



Atelier 10: Analyses multivariées avancées

Série d'ateliers R

Centre de la Science de la Biodiversité du Québec



À propos de cet atelier



Packages requis

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

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

Objectifs d'apprentissage

Utiliser R pour faire des ordinations non-constraines

Introduction

Introduction

L'atelier précédent (#9) a offert un aperçu des analyses multivariées de base:

- Mesures de distances et transformations de données
- Groupement hiérarchique
- Ordinations sans contraintes (PCA, PCoA, CA, nmDS)

Celles-ci permettent de relever des **tendances** dans la structure des communautés d'espèces ou des descripteurs.

L'atelier #10 montre des analyses permettant d'explorer comment les variables environnementales **expliquent** ces tendances.

Introduction

Le présent atelier se concentrera sur les analyses **sous contraintes**:

- Analyse canonique de redondances (ACR ou RDA)
- Partitionnement de la variation
- Arbre de régression multivarié (ARM ou MRT)
- Analyse linéaire discriminante (ALD ou LDA)

Ces analyses nous permettront de **décrire** et de **prédir** les relations entre la structure des communautés et les variables environnementales.

On pourra alors **tester des hypothèses!**

Code et données

Lien vers le code et les jeux de données: qcbs.ca/wiki/r_workshop10

Téléchargez le code R et les données requises pour cet atelier:

- Code R
- Données:
 - DoubsEnv
 - DoubsSpe
 - DoubsSpa
 - Données test pour l'analyse linéaire discriminante

Librairies

Assurez-vous d'installer et d'importer les librairies suivantes dans R Studio (procédure fournie dans le code R):

- *vegan* (for multivariate analyses)
- *labdsv* (pour identification d'espèces indicatrices pour l'arbre de régression multivarié))
- *plyr* (classification pour l'analyse linéaire discriminante)
- *MASS* (for l'analyse linéaire discriminante)
- *rdaTest* package (**voir code R**)
- *mvpart* package (**voir code R**)
- *MVPARTwrap* package (**voir code R**)

Suivre l'atelier

Quelques conseils:

- Créez votre propre code (ou commentez le code R fourni)
- Évitez de copier-coller, ou d'exécuter le code directement du script fourni
- N'oubliez pas de bien définir votre répertoire de travail (le dossier contenant les fichiers requises pour l'atelier)

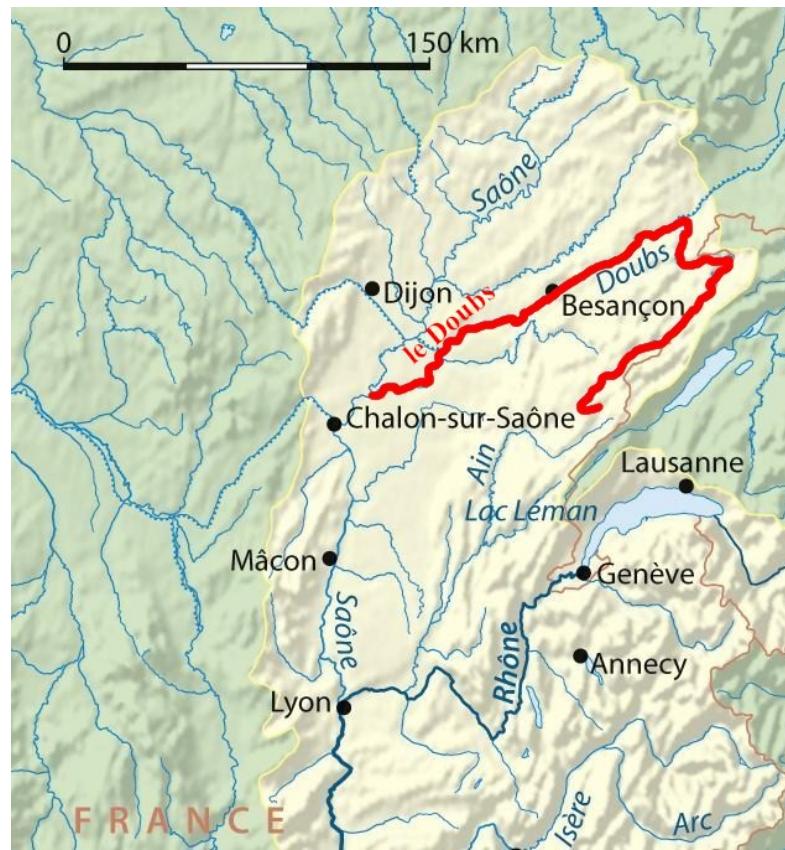
Exploration et préparation des données

Introduction aux données

Rivière Doubs (Verneaux 1973)

Données d'abondances d'espèces de communautés de poissons de la rivière Doubs.

- 27 espèces
- 30 sites
- 11 variables environnementales



Charger les données

Assurez-vous que les fichiers se trouvent dans votre répertoire de travail!

Chargez la matrice d'abondances d'espèces (*doubsspe.csv*):

```
# Assurez vous que les fichiers se trouvent dans votre répertoire de travail!
spe ← read.csv("data/doubsspe.csv", row.names = 1)
spe ← spe[-8,] # Supprimer site 8 (pas d'espèces).
```

Chargez la matrice de données environnementales (*doubsenv.csv*):

```
env ← read.csv("data/doubsenv.csv", row.names = 1)
env ← env[-8,] # Supprimer site 8 (pas d'espèces).
```

Note: N'exécuter qu'une seule fois!

Données d'abondances d'espèces

Explorons la matrice des abondances d'espèces:

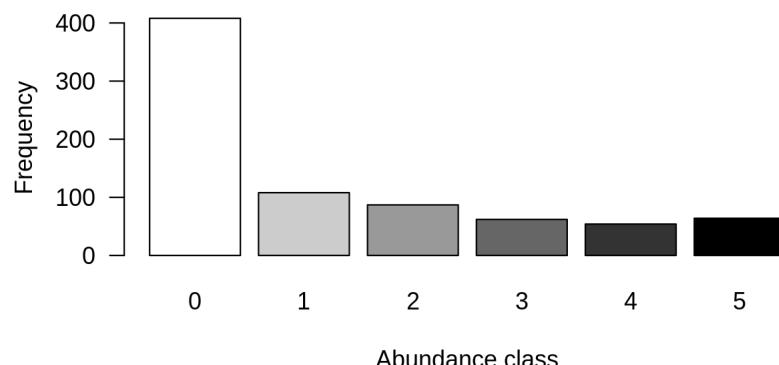
```
names(spe) # voir les noms des colonnes (espèces)
dim(spe) # dimensions de la matrice
head(spe) # 5 premières lignes
str(spe) # structure interne de la matrice
summary(spe) # statistiques descriptives des objets (min, moyenne, max, etc.)
```

[1] 29 27

Distribution des abondances d'espèces

Explorons la structure de la communauté:

```
# Compter la fréquence d'espèces dans chaque classe d'abondance  
ab ← table(unlist(spe))  
# Visualiser cette distribution  
barplot(ab, las = 1,  
        xlab = "Abundance class", ylab = "Frequency",  
        col = grey(5:0/5))
```



Notice: Il y a beaucoup de zéros.

Distribution des abondances d'espèces

Comptez le nombre absences dans les données d'abondances.

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

Regardez la proportion d'absences dans les données d'abondances.

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

Transformation des données d'abondances

Plus de 50% des données d'abondances consiste d'absences. C'est élevé, mais pas inhabituel pour ce type de données.

Par contre, il faut éviter que les **double zéros** soient considérés comme une similarité entre sites.

- Nous appliquerons alors une **transformation Hellinger** aux données d'abondances d'espèces.

```
# la fonction decostand() dans la librairie vegan nous facilite la tâche:
```

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

Données environnementales

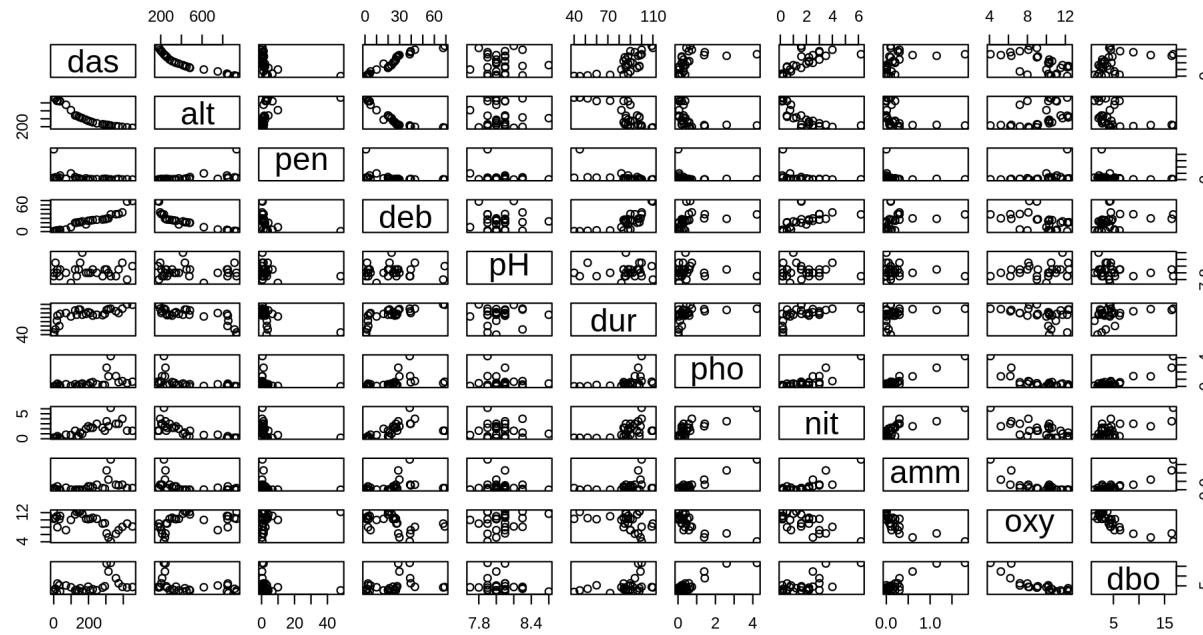
Explorons les données environnementales:

```
names(env) # noms des objets (variables environnementales)
dim(env) # dimensions de la matrice
head(env) # 5 premières lignes
str(env) # structure des objets
summary(env) # statistiques descriptives (min, moyenne, max, etc.)

# [1] "das" "alt" "pen" "deb" "pH"   "dur" "pho" "nit" "amm" "oxy" "dbo"
# [1] 29 11
```

Colinéarité

On peut également détecter (visuellement) les colinéarités entre variables:
pairs(env)



Note: Colinéarité entre quelques variables... (das vs. alt, das vs. deb, das vs. dur, das vs. nit, oxy vs. dbo, etc.)

Standardisation des données

Il est impossible de comparer les effets de variables d'unités différentes.
Avant d'effectuer les analyses qui suivent, les données doivent donc être **standardisées**.

```
# standardiser les données
env.z ← decostand(env, method = "standardize")

# centrer les données (moyenne ~ 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

# réduire les données (écart type = 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
```

Analyses canoniques

Analyses canoniques

Les analyses canoniques nous permettent:

- d'identifier les *relations* entre un ensemble de variables réponses et un ensemble de variables explicatives
- de tester des **hypothèses écologiques** à propos de ces relations
- de faire des *prédictions*

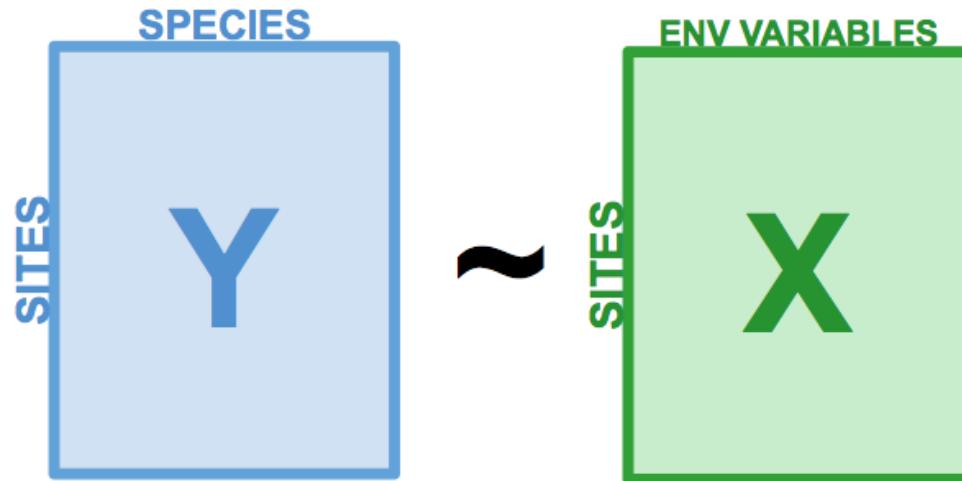
Analyses canoniques

Analyse canonique de redondances (RDA)

Analyse canonique de redondances (RDA)

L'analyse canonique de redondances est une ordination **sous contraintes**.

- extension directe de la régression multiple.
- modélise l'effet d'une matrice X (variables explicatives) sur une matrice Y (variables réponses)



Variables peuvent être quantitatives, qualitatives, ou binaires (0/1).

- **transformez** et **standardisez** les variables avant d'effectuer une RDA.

Effectuer une RDA dans R

Préparer les données

```
# On utilisera nos données explicatives standardisées  
# Enlever la variable "distance from the source" (colinéarity avec plusieurs vari  
env.z ← subset(env.z, select = -das)
```

Effectuer une RDA

```
# Modélise l'effect de tous les variables environnementales sur la composition en  
spe.rda ← rda(spe.hel ~ ., data = env.z)
```

Extraire les résultats de la RDA

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

Sortie d'une RDA

```
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 de Y expliquée par X (**73.41%**)
- **Unconstrained Proportion:** variance in Y non expliquée par (**26.59%**)

Les variables environnementales mesurées expliquent **73.41%** de la variation dans la composition en espèces des communautés de poissons dans la rivière Doubs.

Sélection de variables

Une **sélection progressive** peut être effectuée afin de sélectionner les variables explicatives significatives.

Quelles variables contribuent de façon significative au pouvoir explicatif du modèle?

```
# Sélection progressive de variables:  
fwd.sel ← ordiR2step(rda(spe.hel ~ 1, data = env.z), # modèle le plus simple  
                      scope = formula(spe.rda), # modèle "complet"  
                      direction = "forward",  
                      R2scope = TRUE, # limité par le R2 du modèle "complet"  
                      pstep = 1000,  
                      trace = FALSE) # mettre TRUE pour voir le processus du sélection!
```

Essentiellement, on ajoute une variable à la fois au modèle, et on retient la variable si elle augmente significativement le R2 ajusté du modèle.

Sélection de variables

- Quelles variables ont été sélectionnées?

```
fwd.sel$call  
# rda(formula = spe.hel ~ alt + oxy + dbo, data = env.z)
```

- Quel est le R2 ajusté d'une RDA incluant seulement les variables significatives?

```
spe.rda.signif ← rda(spe.hel ~ alt + oxy + dbo, data = env.z)  
RsquareAdj(spe.rda.signif)  
# $r.squared  
# [1] 0.5894243  
#  
# $adj.r.squared  
# [1] 0.5401552
```

Tester la significativité d'une RDA

Utilisez **anova.cca()** pour tester la significativité globale de notre RDA

```
anova.cca(spe.rda.signif, permutations = 1000)
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 1000
#
# Model: rda(formula = spe.hel ~ alt + oxy + dbo, data = env.z)
#          Df Variance      F   Pr(>F)
# Model     3  0.29619 11.963 0.000999 *** 
# Residual 25  0.20632
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On peut aussi tester la significativité des axes!

```
anova.cca(spe.rda.signif, permutations = 1000, by = "axis")
```

Représentation graphique des RDAs

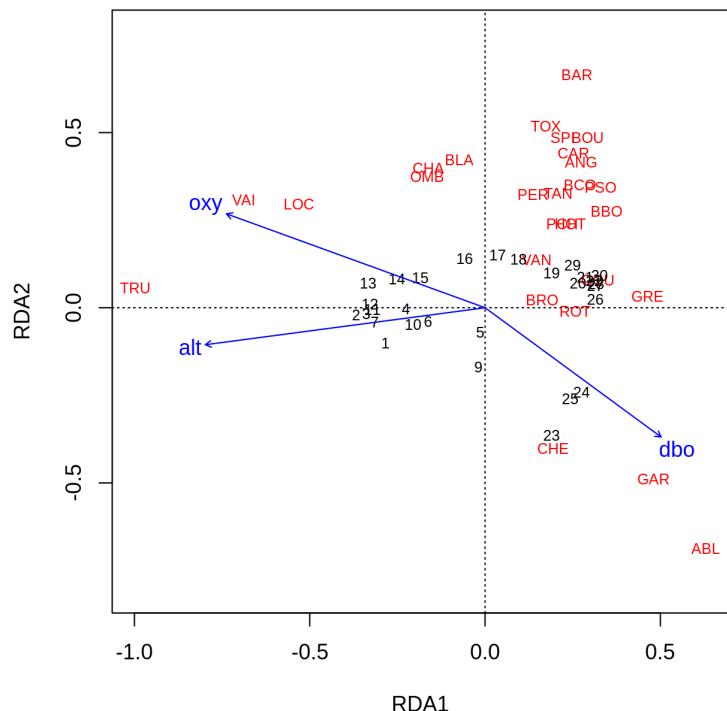
Une RDA permet la **visualization simultanée** des variables réponses et explicatives (*i.e. espèces et variables environnementales*).

Comme pour la PCA, on doit choisir le **cadrage**:

Type 1	Type 2
distances entre objects \approx distances euclidiennes	angles entre variables \approx leur corrélation

Triplot RDA: Cadrage de type 1

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

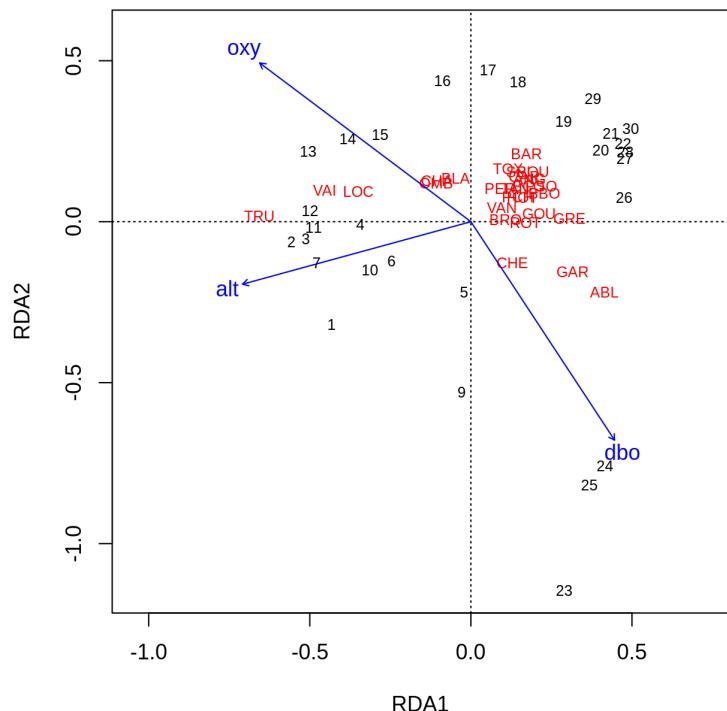


Le cadrage 1 permet d'interpréter les **distances** entre objets (espèces).

- Les communautés dans les sites (chiffres) *plus rapprochés* ont des compositions plus similaires.
- Les espèces *plus rapprochées* occupent souvent les mêmes sites.

Triplot RDA: Cadrage de type 2

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

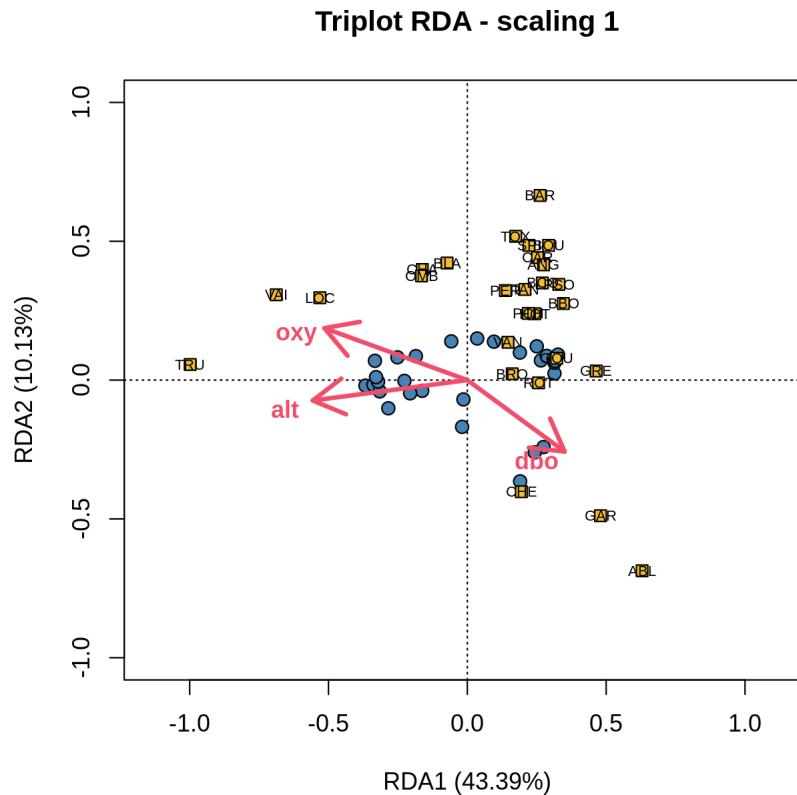


Le cadrage 2 permet d'interpréter les **relations** entre variables X et Y.

- Longues flèches = cette variable explique fortement la variation dans la matrice Y d'abondances
- Flèches pointant des *directions opposées* = relation négative
- Flèches pointant la *même direction* = relation positive

Configuration des triplots RDA

Les fonctions `plot()` et `ordiplot()` produisent des triplots rapidement et facilement, mais on peut aussi configurer les graphiques manuellement.



Voir le wiki pour plus de détails!

Défi 1



Effectuer une RDA pour modéliser les effect des variables environnementales (*mite.env*) sur l'abondance des espèces d'acariens (*mite*).

Chargez les données:

```
# Charger les données d'abondance des espèces d'acariens  
data("mite")  
  
# Charger les données environnementales  
data("mite.env")
```

Rappel de fonctions utiles:

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

Défi 1: Solution

Étape 1: Préparer les données

```
# Transformer les données d'abondances  
mite.spe.hel ← decostand(mite, method = "hellinger")  
  
# Standardiser les données environnementales quantitatives  
mite.env$SubsDens ← decostand(mite.env$SubsDens, method = "standardize")  
mite.env$WatrCont ← decostand(mite.env$WatrCont, method = "standardize")
```

Défi 1: Solution

Étape 2: Sélectionner les variables environnementales

```
# RDA avec tous les variables environnementales
mite.spe.rda ← rda(mite.spe.hel ~ ., data = mite.env)

# Sélection progressive des variables environnementales significatives
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)
```

Défi 1: Solution

Étape 3: Effectuer l'RDA et extraire le R2 ajusté

```
# Refaire la RDA avec seulement les variables significatives
mite.spe.rda.signif ← rda(mite.spe.hel ~ WatrCont + Shrub +
                           Substrate + Topo + SubsDens,
                           data = mite.env)

# Calculer le R2 ajusté
RsquareAdj(mite.spe.rda.signif)$adj.r.squared
# [1] 0.4367038
```

Défi 1: Solution

Étape 4: Tester la signification globale du modèle

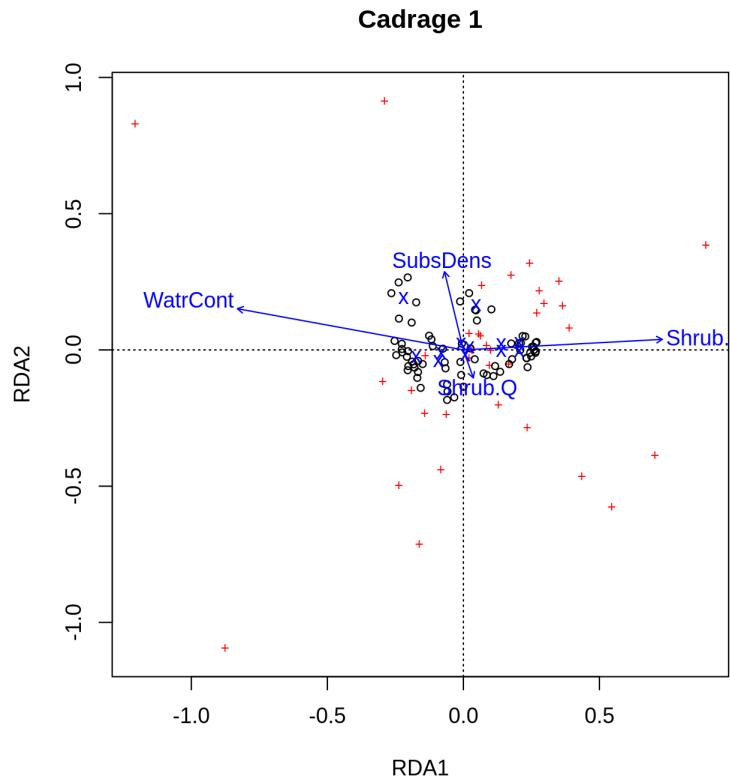
```
anova.cca(mite.spe.rda.signif, step = 1000)
# Permutation test for rda under reduced model
# Permutation: free
# Number of permutations: 999
#
# Model: rda(formula = mite.spe.hel ~ WatrCont + Shrub + Substrate + Topo + SubsD
#           Df Variance      F Pr(>F)
# Model     11  0.20759 5.863  0.001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

Les variables environnementales sélectionnées expliquent **43.7% (p = 0.001)** de la variation dans la composition de communautés des acariens.

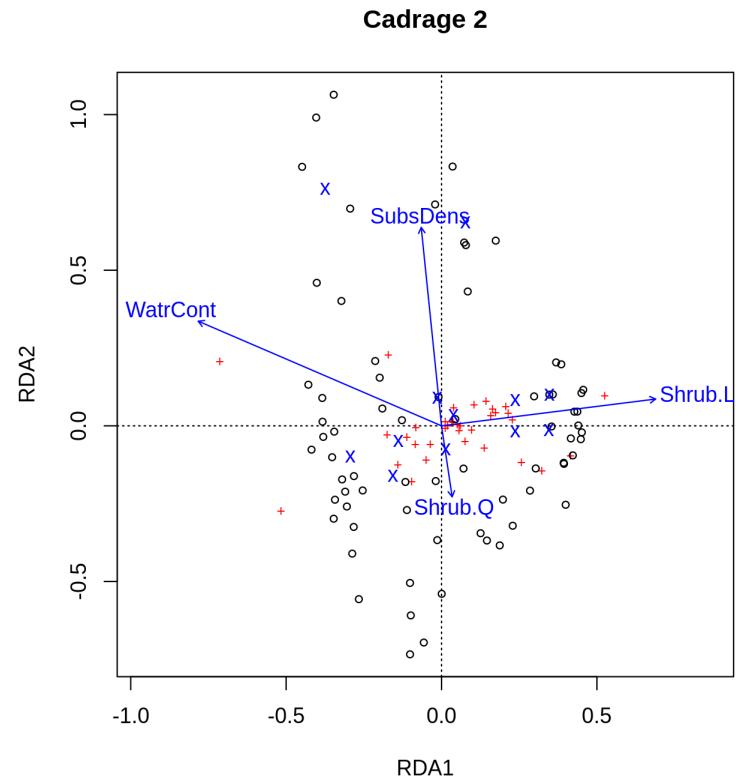
Défi 1: Solution

Étape 5: Triplot!

```
ordiplot(mite.spe.rda.signif,  
         scaling = 1,  
         main = "Cadrage 1")
```



```
ordiplot(mite.spe.rda.signif,  
         scaling = 2,  
         main = "Cadrage 2")
```

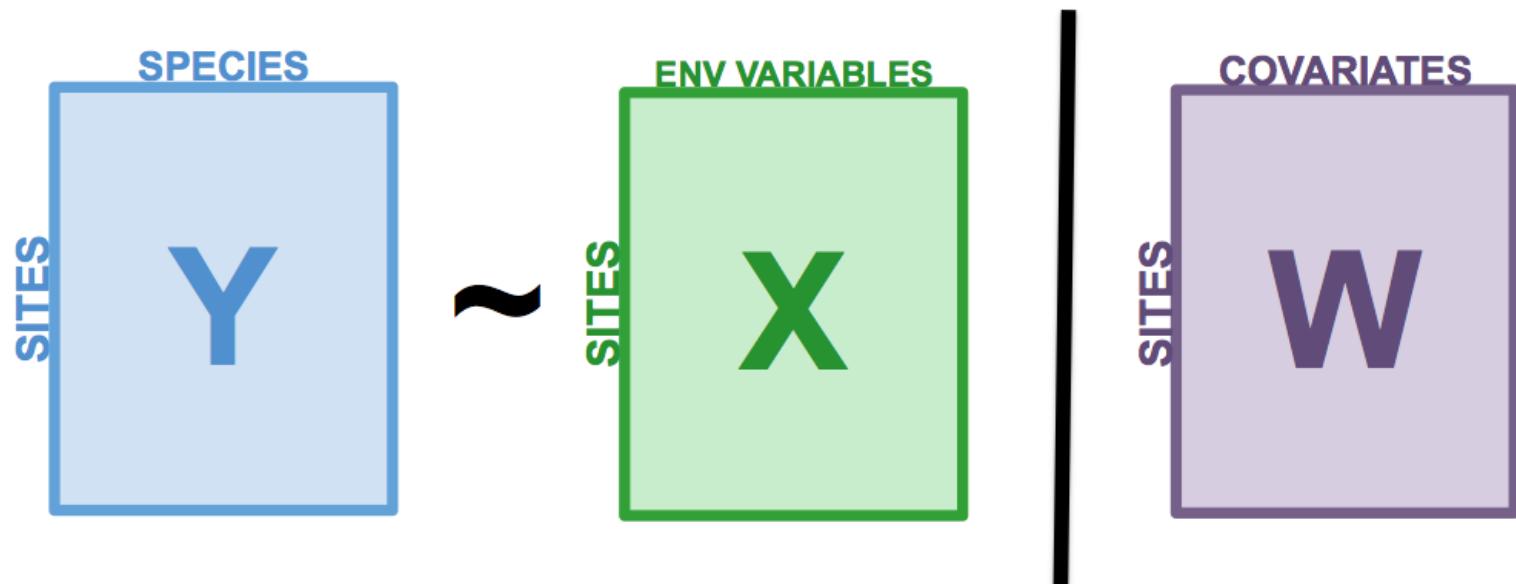


Analyses canoniques

RDA partielle

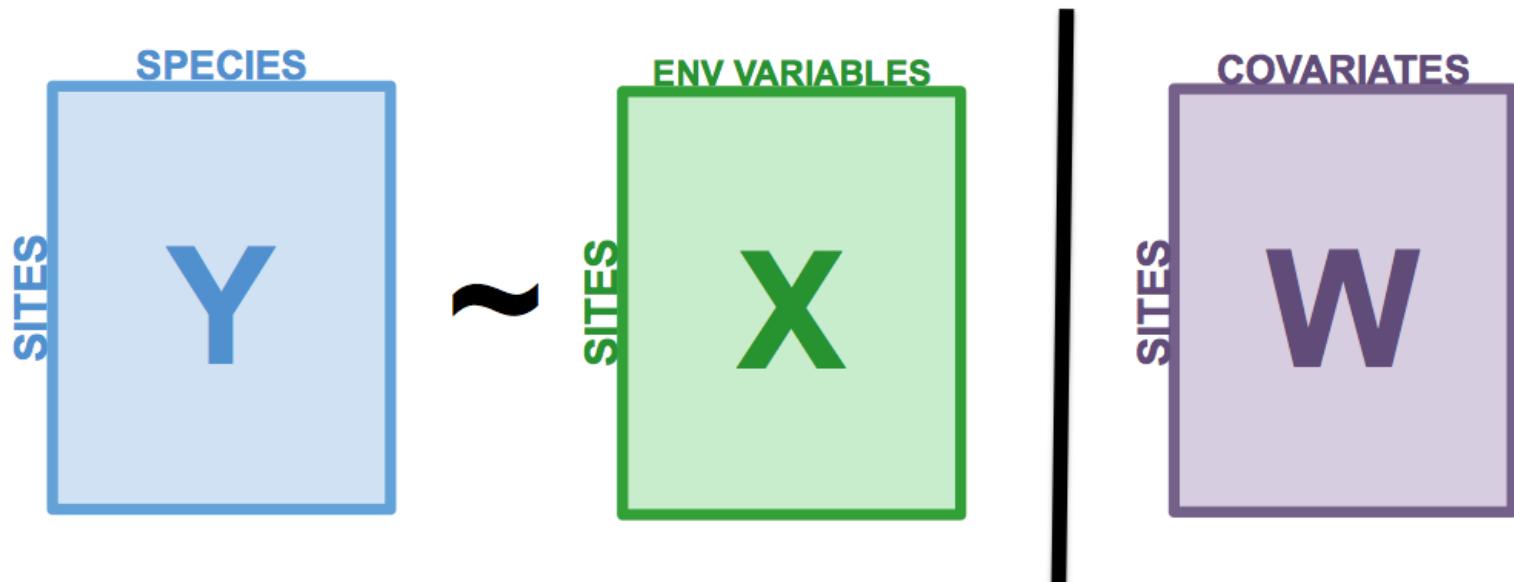
RDA partielle

- Cas particulier de la RDA en présence d'une matrice W de variables explicatives additionnelles, appelées co-variables
 - Modèle linéaire de l'effet de X sur Y, ajusté pour tenir compte de l'effet des **co-variables W**.



Applications de la RDA partielle

- Évaluer l'effet de variables environnementales sur la composition des communautés, prenant en compte l'**effet de covariables** de moindre intérêt.
- **Isoler** les effets d'un ou plusieurs groupes de variables explicatives.



RDA partielle: données Doubs

Évaluons l'effet de la *chimie de l'eau* (X) sur l'abondance des poissons en tenant compte de *covariables topographiques* (W).

```
# Divisez le tableau de données environnementales en deux:  
# variables topographiques et chimiques  
env.topo ← subset(env.z, select = c(alt, pen, deb))  
env.chem ← subset(env.z, select = c(pH, dur, pho, nit, amm, oxy, dbo))  
  
# Faire la RDA partielle  
spe.partial.rda ← rda(spe.hel, env.chem, env.topo)
```

Note: Syntaxe alternative pour une RDA partielle:

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

Résultats d'une RDA partielle

```
# Extraire les résultats  
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 de Y expliquée par W (**41.53%**)
- **Constrained Proportion:** variance de Y expliquée par X (**31.89%**)
- **Unconstrained Proportion:** variance de Y non expliquée (**26.59%**)

*La chimie de l'eau explique **31.89%** de l'abondance des espèces de poissons, tandis que la topographie explique **41.53%** de la variation en abondances des poissons.*

Tester la significativité

Extraire le R2 ajusté du modèle:

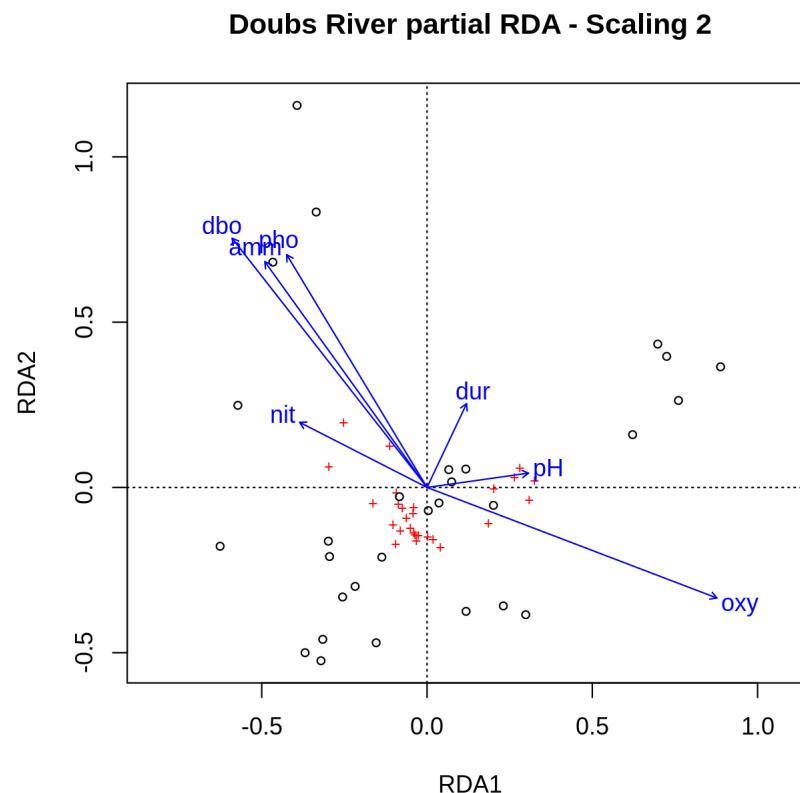
```
RsquareAdj(spe.partial.rda)$adj.r.squared  
# [1] 0.2413464
```

Ensuite, tester la significativité globale de notre RDA partielle

```
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
```

Représentation graphique

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





Défi 2

Effectuez une RDA partielle de l'abondance des espèces de mites en fonction des variables environnementales, tenant compte de l'effet du substrat (SubsDens, WaterCont and Substrate).

- Quel pourcentage de variance est expliqué par les variables environnementales?
- Le modèle est-il significatif?
- Quels sont les axes significatifs?

Rappel des données et fonctions utiles:

```
mite.spe.hel  
mite.env  
rda()  
summary()  
RsquareAdj()  
anova.cca()
```

Défi 2: Solution

Nos données sont déjà transformés et standardisés.

Commençons alors par la RDA partielle:

```
mite.spe.subs ← rda(mite.spe.hel ~ Shrub + Topo  
+ Condition(SubsDens + WatrCont + Substrate),  
data = mite.env)  
  
# Extraire les résultats  
summary(mite.spe.subs, display = NULL)
```

Shrub et Topo expliquent **9.8%** de la variation de l'abondance de mites, tandis que le substrat explique **42.8%** de cette variation.

Défi 2: Solution

- Le modèle est-il significatif?

```
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 + WatrCor
#           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
```

Défi 2: Solution

- Quels sont les axes significatifs?

```
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 + WatrCor
#           Df Variance      F Pr(>F)
# RDA1      1 0.027236 8.4618  0.001 ***
# RDA2      1 0.008254 2.5643  0.019 *
# RDA3      1 0.003193 0.9919  0.437
# Residual 58 0.186688
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

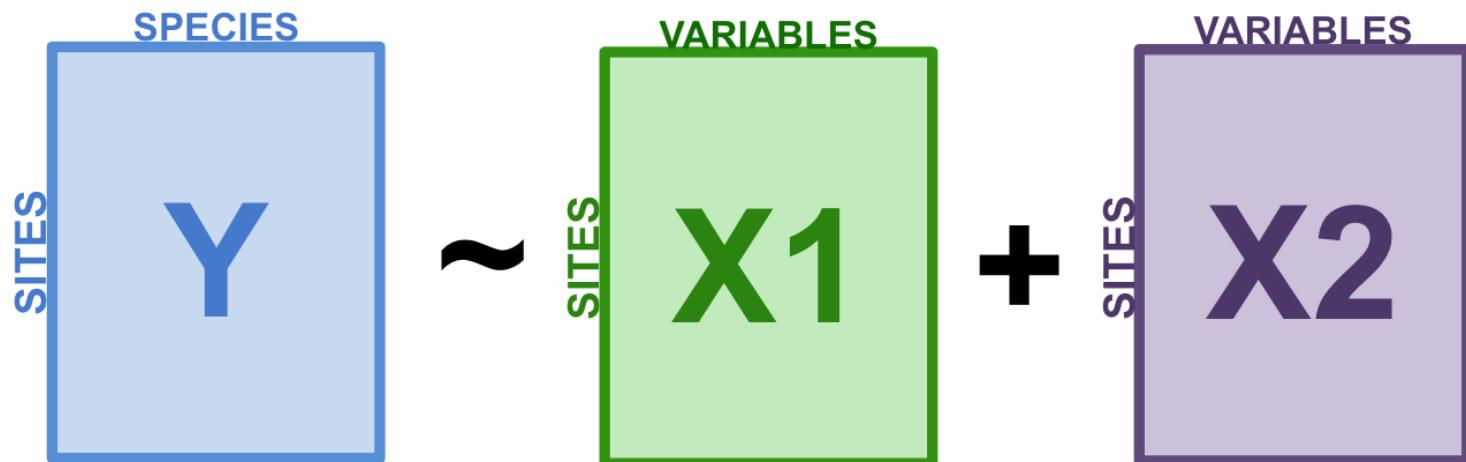
Analyses canoniques

Partitionnement de la variation

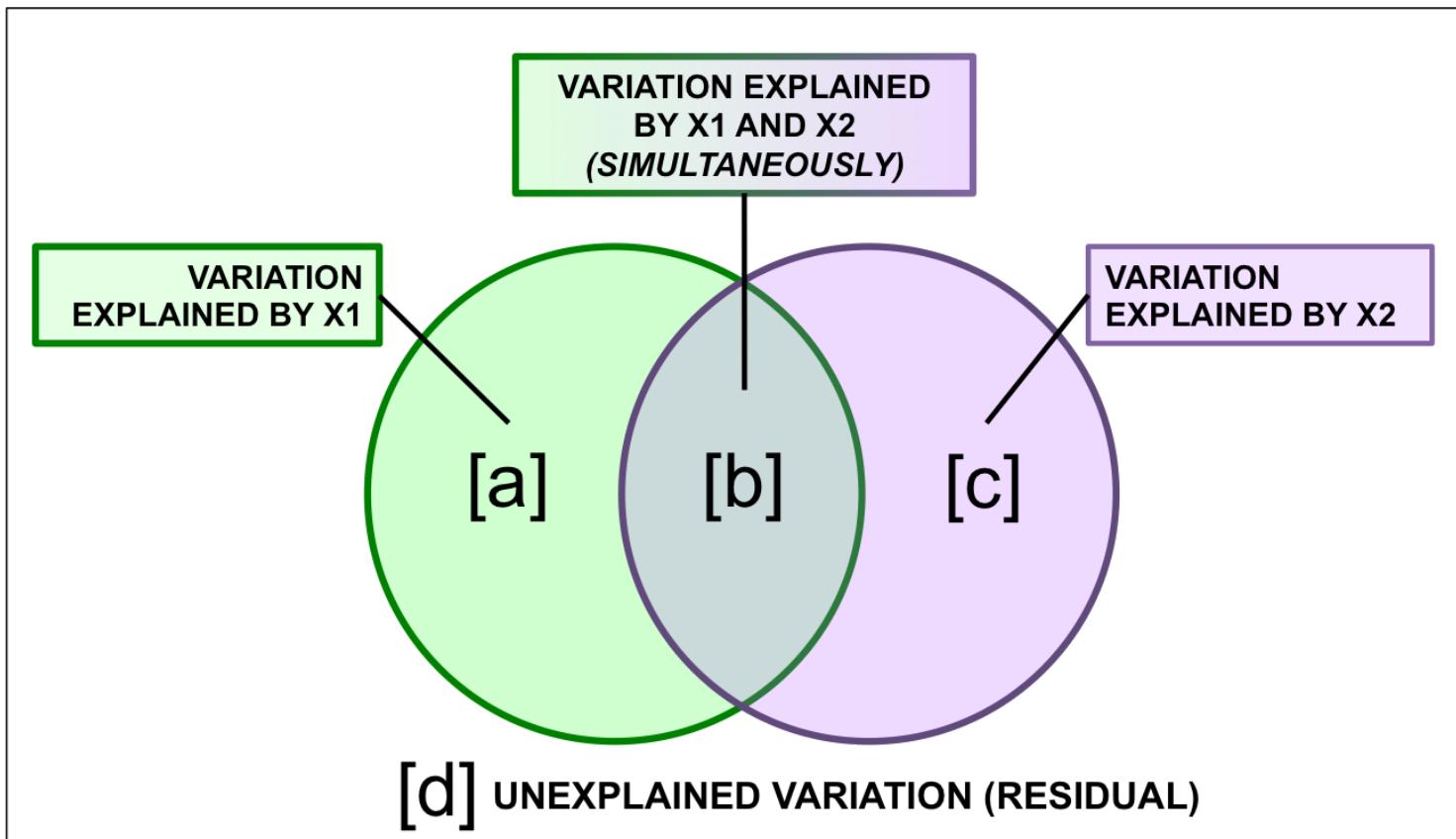
Partitionnement de la variation

Divise la variation d'une matrice de variable réponse en 2, 3, ou 4 matrices de variables explicatives.

- e.g. variables locales vs. à large échelle
- e.g. abiotique vs. biotique



Partitionnement de la variation



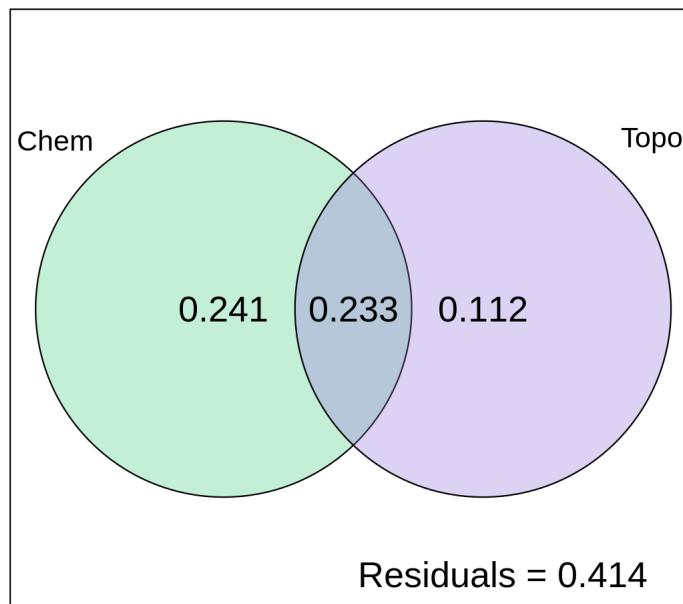
Partitionnement de la variation dans R

Note: Assurez-vous que la librairie *vegan* est chargée!

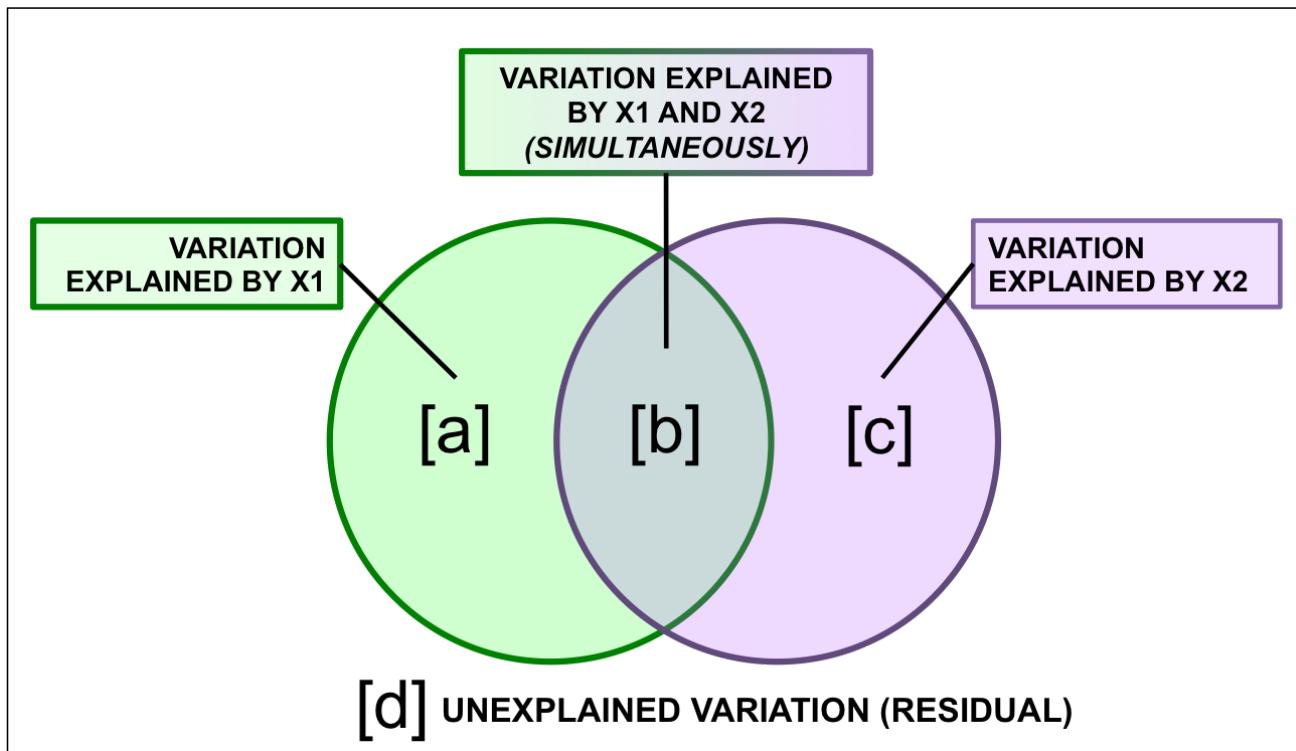
```
spe.part.all <- varpart(spe.hel, env.chem, env.topo)
spe.part.all$part # extraire résultats
# No. of explanatory tables: 2
# Total variation (SS): 14.07
#           Variance: 0.50251
# No. of observations: 29
#
# Partition table:
#                               Df R.squared Adj.R.squared Testable
# [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
```

Diagramme Venn

```
plot(spe.part.all,
  Xnames = c("Chem", "Topo"), # noms des matrices explicatives
  bg = c("seagreen3", "mediumpurple"), alpha = 80,
  digits = 2,
  cex = 1.5)
```

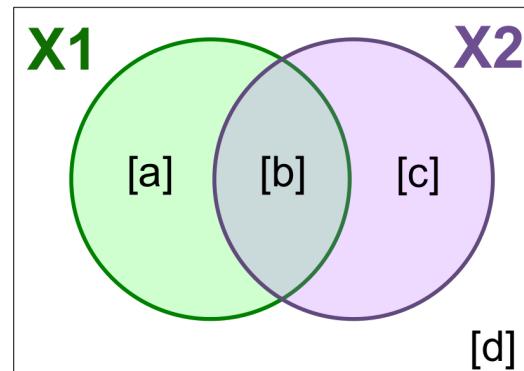


Tester la significativité



- La significativité de la fraction partagée [b] ne peut **pas** être testée.
- Mais, on peut tester la significativité des autres fractions!

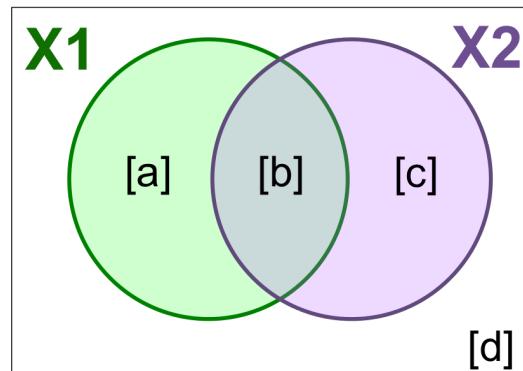
Significativité: X1 [a+b]



[a+b] Chimie sans tenir compte de topographie

```
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
```

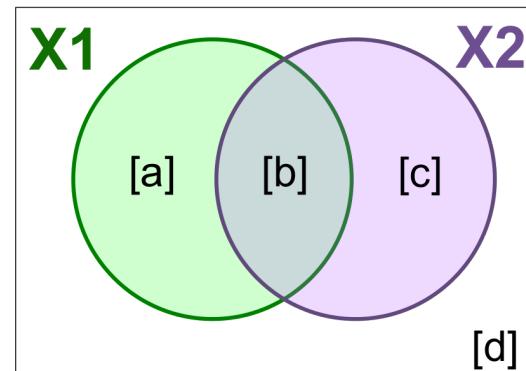
Significativité: X2 [b+c]



[b+c] Topographie sans tenir compte de chimie

```
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
```

Significativité: Fractions individuelles

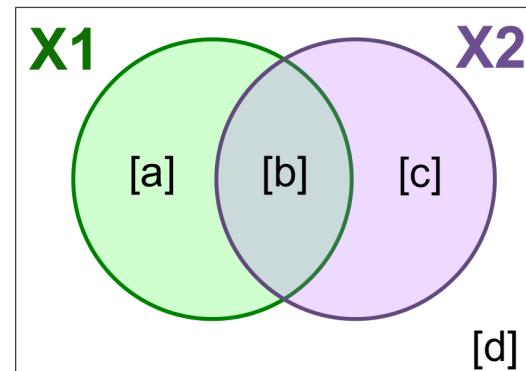


[a] Chimie (ajusté pour tenir compte de topographie)

```
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: Remarquez qu'il s'agit d'une RDA partielle!

Significativité: Fractions individuelles



[c] Topographie (ajusté pour tenir compte de chimie)

```
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
```

Défi 3



Partitionnez la variation de l'abondance des espèces de mites entre des variables de substrat (SubsDens, WaterCont) et des variables spatiales significatives.

- Quelle est la proportion de variance expliquée par le substrat? par l'espace?
- Quelles sont les fractions significatives?
- Diagramme Venn des résultats!

Chargez les variables spatiales:

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

Rappel de fonctions utiles:

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

Défi 3: Solution

Étape 1: Sélection de variables spatiales significatives

```
# Modèle RDA avec tous les variables spatiales
full.spat ← rda(mite.spe.hel ~ ., data = mite.pcnm)

# Sélection progressive des variables spatiales
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)
```

Défi 3: Solution

Étape 2: Créer sous-groupes de variables explicatives

```
# Variables de substrat
mite.subs ← subset(mite.env, select = c(SubsDens, WatrCont))

# Variables spatiales significatives
mite.spat ← subset(mite.pcnm,
                     select = names(spat.sel$terminfo$ordered))
                     # pour rapidement accéder aux variables sélectionnées
```

Défi 3: Solution

Étape 3: Partitionnement de la variation

```
mite.part <- varpart(mite.spe.hel, mite.subs, mite.spat)
mite.part$part$indfract # extraire résultats
# 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
```

- Quelle est la proportion de variance expliquée par le substrat?
 - **5.9%**
- Quelle est la proportion de variance expliquée par l'espace?
 - **19.4%**

Défi 3: Solution

Étape 4: Quelles sont les fractions significatives?

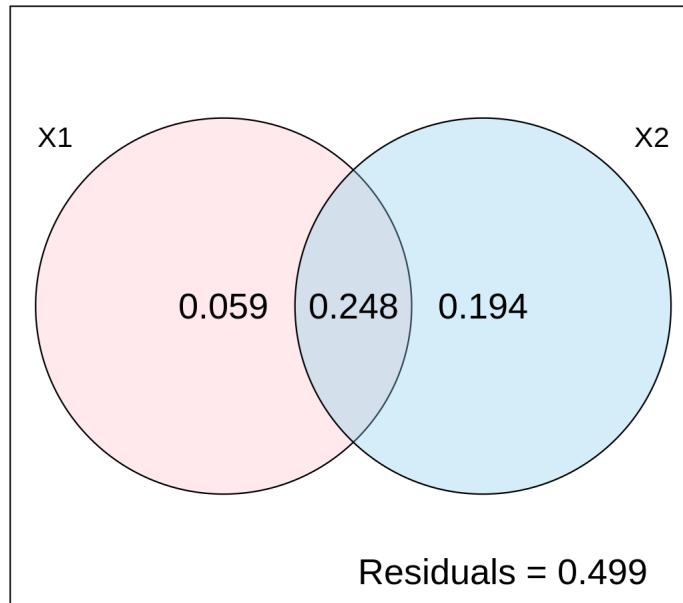
```
# [a]: Substrat seulement  
anova.cca(rda(mite.spe.hel, mite.subs, mite.spat))  
# p = 0.001 ***  
  
# [c]: Espace seulement  
anova.cca(rda(mite.spe.hel, mite.spat, mite.subs))  
# p = 0.001 ***
```

Alors, quels sont les effets de substrat et de l'espace sur les abondances d'espèces de mites?

Défi 3: Solution

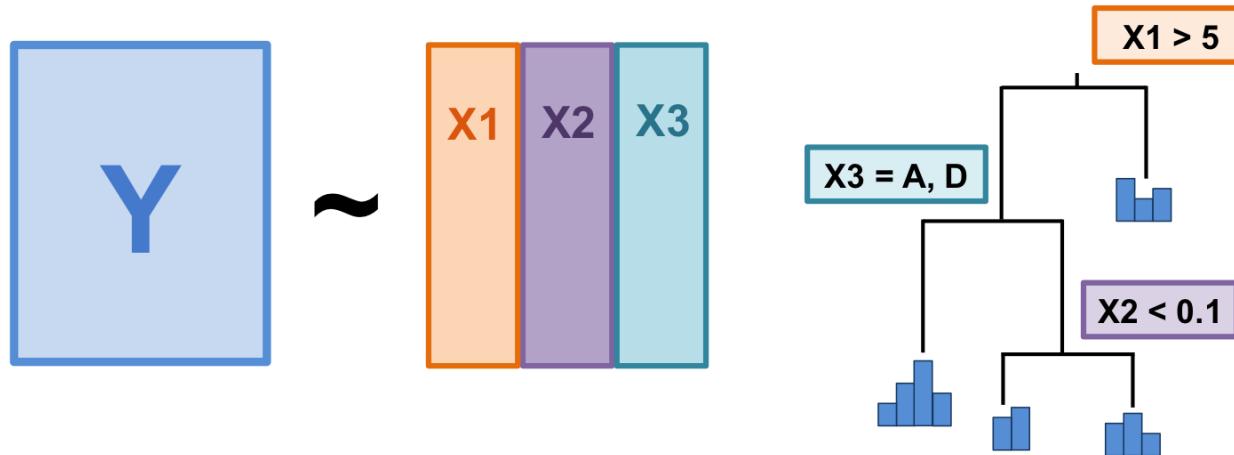
Étape 5: Diagramme Venn

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



Arbre de régression multivarié (MRT)

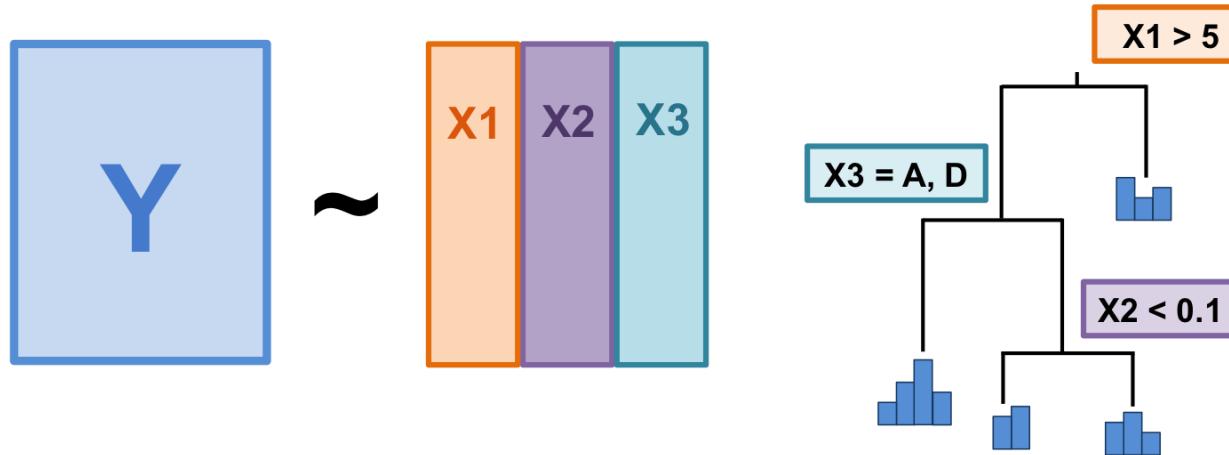
Arbre de régression multivarié (MRT)



L'arbre de régression multivarié (ARM ou MRT) est une méthode de groupement hiérarchique sous contrainte.

- Partitionne une matrice réponse quantitative (Y) en sous-groupes sous la contrainte d'une matrice de variables explicatives (X).

Arbre de régression multivarié (MRT)



L'arbre de régression multivarié consiste de:

- **Branches**: chaque lignée formée par un noeud
- **Noeuds**: Point où les données se divisent en 2 groupes (caractérisé par une valeur limite d'une variable explicative)
- **Feuilles**: groupe terminal de sites

Arbre de régression multivarié (MRT)

Plusieurs avantages:

- N'assume pas de relation linéaire entre les matrices Y et X
- Facile à interpréter et à visualiser
- Robuste en présence de valeurs manquantes ou de colinéarité(s) entre les descripteurs
- Valeurs brutes peuvent être utilisées (sans transformation)

MRT: La méthode

La méthode implique deux volets s'effectuant en parallèle:

1. *Partitionnement* des données sous contrainte
2. *Validation croisée* pour identifier l'arbre ayant le meilleur pouvoir prédictif.

Choisissez l'arbre selon les objectifs de ton étude. Généralement, on veut un arbre:

- *parcimonieux*
- mais avec un nombre *informatif* de groupes
- Essentiellement: quel arbre répond à ta question?

MRT dans R

Dans ce qui suit, nous allons utiliser **mpart** qui est archivé sur le CRAN. Nous l'installons depuis GitHub avec le package remotes:

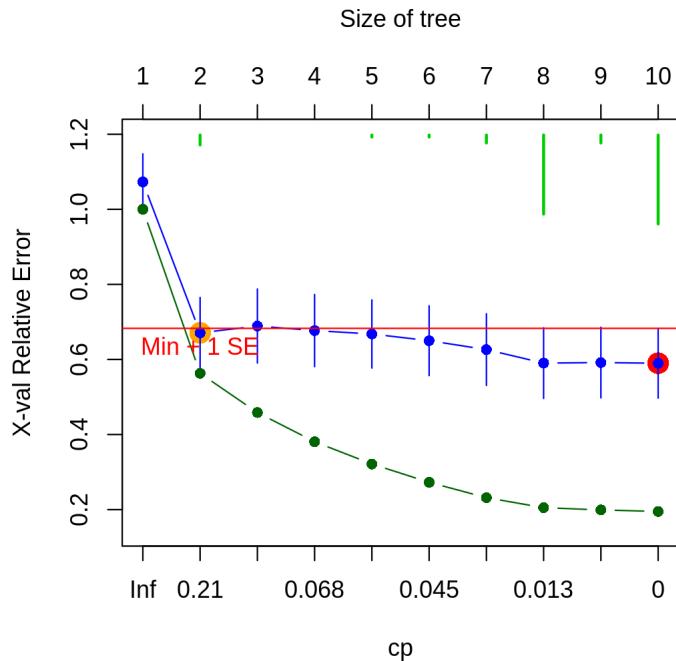
```
remotes::install_github("cran/mvpart")
library(mvpart)
```

MRT dans R

```
# Enlever la variable "distance from source"
env <- subset(env, select = -das)

# Créer l'arbre de regression multivarié
# library(mvpart)
doubts.mrt <- mvpart(as.matrix(spe.hel) ~ ., data = env,
                      xv = "pick", # selection graphique interactive
                      xval = nrow(spe.hel), # nombre de validations
                      xvmult = 100, # nombre de validations multiples
                      which = 4, # identifier les noeuds
                      legend = FALSE, margin = 0.01, cp = 0)
```

MRT dans R: Choisir un arbre

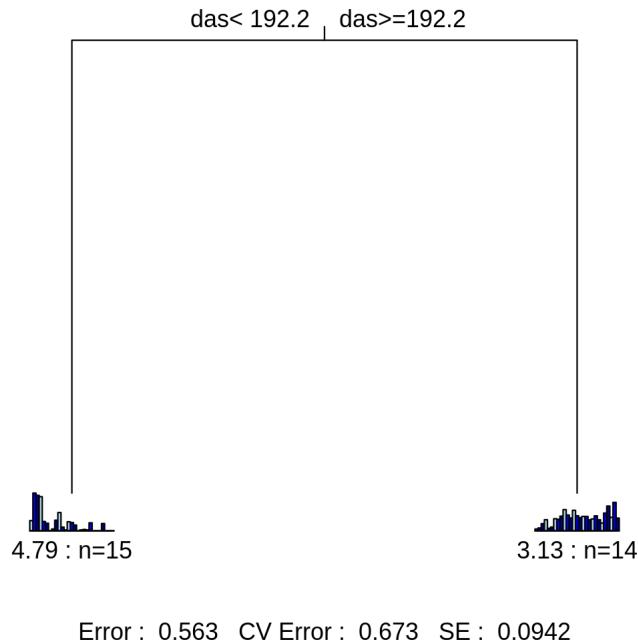


- Points verts: Erreur relative
- Points bleus: Erreur relative de validation croisée (CVRE)
- Point rouge: Arbre avec la valeur minimale de CVRE
- Point orange: l'arbre le plus petit ayant un CVRE à 1 écart type du CVRE minimal
- Barres vertes: # de fois que chaque taille d'arbre a été choisi

MRT dans R: Choisir un arbre

- Cliquez sur le point bleu correspondant à la taille de l'arbre choisie!
- Puisqu'on ne sait pas *a priori* comment partitionner ces données, on choisira *l'arbre le plus petit ayant un CVRE à 1 écart type du CVRE minimal* (i.e. le point orange).

MRT dans R: Visualisation

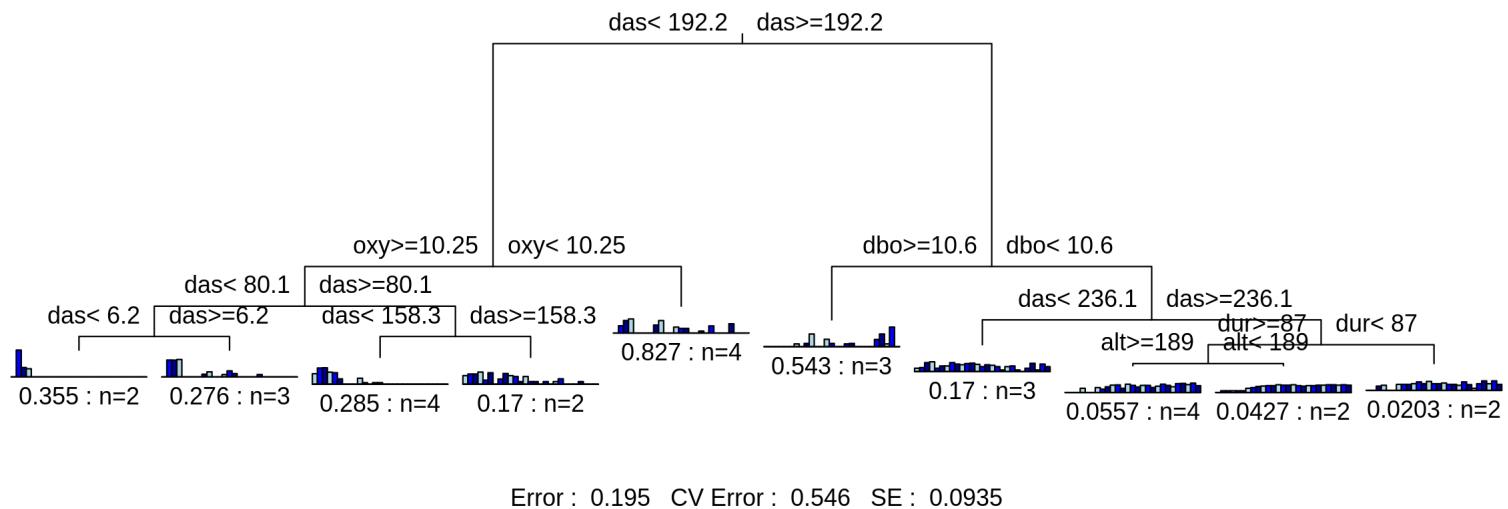


- La matrice d'abondances est partitionnée selon un seuil d'**altitude (361.5)**.
 - "Barplots": abondances d'espèces inclus dans chaque groupe
- Erreur résiduelle = 0.563, alors le R2 du modèle est **43.7%**

MRT dans R: Comparaison d'arbres

Pour choisir un arbre, on peut aussi comparer plusieurs solutions possibles.

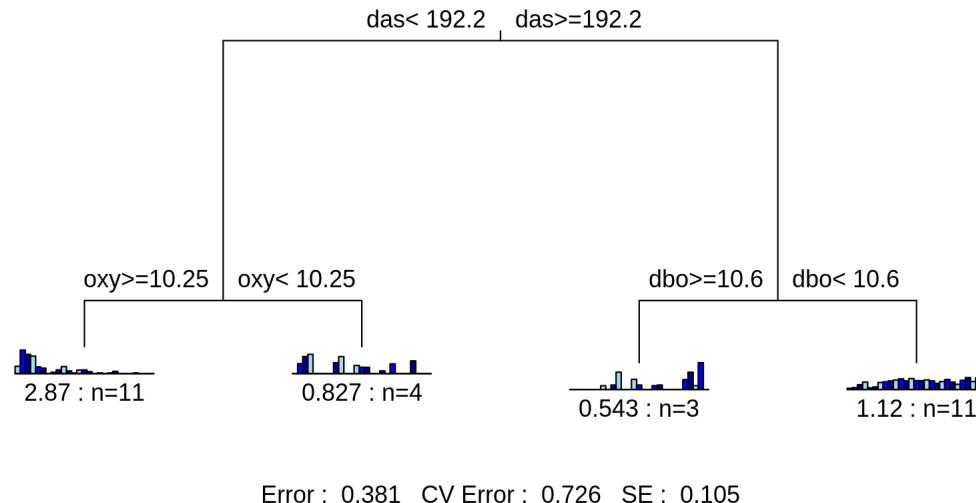
Par exemple, considérons une solution de 10 groupes!



- L'interprétation est **plus difficile**.
 - Plus grand pouvoir explicatif, MAIS le pouvoir prédictif (CV Error = 0.671) ressemble à la solution précédante (CV Error = 0.673).

MRT dans R: Comparaison d'arbres

Considérons une solution avec moins (4) de groupes!



- Plus facile à interpréter!
- Plus grand pouvoir explicatif (**Error**) que notre solution originale
- **Plus grand pouvoir prédictif** que les 2 solutions précédentes (CV Error)

MRT dans R: Paramètre de complexité

Le *paramètre de complexité (CP)* représente la variance expliquée par chaque noeud.

```
doub$mrta$cptable  
#          CP nsplit rel error      xerror      xstd  
# 1 0.4369561    0 1.0000000 1.0754712 0.07512742  
# 2 0.1044982    1 0.5630439 0.6730179 0.09424486
```

- CP @ nsplit 0 = R2 de l'arbre au complet
- CP @ autres noeuds = R2 de chaque noeud (voir sommaire complet pour la valeur de seuil de chaque noeud)

MRT dans R: Sommaire des résultats

Pour accéder au sommaire des résultats:

```
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 1.0754712 0.07512742
# 2 0.1044982      1 0.5630439 0.6730179 0.09424486
#
# Node number 1: 29 observations,    complexity param=0.4369561
# Means=0.07299,0.2472,0.2581,0.2721,0.07133,0.06813,0.06897,0.07664,0.1488,0.2
# 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)
#
# Node number 2: 15 observations
```

MRT dans R: Espèces discriminantes

On peut aussi déterminer quelles espèces contribuent le plus à la variance expliquée par chaque nœud (**espèces discriminantes**), ou quels sites sont inclus dans chaque feuille (groupe).

Pour ceci, on peut utiliser la librairie **MVPARTwrap**

MVPARTwrap est archivé, on utilise remotes pour l'installer depuis GitHub

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

MRT dans R: Espèces discriminantes

```
# Créer un sommaire plus informatif et moins dense
doubs.mrt.wrap ← MRT(doubs.mrt, percent = 10, species = colnames(spe.hel))

# Voir le sommaire
summary(doubs.mrt.wrap)
```

MRT dans R: Espèces discriminantes

Pour voir la contribution de chaque espèce à la variance expliquée par noeud:

```
summary(doubs.mrt.wrap)
# Portion (%) of deviance explained by species for every particular node
#
# ~~~~~
#           --- Node 1 ---
# Complexity(R2) 43.69561
# das< 192.2 das≥192.2
#
# ~ Discriminant species :
#                               TRU          VAI          ABL
# % of expl. deviance 20.03148905 13.16272887 13.3114611
# Mean on the left    0.44630603  0.41943941  0.0000000
# Mean on the right   0.03390824  0.08514225  0.3361805
#
# ~ INDVAL species for this node: : left is 1, right is 2
#      cluster indicator_value probability
# TRU      1          0.8674        0.001
# VAI      1          0.7758        0.001
# LOC      1          0.7042        0.002
# ABL      2          1.0000        0.001
# HOT      2          0.8571        0.001
# GRE      2          0.8571        0.001
```

MRT dans R: Espèces discriminantes

Pour déterminer les espèces discriminantes **significatives** pour chaque groupe:

```
library(labdsv)
```



Défi 4

Créez un arbre de régression multivarié pour les données *mite*.

- Choisir l'arbre le plus petit à 1 écart type du CVRE minimal.
- Quelle est la variance totale expliquée par cet arbre?
- Combien y a-t-il de feuilles?
- Quels sont les 3 principales espèces discriminantes?

Rappel: chargez les données!

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

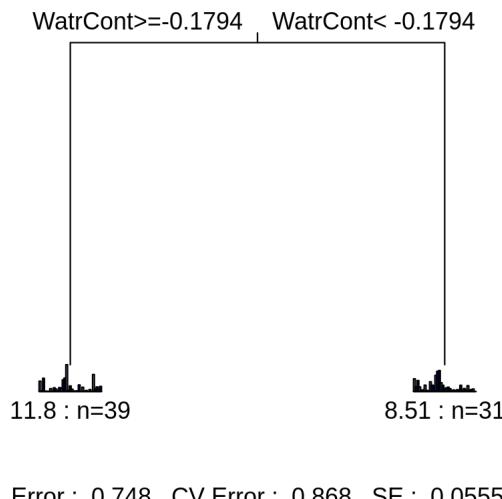
Rappel de fonctions utiles:

```
?mvpard() # argument 'xv' !
?MRT()
summary()
```

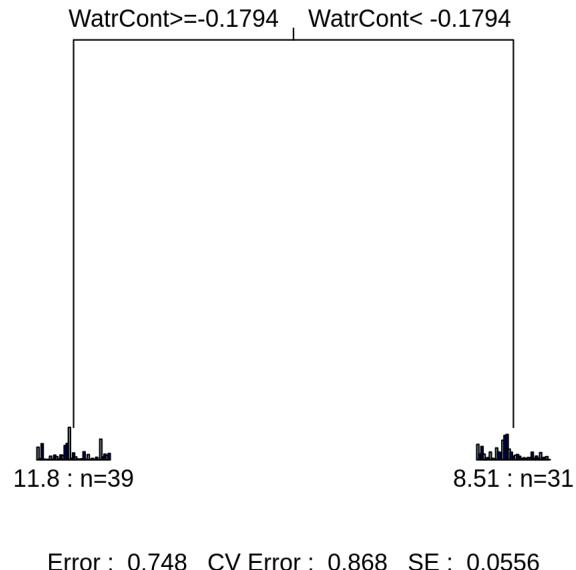
Défi 4: Solution

Étape 1: Créer un arbre de régression multivarié

```
mite.mrt ← mpart(as.matrix(mite.spe.hel) ~ ., data = mite.env,
                    xv = "1se",
                    xval = nrow(mite.spe.hel),
                    xvmult = 100,
                    which = 4, legend = FALSE, margin = 0.01, cp = 0,
                    prn = FALSE)
```



Défi 4: Solution



- Quelle est la variance totale expliquée (R^2) par cet arbre?
 - $1 - \text{Error} = 0.252$, alors l'arbre explique **25.2%** de la variation dans la matrice d'abondances.
- Combien y a-t-il de feuilles?
 - 2 feuilles

Défi 4: Solution

Quels sont les 3 principales espèces discriminantes pour **noeud #1**?

```
# Créer sommaire plus informatif
mite.mrt.wrap ← MRT(mite.mrt,
                      percent = 10,
                      species = colnames(mite.spe.hel))

# Voir sommaire (pour voir les espèces discriminantes)
summary(mite.mrt.wrap)
```

Défi 4: Solution

Quels sont les 3 principales espèces discriminantes pour **noeud #1?**

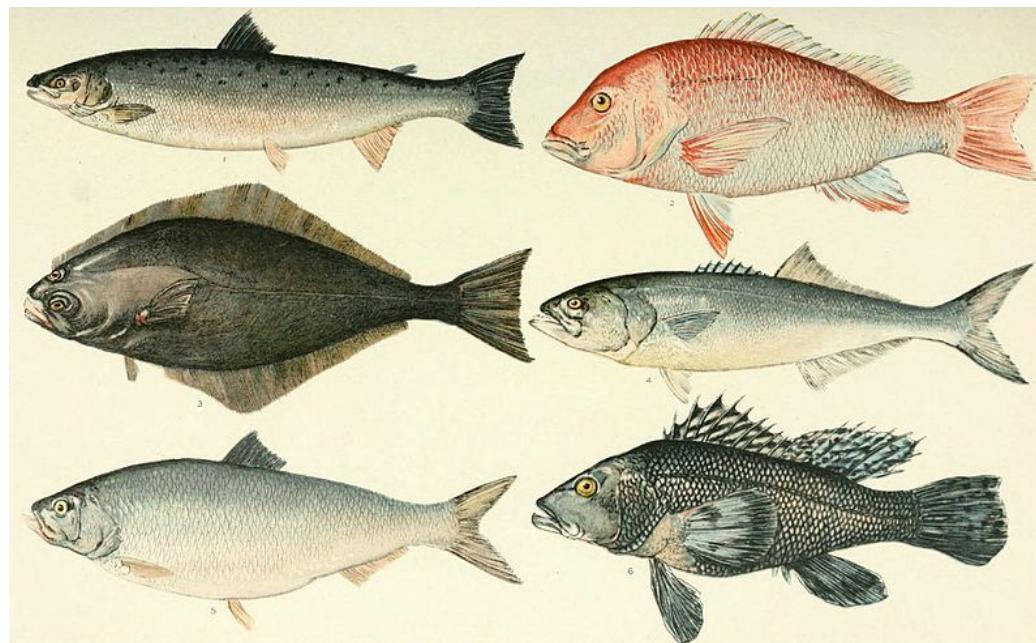
- LCIL, LRUG, Ceratoz1

~ 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

Analyse discriminante linéaire (LDA)

Analyse discriminante linéaire (LDA)

- Détermine si une matrice de variables indépendantes explique bien un groupement établi *a priori*
 - e.g. prédire l'appartenance d'une espèce de poisson à un groupe (marin vs. eau douce) selon sa morphologie



LDA dans R: Rivière Doubs

Généralement, les variables environnementales changent avec la latitude.

Si on classifie les sites de la rivière Doubs selon la latitude, à quel point les variables environnementales expliquent-elles ces groupements?

- Une LDA permet de répondre à cette question.

LDA dans R: Rivière Doubs

Commençons par charger les données spatiales des sites:

```
# charger les données spatiales des sites Doubs:  
spa <- read.csv("data/doubsspa.csv", row.names = 1)  
spa$site <- 1:nrow(spa) # assigner un chiffre par site  
spa <- spa[-8,] # enlever le site #8
```

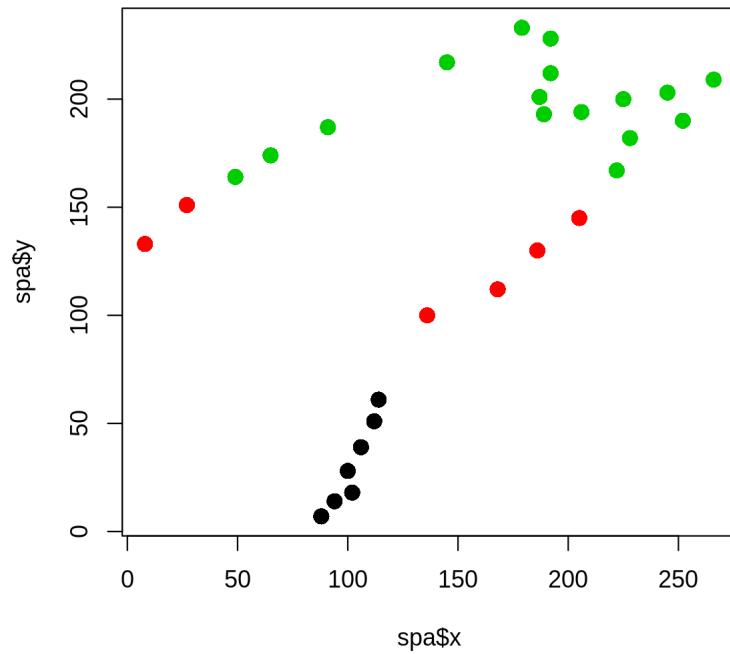
Ensuite, on peut classifier les sites dans 3 groupes de latitudes:

```
spa$group <- NA # créer colonne "group"  
spa$group[which(spa$y < 82)] <- 1  
spa$group[which(spa$y > 82 & spa$y < 156)] <- 2  
spa$group[which(spa$y > 156)] <- 3
```

LDA dans R: Rivière Doubs

Visualisons ces regroupements par latitude:

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



LDA dans R

Note: Normalement, nous devons vérifier que les matrices de covariance des variables explicatives sont homogènes (voir Borcard et al. 2011).

Pour les besoins de l'atelier, on passera directement à la LDA:

```
# charger la librairie requise  
library(MASS)  
  
# faire la LDA  
LDA ← lda(env, spa$group)
```

LDA dans R: Vérification

On peut ensuite voir comment les sites sont classifiés, et si cette classification est exacte.

```
# classification des objets en fonction de la LDA
spe.class ← predict(LDA)$class

# probabilités que les objets appartiennent à chaque groupe a posteriori
spe.post ← predict(LDA)$posterior

# tableau des classifications a priori et prédites
(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 de classification correcte
diag(prop.table(spe.table, 1))
# 1 2 3
# 1 1 1
```

Tous les sites ont été correctement classifiés dans leur groupe de latitude en fonction des variables environnementales.

LDA dans R: Prédictions

On peut maintenant utiliser la LDA pour classifier de nouveaux sites dans les groupes de latitude. Tentons de **prédirer la classification** de 5 nouveaux sites à l'aide de notre LDA:

```
# charger les nouvelles données
classify.me ← read.csv("data/classifyme.csv", header = TRUE)
# classify.me ← classify.me[,-1] # remove das variable

# prédirer le groupement des nouvelles données
predict.group ← predict(LDA, newdata = classify.me)

# donner la classification pour chaque site
predict.group$class
# [1] 1 1 1 3 3
# Levels: 1 2 3
```

Défi 5



Créez 4 groupes de latitude à partir des données *mite.xy*. Ensuite, une LDA sur les données environnementales (*mite.env*) des acariens (*SubsDens* et *WatrCont*).

- Quelle proportion de sites ont été classifiés correctement au groupe 1? Au groupe 2?

Chargez *mite.xy*:

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

Rappel de fonctions utiles:

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

Défi 5: Solution

Étape 1: Créer 4 groupes de latitude

```
# numérotter les sites
mite.xy$site ← 1:nrow(mite.xy)

# trouver une étendue égale de latitudes par groupe
(max(mite.xy[,2])-min(mite.xy[,2]))/4
# [1] 2.4

# classifier les sites dans 4 groupes de latitude
mite.xy$group ← NA # nouvelle colonne "group"
mite.xy$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
```

Étape 2: Faire la LDA

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

Défi 5: Solution

Étape 3: Vérifier la classification

```
# classification des objets en fonction de la LDA
mite.class <- predict(LDA.mite)$class
# tableau de classifications (prior versus predicted)
(mite.table <- table(mite.xy$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 de classifications exactes
diag(prop.table(mite.table, 1))
#           1             2             3             4
# 0.6000000 0.6470588 0.7368421 0.8421053
```

Quelle proportion de sites ont été classifiés correctement au groupe 1? Au groupe 2?

- **60%** des sites ont été classifiés correctement dans group1, et **64.7%** dans group2.

Merci d'avoir participé à cet atelier!

