply(mtcars, 2, scale, center=TRUE, scale=TRUE)

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

res <- pcoa(dis) # library(ape)
biplot(res, cars)
```

PCoA plot

Not standardized

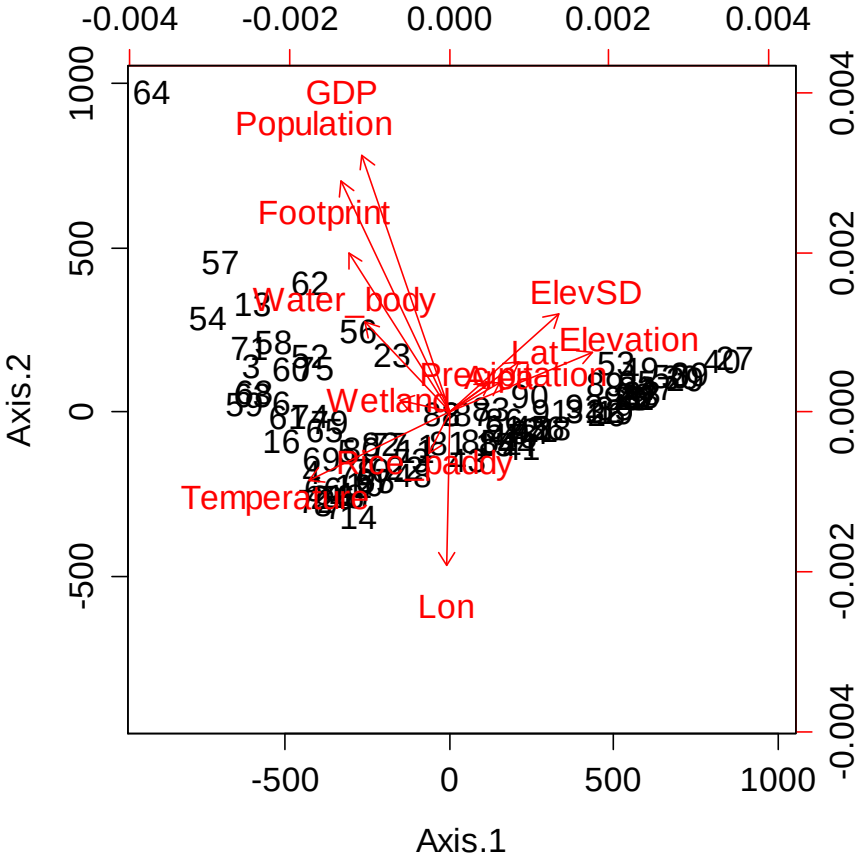


Standardized

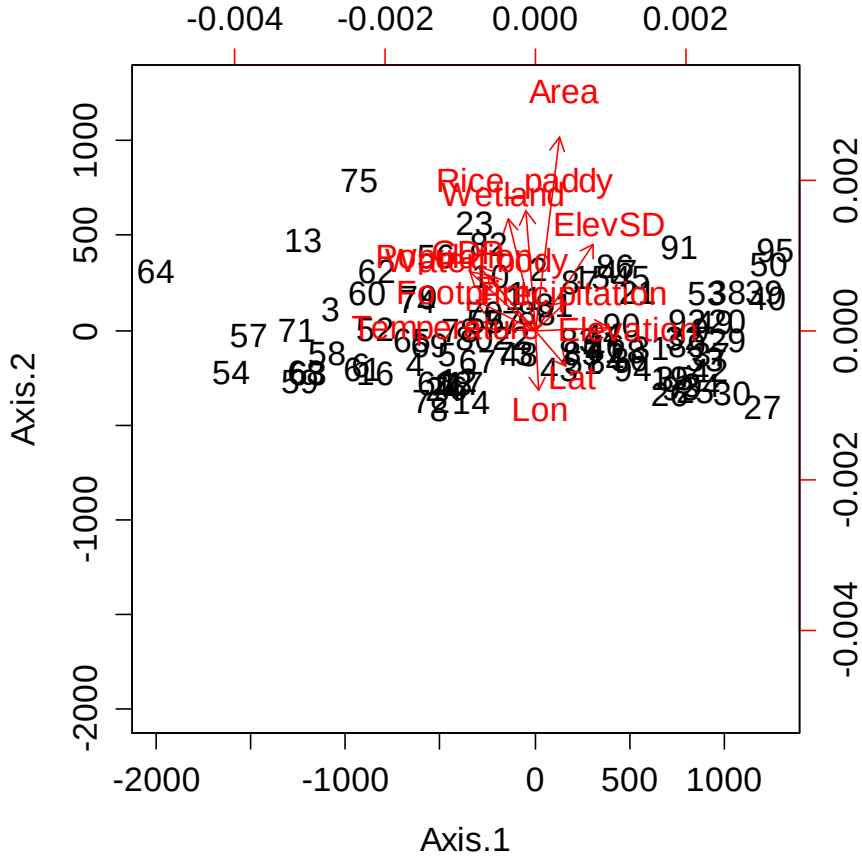


PCoA plot

Using Euclidean distance



Using Manhattan distance



Non-metric multidimensional scaling (NMDS)

Non-metric multidimensional scaling (NMDS)

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

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

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

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

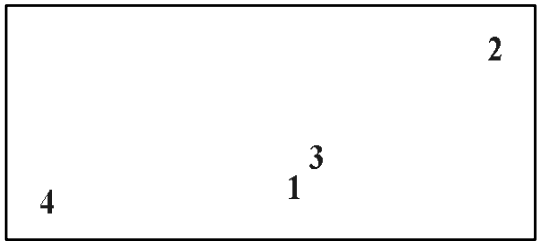
Bray-Curtis similarity (NMDS default)

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

	Species1	Species2	Species3	Species4	Species5	Species6
Site1	0	1.3	1.8	1.7	1.2	0
Site2	0	0	0	3.5	4.3	0
Site3	0	0	2.5	1.9	3.4	0
Site4	1.7	2.1	1.7	0	0	0

	Site1	Site2	Site3	Site4
Site1	100	42.6	68.1	52.5
Site2	42.6	100	67.9	0
Site3	68.1	67.9	100	25.6
Site4	52.5	0	25.6	100

	Site1	Site2	Site3	Site4
Site1	0	4	1	3
Site2	4	0	2	0
Site3	1	2	0	5
Site4	3	6	5	0



Bray-Curtis similarity (NMDS default)

- It is invariant to changes in units
- It is unaffected by additions/removals of species that are not present in two communities
- It is unaffected by the addition of a new community
- It can recognize differences in total abundances when relative abundances are the same

Algorithm of NMDS

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

Goodness-of-fit and STRESS

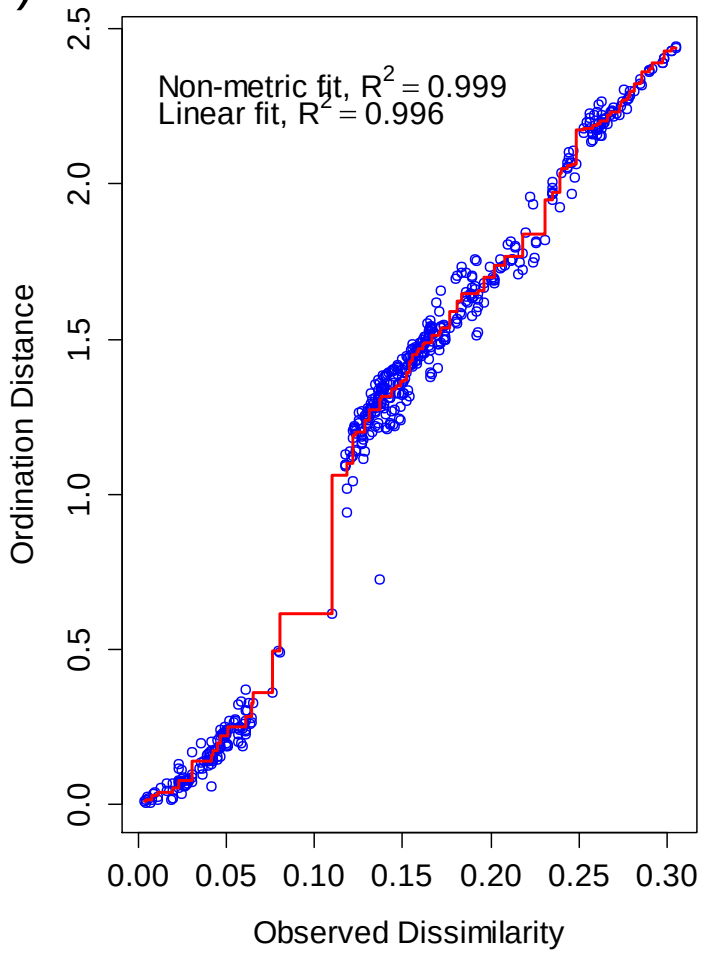
- The STRESS measure (STandardized REsiduals Sum of Squares) is a function of the original and derived distances to evaluate the goodness-of-fit of a MDS solution:

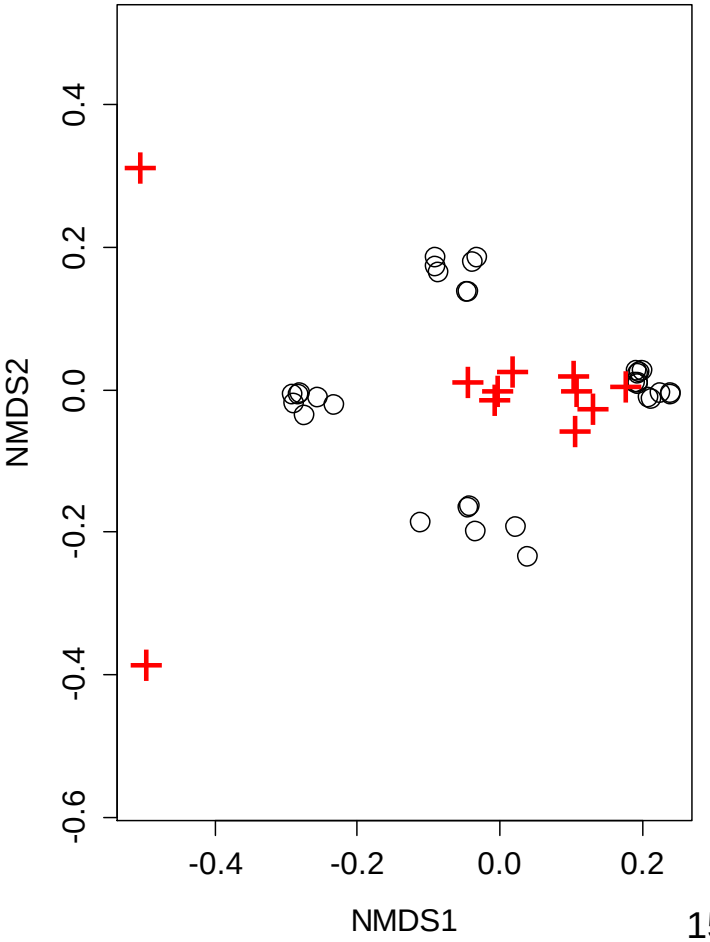
$$STRESS = \sqrt{\frac{\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}(d_{ij}-\hat{d}_{ij})^2}{\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}d_{ij}^2}}$$

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

R code

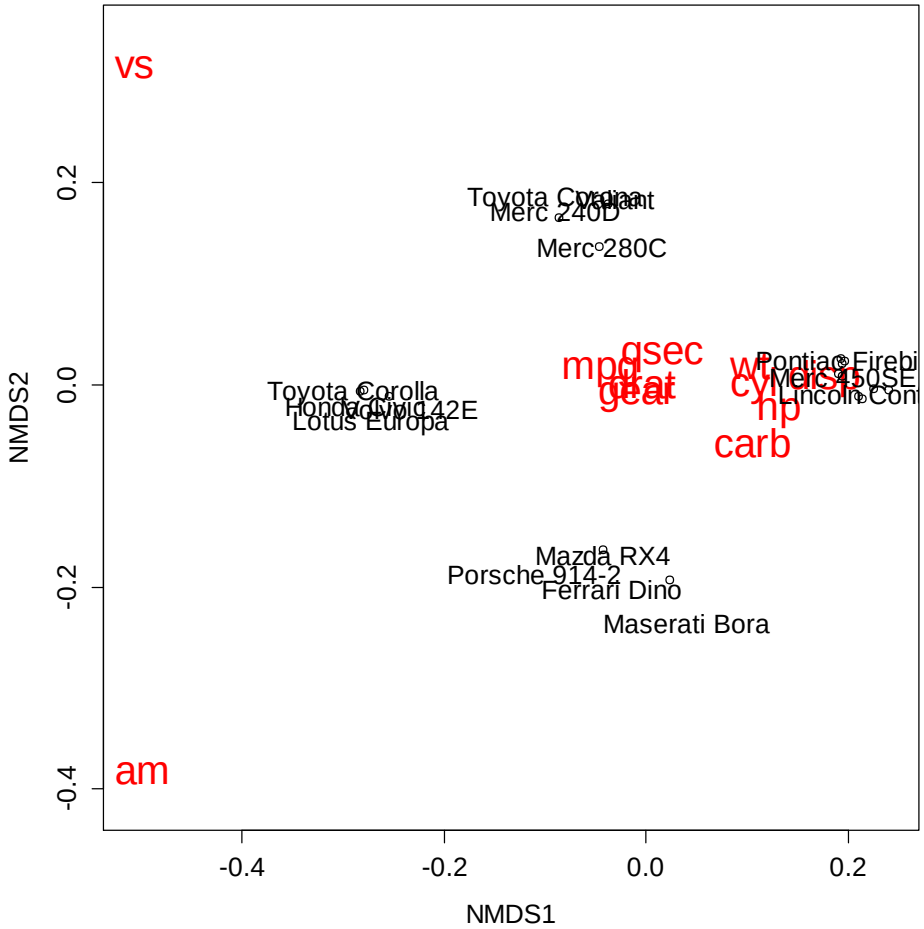
```
NMDS <- metaMDS (mtcars) # vegan
par (mfrow = c(1,2),mar=c(5,4,3,2))
stressplot (NMDS)
plot (NMDS)
```





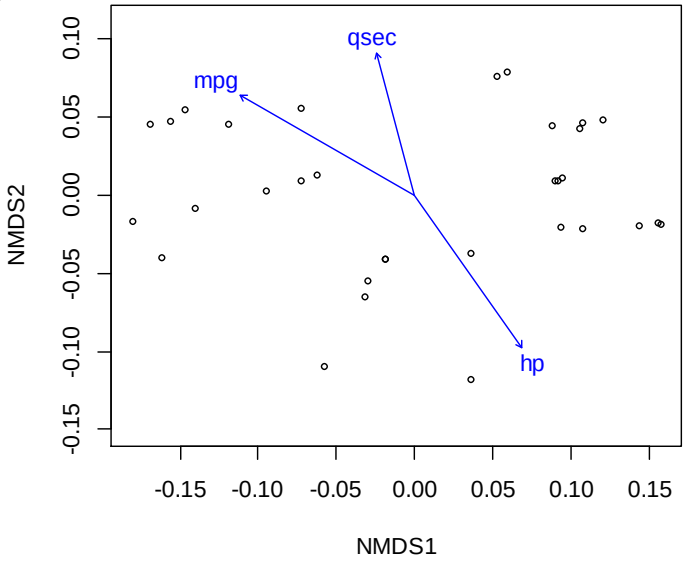
Plots

```
NMDS <- metaMDS (mtcars) # vegan
ordiplot(NMDS, type = "n")
orditorp(NMDS, display = "species", col = "red", air = 0.01, cex = 1.5) # column
orditorp(NMDS, display = "sites", cex = 1, air = 0.01) # row
```

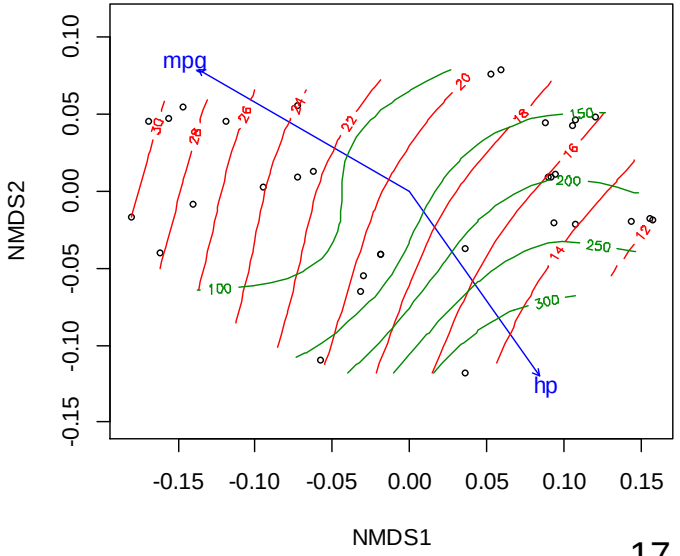


Fits Environmental variables onto an Ordination and Plot Smooth Surfaces of Variables

```
library(permute); library(vegan)
perf = mtcars[,c('mpg', 'hp', 'qsec')]
status = mtcars[,c('cyl', 'disp', 'drat', 'wt', 'gear', 'carb')]

vare.mds<- metaMDS(status)
# Fits an Environmental Vector or Factor onto an Ordination
ef <- envfit(vare.mds, perf, permu = 999)
plot(vare.mds, display = "sites")
plot(ef, p.max = 0.05)
```

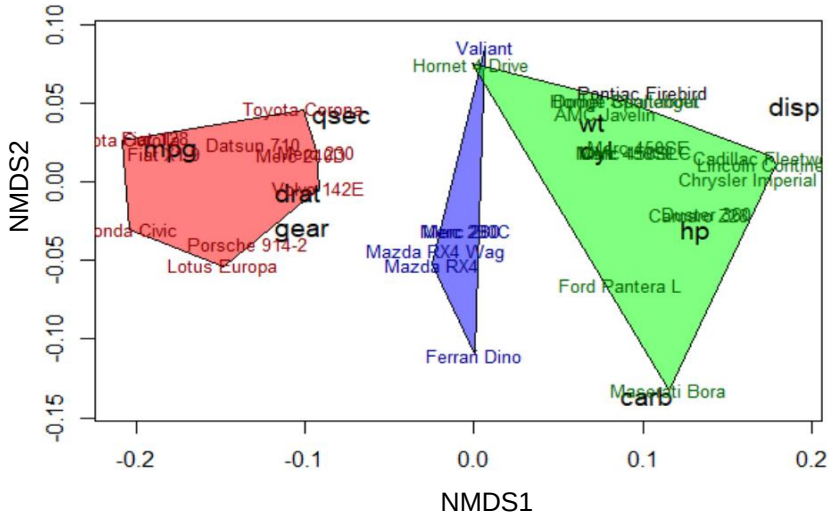



```
ef <- envfit(vare.mds ~ mpg + hp, perf)
plot(vare.mds, display = "sites")
plot(ef) # display mpg and hp
# Fit and Plot Smooth Surfaces of Variables on Ordination
tmp <- with(perf, ordisurf(vare.mds, mpg, add = TRUE))
with(perf, ordisurf(vare.mds, hp, add = TRUE, col = "green4"))
```



Grouping species

```
library(vegan)
mtcars = mtcars[order(mtcars$cyl), ]
mtcars = mtcars[, c('mpg','cyl','disp','hp','drat','wt', 'qsec','gear', 'carb')] # remove vs & am
table(mtcars$cyl) # 11, 6, 14

NMDS = metaMDS(mtcars, k=3) # cyl = 4, 6, 8
treat = c(rep("Treatment1",11), rep("Treatment2",6), rep("Treatment3",14))
colors = c(rep("red",11), rep("blue",6), rep("green",14))

ordiplot(NMDS, type = "n")
for(i in unique(treat)) {
    ordihull(NMDS$point[grep(i, treat),], draw = "polygon", # Plot convex hulls with colors based on treatment
            groups = treat[treat==i], col = colors[grep(i, treat)], label = F) }
orditorp(NMDS, display = "species", col = "black", air = 0.01, cex = 1.2)
colors = c(rep("darkred",11), rep("darkblue",6), rep("darkgreen",14))
orditorp(NMDS, display = "sites", col = colors, air = 0.01, cex = 0.8)
```



STRESS and number of dimensions

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

Redundancy analysis (RA)

Redundancy analysis (RDA)

Examines how much of the variation in one set of variables (X) explains the variation in another set of variables (Y)

Summarizes linear relationships between components of response variables (Y) that are "redundant" with (i.e. "explained" by) a set of explanatory variables (X).

Based on similar principles as principal components analysis and thus makes similar assumptions about the data.
Be appropriate when the expected relationship between X and Y variables is linear.

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

Redundancy analysis (RDA)

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

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

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

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

Redundancy Analysis

y_1 <=> y_2 Correlation Analysis

x => y Simple Regression Analysis

X => y Multiple Regression Analysis

(**X**={x_1,x_2,...})

$\mathbf{Y}_1$ <=> $\mathbf{Y}_2$ Canonical Correlation Analysis

X => **Y** Redundancy Analysis

*How one set of variables (**X**) may explain another set (**Y**)*

Redundancy analysis

The **total variance** of the data set, partitioned into constrained and unconstrained variances, is a standard result. This result shows how much variation in your response variables was redundant with the variation in your explanatory variables.

If the **constrained variance** is much higher than your **unconstrained variance**, the analysis suggests that much of the variation in the response data may be accounted for by your explanatory variables. If, however, there is a large proportion of unconstrained variation (i.e. variation in your response matrix that is non-redundant with the variation in the explanatory matrix), then the results should be interpreted with caution as only a small amount of the variation in your response matrix is displayed.

Information concerning a number of constrained axes (RDA axes) and unconstrained axes (PCA axes) are often presented in the results of an RDA.

R code

```
library(vegan) # rda() is in this library
X <- matrix(rnorm(120),ncol=5)
Y <- matrix(rnorm(120),ncol=5)
colnames(X) = c('X1','X2','X3','X4','X5')
X=data.frame(X, X6 = rep(1:3, each = 8))
colnames(Y) = c('Y1','Y2','Y3','Y4','Y5')
rda.results <- rda(X,Y) # X = Community data matrix; Y = Constraining matrix, typically of environmental variables
plot(rda(X,Y), scaling =1) # Distance triplot
plot(rda(X,Y), scaling =2) # Correlation triplot
# mtcars
stat = scale(mtcars[, c('cyl','disp', 'drat', 'wt', 'vs', 'am', 'gear', 'carb')])
perf = scale(mtcars[, c('mpg','hp', 'qsec')])
RDA <- rda(perf, stat)
stat = as.data.frame(stat); perf = as.data.frame(perf)
RDA <- rda(perf  ~., data= stat) #same

plot(RDA, scaling =1, main="Scaling1") # Distance triplot
plot(RDA, scaling =2, main="Scaling2") # Correlation triplot
plot(RDA) # Correlation triplot
```

Sores and loadings

goodness(RDA)

RDACCAu

	RDA1	RDA2	RDA3
Mazda RX4	0.0419	0.2147	0.2153
Mazda RX4 Wag	0.0268	0.1626	0.2515
Datsun 710	-0.2222	0.0687	-0.0624
Hornet 4 Drive	-0.0888	-0.1432	-0.1957
Hornet Sportabout	0.1247	0.0376	0.0704
Valiant	-0.1279	-0.2278	-0.0679
Duster 360	0.2153	0.0308	-0.0554
Merc 240D	-0.2059	-0.2106	-0.0800
Merc 230	-0.2123	-0.1992	-0.0660
Merc 280	-0.0354	-0.1719	-0.1714
Merc 280C	-0.0354	-0.1719	-0.1714
Merc 450SE	0.0633	-0.1463	0.3326
Merc 450SL	0.0835	-0.0769	0.2843
Merc 450SLC	0.0805	-0.0871	0.2914
Cadillac Fleetwood	0.2197	-0.2395	-0.1470
Lincoln Continental	0.1976	-0.2815	-0.0865
Chrysler Imperial	0.1811	-0.2725	-0.0369
Fiat 128	-0.2446	0.0793	0.0086
Honda Civic	-0.1758	0.2332	-0.1291
Toyota Corolla	-0.2319	0.1528	-0.0195
Toyota Corona	-0.2473	-0.1198	0.0306
Dodge Challenger	0.0885	-0.0222	0.2004
AMC Javelin	0.0752	-0.0023	0.2334
Camaro Z28	0.1827	-0.0146	0.0177
Pontiac Firebird	0.1374	-0.0181	0.0103
Fiat X1-9	-0.2287	0.1336	-0.0299
Porsche 914-2	-0.1323	0.2633	0.2737
Lotus Europa	-0.1243	0.2595	-0.2759
Ford Pantera L	0.2562	0.3310	-0.2478
Ferrari Dino	0.1339	0.2109	0.0692
Maserati Bora	0.3966	0.2259	-0.3420
Volvo 142E	-0.1922	0.0017	-0.1048

RDACCAv

	RDA1	RDA2	RDA3
mpg	-0.5483	0.6658	-0.5061
hp	0.6456	-0.0477	-0.7622
qsec	-0.5316	-0.7446	-0.4036

RDACCAwa

	RDA1	RDA2	RDA3
Mazda RX4	-0.0019	0.1829	0.6454
Mazda RX4 Wag	-0.0225	0.1223	0.5188
Datsun 710	-0.1211	0.0050	0.1975
Hornet 4 Drive	-0.1161	-0.1280	-0.0615
Hornet Sportabout	0.0792	0.0447	-0.0108
Valiant	-0.1135	-0.3060	0.0950
Duster 360	0.2538	0.0335	-0.1530
Merc 240D	-0.2265	-0.0938	0.0937
Merc 230	-0.2768	-0.4594	-0.7941
Merc 280	-0.0342	-0.0701	0.2363
Merc 280C	-0.0405	-0.1751	0.2183
Merc 450SE	0.0969	-0.0633	0.0409
Merc 450SL	0.0794	-0.0591	-0.0799
Merc 450SLC	0.0883	-0.1626	0.0061
Cadillac Fleetwood	0.1723	-0.3026	0.1358
Lincoln Continental	0.1898	-0.2871	0.0608
Chrysler Imperial	0.1736	-0.1232	-0.3768
Fiat 128	-0.2924	0.1922	-0.5029
Honda Civic	-0.2512	0.2401	0.0354
Toyota Corolla	-0.3263	0.1888	-0.7149
Toyota Corona	-0.1534	-0.1844	-0.0542
Dodge Challenger	0.0916	-0.0264	0.5699
AMC Javelin	0.0791	-0.0815	0.4979
Camaro Z28	0.2809	0.0513	0.0281
Pontiac Firebird	0.0724	0.0558	-0.0596
Fiat X1-9	-0.2140	0.1076	0.0542
Porsche 914-2	-0.0892	0.3038	0.3825
Lotus Europa	-0.1204	0.4043	-0.2769
Ford Pantera L	0.3084	0.2180	-0.1876
Ferrari Dino	0.1239	0.2377	0.2486
Maserati Bora	0.3965	0.1714	-0.9325
Volvo 142E	-0.0863	-0.0369	0.1394

coef(RDA)

RDACCAbiplot

	RDA1	RDA2	RDA3
cyl	0.9091	-0.2410	0.1970
disp	0.8311	-0.4033	0.0060
drat	-0.4905	0.5882	-0.2198
wt	0.6850	-0.6922	0.0417
vs	-0.8455	-0.1125	-0.4756
am	-0.2507	0.8413	-0.1414
gear	-0.1595	0.6997	-0.3419
carb	0.7823	0.1245	-0.1551

cor(RDACCAu[,1], RDACCAwa[,1]) # 0.96

spenvcor(RDA)

summary(RDA)

Call:
rda(formula = perf ~ stat)

Partitioning of variance:

	Inertia	Proportion
Total	3.0000	1.000
Constrained	2.6161	0.872
Unconstrained	0.3839	0.128

Importance of components:

	RDA1	RDA2	RDA3	PC1	PC2	PC3
Eigenvalue	2.1051	0.4788	0.03224	0.1875	0.10813	0.08833
Proportion Explained	0.7017	0.1596	0.01075	0.0625	0.03604	0.02944
Cumulative Proportion	0.7017	0.8613	0.87202	0.9345	0.97056	1.00000

Accumulated constrained eigenvalues
Importance of components:

	RDA1	RDA2	RDA3
Eigenvalue	2.1051	0.4788	0.03224
Proportion Explained	0.8047	0.1830	0.01232
Cumulative Proportion	0.8047	0.9877	1.00000

Species scores

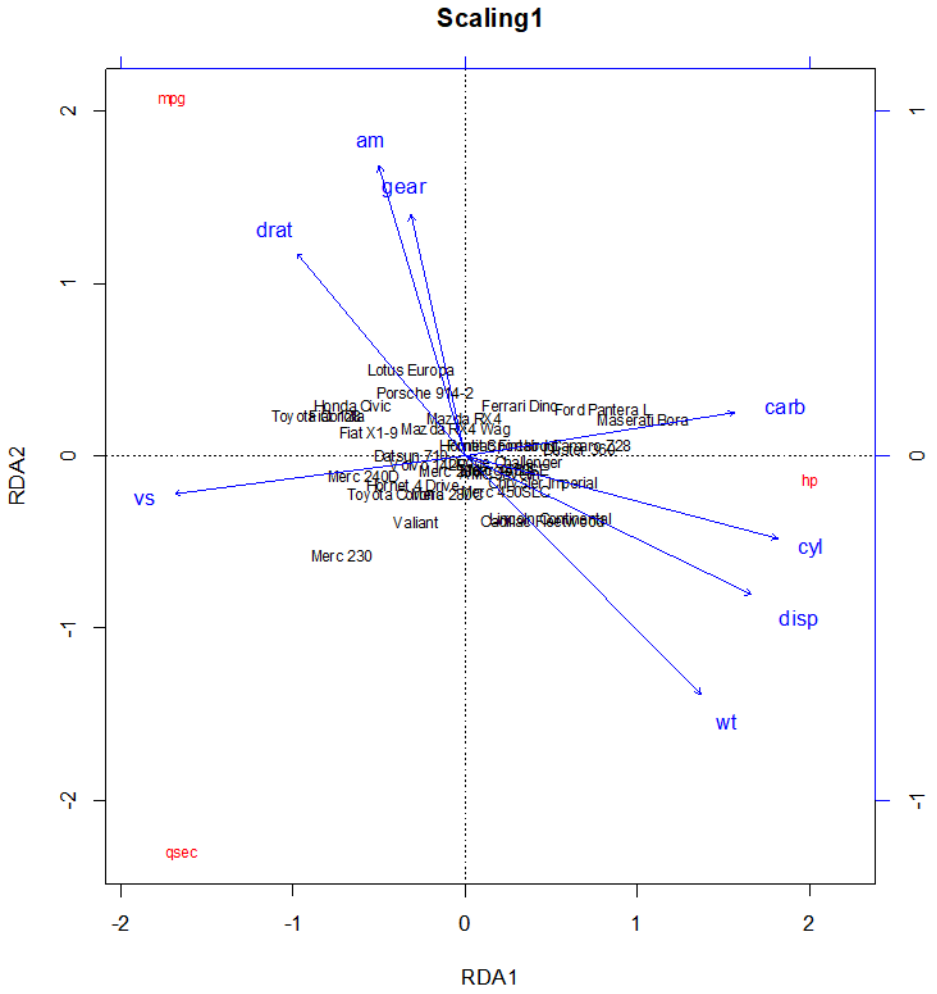
	RDA1	RDA2	RDA3	PC1	PC2	PC3
mpg	-1.426	0.82591	-0.1629	0.5756	-0.2574	0.27151
hp	1.679	-0.05916	-0.2454	-0.2603	0.2456	0.45026
qsec	-1.383	-0.92377	-0.1299	0.4512	0.4701	-0.08656

Site scores (weighted sums of species scores)

	RDA1	RDA2	RDA3	PC1	PC2	PC3
Mazda RX4	-0.005779	0.56804	2.00415	0.215960	-0.167977	-1.004973
Mazda RX4 Wag	-0.069821	0.37993	1.61123	0.340563	-0.008588	-0.805014
Datsun 710	-0.376140	0.01564	0.61320	-1.123391	0.308921	-0.060746
Hornet 4 Drive	-0.360562	-0.39735	-0.19107	0.209371	-0.216872	-0.310089

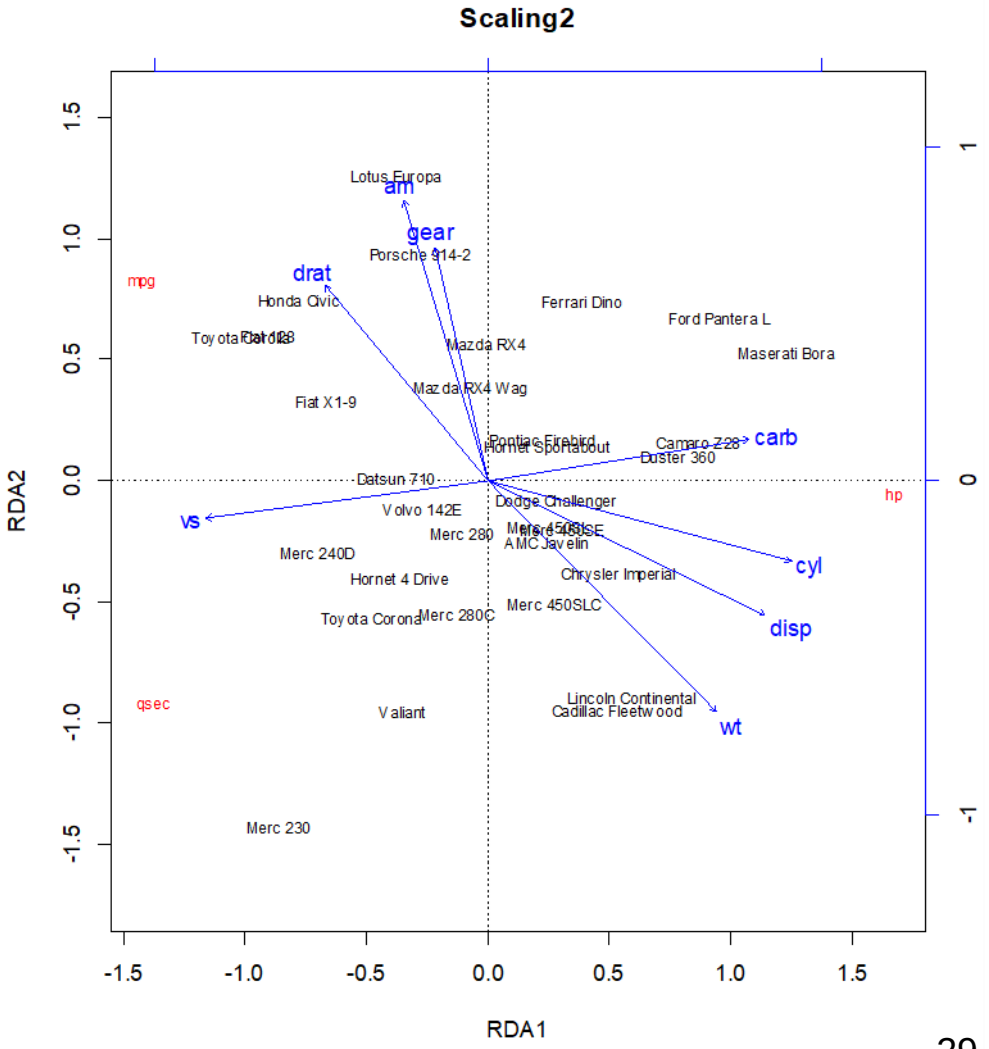
```
plot(RDA, scaling =1, main="Scaling1") # Distance triplot
```

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

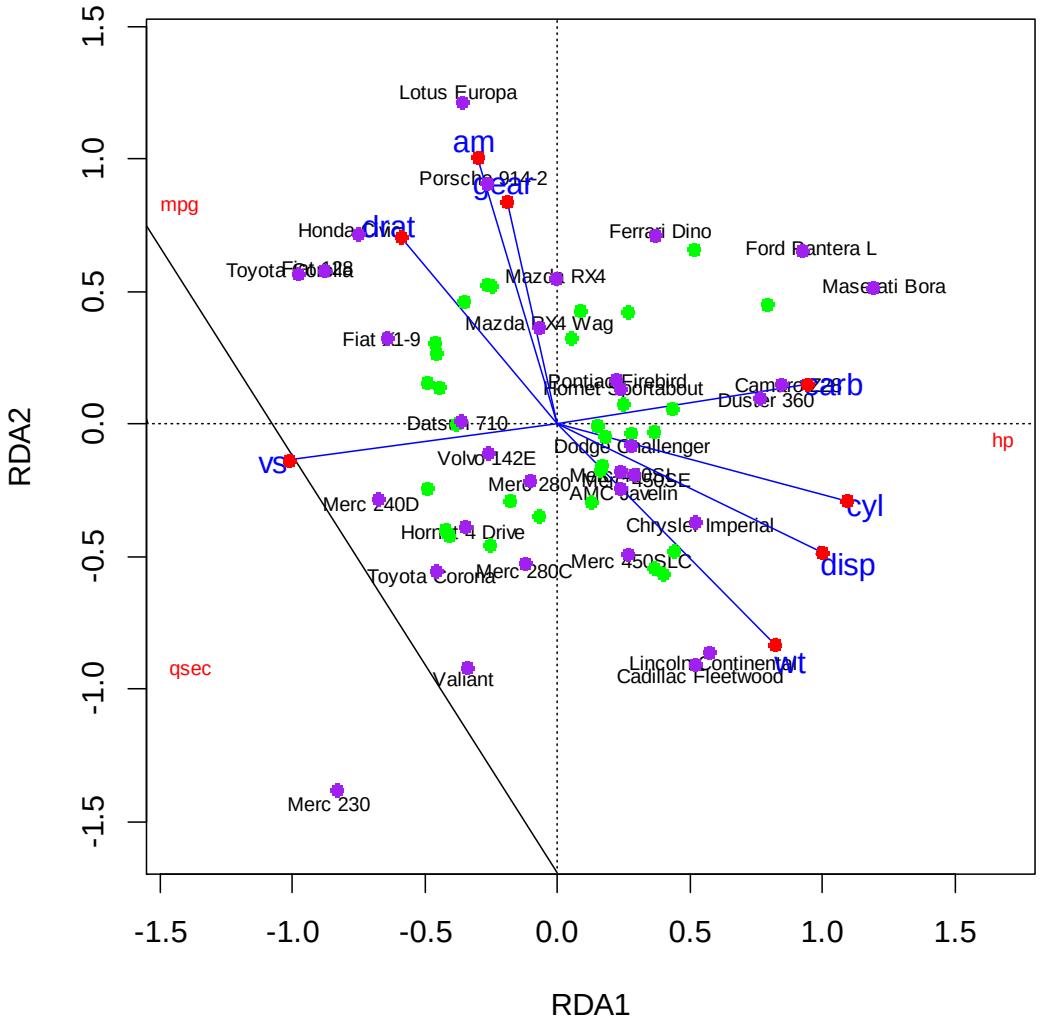


plot(RDA, perf), scaling =2, main="Scaling2") # Correlation triplot

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



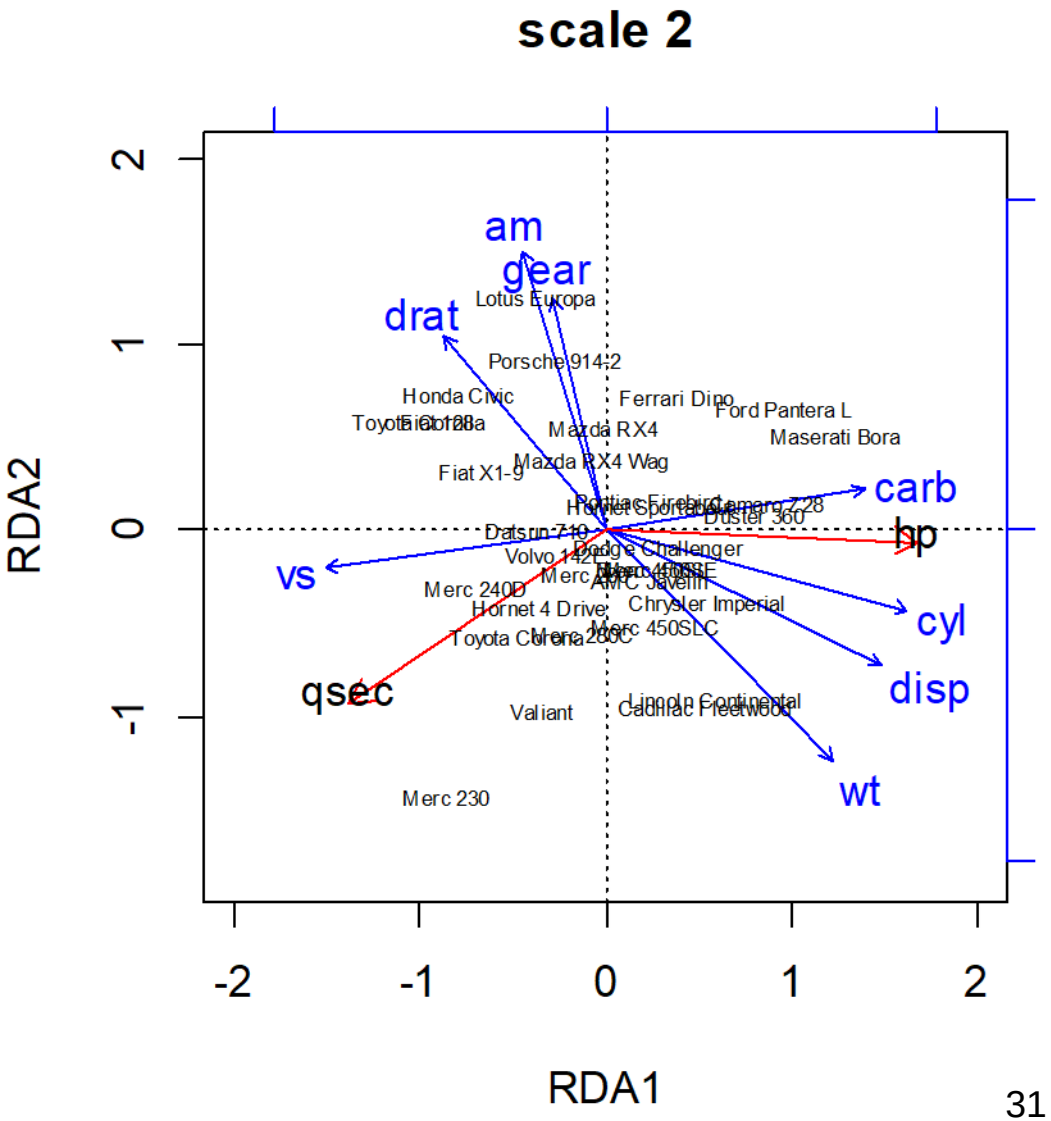
plot(RDA) uses environmental (not species) scores and loadings




```
plot(RDA, scaling =2, main="Scaling2") # Correlation triplot
points(RDA$CCA$u[, c(1,2)], col = "green", pch=16) # no match
points(RDA$CCA$wa[, c(1,2)]*3, col = "purple", pch=16) # match well
points(RDA$CCA$biplot[, c(1,2)]*1.2, col = "red",  pch=16) # match well
```

Plotting

```
plot(RDA, scaling=2, display = c("bp", "cn"),
      main = "scale 2", xlim = c(-2, 2))

spe.sc <- scores(RDA, choices=1:2, display="sp")
# Calculate importance
R2 = goodness(RDA, addprevious = TRUE)[,2]
#select 2 important species/perf
spe.sc=spe.sc[order(-R2),][1:2,]
arrows(0, 0, spe.sc[,1], spe.sc[,2], length=0.1,
         lty=1, col="red")
text(spe.sc[,1],
      spe.sc[,2]+0.05, row.names(spe.sc))

text(scores(RDA)$sites,
      row.names(scores(RDA)$sites), cex=.5)
```



Explained variance

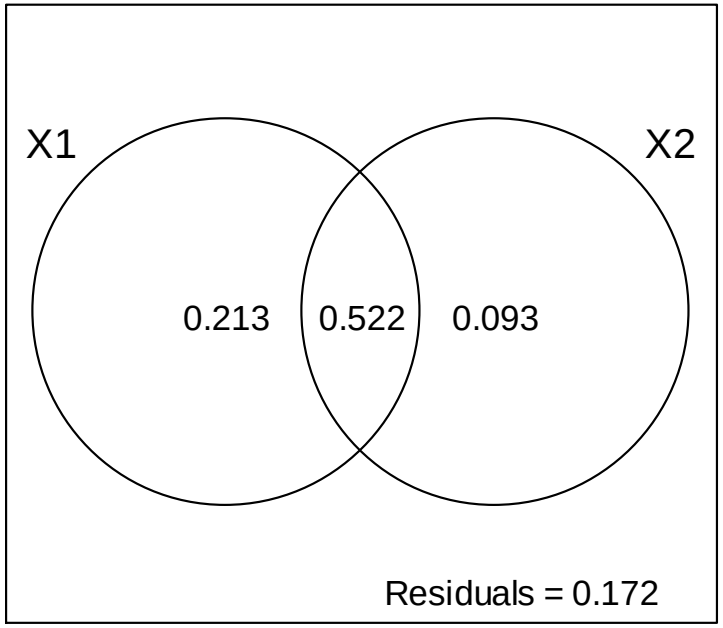
```
(R2 <- RsquareAdj(RDA)$r.squared)
# 0.872

(R2adj <- RsquareAdj(RDA)$adj.r.squared)
# 0.828
```

Partitioning of variance:

	Inertia	Proportion
Total	3.0000	1.000
Constrained	2.6161	0.872
Unconstrained	0.3839	0.128

Variance partitioning

```
ST1 = stat[,1:4]
ST2 = stat[,5:8]
(spe.part <- varpart(perf, ST1, ST2))
plot(spe.part, digits=2)
```



	mpg	hp	qsec	cyl	disp	drat	wt	vs	am	gear	carb
Mazda RX4	21	110	16.46	6	160	3.9	2.62	0	1	4	4
Mazda RX4 Wag	21	110	17.02	6	160	3.9	2.875	0	1	4	4
Datsun 710	22.8	93	18.61	4	108	3.85	2.32	1	1	4	1
Hornet 4 Drive	21.4	110	19.44	6	258	3.08	3.215	1	0	3	1
Hornet Sportabout	18.7	175	17.02	8	360	3.15	3.44	0	0	3	2
Valiant	18.1	105	20.22	6	225	2.76	3.46	1	0	3	1
Duster 360	14.3	245	15.84	8	360	3.21	3.57	0	0	3	4
Merc 240D	24.4	62	20	4	146.7	3.69	3.19	1	0	4	2
Merc 230	22.8	95	22.9	4	140.8	3.92	3.15	1	0	4	2
Merc 280	19.2	123	18.3	6	167.6	3.92	3.44	1	0	4	4
Merc 280C	17.8	123	18.9	6	167.6	3.92	3.44	1	0	4	4
Merc 450SE	16.4	180	17.4	8	275.8	3.07	4.07	0	0	3	3
Merc 450SL	17.3	180	17.6	8	275.8	3.07	3.73	0	0	3	3
Merc 450SLC	15.2	180	18	8	275.8	3.07	3.78	0	0	3	3
Cadillac Fleetwood	10.4	205	17.98	8	472	2.93	5.25	0	0	3	4
Lincoln Continental	10.4	215	17.82	8	460	3	5.424	0	0	3	4
Chrysler Imperial	14.7	230	17.42	8	440	3.23	5.345	0	0	3	4
Fiat 128	32.4	66	19.47	4	78.7	4.08	2.2	1	1	4	1
Honda Civic	30.4	52	18.52	4	75.7	4.93	1.615	1	1	4	2
Toyota Corolla	33.9	65	19.9	4	71.1	4.22	1.835	1	1	4	1
Toyota Corona	21.5	97	20.01	4	120.1	3.7	2.465	1	0	3	1
Dodge Challenger	15.5	150	16.87	8	318	2.76	3.52	0	0	3	2
AMC Javelin	15.2	150	17.3	8	304	3.15	3.435	0	0	3	2
Camaro Z28	13.3	245	15.41	8	350	3.73	3.84	0	0	3	4
Pontiac Firebird	19.2	175	17.05	8	400	3.08	3.845	0	0	3	2
Fiat X1-9	27.3	66	18.9	4	79	4.08	1.935	1	1	4	1
Porsche 914-2	26	91	16.7	4	120.3	4.43	2.14	0	1	5	2
Lotus Europa	30.4	113	16.9	4	95.1	3.77	1.513	1	1	5	2
Ford Pantera L	15.8	264	14.5	8	351	4.22	3.17	0	1	5	4
Ferrari Dino	19.7	175	15.5	6	145	3.62	2.77	0	1	5	6
Maserati Bora	15	335	14.6	8	301	3.54	3.57	0	1	5	8
Volvo 142E	21.4	109	18.6	4	121	4.11	2.78	1	1	4	2

Hierarchical partitioning

```
library(rdacca.hp)
rdacca.hp(perf, stat, method = "RDA", type = "R2")
```

	Unique	Average.share	Individual	I.perc(%)
cyl	0.0073	0.169	0.1763	20.22
disp	0.0266	0.1211	0.1477	16.94
drat	0.0009	0.0484	0.0493	5.65
wt	0.0279	0.0893	0.1172	13.44
vs	0.0088	0.1274	0.1362	15.62
am	0.0056	0.0511	0.0567	6.5
gear	0.0008	0.0435	0.0443	5.08
carb	0.0298	0.1145	0.1443	16.55

Significance test

```
anova.cca(RDA, by="axis", step=1000)
```

```
Model: rda(formula = perf ~ stat)
          Df  Variance      F    Pr(>F)
RDA1       1  2.10506 153.5143  0.001 ***
RDA2       1  0.47875  34.9137  0.001 ***
RDA3       1  0.03224   2.3512  0.088 .
Residual  28  0.38395
```

```
anova.cca(RDA, by="term", step=1000)
```

```
Model: rda(formula = perf ~ cyl + disp + drat + wt + vs + am + gear + carb, data = stat)
          Df  Variance      F    Pr(>F)
cyl        1  1.76872 105.9528  0.001 ***
disp       1  0.09526   5.7062  0.010 **
drat       1  0.21828  13.0755  0.001 ***
wt         1  0.16739  10.0271  0.002 **
vs         1  0.08333   4.9918  0.016 *
am         1  0.10314   6.1782  0.013 *
gear       1  0.09045   5.4181  0.008 **
carb       1  0.08950   5.3616  0.013 *
Residual  23  0.38395
```

Canonical correspondence analysis (CCA)

Canonical correspondence analysis (CCA)

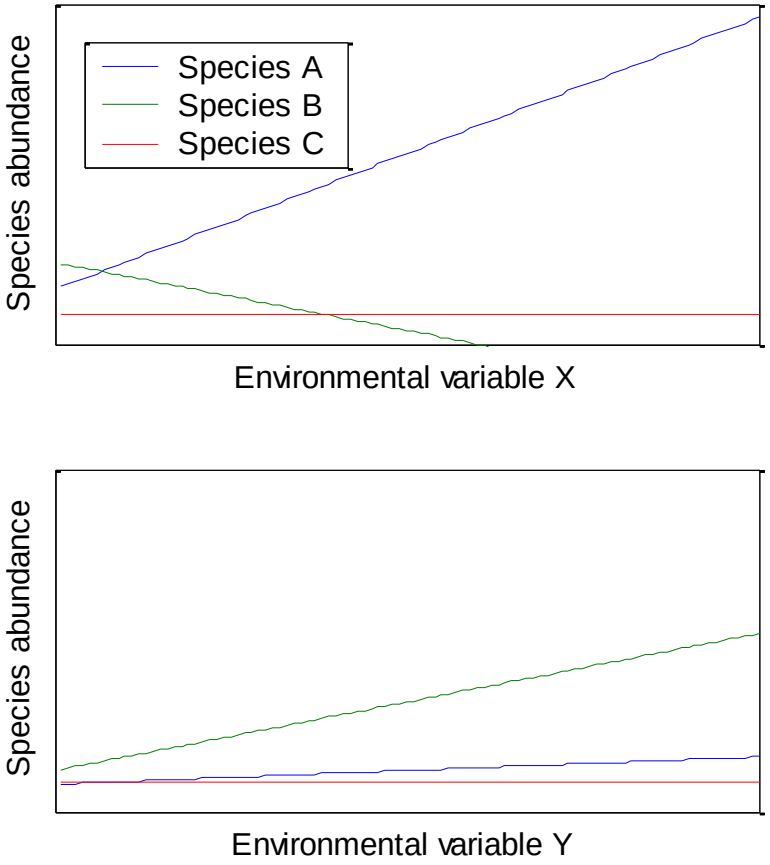
CCA is a multivariate constrained ordination technique that extracts major gradients among combinations of explanatory variables in a dataset.

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

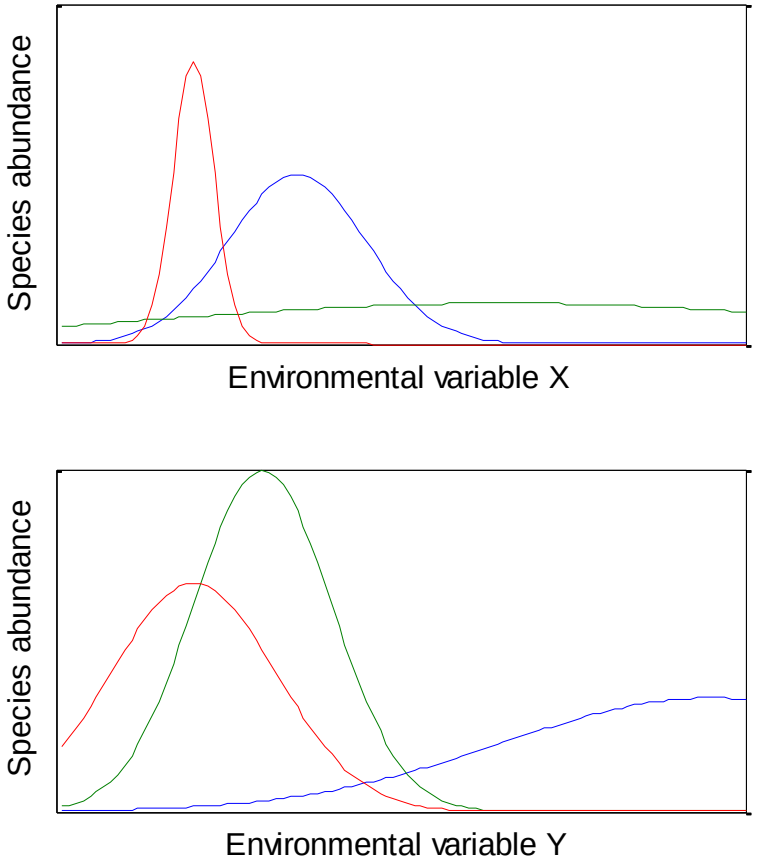
So **CCA** is a **CA** with the axes being linear combinations of the explanatory variables.

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

Canonical Correlation Analysis



Canonical Correspondence Analysis



Data of CCA

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

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

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

Calculation steps

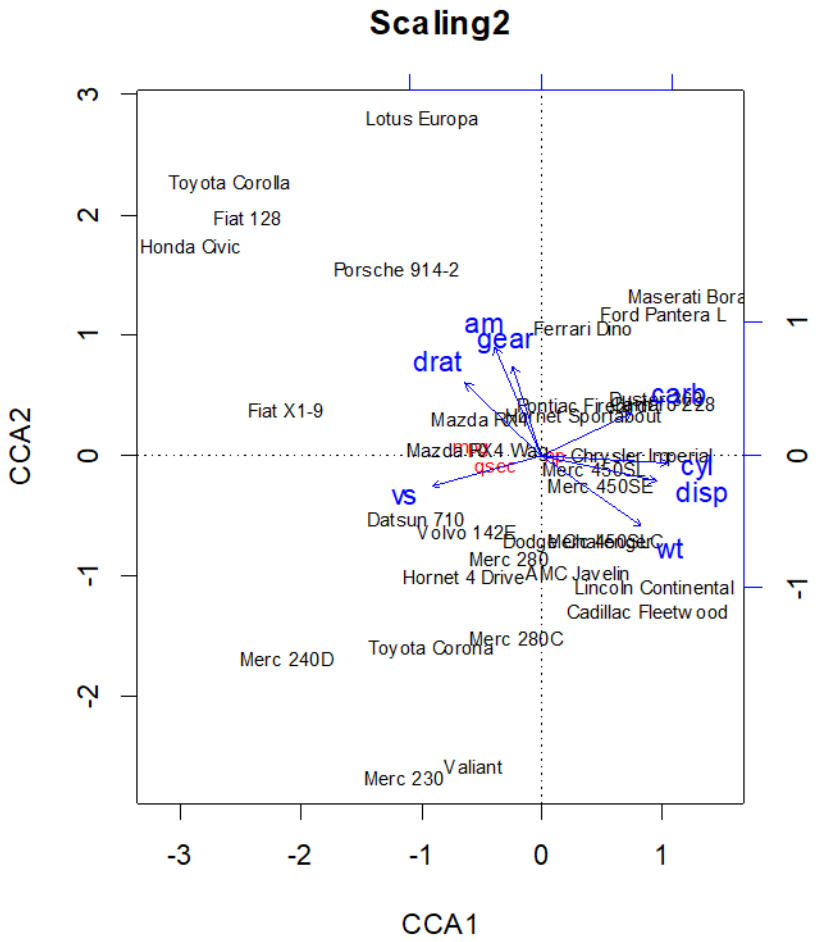
1. Start with a Chi-square species matrix [(actual - predicted)/sqrt(predicted)]
2. Regress the differences from expectation on environmental variables to get fitted values, using a weighted regression where total abundance by plots is used as the weights
3. Calculate the Euclidean distance of the fitted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

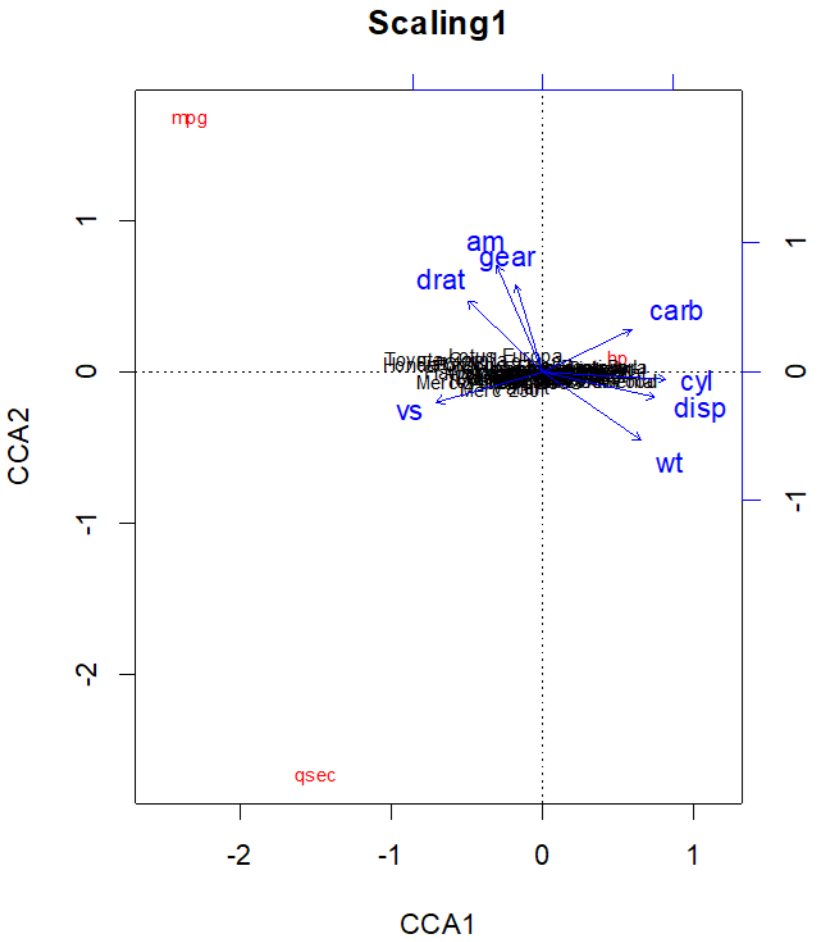
R code

```
library(vegan)
stat = mtcars[, c('cyl','disp', 'drat', 'wt', 'vs', 'am', 'gear', 'carb')]
perf = mtcars[, c('mpg','hp', 'qsec')]
cca.cars <- cca(perf, stat)

# the total variation (inertia) in the data is: 0.07
round(cca.cars $tot.chi, 2)

# the sum of all canonical eigenvalues (Constrained inertia): 0.06
round(cca.cars$CCA$tot.chi, 2)

# all explanatory variables explain 88% of the total variation in the data
cat(round(cca.cars$CCA$tot.chi
      /cca.cars$tot.chi*100), "% of data", "\n")

# the first two (canonical) eigenvalues are: 0.06, 0.0
round(cca.cars$CCA$eig[1:2], 2)

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

# but this is (the first two canonical axes explain) 88% of the total variation in the data
cat(round(sum(cca. cars$CCA$eig[1:2])
      /cca. cars$tot.chi*100), "%", "\n")
```

Triplot

```
plot(cca.cars, scaling = 2, main="Scaling2") # status scores are scaled by eigenvalues
plot(cca.cars, scaling = 1, main="Scaling1") # performance scores are scaled by eigenvalues
```





Triplot

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

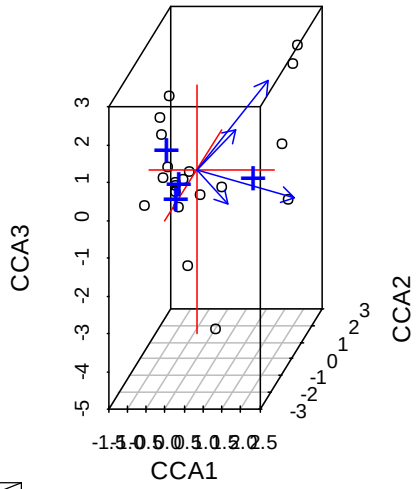
Again the position of a species represents the optimum value in terms of the Gausian response model (niche) along the first and second axes. For this reason, species scores are represented as labels or points.

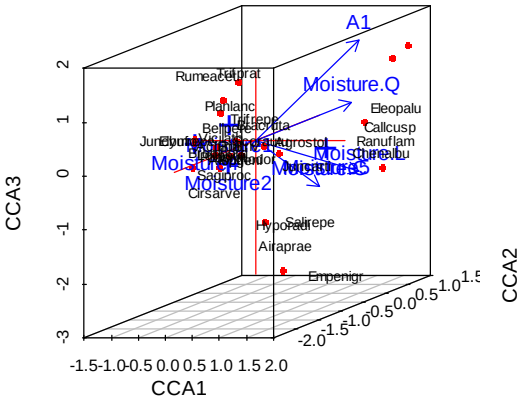
In addition: Species can be projected perpendicular (=orthogonally) on the lines showing the species optima of the respective explanatory variables (in the respective scaling). Projecting sites perpendicular on the lines results in the values of the respective environmental variable at those sites.

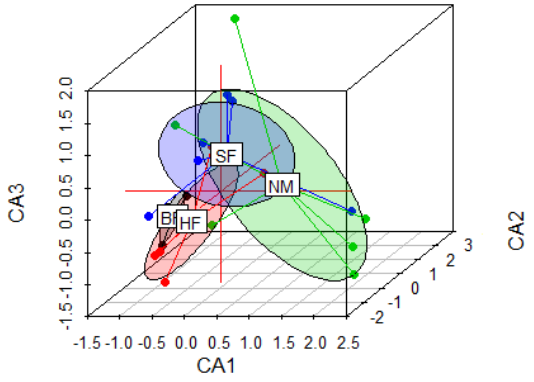
The angle between lines does NOT represent correlation between the variables. Instead if the tip of a line is projected on another line or an axis then the resulting value represents a weighted correlation.

3D plot

```
library(vegan3d)
data(dune, dune.env)
ord <- cca(dune ~ A1 + Moisture, dune.env)
ordiplot3d(ord)
### A boxed 'pin' version
ordiplot3d(ord, type = "h")
### More user control
pl <- ordiplot3d(ord, scaling = "symmetric", angle=15, type="n")
points(pl, "points", pch=16, col="red", cex = 0.7)
### identify(pl, "arrows", col="blue") would put labels in better positions
text(pl, "arrows", col="blue", pos=3)
text(pl, "centroids", col="blue", pos=1, cex = 1)
### Add species using xyz.convert function returned by ordiplot3d
sp <- scores(ord, choices=1:3, display="species", scaling="symmetric")
text(pl$xyz.convert(sp), rownames(sp), cex=0.7, xpd=TRUE)
### Two ways of adding fitted variables to ordination plots
ord <- cca(dune)
ef <- envfit(ord ~ Moisture + A1, dune.env, choices = 1:3)
### 1. use argument 'envfit'
ordiplot3d(ord, envfit = ef)
### 2. use returned envfit.convert function for better user control
pl3 <- ordiplot3d(ord)
plot(pl3$envfit.convert(ef), at = pl3$origin)
### envfit.convert() also handles different 'choices' of axes
pl3 <- ordiplot3d(ord, choices = c(1,3,2))
plot(pl3$envfit.convert(ef), at = pl3$origin)
### vegan::ordiXXXX functions can add items to the plot
ord <- cca(dune)
pl4 <- with(dune.env, ordiplot3d(ord, col = Management, pch=16))
with(dune.env, ordiellipse(pl4, Management, draw = "poly", col = 1:4,
                           alpha = 60))
with(dune.env, ordispider(pl4, Management, col = 1:4, label = TRUE))
```







When to use PCA, CA, RDA or CCA

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

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

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

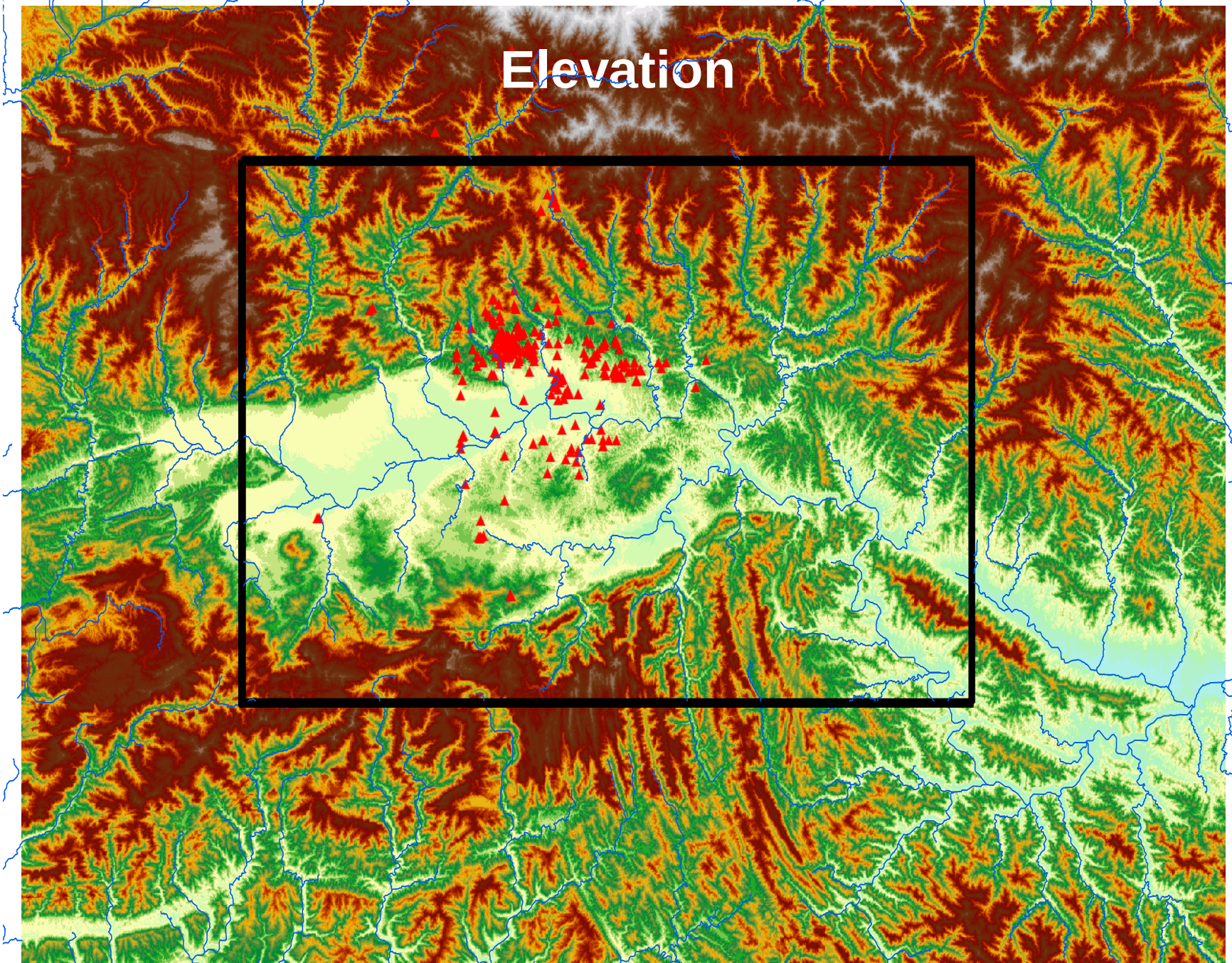
Generalized Joint Attribute Modeling

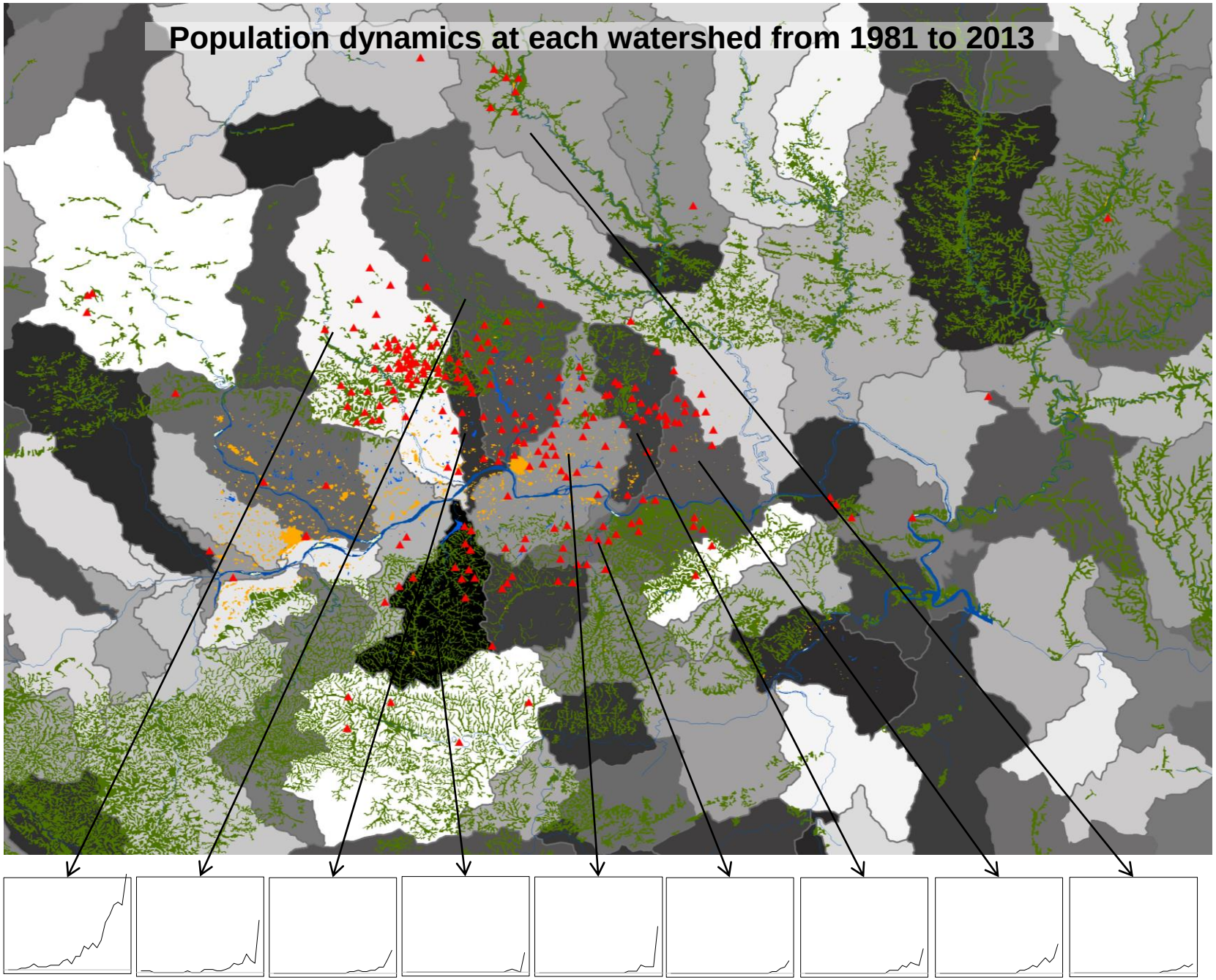
Clark, J.S., D. Nemergut, B. Seyednasrollah, P. Turner, and S. Zhang. 2017. Generalized joint attribute modeling for biodiversity analysis: Median-zero, multivariate, multifarious data, Ecological Monographs, 87, 34–56

James S. Clark
Nicholas Professor of Environmental Science, Duke University
Office: A221 LSRC, Durham, NC 27708
Campus Box: Box 90338, Durham, NC 27708-0338
Phone: (919) 613-8036
Email: jimclark@duke.edu

James S. Clark is Nicholas Professor of the Nicholas School of the Environment and Professor of Statistical Science. Clark's research focuses on how global change affects populations, communities, and ecosystems.

An example




Population dynamics at each watershed from 1981 to 2013

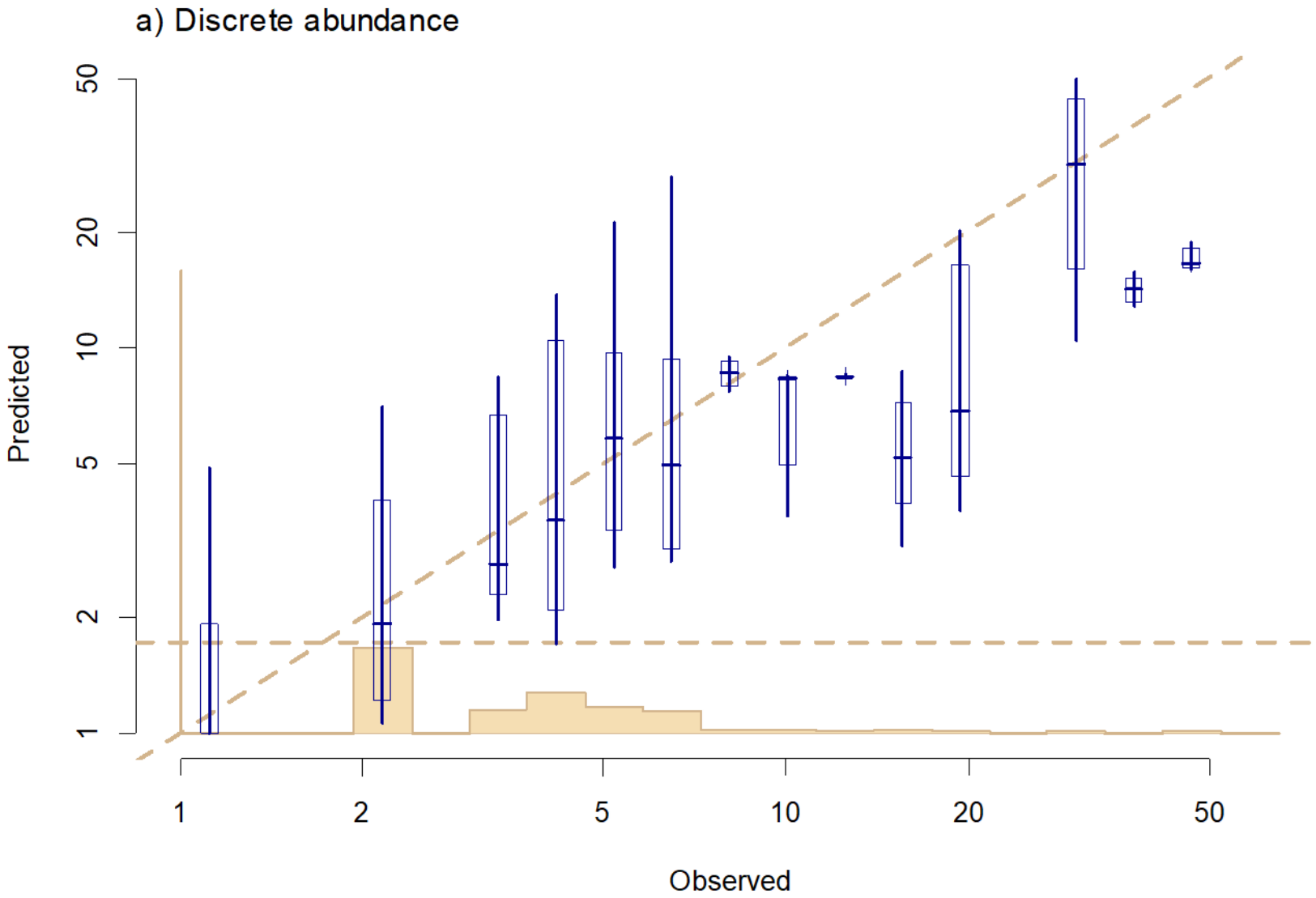
Data: Y matrix ~ X matrix

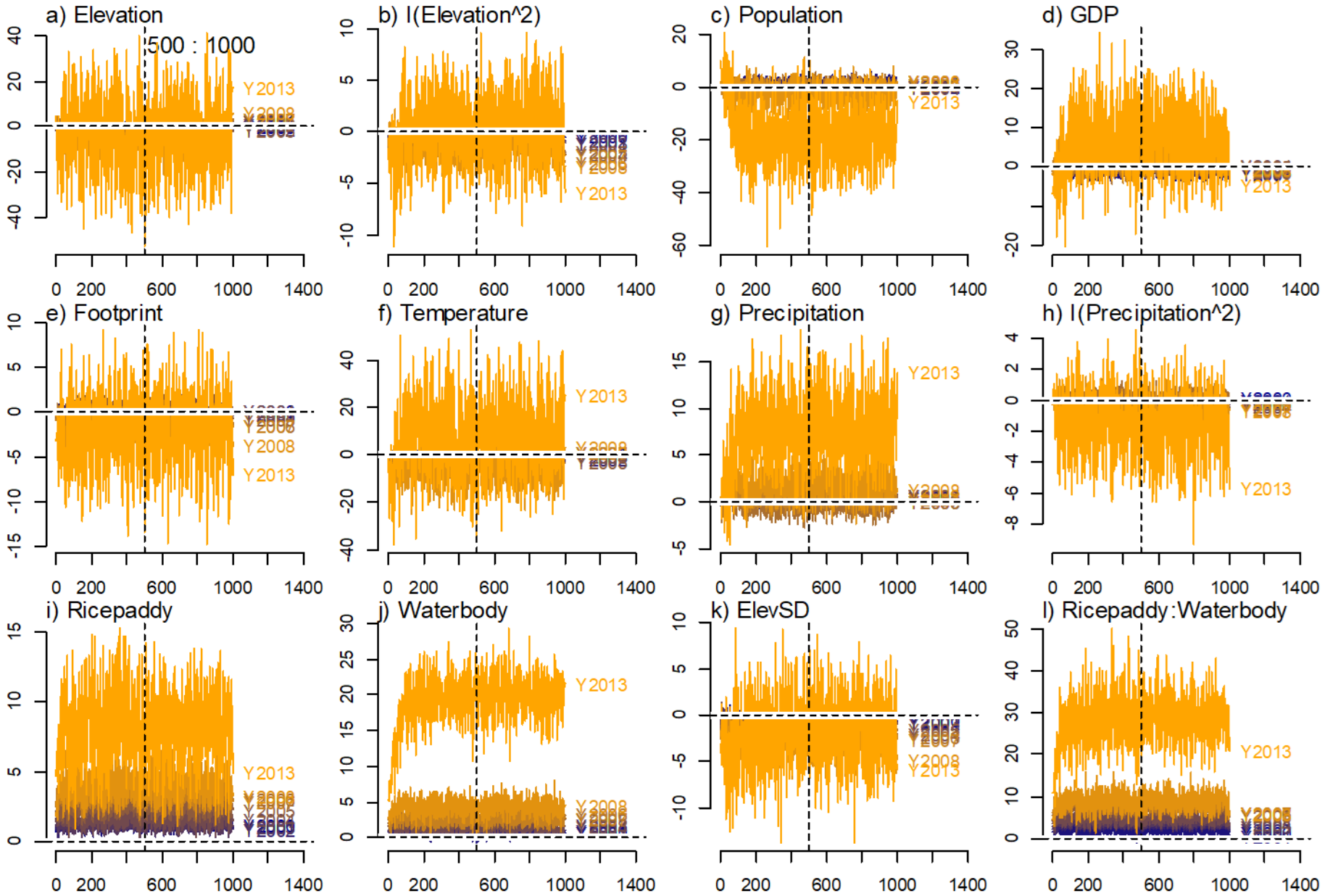
Watersheds	Y2000	Y2001	Y2002	Y2003	Y2004	Y2005	Y2006	Y2007	Y2008	Y2013	lat	lon	Area	Elevation	Population	GDP	Footprint	Temperature	Precipitation	Rice_paddy	Water_body	wet	elevSD
723	14	18	15	20	32	37	44	46	44	65	33.35	107.4	310.35	945.66	147.51	14.98	23.21	12.46	844.89	4.29	0.28	1.2012	143.05
717	1	2	3	6	5	6	12	8	6	34	33.39	107.49	304.05	1034.07	200.83	19.19	25.5	12.16	850.56	2.76	0.29	0.8004	282.96
758	0	0	1	1	1	5	4	4	4	30	33.22	107.57	167.04	548.81	594.57	62.53	34.35	14.59	982.02	2.16	0.63	1.3608	97.17
762	2	2	3	5	4	7	10	7	10	19	33.19	107.7	134.28	600.72	221.82	17.35	27.16	14.3	811.75	2.79	0.29	0.8091	71.2
730	0	2	2	2	5	4	7	6	5	16	33.28	107.65	85.36	703.73	228	20.69	28.15	13.69	950.02	2.2	0.13	0.286	160.02
734	1	2	1	1	2	2	4	4	9	15	33.26	107.49	39.43	596.26	450.98	48.61	31.83	14.33	938.25	1.64	0.2	0.328	111.95
821	0	0	0	0	0	1	2	1	0	13	33.08	107.46	122.58	621.74	88.41	14.74	30.6	14.29	825.3	3.27	0.11	0.3597	58.38
772	0	0	0	0	0	1	1	3	4	8	33.13	107.62	40.91	594.55	116.45	10.09	20.43	14.4	920.58	1.23	0.00712	0.0087576	68.6
884	0	0	0	0	0	1	4	5	5	6	32.95	107.44	341.83	674.96	98.84	15.29	23.93	14.27	841.47	8.45	0.1	0.845	147.63
637	0	0	1	1	2	2	3	5	4	6	33.59	107.57	365.63	1597.55	19.98	1.87	23	9.18	890.54	2.62	0.42	1.1004	360.74
808	0	0	1	1	1	2	2	3	3	5	33.09	107.55	79.77	674.05	30.95	3.53	20	14	837.88	0.75	0.12	0.09	88.39
735	0	1	1	1	1	1	1	1	1	5	33.21	107.84	68.69	690.08	123.4	6.91	20.87	13.88	829.56	0.27	0.18	0.0486	128.13
745	0	0	0	0	0	1	1	0	4	5	33.23	107.3	217.51	609.33	715.48	213.3	30.2	14.29	817.88	1.17	1.03	1.2051	184.58
727	0	1	1	2	4	4	6	7	6	5	33.32	107.74	218.1	869.55	133	8.86	22.57	12.99	837.93	3.23	0.11	0.3553	259.06
690	0	0	0	0	0	0	1	1	1	4	33.42	107.17	423.46	1110.27	49.73	28.76	17.46	11.6	851.05	1.55	0.07	0.1085	270.73
805	0	0	0	0	0	0	0	1	4	3	33.16	107.42	96.8	552.4	340.6	41.32	39.22	14.58	831.14	1.03	0.43	0.4429	78.36
823	0	0	0	0	0	0	0	0	0	2	33.05	107.62	96.06	663.44	112.38	9.6	20.1	14.02	843.17	2.86	0.03	0.0858	81.94
792	0	0	0	0	0	0	0	0	0	2	33.12	107.73	78.9	694.9	96.31	6.73	18.08	13.99	840.09	2.1	0.05	0.105	103.67
779	0	0	0	0	0	0	0	0	0	1	33.16	107.91	87.28	606.99	120.81	5.78	20.54	14.4	824.98	0.93	0.48	0.4464	121.93
736	0	0	1	1	1	1	1	1	1	1	33.26	107.93	121.61	826.89	101.11	5.51	20	13.09	987.19	0.01	0.25	0.0025	423.34
596	0	0	0	0	0	0	0	0	0	1	33.6	107.41	128.23	1500.77	6.48	1.64	21.6	9.97	869.15	0.02	0.31	0.0062	250.67
699	0	0	0	0	0	0	0	0	0	1	33.35	106.96	252	1225.77	35.41	9.76	22.76	11.16	862.64	1.12	0.29	0.3248	327.51
761	0	0	0	0	0	0	0	0	0	1	33.25	107.13	209.48	885.69	388.6	152.91	27.16	12.75	855.86	1.67	0.17	0.2839	454.08
635	0	0	0	0	0	0	0	0	0	1	33.54	107.68	185.83	1527.38	15.19	1.78	20.33	9.84	810.57	0.79	0.19	0.1501	397.83
777	0	0	0	0	0	0	0	0	2	0	33.13	107.36	33.69	508.43	459.07	159.88	35.39	14.78	920.36	0.15	0.6	0.09	53.09

R code

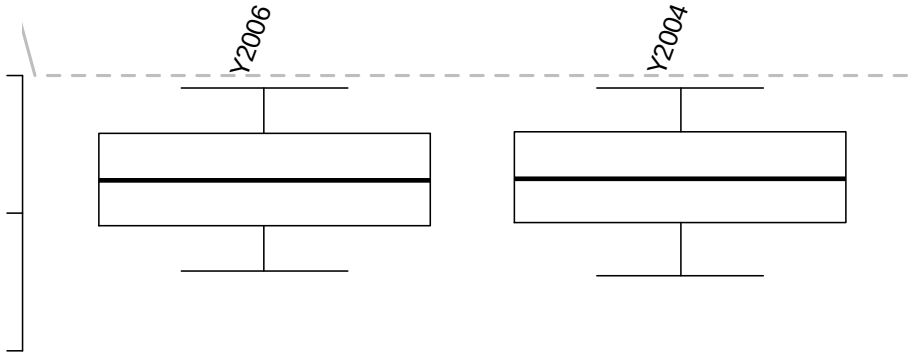
```
library(gjam)
D = read.csv('d:/database/ibisdata/watersheds6-year.csv', header=T)
head(D); dim(D)
xdata <- D[,c(12:24)]
ydata <- D[,c(2:11)]
formula <- as.formula( ~ lat + lon + Area + Elevation + Population + GDP + Footprint +
                         Temperature + Precipitation + Rice_paddy +Water_body + wet+ elevSD )
ml   <- list(ng = 2500, burnin = 500, typeNames = 'DA')
out  <- gjam(formula, xdata = xdata, ydata = ydata, modelList = ml)
summary(out)
specNames <- colnames(ydata)
specColor <- rep('black', ncol(ydata))
specColor[ c(grep('quer', specNames), grep('cary', specNames)) ] <- 'brown'
specColor[ c(grep('acer', specNames), grep('frax', specNames)) ] <- 'darkgreen'
specColor[ c(grep('abie', specNames), grep('pice', specNames)) ] <- 'blue'

pl   <- list(SMALLPLOTS = F, GRIDPLOTS=T, specColor = specColor)
gjamPlot(output = out, plotPars = pl)
```

Generalized Joint Attribute Modeling

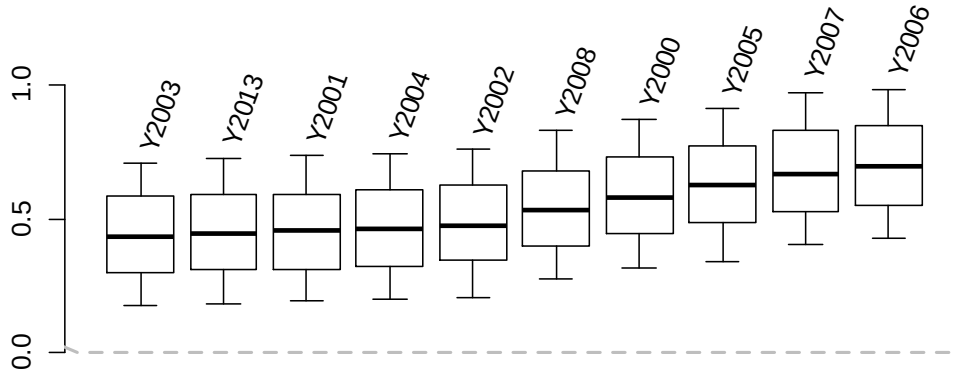



Beta coefficient thinned chains
a) Elevation
500 : 1000
Y2013
b) I(Elevation^2)
Y2013
c) Population
Y2013
d) GDP
Y2013
e) Footprint
Y2008
Y2013
f) Temperature
Y2013
g) Precipitation
Y2013
h) I(Precipitation^2)
Y2013
i) Ricepaddy
Y2013
j) Waterbody
Y2013
Y2008
k) ElevSD
Y2008
Y2013
l) Ricepaddy:Waterbody
Y2013
0 200 600 1000 1400

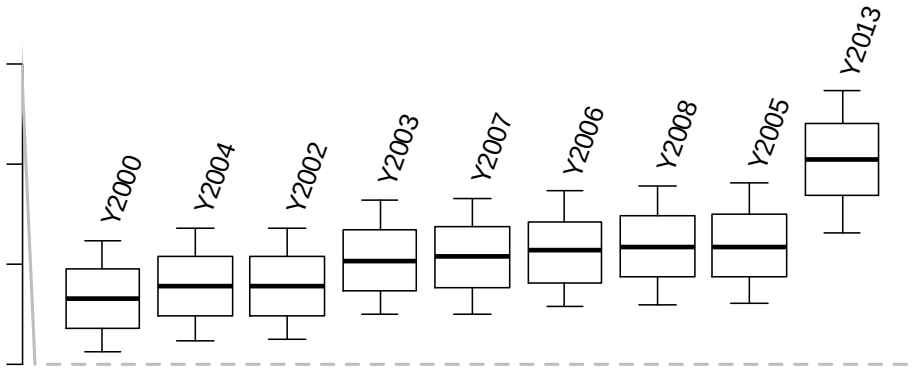
Significant terms (95% posterior)



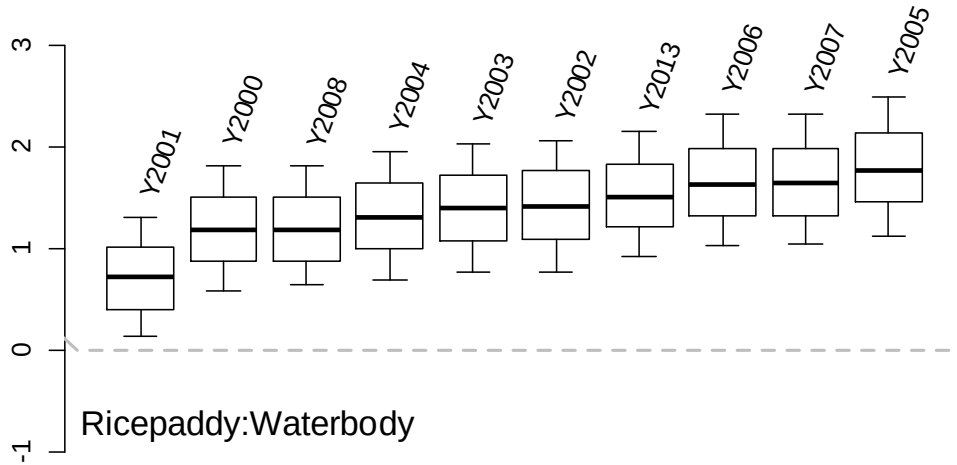
I(Elevation^2)



Ricepaddy

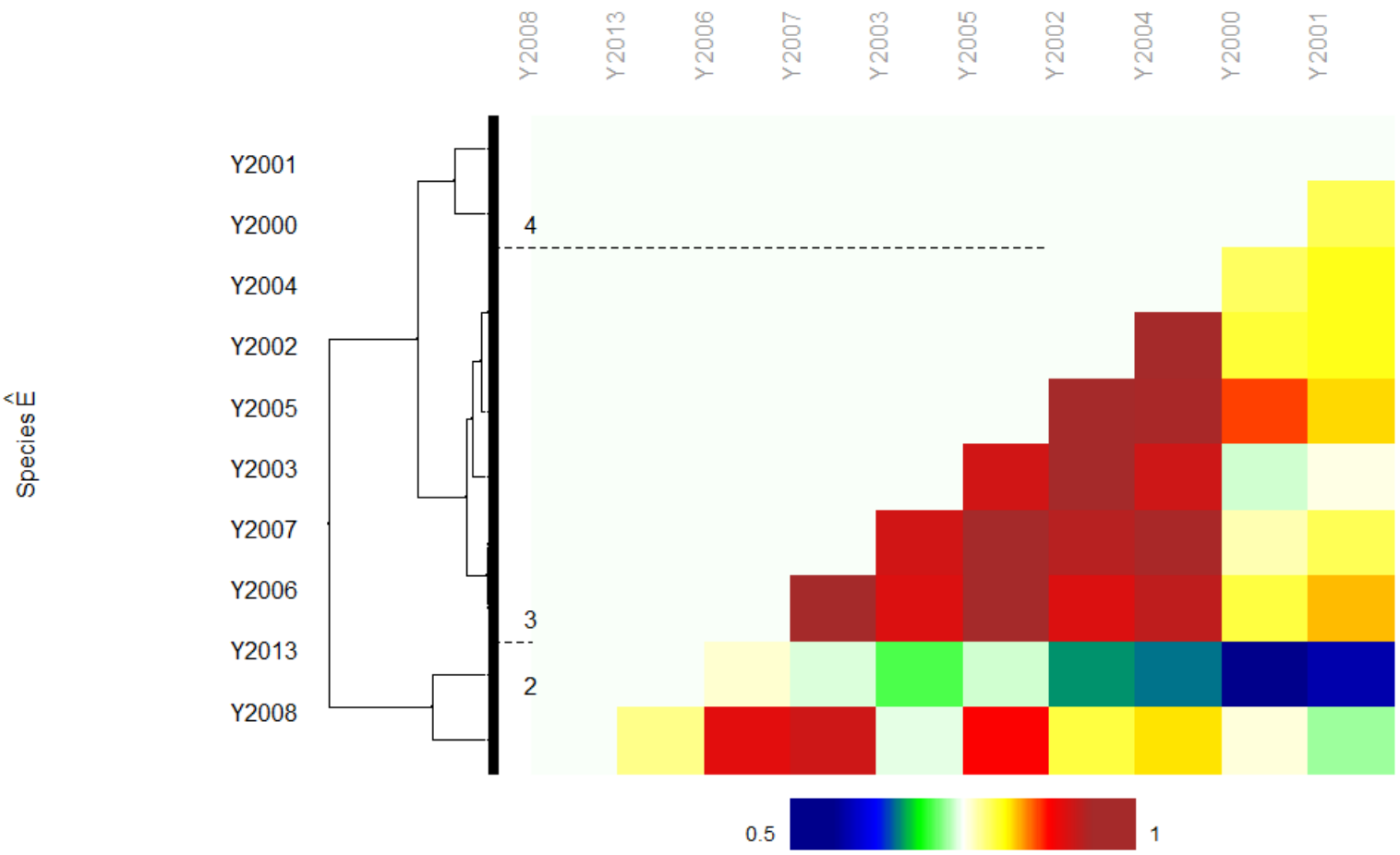


Waterbody

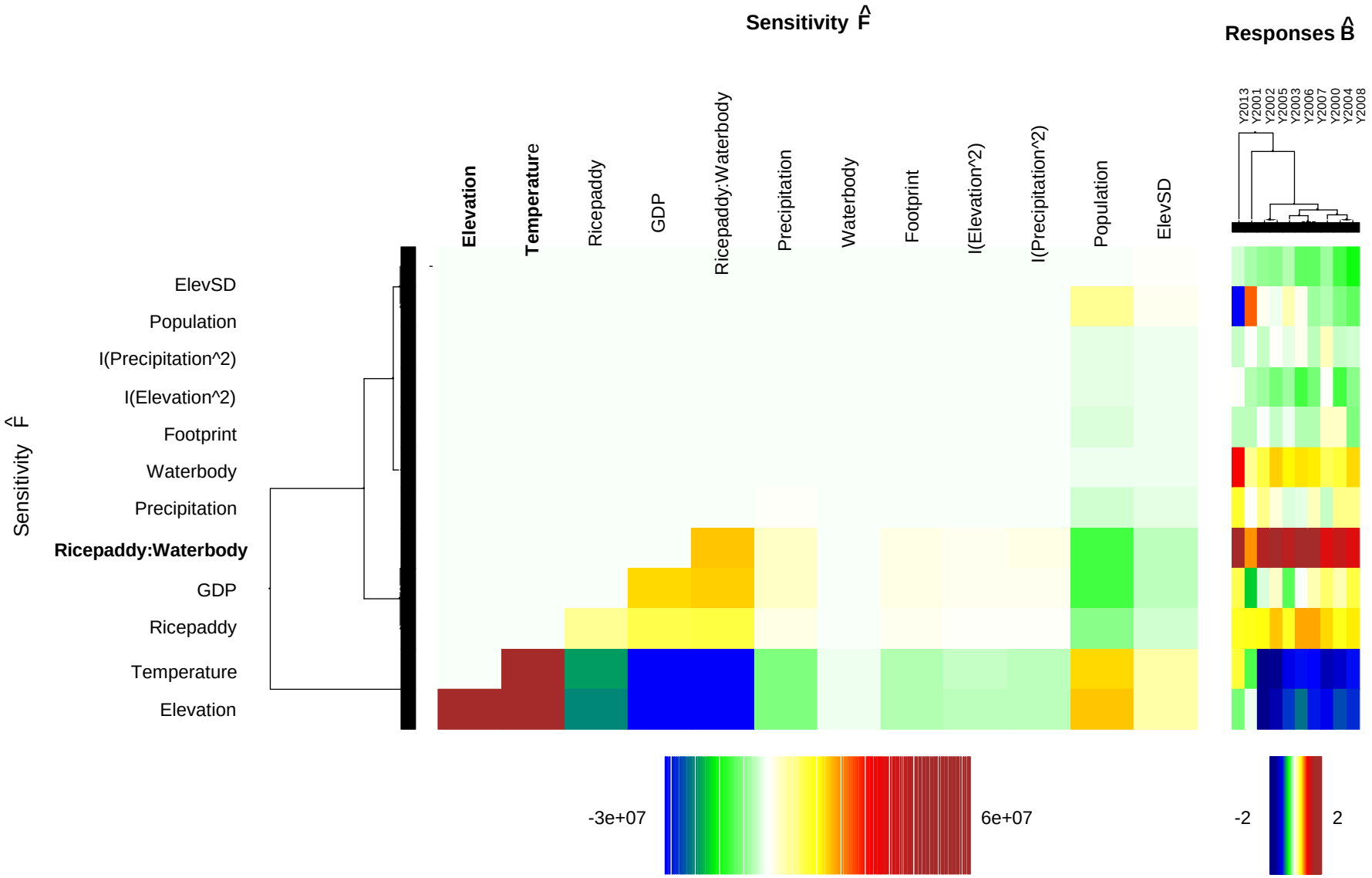


Ricepaddy:Waterbody

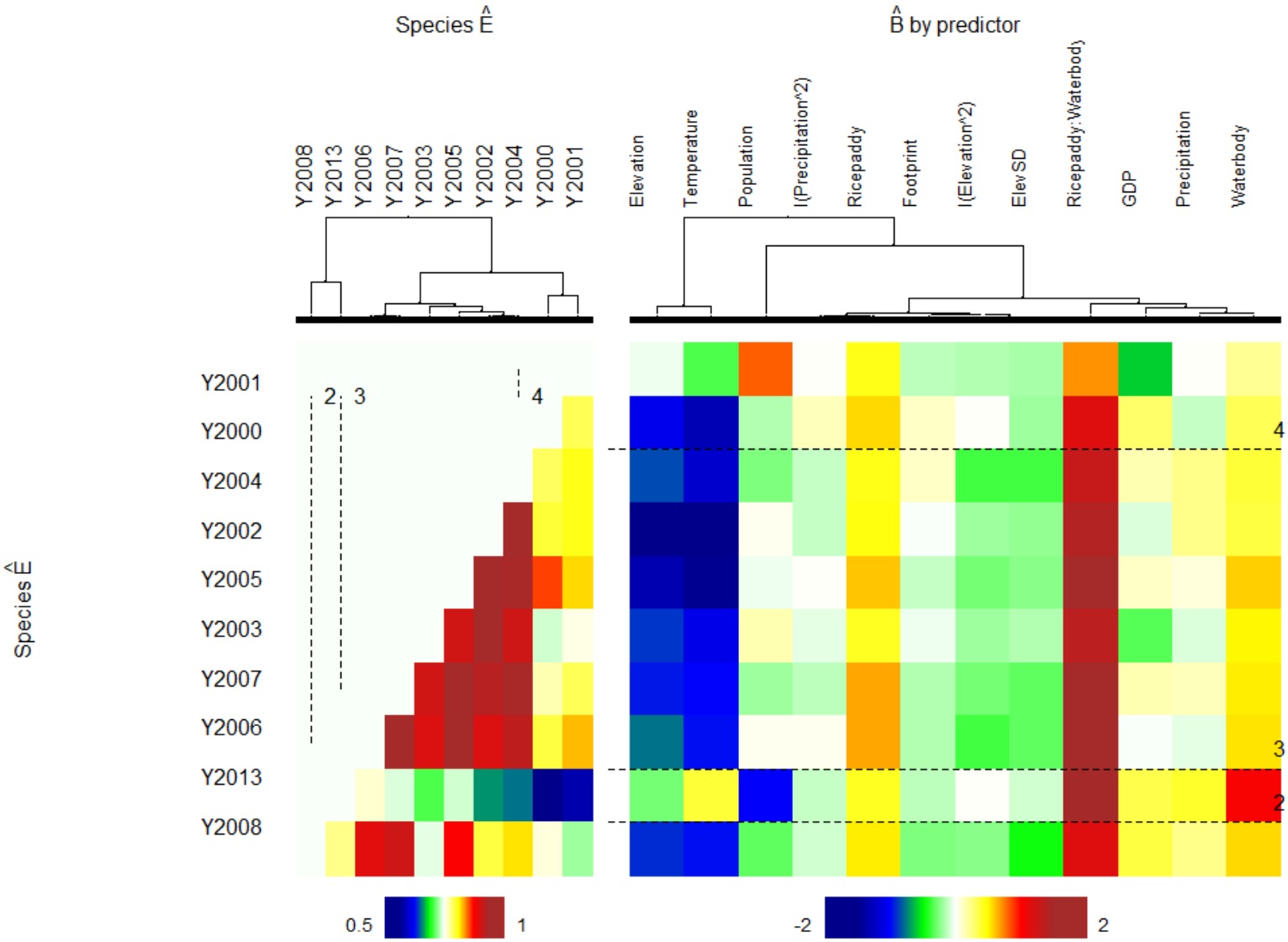
E: model-based response to X



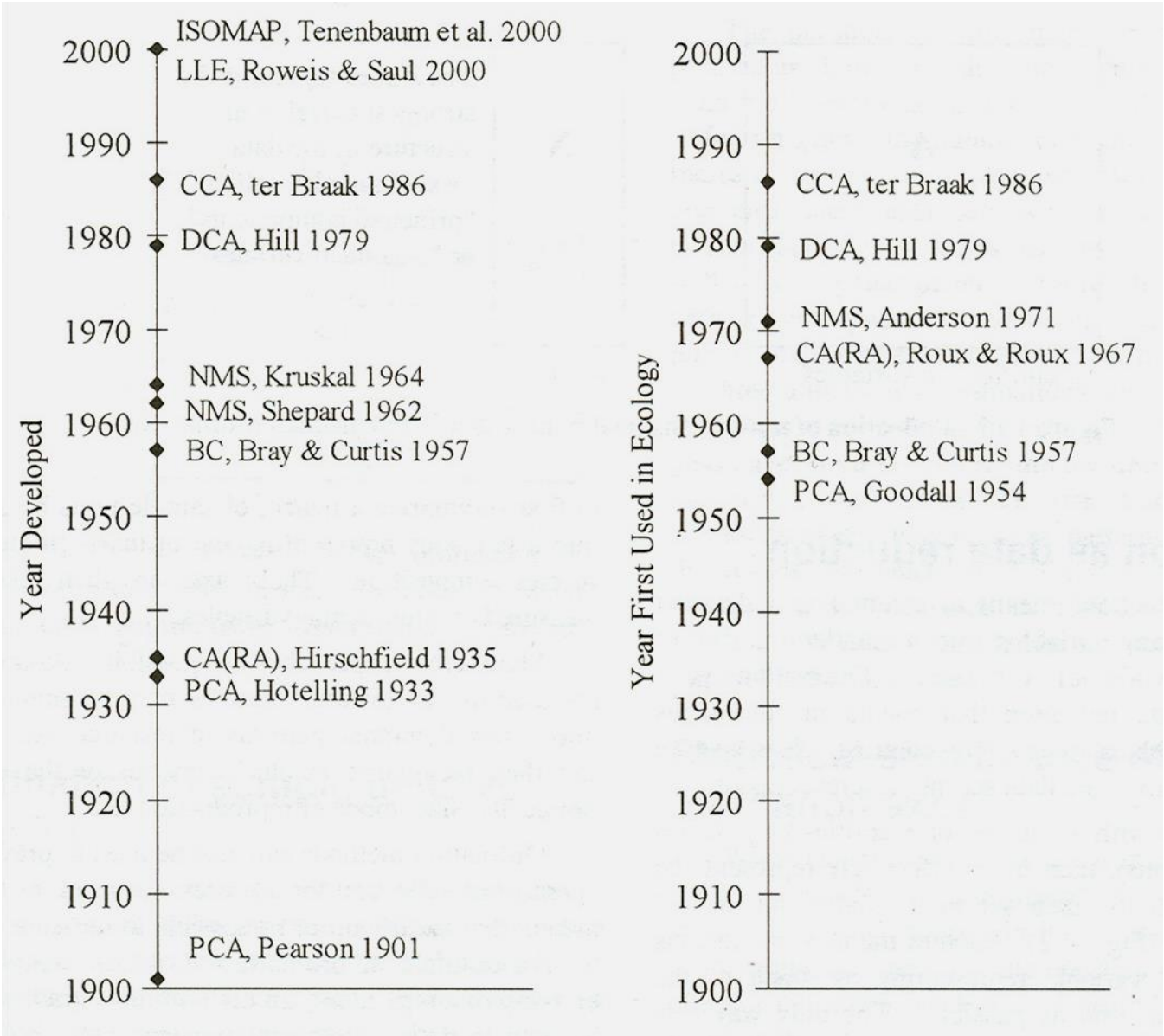
F & beta structure



beta ordered by response to X



History of Ordinations in Ecology



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

Assignment 1

General objectives: learn about NMDS

- Develop a dataset to perform NMDS
- Describe your data, e.g. X1, X2, X3, etc.
- Plot ordinated variables (in columns) and observations (in rows), provide relevant interpretation

Assignment 2

General objectives: learn about CCA

- Develop a dataset to perform CCA
- Describe your data, e.g. X matrix and Y matrix.
- Plot the triplot and provide relevant interpretation.