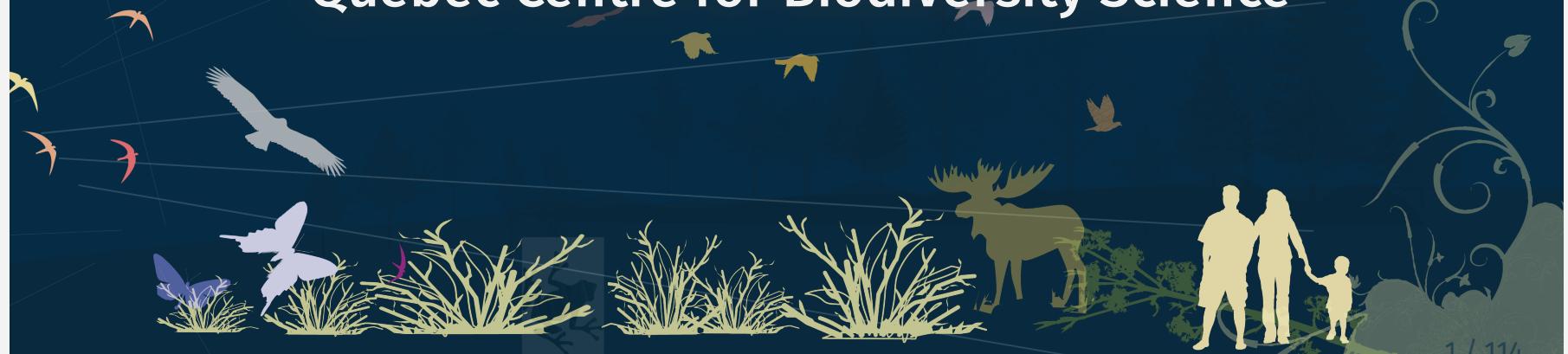




# Workshop 10: Advanced multivariate analyses

QCBS R Workshop Series

Québec Centre for Biodiversity Science



# About this workshop



# Required packages

- `Hmisc`
- `labdsv`
- `MASS`
- `vegan`

```
install.packages(c('Hmisc', 'labdsv', 'MASS', 'vegan'))
```

# Learning objectives

Use R to perform unconstrained ordinations

# Introduction

# Introduction

This workshop is an extension of Workshop 9, which covered the basics of unconstrained analyses:

- Distance metrics and transformations
- Hierarchical clustering
- Unconstrained ordinations (PCA, PCoA, CA, nmDS)

These identify patterns in community composition data or in descriptors, **without** exploring how environmental variables could be driving these patterns.

# Introduction

In this workshop, we will focus instead on **constrained** analyses:

- Redundancy analysis (RDA)
- Partial redundancy analysis
- Variation partitioning
- Multivariate regression tree (MRT)
- Linear discriminant analysis (LDA)

These analyses allow us to **describe** and **predict** relationships between community composition data and environmental variables. (This means we can **test hypotheses!**)

# Download today's script and data

All data and code can be found at [qcbs.ca/wiki/r\\_workshop10](http://qcbs.ca/wiki/r_workshop10)

For this workshop, you will need:

- R script
- Data:
  - DoubsEnv data
  - DoubsSpe data
  - DoubsSpa data
  - Test data for linear discriminant analyses

# Required packages

Please make sure you have downloaded, installed, and loaded these packages:

- vegan (for multivariate analyses)
- rdaTest package (**from wiki**)
- mvpart package (**from wiki**)
- MVPARTwrap package (**from wiki**)
- labdsv (to identify indicator species in the MRT)
- MASS (for LDA)

# Follow along!

As always, we recommend that you:

- create your own script (or add comments to the provided script)
- avoid copy-pasting or running the code directly from the script
- remember to set the working directory to the folder in which your files are stored

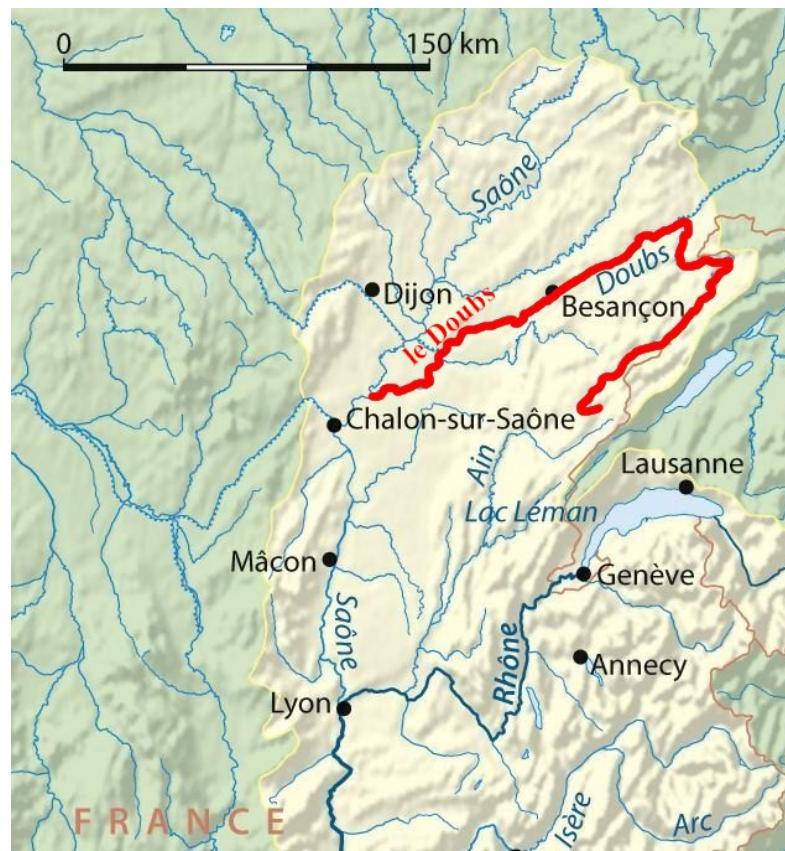
# Data exploration & preparation

# Today's data:

## Doubs River Fish Dataset (Verneaux 1973)

Description of fish communities along the Doubs River:

- 27 species
- 30 sites
- 11 environmental variables



# Load the Doubs River dataset

**Make sure the datasets are in your working directory!**

Load the species data (*doubsspe.csv*):

```
# Make sure the files are in your working directory!
spe <- read.csv("data/doubsspe.csv", row.names = 1)
spe <- spe[-8,] # remove site with no data
```

Load the environmental data (*doubsenv.csv*):

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

**Note:** Only execute once!

# Exploring the fish community dataset

Let's briefly explore the fish community dataset:

```
names(spe) # names of objects (species)
# [1] "CHA" "TRU" "VAI" "LOC" "OMB" "BLA" "HOT" "TOX" "VAN" "CHE" "BAR" "SPI"
# [13] "GOU" "BRO" "PER" "BOU" "PSO" "ROT" "CAR" "TAN" "BCO" "PCH" "GRE" "GAR"
# [25] "BBO" "ABL" "ANG"
dim(spe) # dataset dimensions
# [1] 29 27
```

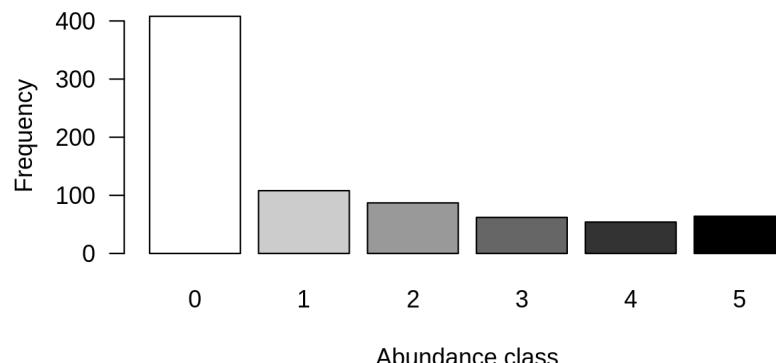
Let's take a closer look at the objects (species):

```
head(spe) # look at first 5 rows
str(spe) # structure of objects in dataset
summary(spe) # summary statistics for all objects (min, mean, max, etc.)
```

# Distribution of species frequencies

Let's take a quick look at how the community is structured.

```
# Count number of species frequencies in each abundance class  
ab <- table(unlist(spe))  
# Plot distribution of species frequencies  
barplot(ab, las = 1, # make axis labels perpendicular to axis  
        xlab = "Abundance class", ylab = "Frequency", # label axes  
        col = grey(5:0/5)) # 5-colour gradient for the bars
```



**Notice:** There are many 0s!

# Distribution of species frequencies

How many 0s are in the dataset?

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

What proportion of the dataset does that represent?

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

# Transforming the community data

**Over 50%** of our dataset consists of 0s, which is common in community datasets.

However, we don't want these common absences to **artificially increase** the similarity between sites.

- To avoid this, we can **transform** the community data.

Let's use the **Hellinger** transformation:

```
# The decostand() function in the vegan package makes this easy for us:  
library(vegan)  
spe.hel <- decostand(spe, method = "hellinger")
```

# Exploring the environmental dataset

Let's briefly explore the environmental dataset:

```
names(env) # names of objects (environmental variables)
# [1] "das" "alt" "pen" "deb" "pH" "dur" "pho" "nit" "amm" "oxy" "dbo"
dim(env) # dataset dimensions
# [1] 29 11
head(env) # look at first 5 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
```

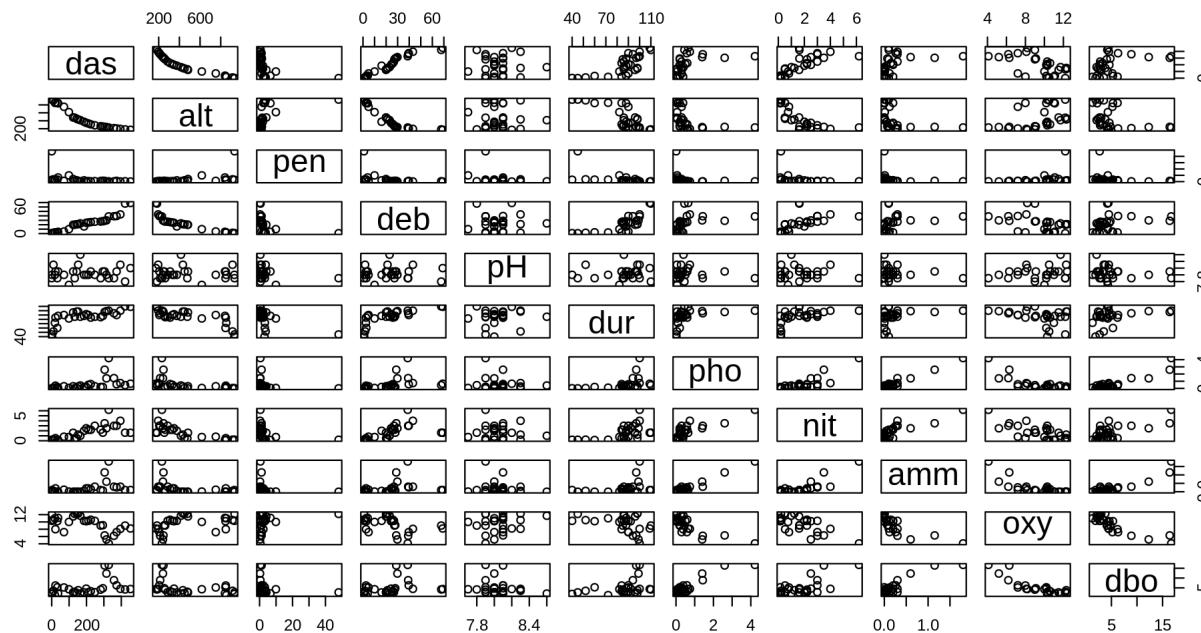
For a closer look at the objects (environmental variables):

```
str(env) # structure of objects in dataset
summary(env) # summary statistics for all objects (min, mean, max, etc.)
```

# Collinearity

```
# We can visually look for correlations between variables:
```

```
pairs(env)
```



**Note:** Some variables look correlated... (das vs. alt, das vs. deb, das vs. dur, das vs. nit, oxy vs. dbo, etc.)

# Standardizing the environmental variables

You cannot compare the effects of variables with different units.

Before moving on to further analyses, **standardizing** your environmental variables is therefore crucial.

```
# Scale and center variables
env.z <- decostand(env, method = "standardize")

# Variables are now centered around a mean of 0:
round(apply(env.z, 2, mean), 1)
# das alt pen deb pH dur pho nit amm oxy dbo
#   0   0   0   0   0   0   0   0   0   0   0   0

# and scaled to have a standard deviation of 1
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   1
```

# Canonical analyses

# Canonical analyses

Canonical analyses allow us to:

- identify relationships between a response matrix and explanatory matrix/matrices
- test hypotheses about these relationships
- make predictions

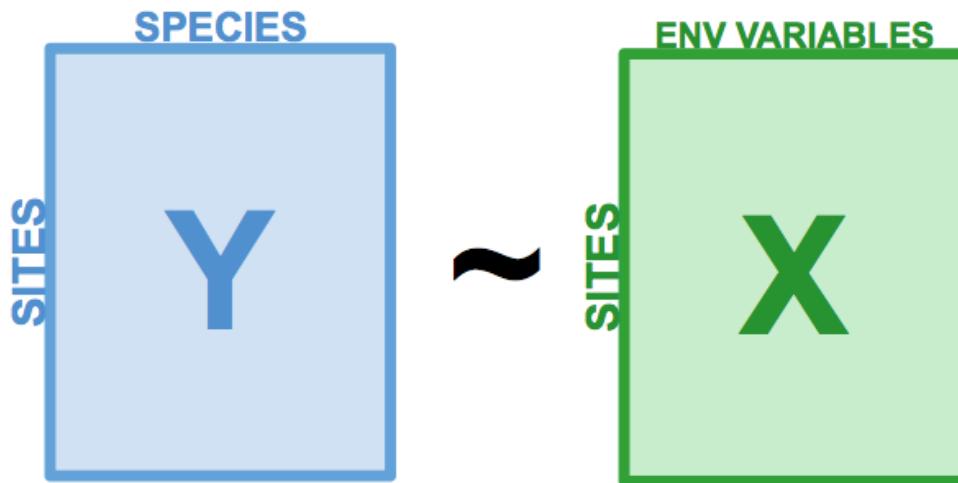
# Canonical analyses

Redundancy analysis (RDA)

# Redundancy analysis (RDA)

RDA is constrained ordination.

- RDA is a direct extension of multiple regression.
- RDA models the effect of an explanatory matrix on a response matrix (*instead of a single response variable*).



Variables can be quantitative, qualitative, or binary (0/1).

- **transform** and/or **standardize** them prior to running an RDA.

# Running an RDA in R

## Prepare the data

```
# We'll use our standardized environmental data  
# But we will remove 'das', which was correlated with many other variables:  
env.z <- subset(env.z, select = -das)
```

## Run the RDA

```
# Model the effect of all environmental variables on fish community composition  
spe.rda <- rda(spe.hel ~ ., data = env.z)
```

## Extract key results of the RDA

```
summary(spe.rda, display = NULL)
```

# RDA output in R

```
Call:  
rda(formula = spe.hel ~ alt + pen + deb + pH + dur + pho + nit +  
amm + oxy + dbo, data = env.z)
```

Partitioning of variance:		
	Inertia	Proportion
Total	0.5025	1.0000
Constrained	0.3689	0.7341
Unconstrained	0.1336	0.2659

- **Constrained Proportion:** variance of Y explained by X (**73.41%**)
- **Unconstrained Proportion:** unexplained variance in Y (**26.59%**)

How would you report these results?

- *The included environmental variables explain **73.41%** of the variation in fish community composition across sites.*

# Selecting variables

Using **forward selection**, we can select the explanatory variables that are statistically "important".

***Which variables significantly contribute to our model's explanatory power?***

```
# Forward selection of variables:  
fwd.sel <- ordiR2step(rda(spe.hel ~ 1, data = env.z), # lower model limit (simple!)  
                      scope = formula(spe.rda), # upper model limit (the "full" model)  
                      direction = "forward",  
                      R2scope = TRUE, # can't surpass the "full" model's R2  
                      pstep = 1000,  
                      trace = FALSE) # change to TRUE to see the selection process!
```

Here, we are essentially adding one variable at a time, and retaining it if it significantly increases the model's adjusted R2.

# Selecting variables

- Which variables are retained by the forward selection?

```
# Check the new model with forward-selected variables  
fwd.sel$call  
# rda(formula = spe.hel ~ alt + oxy + dbo, data = env.z)
```

- What is the adjusted R<sup>2</sup> of the RDA with the selected variables?

```
# Write our new model  
spe.rda.signif <- rda(spe.hel ~ alt + oxy + dbo, data = env.z)  
# check the adjusted R2  
RsquareAdj(spe.rda.signif)  
# $r.squared  
# [1] 0.5894243  
#  
# $adj.r.squared  
# [1] 0.5401552
```

# Significance testing

Use **anova.cca()** to test the significance of your RDA.

```
anova.cca(spe.rda.signif, step = 1000)
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(formula = spe.hel ~ alt + oxy + dbo, data = env.z)
#          Df Variance      F Pr(>F)
# Model     3  0.29619 11.963  0.001 ***
# Residual 25  0.20632
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

You can also test the significance of each variable!

```
anova.cca(spe.rda.signif, step = 1000, by = "term")
```

# RDA plot

One of the most powerful aspects of RDA is the **simultaneous visualization** of your response and explanatory variables (*i.e. species and environmental variables*).

There are 2 scaling options:

## Type 1

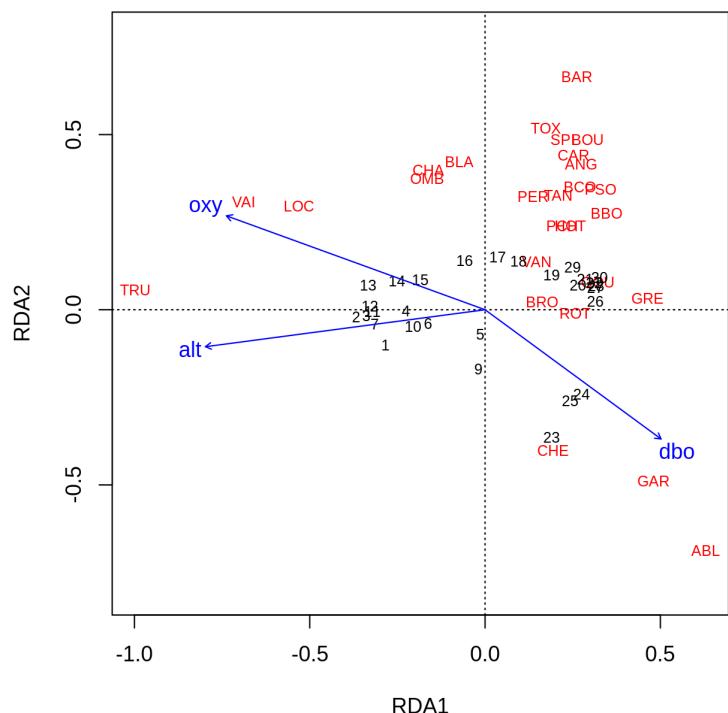
distances among objects reflect their similarities

## Type 2

angles between variables reflect their correlation

## RDA plot: Type 1

```
ordiplot(spe.rda.signif,  
         scaling = 1,  
         type = "text")
```

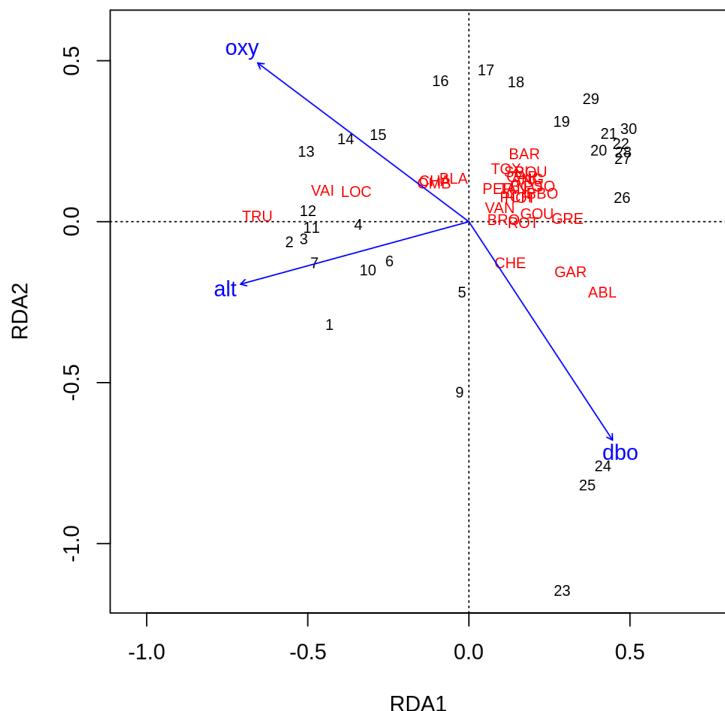


Scaling 1 shows similarities between objects in the *response matrix*.

- Sites (numbers) that are **closer together** have more similar communities.
  - Species that are **closer together** occupy more sites in common.

# RDA plot: Type 2

```
ordiplot(spe.rda.signif,  
        scaling = 2,  
        type = "text")
```

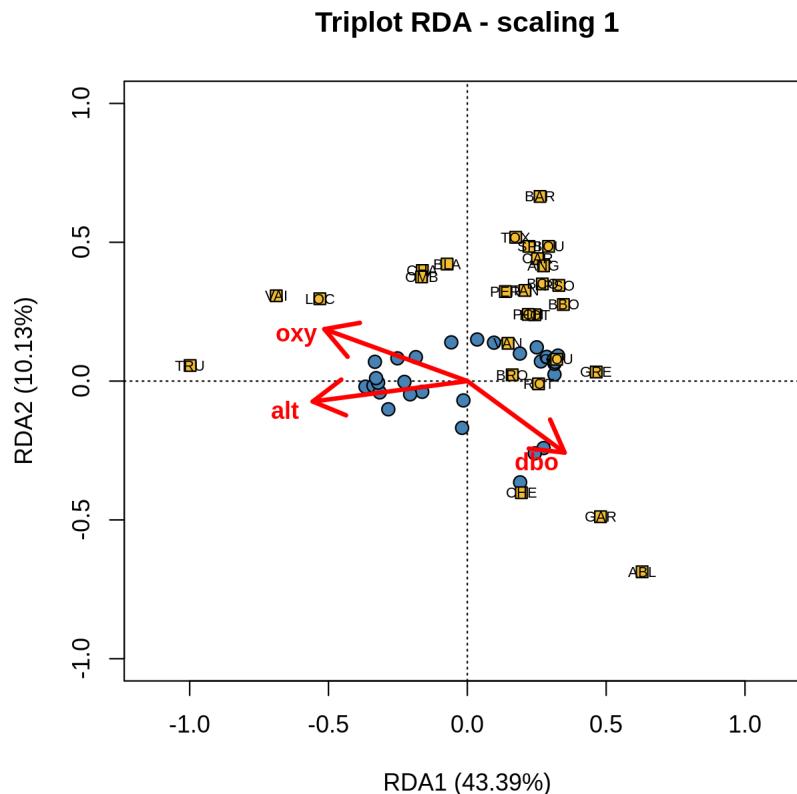


Scaling 2 shows the effects of explanatory variables.

- Longer arrows mean this variable strongly drives the variation in the community matrix.
- Arrows pointing in opposite directions have a negative relationship
- Arrows pointing in the same direction have a positive relationship

# Customizing RDA plots

Both `plot()` and `ordiplot()` make quick and simple ordination plots, but you can customize your plots by manually setting the aesthetics of points, text, and arrows.



See the wiki page for more details!



# Challenge 1

Run an RDA to model the effects of environmental variables on mite species abundances.

First, load the mite data:

```
# Load mite species abundance data  
data("mite")  
  
# Load environmental data  
data("mite.env")
```

Recall some useful functions:

```
decostand()  
rda()  
ordiR2step()  
anova.cca()  
ordiplot()
```

# Challenge 1: Solution

Step 1: Prepare the data

```
# Hellinger transform the community data  
mite.spe.hel <- decostand(mite, method = "hellinger")  
  
# Standardize quantitative environmental data  
mite.env$SubsDens <- decostand(mite.env$SubsDens, method = "standardize")  
mite.env$WatrCont <- decostand(mite.env$WatrCont, method = "standardize")
```

# Challenge 1: Solution

## Step 2: Select environmental variables

```
# Initial RDA with ALL of the environmental data
mite.spe.rda <- rda(mite.spe.hel ~ ., data = mite.env)

# Forward selection of environmental variables
fwd.sel <- ordiR2step(rda(mite.spe.hel ~ 1, data = mite.env),
                       scope = formula(mite.spe.rda),
                       direction = "forward",
                       R2scope = TRUE, pstep = 1000, trace = FALSE)

fwd.sel$call
# rda(formula = mite.spe.hel ~ WatrCont + Shrub + Substrate + Topo,
#      data = mite.env)
```

# Challenge 1: Solution

Step 3: Run RDA and check adjusted R2

```
# Re-run the RDA with the significant variables
mite.spe.rda.signif <- rda(mite.spe.hel ~ WatrCont + Shrub +
                           Substrate + Topo + SubsDens,
                           data = mite.env)

# Find the adjusted R2 of the model with the retained env variables
RsquareAdj(mite.spe.rda.signif)$adj.r.squared
# [1] 0.4367038
```

# Challenge 1: Solution

## Step 4: Test model significance

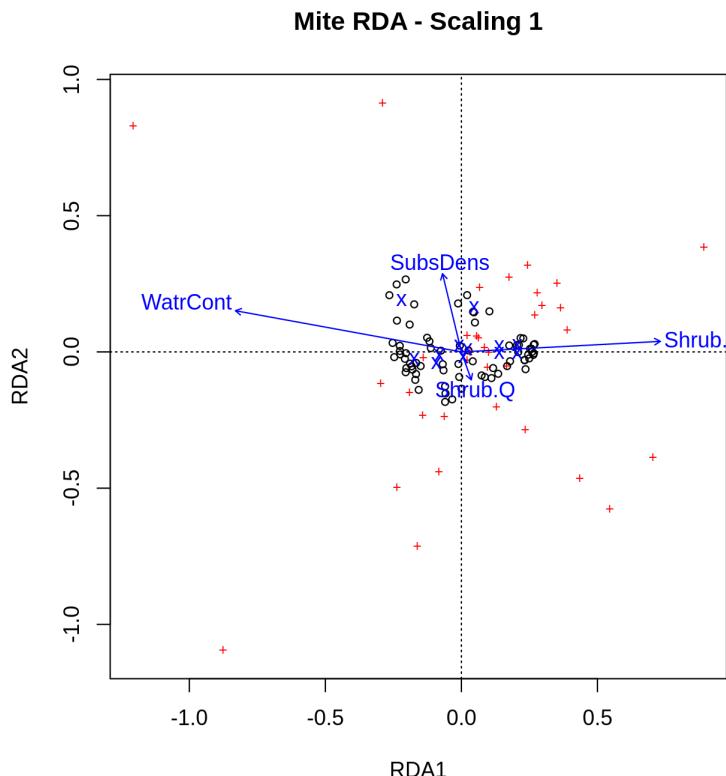
```
anova.cca(mite.spe.rda.signif, step = 1000)
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(formula = mite.spe.hel ~ WatrCont + Shrub + Substrate + Topo + SubsDens)
#          Df Variance      F Pr(>F)
# Model     11  0.20759 5.863  0.001 ***
# Residual  58  0.18669
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

The selected environmental variables significantly explain **43.7% (p = 0.001)** of the variation in mite species abundances.

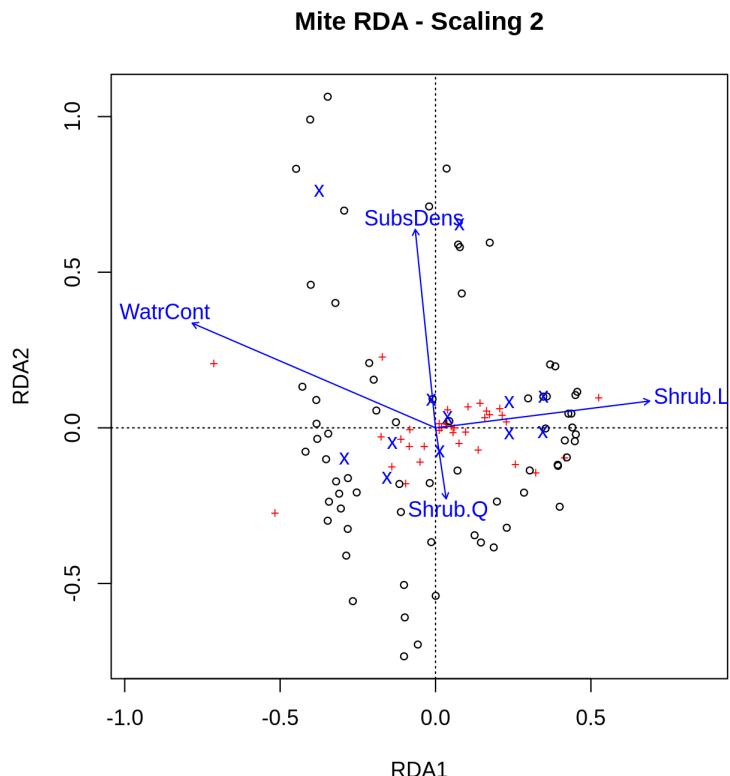
# Challenge 1: Solution

Step 5: Plot the RDA!

```
ordiplot(mite.spe.rda.signif,  
         scaling = 1,  
         main = "Mite RDA - Scaling 1")
```



```
ordiplot(mite.spe.rda.signif,  
         scaling = 2,  
         main = "Mite RDA - Scaling 2")
```

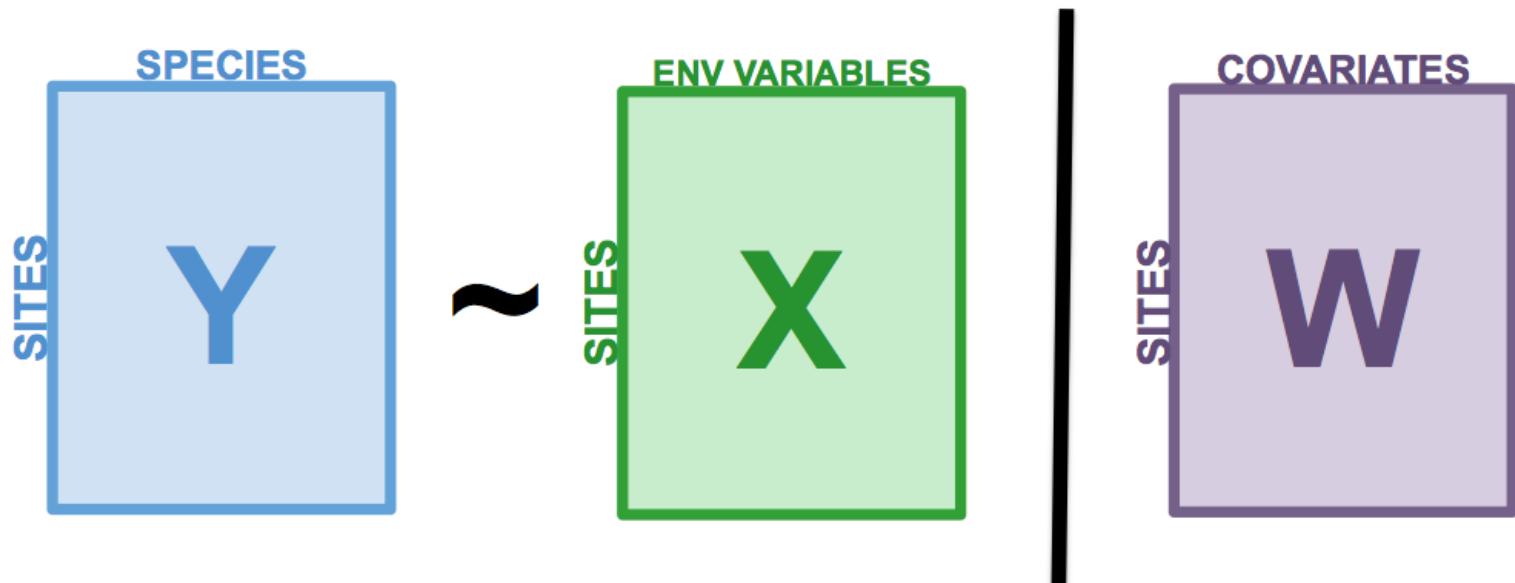


# Canonical analyses

Partial RDA

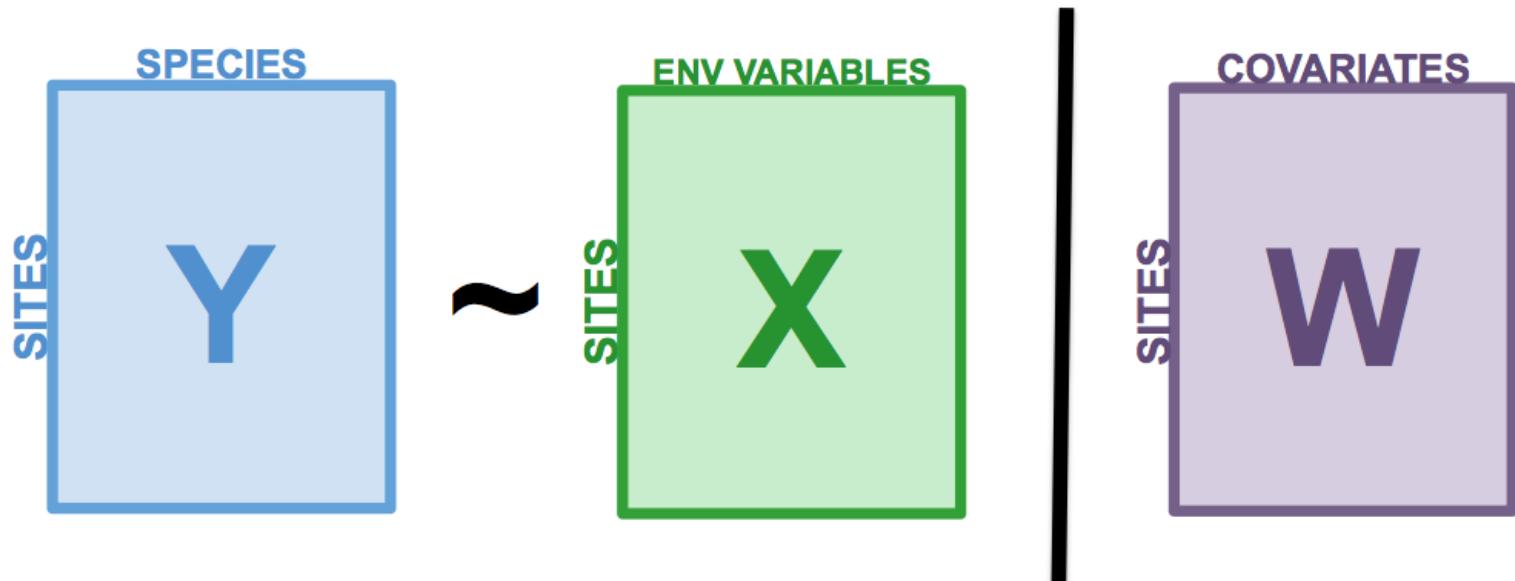
# Partial RDA

- Special case of RDA involving covariates
  - Models the linear effects of matrix X on matrix Y while controlling for a matrix W of **covariates**.



# Applications of partial RDA

- Assess effects of environmental variables on community composition while **accounting for variation** that isn't the focus of the study.
- **Isolate** effect of one or more groups of explanatory variables



# Partial RDA on Doubs River data

Let's assess the effect of water chemistry on fish species abundances while controlling for the effect of topography.

```
# Subset environmental data into topography variables and chemistry variables  
env.topo <- subset(env.z, select = c(alt, pen, deb))  
env.chem <- subset(env.z, select = c(pH, dur, pho, nit, amm, oxy, dbo))  
  
# Partial RDA  
spe.partial.rda <- rda(spe.hel, env.chem, env.topo)
```

**Note:** Alternative syntax for the partial RDA:

```
spe.partial.rda <- rda(spe.hel ~ pH + dur + pho + nit + amm + oxy + dbo +  
                        Condition(alt + pen + deb),  
                        data = env.z)
```

# Partial RDA output in R

```
summary(spe.partial.rda, display = NULL)
```

```
Call:  
rda(X = spe.hel, Y = env.chem, Z = env.topo)  
  
Partitioning of variance:  
          Inertia Proportion  
Total      0.5025    1.0000  
Conditioned 0.2087    0.4153  
Constrained 0.1602    0.3189  
Unconstrained 0.1336   0.2659
```

- **Conditioned Proportion:** variance of Y explained by W (**41.53%**)
- **Constrained Proportion:** variance of Y explained by X (**31.89%**)
- **Unconstrained Proportion:** unexplained variance in Y (**26.59%**)

How would you report these results? Water chemistry explains 31.9% of the variation in fish community composition across sites, while topography explains 41.5% of this variation.

# Significance testing

First, let's extract the model's adjusted R2.

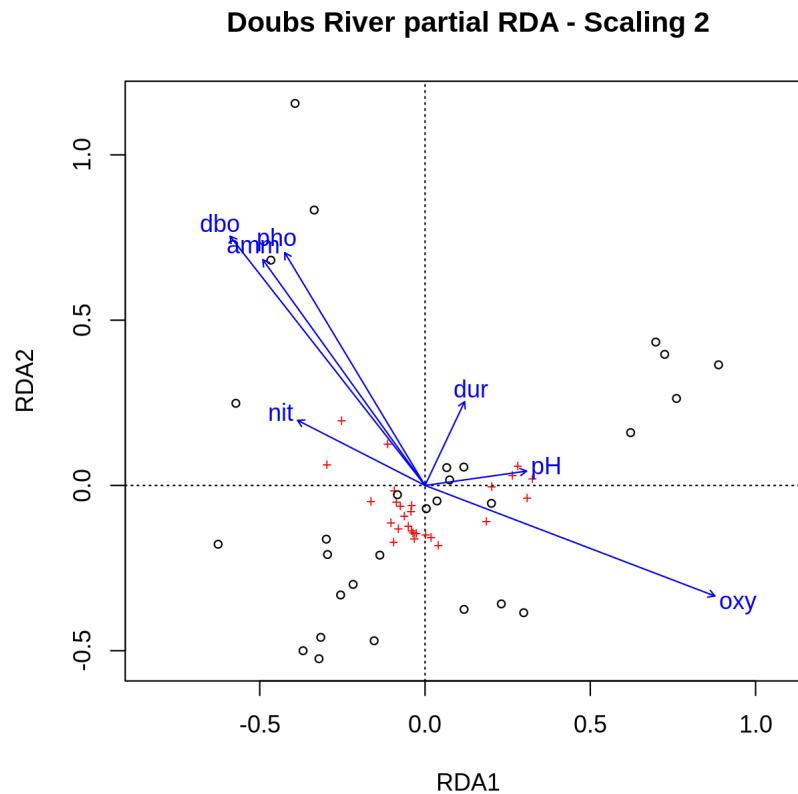
```
RsquareAdj(spe.partial.rda)$adj.r.squared  
# [1] 0.2413464
```

Then, let's test whether the model is statistically significant.

```
anova.cca(spe.partial.rda, step = 1000)  
# Permutation test for rda under reduced model  
# Permutation: free  
# Number of permutations: 999  
#  
# Model: rda(X = spe.hel, Y = env.chem, Z = env.topo)  
#          Df Variance      F Pr(>F)  
# Model     7  0.16024 3.0842  0.001 ***  
# Residual 18  0.13360  
# ---  
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

# Plot the partial RDA

```
ordiplot(spe.partial.rda, scaling = 2,  
         main = "Doubs River partial RDA - Scaling 2")
```





# Challenge 2

Run a partial RDA to model the effects of environmental variables on mite species abundances, while controlling for substrate variables (SubsDens, WatrCont, and Substrate).

- What is the variance explained by substrate variables?
- Is the model significant?
- Which axes are significant?

Recall some useful functions:

```
rda()  
summary()  
RsquareAdj()  
anova.cca() # hint: see the 'by' argument in ?anova.cca
```

# Challenge 2: Solution

Our species abundance and environmental datasets have already been transformed and standardized.

So, we can start with the partial RDA:

```
# Compute partial RDA
mite.spe.subs <- rda(mite.spe.hel ~ Shrub + Topo
                      + Condition(SubsDens + WatrCont + Substrate),
                      data = mite.env)

# Check summary
summary(mite.spe.subs, display = NULL)
```

Shrub and Topo explain **9.8%** of the variation in mite species abundances, while substrate covariables explain **42.8%** of this variation.

# Challenge 2: Solution

- What is the variance explained by substrate variables?

```
RsquareAdj(mite.spe.subs)$adj.r.squared  
# [1] 0.08327533
```

- Is the model significant?

```
anova.cca(mite.spe.subs, step = 1000)  
# Permutation test for rda under reduced model  
# Permutation: free  
# Number of permutations: 999  
  
# Model: rda(formula = mite.spe.hel ~ Shrub + Topo + Condition(SubsDens + WatrCont +  
#           Df Variance      F Pr(>F)  
# Model     3 0.038683 4.006  0.001 ***  
# Residual 58 0.186688  
# ---  
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

# Challenge 2: Solution

- Which axes are significant?

```
anova.cca(mite.spe.subs, step = 1000, by = "axis")
# Permutation test for rda under reduced model
# Forward tests for axes
# Permutation: free
# Number of permutations: 999
#
# Model: rda(formula = mite.spe.hel ~ Shrub + Topo + Condition(SubsDens + WatrCont +
#                 Df Variance      F Pr(>F)
# RDA1      1 0.027236 8.4618  0.001 ***
# RDA2      1 0.008254 2.5643  0.018 *
# RDA3      1 0.003193 0.9919  0.439
# Residual 58 0.186688
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

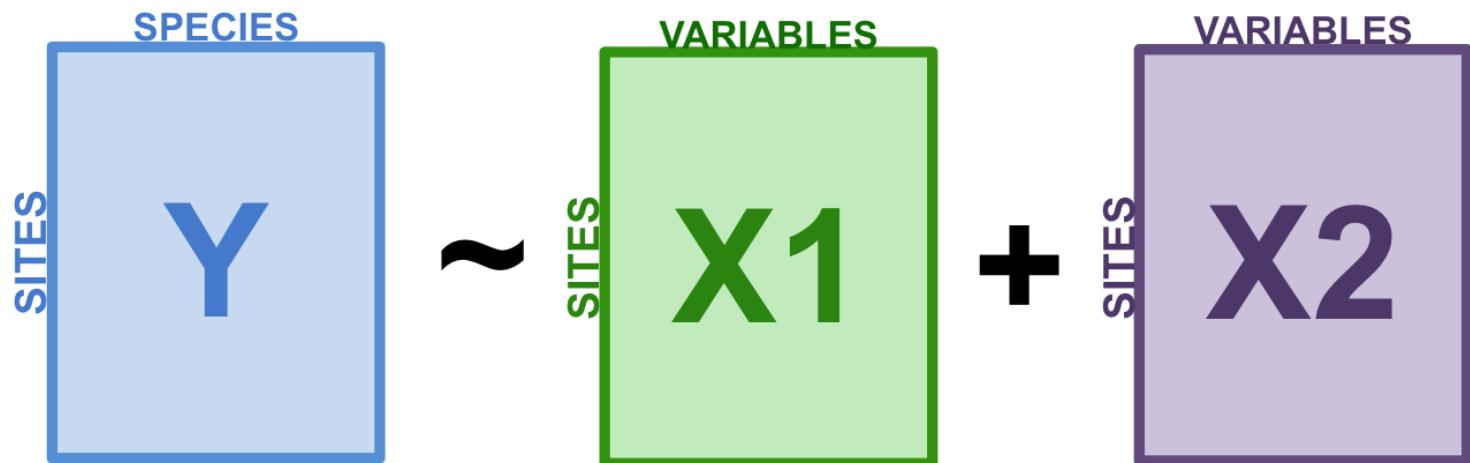
# Canonical analyses

## Variation partitioning

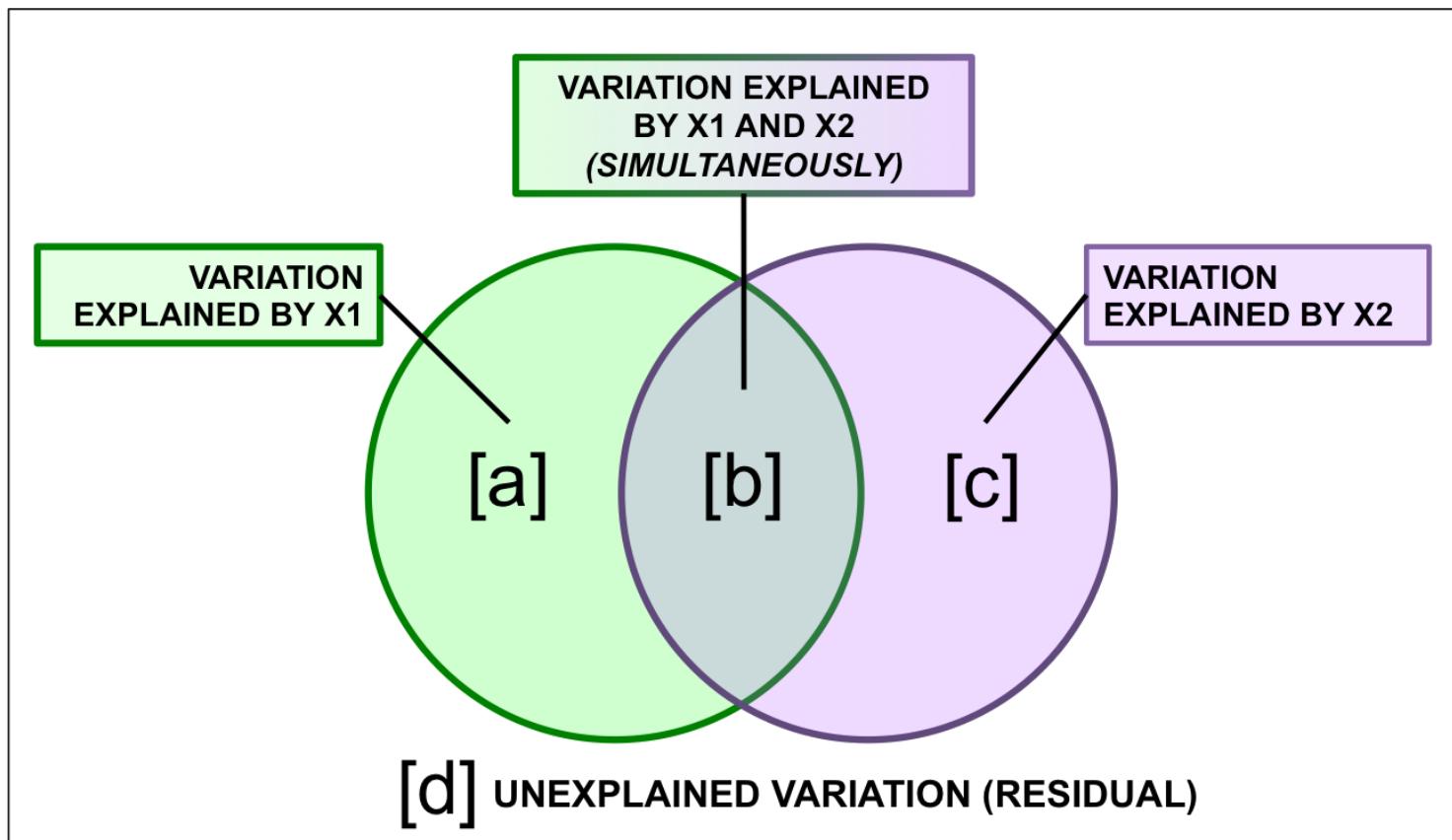
# Variation partitioning

Partitions the variation of response variables among 2, 3, or 4 explanatory datasets.

- e.g. large-scale and small-scale
- e.g. abiotic and biotic



# Variation partitioning



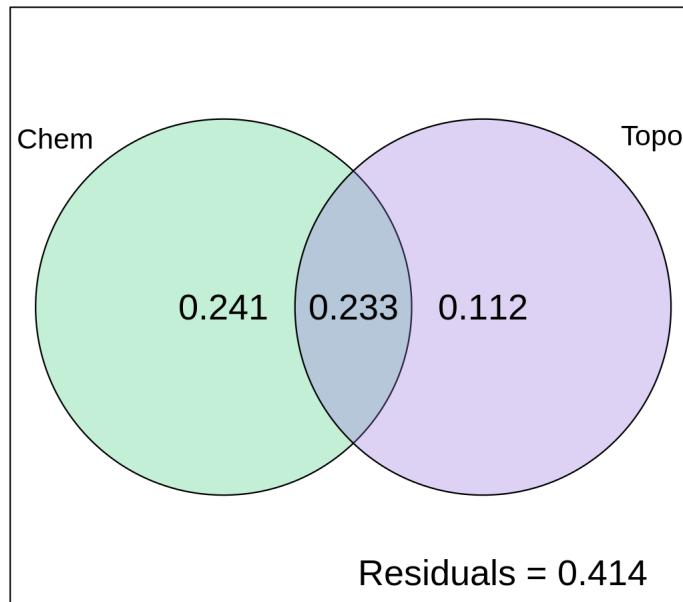
# Variation partitioning in R

**Note:** Make sure you've loaded the *vegan* package!

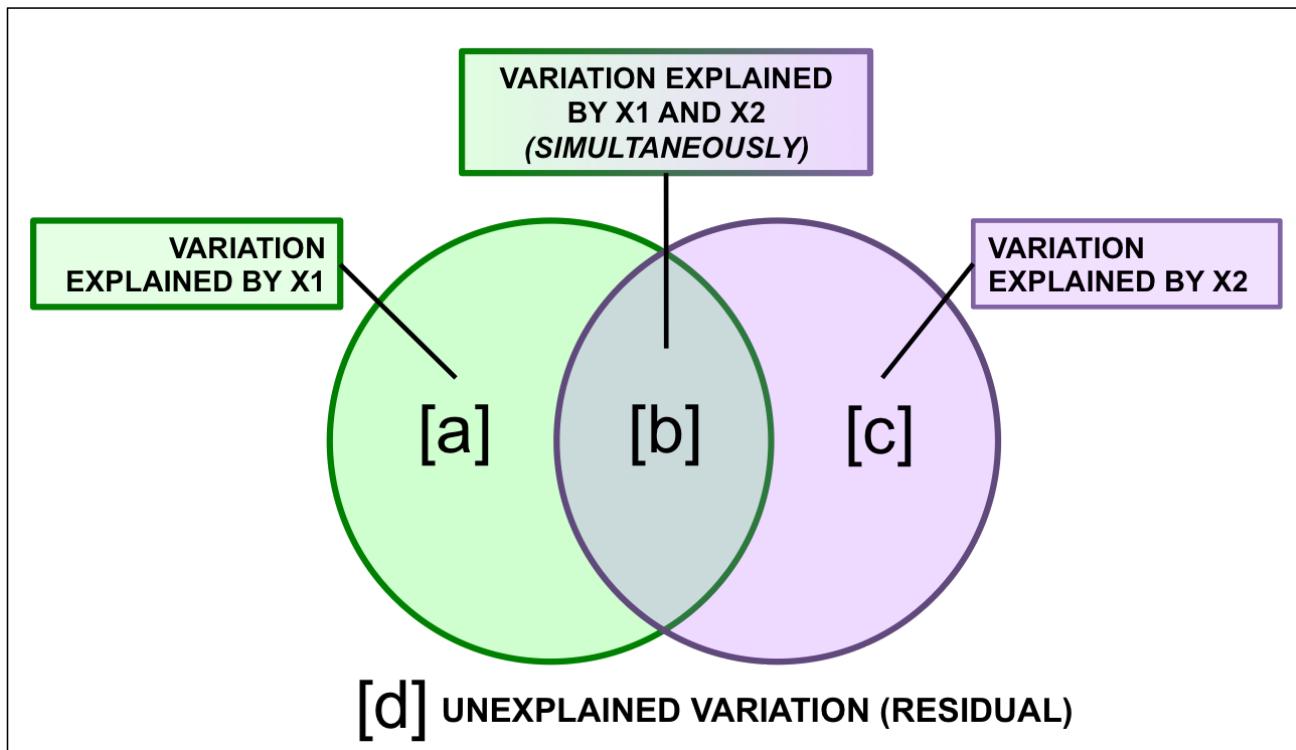
```
spe.part.all <- varpart(spe.hel, env.chem, env.topo)
spe.part.all$part # access results!
# No. of explanatory tables: 2
# Total variation (SS): 14.07
#           Variance: 0.50251
# No. of observations: 29
#
# Partition table:
#                               Df R.squared Adj.R.squared Testable
# [a+b] = X1                  7  0.60579    0.47439    TRUE
# [b+c] = X2                  3  0.41526    0.34509    TRUE
# [a+b+c] = X1+X2            10  0.73414    0.58644    TRUE
# Individual fractions
# [a] = X1|X2                 7           0.24135    TRUE
# [b]                         0           0.23304   FALSE
# [c] = X2|X1                 3           0.11205    TRUE
# [d] = Residuals              0           0.41356   FALSE
# ---
# Use function 'rda' to test significance of fractions of interest
```

# Variation partitioning plot

```
plot(spe.part.all,  
      Xnames = c("Chem", "Topo"), # name the partitions  
      bg = c("seagreen3", "mediumpurple"), alpha = 80, # colour the circles  
      digits = 2, # only show 2 digits  
      cex = 1.5)
```

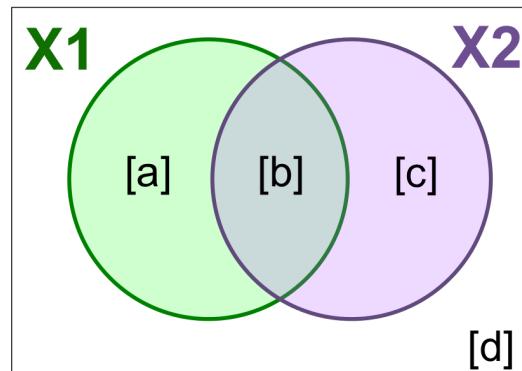


# Significance testing



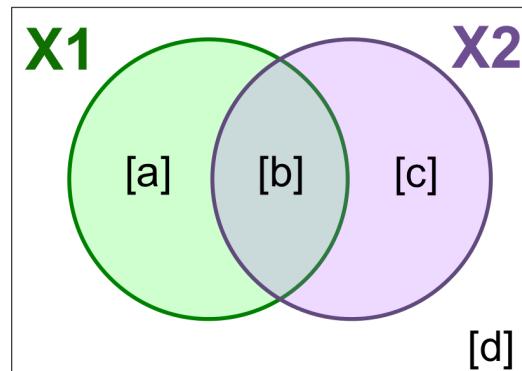
- The shared fraction [b] **cannot** be tested for significance.
- But, we can test the significance of the remaining fractions!

# Significance testing: X1 [a+b]



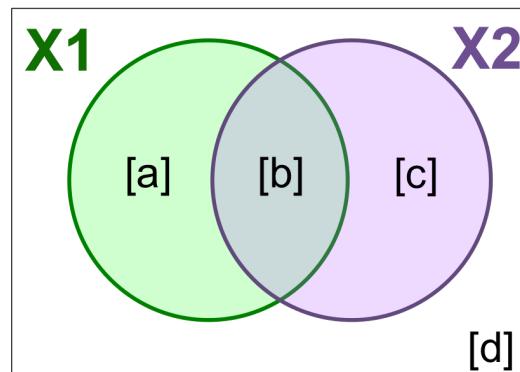
```
# [a+b] Chemistry without controlling for topography
anova.cca(rda(spe.hel, env.chem))
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(X = spe.hel, Y = env.chem)
#          Df Variance      F Pr(>F)
# Model     7  0.30442 4.6102  0.001 ***
# Residual 21  0.19809
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

# Significance testing: X2 [b+c]



```
# [b+c] Topography without controlling for chemistry
anova.cca(rda(spe.hel, env.topo))
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(X = spe.hel, Y = env.topo)
#          Df Variance      F Pr(>F)
# Model     3  0.20867 5.918  0.001 ***
# Residual 25  0.29384
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 ' ' 1
```

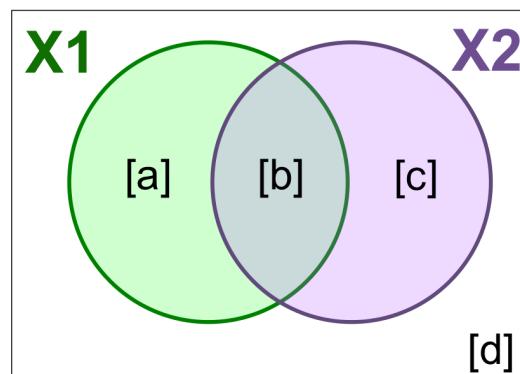
# Significance testing: Individual fractions



```
# [a] Chemistry alone
anova.cca(rda(spe.hel, env.chem, env.topo))
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(X = spe.hel, Y = env.chem, Z = env.topo)
#          Df Variance      F Pr(>F)
# Model     7  0.16024 3.0842  0.001 ***
# Residual 18  0.13360
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

**Note:** Recognize this? It's a partial RDA!

# Significance testing: Individual fractions



```
# [c] Topography alone
anova.cca(rda(spe.hel, env.topo, env.chem))
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(X = spe.hel, Y = env.topo, Z = env.chem)
#          Df Variance      F Pr(>F)
# Model     3 0.064495 2.8965  0.001 ***
# Residual 18 0.133599
# ---
# Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 ' ' 1
```



# Challenge 3

Partition the variation in the mite species data according to substrate variables (SubsDens, WatrCont) and significant spatial variables.

- What proportion of the variation is explained by substrate variables? By space?
- Which individual fractions are significant?
- Plot your results!

Load the spatial variables:

```
data("mite.pcnm")
```

Recall some useful functions:

```
ordiR2step()  
varpart()  
anova.cca(rda())  
plot()
```

# Challenge 3: Solution

Step 1: Forward selection of significant spatial variables

```
# Write full RDA model with all variables
full.spat <- rda(mite.spe.hel ~ ., data = mite.pcnm)

# Forward selection of spatial variables
spat.sel <- ordiR2step(rda(mite.spe.hel ~ 1, data = mite.pcnm),
                        scope = formula(full.spat),
                        R2scope = RsquareAdj(full.spat)$adj.r.squared,
                        direction = "forward",
                        trace = FALSE)

spat.sel$call
# rda(formula = mite.spe.hel ~ V2 + V3 + V8 + V1 + V6 + V4 + V9 +
#      V16 + V7 + V20, data = mite.pcnm)
```

# Challenge 3: Solution

## Step 2: Create variable groups

```
# Subset environmental data to retain only substrate variables  
mite.subs <- subset(mite.env, select = c(SubsDens, WatrCont))  
  
# Subset to keep only selected spatial variables  
mite.spat <- subset(mite.pcnm,  
                      select = names(spat.sel$terminfo$ordered))  
# a faster way to access the selected variables!
```

# Challenge 3: Solution

## Step 3: Variation partitioning

```
mite.part <- varpart(mite.spe.hel, mite.subs, mite.spat)
mite.part$part$indfract # access results!
#           Df R.squared Adj.R.squared Testable
# [a] = X1|X2    2      NA     0.05901929    TRUE
# [b]            0      NA     0.24765221   FALSE
# [c] = X2|X1    10     NA     0.19415929    TRUE
# [d] = Residuals NA     NA     0.49916921   FALSE
```

- What proportion of the variation is explained by substrate variables?
  - **5.9%**
- What proportion of the variation is explained by spatial variables?
  - **19.4%**

# Challenge 3: Solution

Step 4: Which individual fractions are significant?

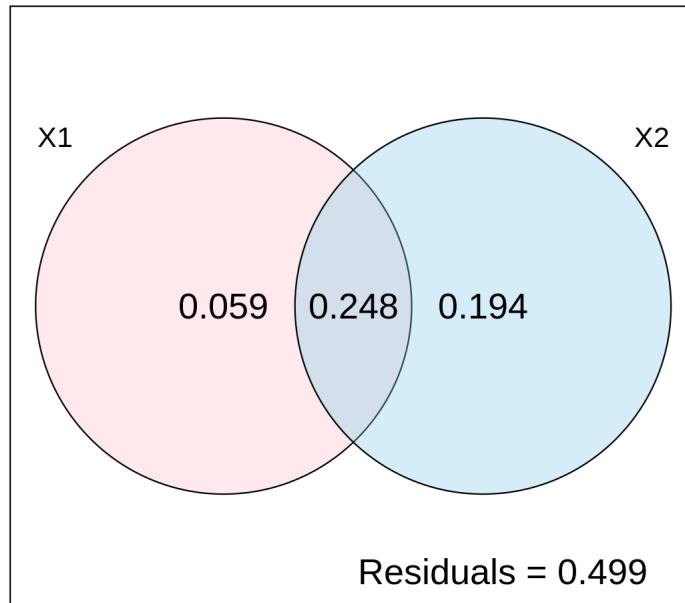
```
# [a]: Substrate only  
anova.cca(rda(mite.spe.hel, mite.subs, mite.spat))  
# p = 0.001 ***  
  
# [c]: Space only  
anova.cca(rda(mite.spe.hel, mite.spat, mite.subs))  
# p = 0.001 ***
```

*So, what can you say about the effects of substrate and space on mite species abundances?*

# Challenge 3: Solution

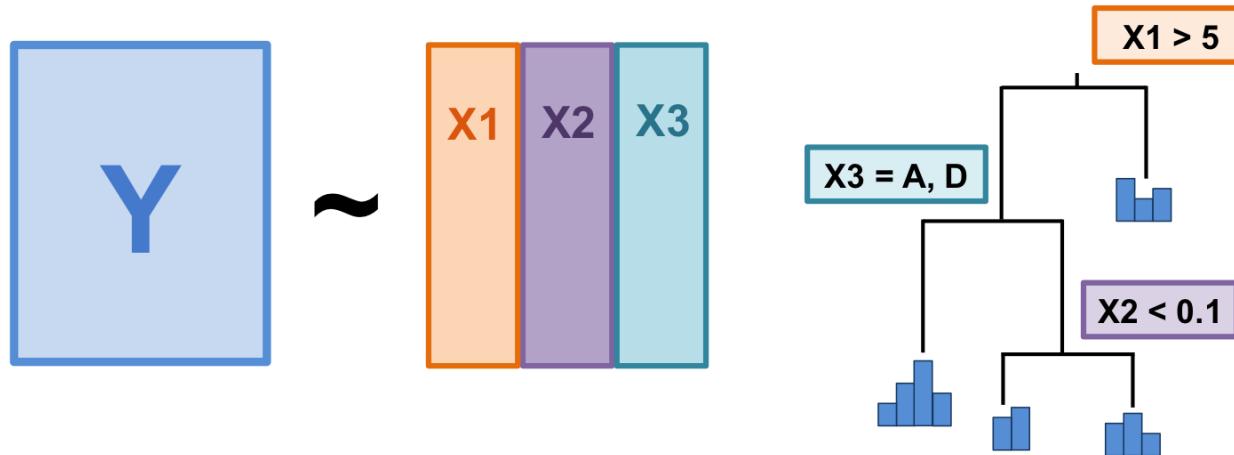
Step 5: Plot the variation partitioning results

```
plot(mite.part, digits = 2, cex = 1.5,  
bg = c("pink", "skyblue"), alpha = 90) # add colour!
```



# Multivariate regression tree (MRT)

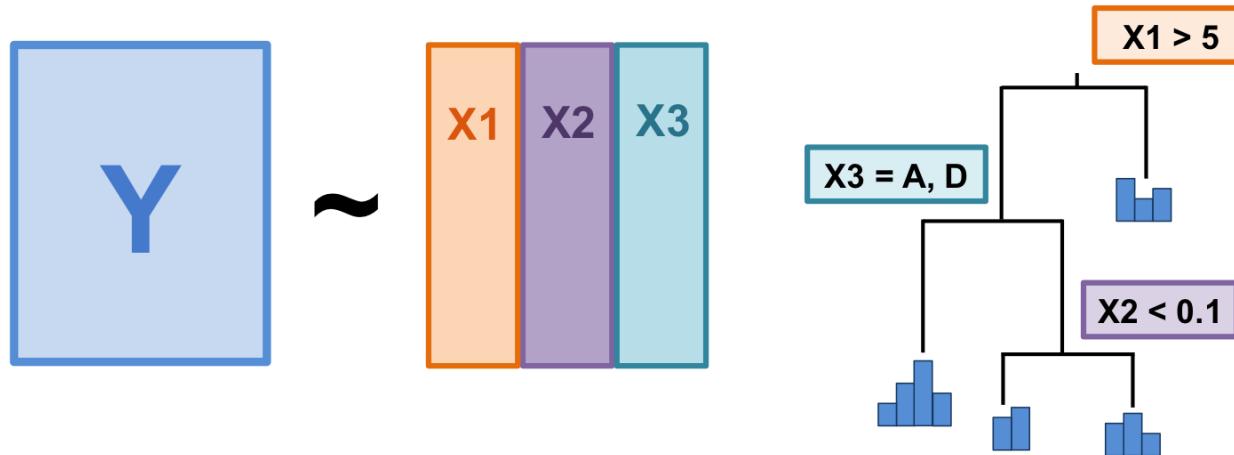
# Multivariate regression tree (MRT)



MRT is a constrained clustering technique.

- Splits a response matrix ( $Y$ ) into clusters based on thresholds of explanatory variables ( $X$ )

# Multivariate regression tree (MRT)



An MRT consists of:

- **Branch**: each group formed by a split
- **Node**: splitting point (threshold value of an explanatory variable)
- **Leaf**: terminal group of sites

# Multivariate regression tree (MRT)

MRT has many advantages:

- Doesn't assume a linear relationship between Y and X matrices
- Results are easy to visualize (it's a tree!)
- Clearly identifies importance of explanatory variables
- Robust (missing values, collinearity)
- Can handle raw explanatory variables

# MRT: Tree selection

When you run an MRT, 2 things happen:

1. Constrained partitioning of the data
2. **Cross-validation** to identify best predictive tree

The "best" tree varies depending on your study goals. Usually you want a tree:

- that is *parsimonious*
- but still has an *informative* number of groups
- Basically: what makes sense for your question?

# MRT in R

In what follows we will be using **mpart** **that is currently archived on CRAN**. We install it from GitHub using the package remotes:

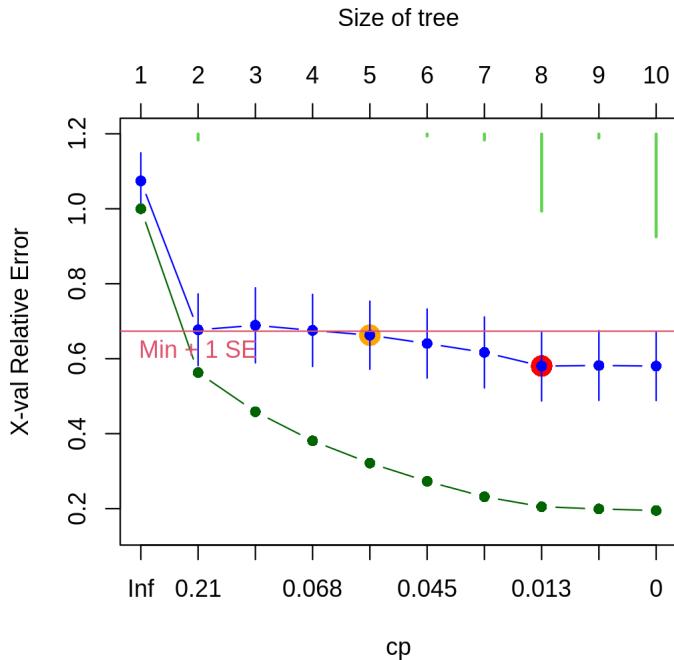
```
remotes::install_github("cran/mvpart")
library(mvpart)
```

# MRT in R

```
# First, remove the "distance from source" variable
env <- subset(env, select = -das)

# Create multivariate regression tree
# library(mvpart)
doubts.mrt <- mvpart(as.matrix(spe.hel) ~ ., data = env,
                      xv = "pick", # interactively select best tree
                      xval = nrow(spe.hel), # number of cross-validations
                      xvmult = 100, # number of multiple cross-validations
                      which = 4, # plot both node labels
                      legend = FALSE, margin = 0.01, cp = 0)
```

# MRT in R: Tree selection

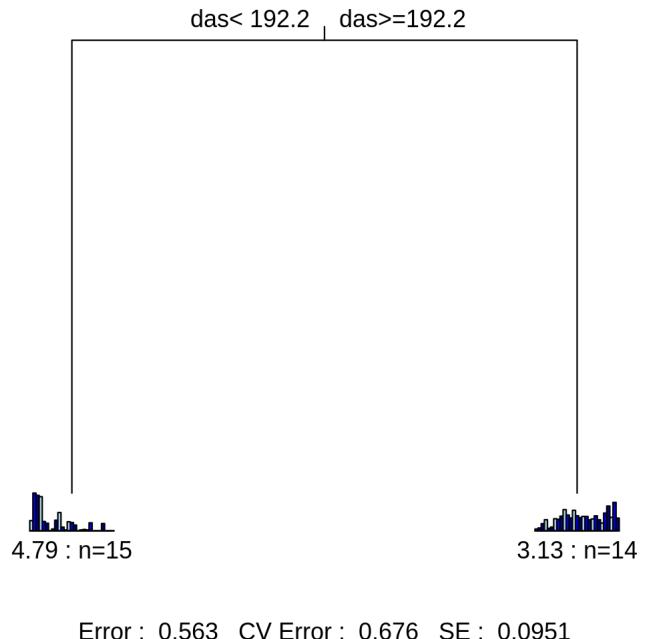


- Green points: Relative error
- Blue points: Cross-validated relative error (CVRE)
- Red dot: Which tree has the smallest CVRE
- Orange dot: Smallest tree within one standard error of the CVRE
- Lime green bars: # of times each tree size was chosen

# MRT in R: Tree selection

- Pick the "best" tree by clicking on a blue dot that corresponds to your chosen tree size!
- We don't have an *a priori* expectation about how to partition this data, so we'll select the *smallest tree within 1 standard error of the overall best-fit tree* (i.e. the orange dot).

# MRT in R: Tree plot

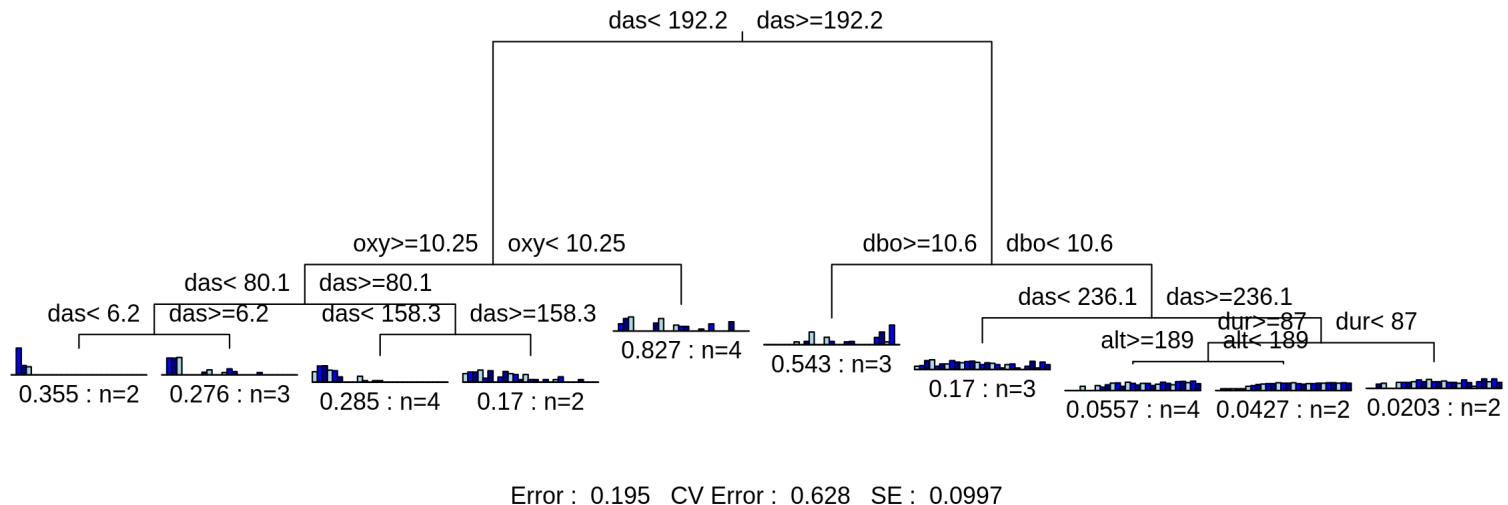


- The species matrix is partitioned according to an **altitude threshold (361.5)**
  - Barplots: species abundances in the sites included in each group
- Residual error = 0.563, which means the model's R<sup>2</sup> is **43.7%**

# MRT in R: Comparing trees

We can also compare solutions, to help us chose the best tree.

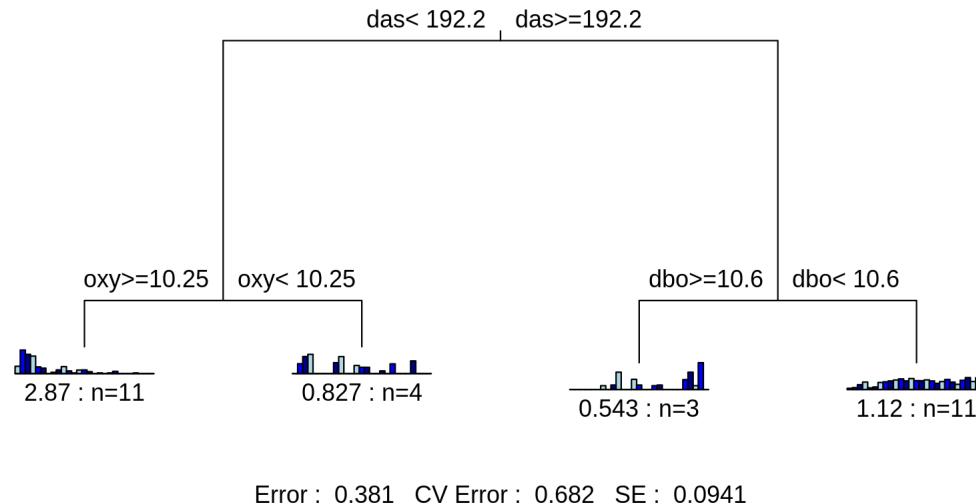
For example, let's look at a 10-group solution!



- This is much **harder to interpret** (so many groups!)
- Higher explanatory power, **BUT** predictive power (CV Error = 0.671) is basically the same as the previous solution (CV Error = 0.673).

# MRT in R: Comparing trees

Let's look at a solution with fewer (4) groups!



- This is easier to interpret!
- Higher explanatory power (**lower Error**) than our original solution
- **Higher predictive power** than both previous solutions (CV Error)

# MRT in R: Complexity parameter

To find out how much variance is explained by each node in the tree, we need to look at the complexity parameter (CP).

```
doub$mr$cptable
#          CP nsplit rel error      xerror      xstd
# 1 0.4369561      0 1.0000000 1.0768059 0.07491763
# 2 0.1044982      1 0.5630439 0.6760624 0.09510184
```

- CP @ nsplit 0 = R<sup>2</sup> of the whole tree
- CP at subsequent nodes = R<sup>2</sup> of each node (see full summary to see which node corresponds to which variable threshold)

# MRT in R: Summary output

We can access more information about the tree (such as which node corresponds to which variable threshold):

```
summary(doubs.mrt)
# Call:
# mpart(form = as.matrix(spe.hel) ~ ., data = env, xv = "1se",
#       xval = nrow(spe.hel), xvmult = 100, xvse = 1, margin = 0.01,
#       which = 4, legend = FALSE, prn = FALSE, cp = 0)
# n= 29
#
#          CP nsplit rel error      xerror        xstd
# 1 0.4369561      0 1.0000000 1.0768059 0.07491763
# 2 0.1044982      1 0.5630439 0.6760624 0.09510184
#
# Node number 1: 29 observations,    complexity param=0.4369561
#   Means=0.07299,0.2472,0.2581,0.2721,0.07133,0.06813,0.06897,0.07664,0.1488,0.2331
#   left son=2 (15 obs) right son=3 (14 obs)
# Primary splits:
#   das < 192.2 to the left,  improve=0.4369561, (0 missing)
#   alt < 361.5 to the right, improve=0.4369561, (0 missing)
#   deb < 23.65 to the left,  improve=0.4369561, (0 missing)
#   amm < 0.06  to the left,  improve=0.3529830, (0 missing)
#   nit < 1.415 to the left,  improve=0.3513335, (0 missing)
```

# MRT in R: Discriminant species

You might also want to know which species are contributing most to the explained variance at each split (i.e. **discriminant species**), or which sites are included within each leaf (group).

To do this, we have the *MVPARTwrap* package.

**MVPARTwrap is archived, we use the package remotes to install it from GitHub:**

```
remotes::install_github("cran/MVPARTwrap")
library(MVPARTwrap)
```

# MRT in R: Discriminant species

```
# Generate a nicer and more informative output
doubts.mrt.wrap <- MRT(doubts.mrt, percent = 10, species = colnames(spe.hel))

# Access the full output:
summary(doubts.mrt.wrap)
```

# MRT in R: Discriminant species

To see each species' contribution to explained variance at each node:

```
summary(doubs.mrt.wrap)
#               Length Class  Mode
# nodes          1    -none- numeric
# pourct        27    -none- numeric
# R2            27    -none- numeric
# obj           13    rpart  list
# percent       1    -none- numeric
# MOYs          54    -none- numeric
# RWHERE         1    -none- list
# LWHERE         1    -none- list
# TABLE1        84    -none- numeric
# LABELS         1    -none- logical
# mat_labels     3    -none- character
```

# MRT in R: Indicator species

You might also be interested in finding out which species are significant **indicator species** for each grouping of sites.

```
library(labdsrv

# Calculate indicator values (indval) for each species
doubs.mrt.indval <- indval(spe.hel, doubs.mrt$where)

# Extract the significant indicator species (and which node they represent)
doubs.mrt.indval$maxcls[which(doubs.mrt.indval$pval <= 0.05)]
# TRU VAI LOC HOT TOX BAR SPI GOU BRO PER BOU PSO ROT CAR BCO PCH GRE GAR BBO ABL
#   1   1   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
# ANG
#   2

# Extract their indicator values
doubs.mrt.indval$indcls[which(doubs.mrt.indval$pval <= 0.05)]
#      TRU      VAI      LOC      HOT      TOX      BAR      SPI      GOU
# 0.8674301 0.7758443 0.7042392 0.8571429 0.6185282 0.6363569 0.7347359 0.6442950
#      BRO      PER      BOU      PSO      ROT      CAR      BCO      PCH
# 0.5533235 0.5449488 0.7857143 0.8070918 0.6352865 0.7307582 0.6428571 0.5000000
#      GRE      GAR      BBO      ABL      ANG
# 0.8571429 0.7726181 0.7142857 1.0000000 0.7857143
```



# Challenge 4

Create a multivariate regression tree for the mite data.

- Select the smallest tree within 1 SE of the CVRE.
- What is the proportion of variance ( $R^2$ ) explained by this tree?
- How many leaves does it have?
- What are the top 3 discriminant species?

Remember to load the mite data:

```
data("mite")
data("mite.env")
```

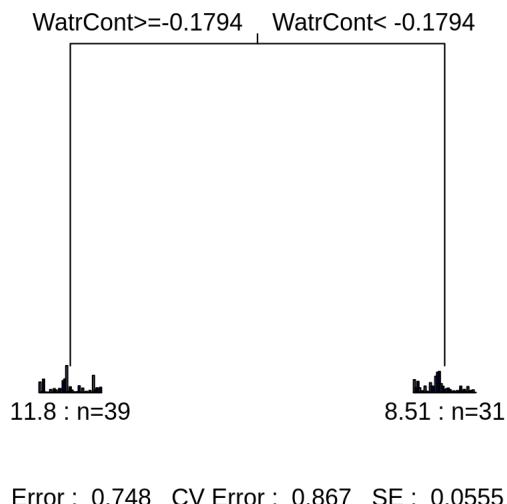
Recall some useful functions:

```
?mvpard() # hint: pay attention to the 'xv' argument!
?MRT()
summary()
```

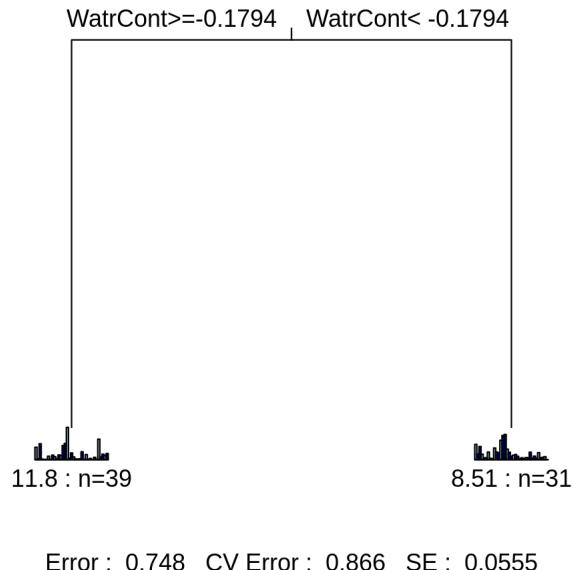
# Challenge 4: Solution

Step 1: Create the multivariate regression tree

```
mite.mrt <- mpart(as.matrix(mite.spe.hel) ~ ., data = mite.env,
                    xv = "1se", # choose smallest tree within 1 SE
                    xval = nrow(mite.spe.hel),
                    xvmult = 100,
                    which = 4, legend = FALSE, margin = 0.01, cp = 0,
                    prn = FALSE)
```



# Challenge 4: Solution



- What is the proportion of variance ( $R^2$ ) explained by this tree?
  - $1 - \text{Error} = 0.252$ , so the tree explains **25.2%** of the variance in the species matrix.
- How many leaves does it have?
  - 2 leaves

# Challenge 4: Solution

What are the top 3 discriminant species **for node 1**?

```
# Generate nicer MRT output
mite.mrt.wrap <- MRT(mite.mrt,
                      percent = 10,
                      species = colnames(mite.spe.hel))

# Look at discriminant species table from MRT output
summary(mite.mrt.wrap)
```

# Challenge 4: Solution

What are the top 3 discriminant species **for node 1**?

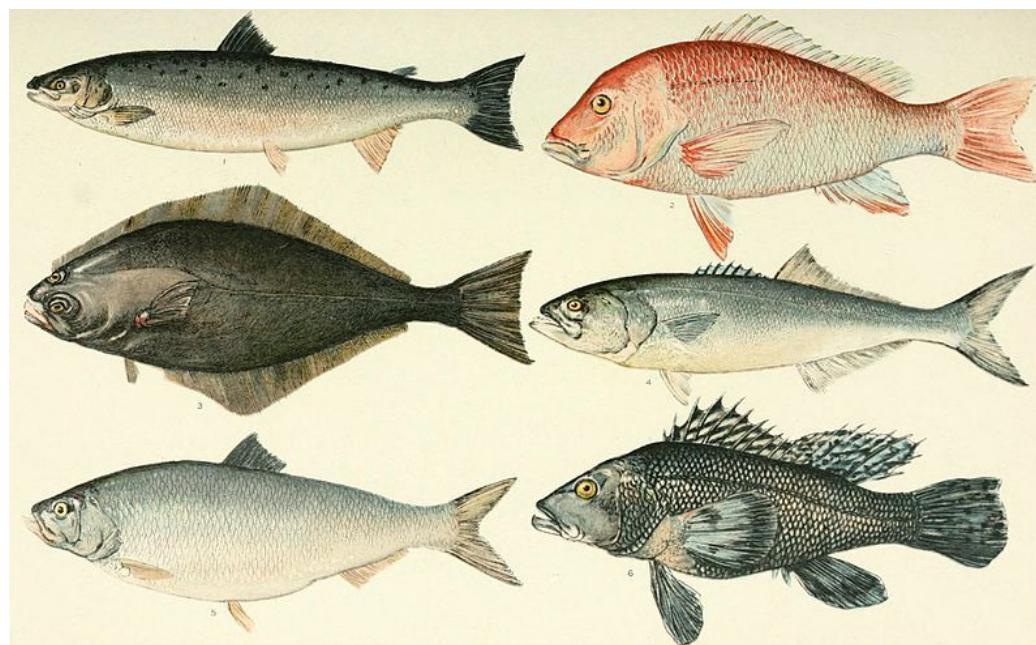
- LCIL, LRUG, Ceratoz1

~ INDVAL species for this node: : left is 1, right is 2			
	cluster	indicator_value	probability
LCIL	1	0.7152	0.001
LRUG	1	0.6683	0.001
Ceratoz1	1	0.4745	0.023
Trhypch1	1	0.4546	0.005
NCOR	1	0.4540	0.005
Trimalc2	1	0.4359	0.002
Ceratoz3	1	0.3963	0.015
TVIE	1	0.3793	0.005
TVEL	2	0.7412	0.001
HMIN	2	0.6421	0.001
FSET	2	0.6361	0.001
ONOV	2	0.6312	0.001
HMIN2	2	0.6193	0.001
SUCT	2	0.6088	0.001
Oribat1	2	0.5978	0.001
Galumna1	2	0.5974	0.001
MEGR	2	0.5770	0.001
RARD	2	0.5585	0.001
PHTH	2	0.5318	0.001
Stgnncrs2	2	0.3898	0.001
Protopl	2	0.2518	0.007
SSTR	2	0.2257	0.007
SLAT	2	0.2249	0.006
Lepidzts	2	0.2108	0.006
Miniglmn	2	0.1880	0.023
MPRO	2	0.1568	0.037

# Linear discriminant analysis (LDA)

# Linear discriminant analysis (LDA)

- Determine how well your descriptor variables explain an *a priori* grouping of your response variable
- Make predictions about how to classify new data
  - e.g. classifying whether a fish comes from a lake or ocean population, based on morphology



# LDA in R: Doubs fish dataset

We know that environmental variables generally change with latitude.

*If we classify our Doubs River sites according to latitude, how well do environmental variables explain these groupings?*

- We can use an LDA to answer this question.

# LDA in R: Doubs fish dataset

Let's begin by loading spatial coordinates for the Doubs sites:

```
# load spatial data for Doubs sites
spa <- read.csv("data/doubsspa.csv", row.names = 1)
spa$site <- 1:nrow(spa) # add site numbers
spa <- spa[-8,] # remove site 8
```

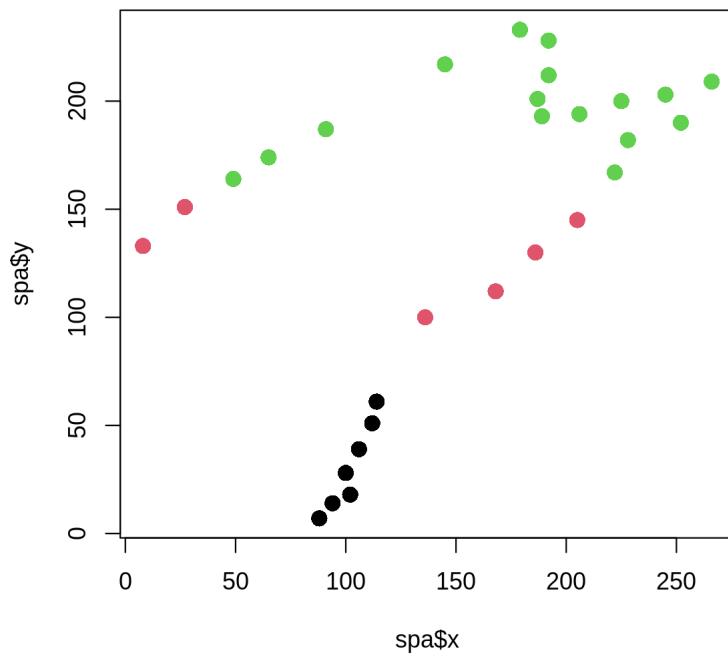
We can then assign sites to 3 latitude groups:

```
# group sites based on latitude
spa$group <- NA # create "group" column
spa$group[which(spa$y < 82)] <- 1
spa$group[which(spa$y > 82 & spa$y < 156)] <- 2
spa$group[which(spa$y > 156)] <- 3
```

# LDA in R: Latitude groups

Let's quickly plot the latitude groupings to see if they make sense:

```
plot(spa$x, spa$y, col = spa$group, pch = 16, cex = 1.5)
```



# LDA in R

**Note:** Usually we would check the multivariate homogeneity of within-group variances before proceeding (see Borcard et al. 2011).

For the purposes of this workshop, we will move straight to doing the LDA:

```
# load required library  
library(MASS)  
  
# run the LDA grouping sites into latitude groups based on env data  
LDA <- lda(env, spa$group)
```

# LDA in R: Grouping accuracy

We can then determine how sites were grouped, and whether this grouping is accurate.

```
# Classification of the objects based on the LDA
spe.class <- predict(LDA)$class

# Posterior probabilities that the objects belong to those groups
spe.post <- predict(LDA)$posterior

# Table of prior vs. predicted classifications
(spe.table <- table(spa$group, spe.class))
#      spe.class
#      1 2 3
# 1 7 0 0
# 2 0 6 0
# 3 0 0 16

# Proportion of corrected classification
diag(prop.table(spe.table, 1))
# 1 2 3
# 1 1 1
```

All sites were correctly classified into the latitude groups based on environmental variables.

# LDA in R: Predictions

We can now use this relationship to *classify new sites into latitude groups*.

Let's **predict the grouping** of 5 new sites using our LDA results:

```
# Load the new site data
classify.me <- read.csv("data/classifyme.csv", header = TRUE)
# classify.me <- classify.me[,-1] # remove das variable

# Predict grouping of new sites
predict.group <- predict(LDA, newdata = classify.me)

# View site classification
predict.group$class
# [1] 1 1 1 3 3
# Levels: 1 2 3
```



# Challenge 5

Create 4 latitude groups in the *mite.xy* dataset. Then, run an LDA to classify mite sites into latitude groupings based on environmental variables (*SubsDens* and *WatrCont*).

- What proportion of sites was correctly classified in group1? in group2?

Load *mite.xy* data:

```
data(mite.xy)
```

Recall some useful functions:

```
lda()  
predict()  
table()  
diag()
```

# Challenge 5: Solution

Step 1: Create 4 latitude groups

```
# assign numbers to sites
mite.xy$site <- 1:nrow(mite.xy)

# find latitudinal range for each group
(max(mite.xy[,2])-min(mite.xy[,2]))/4
# [1] 2.4

# group sites into 4 latitude groups
# group sites based on latitude
mite.xy$group <- NA # create "group" column
mite.xy$group[which(mite.xy$y < 2.5)] <- 1
mite.xy$group[which(mite.xy$y >= 2.5 & mite.xy$y < 4.9)] <- 2
mite.xy$group[which(mite.xy$y >= 4.9 & mite.xy$y < 7.3)] <- 3
mite.xy$group[which(mite.xy$y >= 7.3)] <- 4
```

Step 2: Run the LDA

```
LDA.mite <- lda(mite.env[,1:2], mite.xy$group)
```

# Challenge 5: Solution

Step 3: Check whether the groupings are correct

```
# classification of the objects based on LDA
mite.class <- predict(LDA.mite)$class
# table of prior versus predicted classifications
(mite.table <- table(mite.xy$group, mite.class))
#     mite.class
#   1   2   3   4
# 1  9   4   2   0
# 2  2  11   4   0
# 3  1   2  14   2
# 4  0   0   3  16
# proportion of correct classification
diag(prop.table(mite.table, 1))
#           1             2             3             4
# 0.6000000 0.6470588 0.7368421 0.8421053
```

What proportion of sites was correctly classified in group1? in group2?

- **60%** were correctly classified into group1, and **64.7%** were classified into group2.

**Thank you for attending this workshop!**

