Maxime HERVÉ

Aide-mémoire de statistique appliquée à la biologie

*Construire son étude et analyser les résultats à l'aide du logiciel **R***

Version finale (2016)

AVANT-PROPOS

Il n'est pas besoin de grandes enquêtes d'opinions pour se rendre compte que les biologistes sont globalement frileux à se frotter aux statistiques. L'étape de l'analyse des résultats est souvent vécue comme une contrainte, un passage obligé mais désagréable, voire même parfois un calvaire. Pourtant, le premier objectif des statistiques est bien de révéler ce que les données ont a nous dire. Passer à côté d'une bonne analyse par manque de temps, de motivation ou de compétence, c'est surtout prendre le risque de rater un phénomène intéressant qui était pourtant là, sous nos yeux.

L'objectif de cet aide-mémoire est de guider tout biologiste qui en sentirait le besoin dans sa démarche statistique, depuis la construction du protocole expérimental jusqu'à l'analyse des résultats qui en découlent. Il doit permettre de s'en sortir seul, tout en assurant une analyse appropriée et rigoureuse. Bien entendu, il ne dispense pas de se poser des questions et il est toujours nécessaire d'adapter un minimum le code proposé à ses propres données. Pour imager les choses, considérez que vous apprenez à faire du vélo et que ce document est la paire de roulettes qui vous évite de chuter. C'est rassurant, mais n'oubliez pas qu'avant tout c'est vous qui pédalez.

Depuis la rédaction de la première version de cet aide-mémoire il y a six ans déjà, j'ai beaucoup enseigné, formé et conseillé en statistique appliquée à la biologie. Ces nombreuses interactions avec des collègues et étudiants m'ont conduit à la structure de cette sixième et dernière version, qui je crois est celle qui est la plus adaptée à une bonne démarche d'analyse.

Un assez grand nombre de méthodes sont couvertes par ce document, et j'ai tout fait pour simplifier l'orientation dans cette « jungle ». Malgré tout, une bonne part du cheminement est dépendante du biologiste lui-même. Une bonne analyse est une analyse qui répond pertinemment à une question précise. La règle d'or est donc avant tout de bien identifier cette question ou série de questions, et de ne jamais l'oublier car le choix de la méthode statistique la plus pertinente en dépend et il est facile de se perdre en analysant ses données.

Cet aide-mémoire est directement associé au package `RVAideMemoire`*. La présente version du document correspond aux versions ≥ 0.9-60 du package.*

J'espère sincèrement que ce document comblera vos attentes et qu'il vous permettra de vous sentir moins seul dans le monde pas si cauchemardesque des statistiques.

Le 22 août 2016

Maxime HERVÉ

SOMMAIRE

I Bases du fonctionnement de R **4**

II Théorie statistique élémentaire **16**

III Analyser les résultats d'une étude **36**

PARTIE I – BASES DU FONCTIONNEMENT DE R

Ce document n'est pas à proprement parler une introduction à R. Cette partie rappelle seulement quelques notions essentielles comme la manipulation des objets courants (vecteurs, tableaux, matrices et listes), la construction et l'importation d'un jeu de données, la gestion des packages et diverses autres choses comme des « bonnes pratiques ».

PARTIE II – THÉORIE STATISTIQUE ÉLÉMENTAIRE

Ce document n'est pas non plus une introduction aux statistiques. Cependant certaines bases théoriques sont indispensables pour construire une étude proprement et en analyser correctement les résultats : types de variable, plan d'échantillonnage ou d'expérience, fonctionnement d'un test, taille de l'échantillon à constituer. Accessoirement, les lois de distribution les plus courantes sont présentées.

PARTIE III – ANALYSER LES RÉSULTATS D'UNE ÉTUDE

L'essentiel du document est dans cette partie, qui détaille comment représenter, synthétiser et analyser des données à une, deux ou plus de deux dimensions.

Première partie

Bases du fonctionnement de R

MANIPULATION DES OBJETS COURANTS

1. Les vecteurs
2. Les tableaux
3. Les matrices
4. Les listes

PRÉPARATION ET IMPORTATION DES DONNÉES

5. La construction d'un tableau de données
6. L'importation d'un tableau de données dans R

DIVERS

7. Bonnes pratiques
8. Installer, charger et mettre à jour des packages
9. Citer R et ses packages
10. Changer de version de R

1. Les vecteurs

Le vecteur est à la fois l'objet le plus simple et le plus fondamental du langage **R**. Il se crée grâce à la fonction `c()`, qui prend comme arguments les éléments du vecteur. Tous ces éléments doivent être *du même type* : valeurs numériques, chaînes de caractères ou encore niveaux d'un facteur.

EXEMPLE(S)

Pour créer un vecteur numérique :

```
> vecteur <- c(7,9,4,12,18)
> vecteur
[1] 7 9 4 12 18
```

Pour créer un vecteur de chaînes de caractères :

```
> vecteur <- c("H","C","I","G","F")
> vecteur
[1] "H" "C" "I" "G" "F"
```

Pour créer un facteur :

```
> vecteur <- factor(c("niv1","niv2","niv2","niv3","niv1"))
> vecteur
[1] niv1 niv2 niv2 niv3 niv1
Levels: niv1 niv2 niv3
```

Il existe des fonctions ou des abréviations qui permettent de simplifier la création de certains vecteurs usuels :

EXEMPLE(S)

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> seq(from=1,to=3,by=0.25)
[1] 1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00
> LETTERS[1:5]
[1] "A" "B" "C" "D" "E"
```

Pour accéder au(x) $i^{ème(s)}$ élément(s) d'un vecteur, écrire `vecteur[i]`, où `i` peut être une valeur unique ou lui-même un vecteur :

EXEMPLE(S)

```
> vecteur <- seq(from=2,to=16,by=2)
> vecteur
[1] 2 4 6 8 10 12 14 16
> vecteur[5]
[1] 10
> vecteur[c(2,5,8)]
[1] 4 10 16
> vecteur[-c(2,5,8)]
[1] 2 6 8 12 14
> vecteur[6:3]
[1] 12 10 8 6
```

2. Les tableaux

Les tableaux sont simplement un moyen de regrouper (en colonnes) des vecteurs dans le même objet, chaque colonne étant *indépendante*. L'unique contrainte est que tous les vecteurs doivent avoir *la même longueur*.

Pour créer un tableau, utiliser la fonction `data.frame()`, qui prend en arguments les différentes colonnes (de gauche à droite). On peut préciser le titre des colonnes. Dans le cas d'un vecteur de chaînes de caractères, celui-ci est automatiquement transformé en facteur lorsqu'il est intégré au tableau.

EXEMPLE(S)

```
> variable1 <- 1:5
> variable2 <- LETTERS[1:5]
> tableau <- data.frame(variable1,variable2)
> tableau
  variable1 variable2
1         1         A
2         2         B
3         3         C
4         4         D
5         5         E
```

Le tableau peut être créé directement *via* :

```
> tableau <- data.frame(variable1=1:5,variable2=LETTERS[1:5])
```

Pour accéder à un (ou plusieurs) élément(s) d'un tableau, le principe est le même que pour les vecteurs (voir fiche 1) excepte qu'il n'y a pas une mais *deux* dimensions à l'objet (*i.e.* les lignes et les colonnes). Le principe d'indexation est valable pour tous les objets à deux dimensions et est celui-ci : `tableau[ligne(s),colonne(s)]`, où `ligne(s)` et `colonne(s)` sont soit des valeurs uniques, soit des vecteurs. Si rien n'est mis avant la virgule toutes les lignes sont sélectionnées, si rien n'est mis après toutes les colonnes sont sélectionnées.

EXEMPLE(S)

```
> tableau[c(1,3),]
  variable1 variable2
1         1         A
3         3         C

> tableau[c(3,5),2]
[1] C E
Levels: A B C D E
```

Dans le cas particulier de la sélection d'une colonne entière, il y a trois autres possibilités :

— `tableau$colonne` où `colonne` est le *nom* de la colonne
— `tableau$"colonne"` où `colonne` est le *nom* de la colonne, entre guillemets
— `tableau[,"colonne"]` où `colonne` est le *nom* de la colonne, entre guillemets.

3. Les matrices

À la différence des tableaux (voir fiche **2**), les matrices sont un tout cohérent, *i.e.* les colonnes ne sont pas indépendantes. Cela implique que *tous les éléments d'une matrice sont de même type* : numérique, texte, niveaux de facteur…

Pour créer une matrice, utiliser la fonction `matrix()`, qui prend comme arguments obligatoires les valeurs qui doivent la remplir, et le nombre de lignes et/ou de colonnes. Par défaut les matrices sont remplies en colonnes, pour les remplir en lignes ajouter l'argument `byrow=TRUE`. Pour donner un nom aux lignes et aux colonnes, utiliser l'argument `dimnames=list(lignes,colonnes)`, où `lignes` et `colonnes` sont des vecteurs :

EXEMPLE(S)

```
> matrice <- matrix(1:8,nrow=2)
> matrice
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
> matrice <- matrix(1:8,nrow=2,byrow=TRUE)
> matrice
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
> matrice <- matrix(1:8,nrow=2,dimnames=list(letters[1:2],LETTERS[1:4]))
> matrice
  A B C D
a 1 3 5 7
b 2 4 6 8
```

Il est également possible de créer des matrices à partir de plusieurs vecteurs qui doivent en constituer les lignes ou les colonnes. Utiliser pour cela les fonctions `rbind()` ou `cbind()`, qui assemblent les vecteurs respectivement en lignes et en colonnes :

EXEMPLE(S)

```
> vecteur1 <- 1:3
> vecteur2 <- 4:6
> matrice <- rbind(vecteur1,vecteur2)
> matrice
         [,1] [,2] [,3]
vecteur1    1    2    3
vecteur2    4    5    6
> matrice <- cbind(vecteur1,vecteur2)
> matrice
     vecteur1 vecteur2
[1,]        1        4
[2,]        2        5
[3,]        3        6
```

Les matrices étant des objets à *deux* dimensions (les lignes et les colonnes), leur indexation est identique à celle des tableaux (voir fiche **2**).

4. Les listes

Les listes sont des objets à compartiments, où chaque compartiment est totalement *indépendant* des autres. Une liste peut donc à la fois contenir un vecteur dans un compartiment, un tableau dans un autre, et même une liste dans un troisième.

Pour créer une liste, utiliser la fonction `list()`, qui prend en arguments ce que l'on veut mettre dans chaque compartiment (du premier au dernier). On peut préciser un nom à chaque compartiment, ce qui aide grandement à s'y retrouver.

EXEMPLE(S)

```
> vecteur <- 1:5
> tableau <- data.frame(v1=1:3,v2=LETTERS[1:3])
> list(vecteur,tableau)
[[1]]
[1] 1 2 3 4 5

[[2]]
 v1 v2
1 1  A
2 2  B
3 3  C
```

La liste peut être créée directement *via* :

```
> list(1:5,data.frame(v1=1:3,v2=LETTERS[1:3]))
```

Pour donner un nom aux compartiments :

```
> liste <- list(A=1:5,B=data.frame(v1=1:3,v2=LETTERS[1:3]))
> liste
$A
[1] 1 2 3 4 5

$B
 v1 v2
1 1  A
2 2  B
3 3  C
```

Pour accéder à un compartiment d'une liste (toujours un seul à la fois), on peut utiliser le numéro du compartiment entre *doubles* crochets : `liste[[i]]` où `i` est une valeur numérique unique. Si les compartiments de la liste ont un nom, on peut également utiliser la syntaxe `liste$nom` où `nom` est le nom du compartiment.

EXEMPLE(S)

```
> liste[[1]]
[1] 1 2 3 4 5
```

Ou, puisque les compartiments de cette liste ont un nom :

```
> liste$A
[1] 1 2 3 4 5
```

Pour accéder à un (ou des) élément(s) particulier(s) d'un compartiment d'une liste, il suffit de cumuler l'indexation de la liste et l'indexation de l'objet contenu dans le compartiment en question.

EXEMPLE(S)

```
> liste[[1]][c(2,4)]
[1] 2 4
> liste$B$v1[3]
[1] 3
```

5. La construction d'un tableau de données

La construction d'un tableau de données correctement structuré est une étape importante de l'étude, car si elle est mal réalisée elle peut mener à des résultats faux, ou le plus souvent à des erreurs une fois dans **R**.

Cette construction nécessite de se poser une question essentielle : quelles sont les variables prises en compte dans l'étude ? Y répondre implique d'identifier les variables quantitatives et les facteurs, ainsi que les classes des facteurs. Si les choses sont claires, l'analyse statistique le sera également.

D'une manière générale, il est conseillé de toujours construire son tableau de données dans un tableur. Cela permet d'enregistrer le jeu de données dans un fichier externe à **R**, et donc de toujours pouvoir y revenir puisque **R** ne modifie pas les fichiers externes (sauf si on le lui demande explicitement).

Dans le tableur, la règle est simple : les individus doivent être placés en *lignes* et les variables en *colonnes*.

Il est conseillé de donner un titre à chaque colonne, qui deviendra le nom de la variable dans **R**. Il est indispensable cependant de respecter certaines règles : les noms de variable ne doivent contenir ni espace, ni caractère accentué, ni symbole (ceci est une règle pour tous les noms d'objet dans **R**). Si un nom de variable doit contenir deux mots, ils peuvent être séparés par un point (.) ou un tiret bas (_). Mieux vaut également privilégier les noms courts mais clairs, car une fois dans **R** taper sans cesse des noms de variable longs est vite fastidieux.

Le tableau de données doit absolument obéir à une autre règle : *aucune case ne doit être vide*. La seule exception possible est celle en haut à gauche si les colonnes ont un titre, auquel cas la 1ère colonne sera comprise par **R** comme le nom des lignes. S'il manque une donnée pour un individu, il faut se demander d'où elle vient :

— si c'est une donnée inutilisable (mesure ratée, mal retranscrite...), pas de problème. On dit alors qu'on a une « donnée manquante », que l'on doit noter `NA` (pour *Not Available, i.e.* donnée manquante). Le tableur comme **R** reconnaissent le `NA`, qu'ils interprètent correctement.

— si la situation est autre, c'est que le tableau est mal construit et qu'en particulier les variables n'ont pas été bien définies. La réflexion s'impose donc pour identifier les variables et reconstruire un tableau de données.

Il est déconseillé de coder les niveaux d'un facteur avec uniquement des chiffres. **R** comprendrait cette variable comme numérique (et non comme un facteur), ce qui pourrait sérieusement perturber voire empêcher l'analyse.

Si des analyses dans **R** doivent se faire uniquement sur un sous-ensemble du tableau de données, ou si pour certaines analyses le tableau de données serait plus facile à utiliser s'il était construit autrement, il est conseillé de construire plusieurs tableaux de données séparés. Il est toujours possible de manipuler le tableau initial dans **R** pour en extraire une partie ou pour le transformer, mais il est clairement plus facile (et surtout moins source d'erreur) lorsque l'on n'a pas l'habitude de le faire en amont, dans le tableur.

6. L'importation d'un tableau de données dans R

Il existe de nombreuses méthodes pour importer ses données dans R. Une seule est présentée ici, qui est à la fois très simple, fonctionne dans la plupart des situations et peut être utilisée sur toutes les plateformes.

La procédure se fait en trois étapes :

1. Dans le tableur, sélectionner toutes les cases constituant le tableau de données.
2. Copier ce tableau dans le bloc-notes et enregistrer le fichier en format `.txt`.
3. Dans R, charger le tableau de données grâce à la fonction `read.table()` et le stocker dans un objet : `tableau<-read.table("fichier")` où `fichier` est le nom du fichier texte (avec l'extension `.txt` et éventuellement le chemin qui mène à ce fichier), entre guillemets.

R étant un logiciel anglo-saxon, le séparateur décimal qu'il utilise est le point. Or dans les tableurs français (et donc dans le fichier texte) le séparateur décimal est la virgule. Si le tableau de données contient des valeurs décimales, il est donc nécessaire de préciser à R qu'il interprète la virgule comme séparateur décimal. Ajouter pour cela l'argument `dec=","` à la fonction `read.table()`.

Si les colonnes du tableau de données ont un titre, qui doit donc être interprété comme le nom de la variable, ajouter l'argument `header=TRUE`.

Une fois le tableau importé, il est indispensable de vérifier qu'il n'y a pas eu d'erreur pendant son chargement. Pour cela appeler le résumé du tableau *via* `summary(tableau)`. R renvoie un résumé de chaque variable :

— pour une variable numérique, R donne des indications sur sa distribution : minimum, 1er quartile, médiane, moyenne, 3ème quartile et maximum.
— pour un facteur, R donne le nombre d'individus par classe.

Si un facteur est codé numériquement (par exemple un facteur binaire ou un facteur ordinal), R l'interprète comme une variable numérique. Pour transformer cette variable en facteur, taper `tableau$variable<-factor(tableau$variable)` où `variable` est le nom de la variable.

7. Bonnes pratiques

Il existe des dizaines d'astuces ou de façon de procéder pour utiliser R au mieux. Certaines, importantes, sont résumées ici.

Logiciel et packages

Les packages ne sont pas gravés dans le marbre mais évoluent avec le temps. Les auteurs corrigent des erreurs, ajoutent de nouvelles fonctions... Pour bénéficier de ces améliorations/ajouts/corrections, il est nécessaire de mettre à jour régulièrement ses packages (une fois par mois est un bon rythme). Voir fiche **8** pour la procédure, très simple.

Le logiciel R lui-même évolue. Le numéro de la version installée est toujours donné dans le message d'accueil au lancement du logiciel, par exemple 3.3.1. Le premier chiffre de ce numéro est très important. En effet, toutes les versions des packages créées après la sortie d'une version V.x.x. de R ne sont *pas disponibles* pour les utilisateurs des versions [V-1].x.x. Ces mises à jour dites « majeures » du logiciel ont donc une importance réelle. Il est fortement recommandé de suivre ces mises à jours majeures, qui sont tout de même relativement rares. Certains packages exigent cependant une version minimale de R pour pouvoir être mis à jour. Installer une nouvelle version du logiciel régulièrement (une fois tous les 1 à 2 ans) permet d'avoir accès à toutes les mises à jour récentes (voir fiche **10**).

Création des objets

Au-delà des règles obligatoires sur les noms d'objet (ne pas commencer par un chiffre, pas d'espace...), quelques principes permettent d'éviter des erreurs qui peuvent parfois conduire à une analyse statistique biaisée (ou en tout cas à se compliquer la vie) :

- toujours donner un nom informatif aux objets, pour s'y retrouver plus facilement (attention tout de même aux noms à rallonge qui au final font perdre du temps !)
- toujours créer ses objects avec la syntaxe `nom <- contenu`, et pas `nom = contenu`
- ne jamais appeler deux objets par le même nom
- ne jamais donner à un objet le nom d'une fonction
- toujours coder les facteurs avec au moins une lettre (et pas sous forme numérique), pour que R reconnaisse bien ces variables comme des facteurs.

Utilisation des fonctions

Dès qu'une fonction accepte comme argument une *formule* (voir fiche **40** bien que les formules soient utilisées aussi dans d'autres fonctions que celles créant un modèle), elle a un argument `data`. Celui-ci permet de préciser le tableau de données dans lequel aller chercher les variables contennues dans la formule, ce qui simplifie la rédaction la formule (donc sa clarté) mais évite aussi de provoquer des erreurs.

EXEMPLE(S)

Au lieu de (peu importe le nom de la fonction) :

```
> lm(tableau$y~tableau$x+tableau$z)
```

Préférer la syntaxe :

```
> lm(y~x+z,data=tableau)
```

Quand une fonction accepte plusieurs arguments, mieux vaut les appeler explicitement par leur nom pour éviter d'attribuer à un argument une valeur qui était destinée à un autre argument. En effet, sans utiliser les noms d'arguments il faut *strictement* respecter l'ordre des arguments tels que précisés dans la page d'aide de la fonction, ce qui est souvent source d'erreurs. Le premier ou les deux premiers arguments échappent à cette règle car ils sont souvent tellement logiques que l'on risque peu de se tromper.

Divers

Éviter d'utiliser les fonctions `attach()` et `detach()`, qui sont source de bien des erreurs. Grâce à l'argument `data` des fonctions acceptant une formule, ou de certaines fonctions telles que `with()`, on peut se rendre la vie aussi simple sans risquer quoi que ce soit.

Enfin, il est très fortement recommandé d'enregistrer les scripts de ses analyses afin de pouvoir y revenir plus tard (ou au moins d'être sûr de ce que l'on a fait !). Il est une très bonne chose d'aérer ses scripts et de les commenter (tout ce qui est situé après le symbole **#** est reconnu par **R** comme un commentaire), pour gagner en clarté et en compréhensivité par d'autres personnes (ou soi-même plus tard). Ne pas oublier de préciser les packages nécessaires à l'analyse dans le script.

8. Installer, charger et mettre à jour des packages

Installer un package

Il est nécessaire d'être connecté à internet pour installer un package, car celui-ci doit être téléchargé depuis un serveur. L'installation ne se fait qu'*une seule fois*.

Si **R** est utilisé depuis la console **R**, taper `install.packages("package")` où `package` est le nom du package désiré, entre guillemets. Il est demandé ensuite de choisir un serveur, Lyon 1 par exemple.

Si **R** est utilisé depuis la console système, la procédure se fait en deux étapes :

1. Télécharger les sources du package à partir de son site de dépôt, le site principal étant le CRAN (*the Comprehensive R Archive Network*) : *http://cran.r-project.org* rubrique *Packages*.
2. Installer le package en tapant `R CMD INSTALL package` où `package` est le nom du fichier `tar.gz` contenant les sources.

La procédure expliquée ici est la plus simple, mais il existe de nombreuses variantes. Voir la **R** FAQ pour plus d'informations : *http://cran.r-project.org/doc/manuals/R-admin.html#Installing-packages*.

Charger un package

Le chargement d'un package doit se faire *à chaque session où il doit être utilisé*. La commande est simple : `library(package)` où `package` est le nom du package, sans guillemets.

Mettre à jours les packages installés

Pour mettre à jour automatiquement tous les packages installés, taper `update.packages(ask=FALSE)`. **R** télécharge alors toutes les mises à jour et les installe. Attention, seuls les packages *non chargés* sont mis à jour. Mieux vaut donc faire les mises à jour avant toute autre chose lorsque **R** est ouvert.

Il est recommandé de mettre régulièrement à jour ses packages afin de profiter de leur évolution, souvent rapide.

9. Citer R et ses packages

Lors de l'écriture d'un document scientifique, il est une évidence de citer ses sources bibliographiques. Il doit également en être une de citer les logiciels utilisés lors de la réalisation de l'étude. **R** est certes gratuit, mais il n'en reste pas moins que des dizaines de personnes s'impliquent dans son développement, et qu'il est normal de faire honneur à leur travail en les citant.

R doit être cité dès lors qu'il est utilisé. Pour savoir comment le citer, il suffit de taper `citation()` et de recopier ce qui figure après `To cite R in publications use:`.

Concernant les packages, la règle implicite est de citer tous ceux qui ne sont pas chargés au démarrage de **R**. Cela comprend les packages installés avec **R** mais non chargés automatiquement, ainsi que ceux installés par l'utilisateur. Pour savoir comment les citer, taper `citation("package")` où `package` est le nom du package, entre guillemets. Recopier ce qui figure après `To cite the xxx package in publications use:`.

10. Changer de version de R

Pour installer une version plus récente de R, qui donne accès à des mises à jour de packages pas forcément disponibles avec une version plus ancienne (voir fiche 7), la procédure se fait en plusieurs étapes :

1. Récupérer le chemin vers le dossier où sont installés les packages, *via* `.libPaths()` (dans ce dossier, il y a un sous-dossier par package).
2. Désinstaller R, comme n'importe quel autre logiciel. Tous les fichiers sont ainsi supprimés *à l'exception des packages installés*.
3. Installer la nouvelle version de R, téléchargeable depuis le site officiel : *https://cran.r-project.org*.
4. Ouvrir R et récupérer le chemin vers le dossier où sont installés les packages de cette nouvelle version, *via* `.libPaths()`. Ce chemin *n'est pas le même* que celui récupéré à l'étape 1, car il est spécifique à chaque version de R.
5. Copier tous les sous-dossiers (*i.e.* un par package) présents dans le dossier identifié à l'étape 1, et les coller dans le dossier identifié à l'étape 4. Ainsi tous les packages qui étaient installés sous la précédente version de R le sont désormais sous la nouvelle version.
6. Supprimer le dossier identifié l'étape 1, qui n'a plus d'utilité.
7. Mettre à jour tous les packages avant de faire quoi que ce soit d'autre (voir fiche 8).

Deuxième partie

Théorie statistique élémentaire

NOTIONS GÉNÉRALES

11. Les différents types de variable
12. Le plan d'échantillonnage
13. Le plan d'expérience

FONCTIONNEMENT D'UN TEST STATISTIQUE

14. Principe des tests statistiques et risques associés à la conclusion
15. Le risque ou seuil de rejet α
16. La correction du seuil de rejet α (ou des *p-values*)
17. Risque β et puissance d'un test
18. Déterminer la taille de l'échantillon à constituer

LOIS DE PROBABILITÉS USUELLES

Lois de probabilité discontinues

19. Les lois de probabilité discontinues – généralités
20. La loi binomiale
21. La loi de Poisson
22. La loi binomiale négative

Lois de probabilité continues

23. Les lois de probabilité continues – généralités
24. La loi normale
25. La loi exponentielle
26. La loi de χ^2
27. La loi de Fisher - Snedecor
28. La loi de Student

11. Les différents types de variable

Il existe deux types de variable :

1. Quantitatives : leurs valeurs représentent une *grandeur*, quantifiable et le plus souvent associée à une unité de mesure. On peut effectuer des opérations mathématiques avec ces variables. Elles peuvent être de deux types :
   — continues, lorsqu'elles peuvent prendre une infinité de valeurs (dans un intervalle donné) : masse, temps, distance, volume…
   — discrètes, lorsqu'elles ne peuvent prendre que certaines valeurs (dans un intervalle donné) : nombre d'individus, d'évènements… Ces variables sont liées le plus souvent à des processus de comptage (où les valeurs prises ne peuvent être qu'entières et positives ou nulles).
2. Qualitatives : leurs valeurs ne représentent pas une quantité mais une *catégorie*. On ne peut donc pas effectuer d'opérations mathématiques avec ces variables. On les appelle des *facteurs*, et les valeurs qu'elles peuvent prendre des *classes*, *niveaux* ou *modalités*. Les variables qualitatives peuvent être de deux types :
   — ordinales, lorsque les classes peuvent être ordonnées : rang dans un classement, degré de satisfaction…
   — nominales, lorsque les classes ne peuvent pas être ordonnées : sexe, pays…

Les variables qualitatives peuvent être codées numériquement, mais il est très important de les différencier des variables quantitatives (car elles sont traitées différemment lors de l'analyse statistique). Un principe simple pour faire la différence : si l'on peut remplacer les valeurs d'une variable numérique par des énoncés, alors elle est qualitative. Par exemple, si l'on juge l'état d'une plante par une note allant de 0 à 5, on peut remplacer la note chiffrée par une appréciation du type « mauvais », « satisfaisant »… Cette variable est donc qualitative. Pour ne pas prendre de risque, il est conseillé de ne jamais coder les variables qualitatives numériquement.

Il existe deux types de facteur :

— à effet fixe (« facteur fixe ») : un facteur est fixe si ses classes ont été délibérément choisies, et si le but de l'étude est de les comparer. Par exemple, si l'on veut comparer la taille des individus entre trois espèces, le facteur « espèce » est fixe (à trois classes).
— à effet aléatoire (« facteur aléatoire ») : un facteur est aléatoire si ses classes ont été choisies parmi un grand nombre de classes possibles, et si le but de l'étude n'est pas de les comparer mais simplement de prendre en compte la variabilité qu'il existe entre elles. Par exemple, si les mesures de taille des trois espèces sont réalisées par deux personnes différentes, on peut considérer un facteur « expérimentateur », aléatoire. L'objectif ici n'est en effet pas de comparer les mesures réalisées par les deux personnes, mais de prendre en compte le fait que la façon de réaliser les mesures peut varier entre les deux.

Il y a deux choses à bien garder à l'esprit : (i) la décision de déclarer un facteur comme fixe ou aléatoire est *fondamentale* pour l'analyse des données, car ce ne sont pas les mêmes analyses qui sont réalisées dans les deux cas ; (ii) cette décision doit être prise *selon l'objectif de l'étude, i.e.* la question à laquelle l'étude doit répondre, car aucun facteur n'est fixe ou aléatoire dans l'absolu. Il est donc indispensable de bien réfléchir avant de déclarer un facteur comme fixe ou aléatoire.

Que ce soit pour des variables qualitatives ou quantitatives, si certaines mesures ne sont pas indépendantes entre elles, elles constituent des *séries appariées*. Le cas le plus simple est celui où plusieurs mesures sont réalisées sur un même individu (par exemple avant et après un traitement). Mais d'autres cas plus subtils peuvent se présenter : si des mesures sont réalisées sur des individus apparentés (ces mesures ne sont pas indépendantes car il existe une corrélation d'origine génétique entre elles), si des séries de mesures sont réalisées à des localisations différentes (ces mesures ne sont pas indépendantes car chaque série est influencée par l'environnement local) ou encore si des séries de mesures sont réalisées à des temps différents (ces mesures ne sont pas indépendantes car chaque série est influencée par ce qu'il a pu se passer avant). Il est très important d'identifier les séries appariées lorsqu'elles existent, car des analyses statistiques adaptées doivent alors être utilisées. Les séries appariées sont le plus souvent identifiées par l'introduction d'un facteur aléatoire. Pour les exemples précédents, on a donc respectivement un facteur « individu », un facteur « famille », un facteur « localisation » et un facteur « moment ».

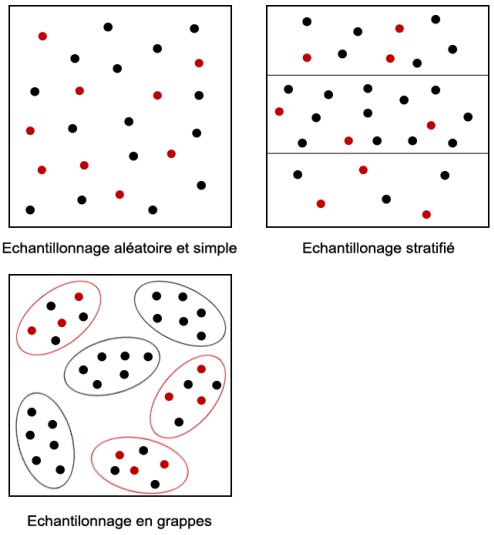
12. Le plan d'échantillonnage

On utilise un plan d'échantillonnage lorsque l'on réalise une étude par enquête, *i.e.* lorsque l'on collecte des informations sur un groupe d'individus dans leur milieu habituel, mais que tous les individus ne sont pas accessibles (par choix ou par contrainte).

Les principales méthodes d'échantillonnage peuvent être regroupées en deux ensembles :

1. L'échantillonnage aléatoire : tous les individus (au sens statistique) ont la même probabilité d'être choisis, et le choix de l'un n'influence pas celui des autres. Différentes méthodes d'échantillonnage aléatoire existent (voir les illustrations) :
   — l'échantillonnage aléatoire et simple : le choix se fait parmi tous les individus de la population (au sens statistique), qui ne forme qu'un grand ensemble.
   — l'échantillonnage stratifié : si la population est très hétérogène, elle peut être divisée en sous-ensembles exclusifs (ou strates). Au sein de ces strates l'échantillonnage est ensuite aléatoire et simple. Les strates sont identifiées dans l'analyse statistique comme les niveaux d'un facteur fixe.
   — l'échantillonnage en grappes : si les strates sont très nombreuses, on en choisit certaines au hasard (les grappes). Au sein de ces grappes l'échantillonnage est ensuite aléatoire et simple. Les grappes sont identifiées dans l'analyse statistique comme les niveaux d'un facteur aléatoire.
   — l'échantillonnage par degrés : il est une généralisation de l'échantillonnage en grappes (qui est en fait un échantillonnage du premier degré). Au sein de la population on choisit des grappes « primaires », puis à l'intérieur de celles-ci des grappes « secondaires » (toujours au hasard), et ainsi du suite… Au dernier niveau l'échantillonnage est aléatoire et simple.
2. L'échantillonnage systématique : un premier individu est choisi aléatoirement, puis les autres sont choisis de façon régulière à partir du précédent (dans le temps ou l'espace). L'analyse de ce type d'échantillonnage, qui fait appel à la statistique spatiale ou à l'analyse des séries chronologiques, n'est pas abordée dans cet ouvrage.

Il est important d'identifier la méthode mise en œuvre car les analyses statistiques doivent être adaptées. Seule l'analyse de plans d'échantillonnage aléatoires est abordée dans cet ouvrage.



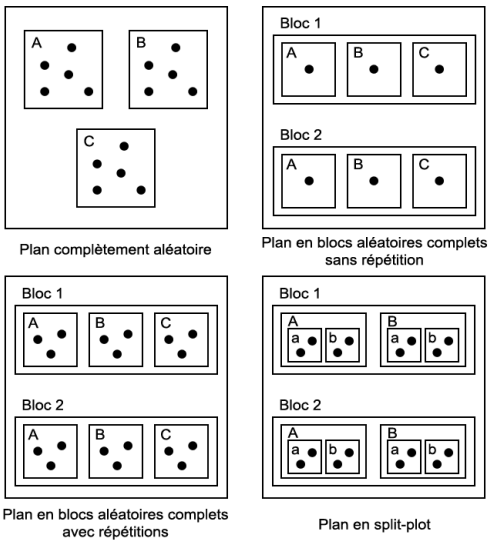
Chaque point représente un individu. Les individus échantillonnés sont représentés en rouge.

13. Le plan d'expérience

On utilise un plan d'expérience lorsque l'on réaliste une étude par expérimentation, *i.e.* lorsque l'on provoque volontairement les faits à étudier. Le plan d'expérience comprend notamment le(s) facteur(s) à faire varier, le nombre de répétitions à réaliser et le dispositif expérimental à mettre en place. L'association des classes de plusieurs facteurs constitue un *traitement*.

Il existe de nombreux types de dispositif expérimental, dont les principaux sont (voir les illustrations) :
— le plan d'expérience complètement aléatoire : chaque individu (au sens statistique) est affecté à un traitement aléatoirement.
— le plan d'expérience en blocs aléatoires complets : s'il y a (ou s'il peut y avoir) une grande hétérogénéité entre les individus, ils sont réunis en groupes aussi homogènes que possibles (ou blocs). Au sein de ces blocs chaque individu est ensuite affecté aléatoirement à un traitement, de manière à ce que tous les traitements soient présents dans chacun des blocs. Les blocs sont identifiés dans l'analyse statistique comme les niveaux d'un facteur aléatoire.
— le plan d'expérience en blocs aléatoires incomplets : dans ce cas tous les traitements ne sont pas présents dans chacun des blocs.
— le plan d'expérience en split-plot : le principe du split-plot est le plus souvent associé à celui des blocs aléatoires complets. Dans ce cas, dans chacun des blocs sont créés autant de sous-blocs qu'il y a de classes au premier facteur étudié. À chacun de ces sous-blocs est associée une classe. Puis chaque sous-bloc est divisé en autant d'unités qu'il y a de classes au second facteur étudié. À chacun de ces « sous-sous-blocs » est associée une classe. Pour plus de deux facteurs, la situation est plus complexe.

Quelle que soit la méthode employée, elle doit être clairement définie car elle doit être prise en compte dans les analyses statistiques.



Chaque point représente un individu. Chaque carré représente un traitement. Les niveaux du 1er facteur sont notés en majuscule, les niveaux du 2nd facteur sont notés en minuscule.

14. Principe des tests statistiques et risques associés à la conclusion

Le principe de réalisation de tout test statistique est le suivant :

1. On pose une hypothèse nulle H_0, de type « rien à signaler » (ex : les moyennes μ_A et μ_B sont égales) ou « valeur ponctuelle » (ex : la moyenne $\mu = 10$, la proportion $p = 50$ %).
2. On pose une hypothèse H_1, de telle manière que H_0 et H_1 soient exclusives (ex : les moyennes μ_A et μ_B sont différentes, la moyenne $\mu \neq 10$).
3. On calcule la valeur de la Variable de Test (VT), d'après une formule qui dépend du test utilisé.
4. On utilise la valeur de la VT calculée pour déterminer une *p-value, i.e.* une probabilité d'obtenir la valeur mesurée (moyenne, pourcentage…) si H_0 est vraie.
5. On conclut sur les deux hypothèses posées grâce à cette *p-value* :
   — si la *p-value* est supérieure au seuil α fixé avant le test (5 % en général, voir fiche **15**), on ne rejete pas H_0.
   — si la *p-value* est inférieure au seuil α, on rejete H_0.

Conclure sur les deux hypothèses présente deux risques :
— le risque de 1ère espèce ou risque α, qui correspond au risque de rejeter de H_0 si celle-ci est réellement vraie. On fixe ce risque, le plus souvent à 5 % (voir fiche **15**).
— le risque de 2ème espèce ou risque β, qui correspond au risque de ne pas rejeter de H_0 si celle-ci est réellement fausse. On ne peut pas fixer ce risque (voir fiche **17**).

	Réalité (inconnue le plus souvent)	
Décision	H_0 vraie	H_0 fausse
H_0 non rejetée	Bonne décision	Erreur β
H_0 rejetée	Erreur α	Bonne décision

La probabilité associée au fait de rejeter H_0 si celle-ci est fausse (soit $1-\beta$) est appelée *puissance* du test (voir fiche **17**).

Il est important de distinguer les notions d'effet *statistiquement* significatif et *biologiquement* significatif. Statistiquement, on peut toujours rejeter H_0 moyennant d'avoir un échantillon suffisamment grand (voir fiche **17**). D'un point de vue statistique, H_0 n'est donc ni « vraie » ni « fausse ». Cette notion de vrai ou faux se pose uniquement en terme biologique. En pratique, on observe toujours un effet, aussi petit soit-il. La question est de savoir si cet effet (ou plutôt l'effet réel, qui n'est qu'estimé par l'expérimentation) est suffisamment grand pour qu'il ait une signification biologique. Si oui alors H_0 est réellement fausse, si non H_0 est réellement vraie.

15. Le risque ou seuil de rejet ®

Le risque ®, ou seuil de rejet ou encore seuil de signi cation de l'hypothèse H_0, est une valeur xée arbitrairement avant la réalisation de tout test statistique . Elle correspond à un risque assumé de se tromper, celui de rejeter H_0 si celle-ci est réellement vraie (ce que bien sûr on ne sait jamais).

Lorsque l'on calcule une p-value suite à un test (voir che 14), on la compare au seuil ® choisi :

— si la p-value est inférieure au seuil ®, on rejette H_0. Il faut être conscient que l'on prend un risque ® de se tromper.
— si la p-value est supérieure au seuil ®, on ne rejette pas H_0.

Obtenir une p-value supérieure au seuil ® peut avoir deux origines :

— une réelle véracité de H_0 (au sens biologique, voir che 14)
— un manque de puissance du test (voir che 17), i.e. un risque (¯, voir che 14) important de « passer à côté » d'une réelle fausseté de H_0. L'effet observé peut donc être biologiquement signi catif, mais la taille de l'échantillon insuf sante pour le montrer d'un point de vue statistique. D'où l'importance de déterminer la taille de l'échantillon en fonction de ce seuil de signi cativité biologique , dès que ce seuil est estimable (voir che 18).

X_{seuil} : valeur de la Variable de Test (VT) X qui donne une fonction de répartition à droite égale au seuil ® (test unilatéral droit ici).

X_{calc} : valeur de la VT X calculée à partir de l'échantillon testé.

À gauche l'hypothèse H_0 est rejetée, à droite elle ne l'est pas.

D'un point de vue pratique, on utilise souvent un seuil ® de 5 %. Cela veut dire qu'on assume un risque de 5 % de rejeter H_0 si celle-ci est réellement vraie.

Qualité de l'analyse

Test(s)

L'effet des variables explicatives est testé *via* `MVA.anova(RDA)`[2]. Un test F par permutation est réalisé.

Si au moins une variable explicative a un effet significatif, on peut se baser sur les résultats de l'ACP *contrainte* pour l'interprétation. Si aucune variable explicative n'a d'effet significatif, interpréter les résultats de cette ACP n'a pas beaucoup d'intérêt puisqu'aucun effet n'est montré.

En lien avec cette ACP contrainte, on peut réaliser des comparaisons multiples entre modalités d'un facteur (ou combinaisons de modalités d'une interaction entre facteurs) à effet significatif. Pour réaliser le test : `pairwise.factorfit(RDA,facteur)`[2] où `facteur` est le facteur d'intérêt.

Synthèse

Si au moins une variable explicative a un effet significatif, on s'intéresse à l'ACP contrainte. Comme pour une ACP classique (voir fiche **96**), on estime la qualité de cette analyse par le pourcentage de variance expliqué par chaque axe. Ces pourcentages sont obtenus *via* `MVA.synt(RDA)`[2], dans le deuxième tableau renvoyé par la fonction.

Remarque 1 : il s'agit ici de pourcentages de variance *contrainte*, pas totale comme en ACP classique.

Remarque 2 : les pourcentages de variance sont toujours en ordre décroissant (*i.e.* l'axe 1 explique plus de variance que l'axe 2, qui en explique lui-même plus que l'axe 3...).

Remarque 3 : il n'y a pas de règle absolue sur le nombre d'axes à retenir pour l'interprétation. Il s'agit toujours d'un compromis entre une bonne synthèse de l'information (qui augmente avec le nombre d'axes) et une facilité à interpréter (qui diminue avec le nombre d'axes).

Représentations graphiques

En RDA on a deux représentations possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Graphe des individus

Pour tracer le graphe : `MVA.plot(RDA)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Par défaut c'est l'ACP contrainte qui est représentée. Pour représenter l'ACP non contrainte, ajouter l'argument `space=2`.

Pour ajouter des groupes sur le graphe, utiliser l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe de chaque individu. Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser une telle représentation.

Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(RDA,"corr")`[2]. Comme pour le graphe des individus c'est l'ACP contrainte qui est représentée, l'argument `space=2` permettant de représenter l'ACP non contrainte. Par défaut à la fois les variables à expliquer et les variables explicatives quantitatives sont représentées. Pour ne représenter que les variables explicatives quantitatives ajouter l'argument `set=1`, pour les variables à expliquer `set=2`.

Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

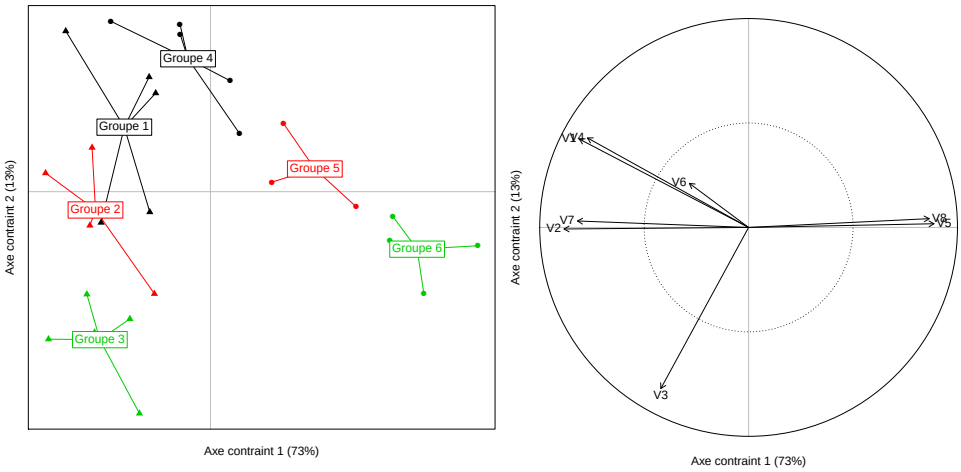
Interprétation

On ne considère que l'ACP contrainte puisque par définition c'est la seule qui permette d'interpréter les résultats en lien avec les variables explicatives.

Le graphe des individus permet d'identifier la structuration des données du tableau à expliquer qui est due aux variables explicatives. On y repère comment les modalités d'un facteur à effet significatif se répartissent, ou des gradients linéaires.

Dans un second temps, le cercle des corrélations permet d'identifier les variables (i) qui différencient d'éventuels groupes et/ou (ii) qui expliquent d'éventuels gradients. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe des individus (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions

sur le cercle des corrélations (voir fiche **89**). La façon de procéder est la même pour les variables à expliquer que pour les variables explicatives quantitatives, sauf bien sûr sur que les variables à expliquer sont influencées par les variables explicatives mais pas l'inverse.



104. L'analyse discriminante linéaire

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]vegan, [2]MASS, [3]RVAideMemoire

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse discriminante linéaire (ADL)
Anglais – *Linear discriminant analysis* (*LDA*)
Canonical variate analysis (*CVA*)
Discriminant function analysis (*DFA*)

Préparation des données

Les variables quantitatives doivent avoir une distribution normale multivariée. Voir fiche **65** pour tester cette condition. La LDA est toutefois assez robuste si elle n'est pas tout à fait respectée.

Il est également nécessaire que les matrices de variance-covariance (l'équivalent multivarié de la variance) soient homogènes entre les différents groupes. Pour le tester : `anova(betadisper(dist(tableau),facteur)`[1]`)` où `tableau` est le tableau de variables quantitatives et `facteur` le facteur définissant les groupes. Là encore la LDA est assez robuste à un non respect (modéré) de cette condition.

Si ces deux conditions ne sont pas du tout respectées, une transformation préalable des données peut grandement aider à améliorer la situation (voir fiche **88**).

Remarque : les variables quantitatives doivent être standardisées (voir fiche **88**), mais en pratique les fonctions réalisant la LDA le feront automatiquement

Réalisation de l'analyse

Pour réaliser la LDA : `LDA<-lda(tableau,facteur)`[2].

Qualité de l'analyse

Pourcentage d'erreur de classification

Une des fonctions de toute analyse discriminante est de faire de la *prédiction, i.e.* de prédire le groupe auquel appartient un individu dont on connaît seulement la valeur pour les variables quantitatives. Une manière d'estimer la qualité d'une analyse discriminante est donc de tester à quel point elle permet de classer un individu de groupe inconnu sans erreur. On utilise pour cela une méthode de *validation croisée*, qui va générer plusieurs sous-modèles chacun sur une partie du jeu de données (définie aléatoirement) et prédire le groupe des individus non pris en compte dans le modèle. Pour plus de fiabilité l'ensemble de la procédure peut être répété plusieurs fois (avec à chaque fois un découpage aléatoire du jeu de données).

Pour réaliser la validation croisée : `MVA.cv(tableau,facteur,model="LDA")`[3]. Par défaut la fonction découpe le jeu de données en 7 parties (*7-fold cross-validation*), mais ce chiffre peut être modifié *via* l'argument `k=nb` où `nb` est le nombre de sous-jeux de données à générer. La fonction peut également ajuster ce chiffre automatiquement si au moins un groupe contient moins de 7 individus. Par défaut l'ensemble de la procédure est répété 10 fois (argument `repet`), ce qui au final génère $7 \times 10 = 70$ sous-modèles.

Si l'on souhaite utiliser la LDA dans un but prédictif, stocker le résultat de la fonction `MVA.cv()`[3] dans un objet.

Test(s)

Le test à réaliser est une MANOVA (*Multivariate ANalysis Of VAriance*), une extenstion de l'ANOVA au cas multivarié. Les conditions d'emploi de ce test sont les mêmes que celles de la LDA. Voir fiche **108** pour réaliser le test.

Si le facteur a un effet significatif et qu'il y a plus de deux groupes, il est nécessaire de réaliser des comparaisons multiples pour identifier les modalités qui diffèrent. Voir fiche **43** pour réaliser ces comparaisons.

Remarque 1 : pourcentage d'erreur de classification et significativité du facteur sont toujours cohérents : si le pourcentage d'erreur est faible, le facteur a un effet significatif (et inversement).

Remarque 2 : si le facteur n'a pas d'effet significatif, l'interprétation des résultats de la LDA n'a pas vraiment de sens et toute tentative de prédiction est inutile.

Représentations graphiques

En LDA on a deux représentations possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Remarque : s'il n'y a que deux groupes, la LDA ne produit qu'un seul axe. Le graphe des individus et le cercle des corrélations se réduisent donc à une seule dimension.

Graphe des individus

Pour tracer le graphe : `MVA.plot(LDA,fac=facteur)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut (s'il y a au moins deux axes), ils peuvent être changés grâce aux arguments `xax` et `yax`.

Dans le cas d'un graphe à deux dimensions, les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser la représentation. Dans le cas d'un graphe à une seule dimension, les arguments `col`, `legend` et `legend.lab` sont intéressants. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(LDA,"corr")`[2].

Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

Le graphe des individus permet d'identifier comment les groupes se structurent, *i.e.* si tout ou partie des groupes se séparent ou au contraire se chevauchent.

Dans un second temps, le cercle des corrélations permet d'identifier les variables qui différencient ces groupes. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe des individus (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions sur le cercle des corrélations (voir fiche **89**).

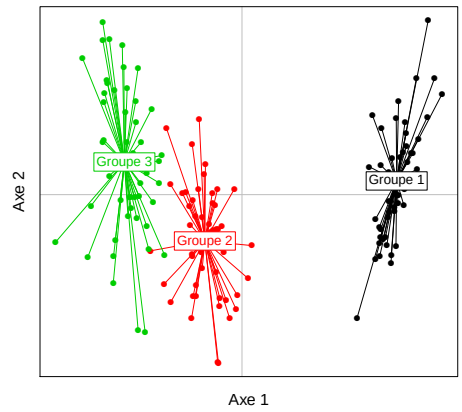
Prédiction

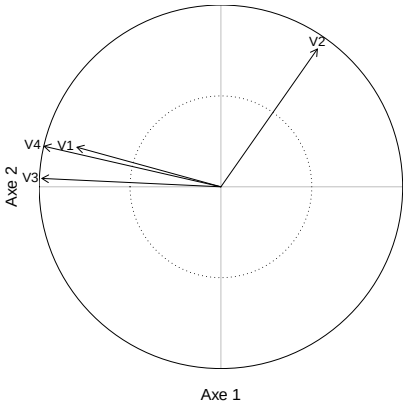
L'un des intérêts des analyses discriminantes est à la fois de comprendre comment les groupes se différencient, mais également de pouvoir prédire le groupe d'un individu pour des valeurs *connues* des variables quantitatives. Prédire un groupe nécessite donc de fixer la valeur de *toutes* les variables quantitatives.

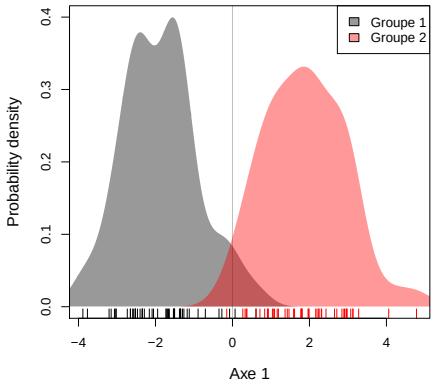
Cette prédiction se fait en trois étapes :

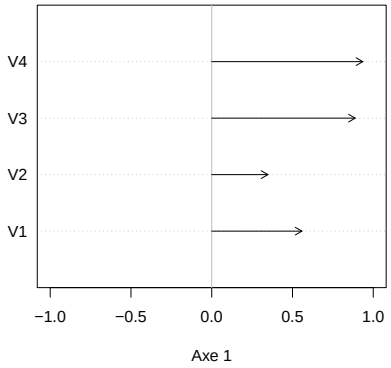
1. Générer une série de sous-modèles à partir du jeu de données initial, par validation croisée.
2. Créer un tableau contenant une colonne par variable quantitative (les noms de colonnes doivent être *strictement identiques* aux noms des variables du jeu de données initial), et remplir chaque ligne en fonction des valeurs pour lesquelles la prédiction doit être faite (il y a donc une prédiction par ligne de ce tableau). Si le jeu de données initial avait été standardisé, il faut également standardiser le jeu de données servant à la prédiction. Pour cela : `new.tab<-stand(new.tab,tableau)`[1] où `new.tab` est le tableau des individus à classer.
3. Réaliser la prédiction : `predict(LDA.vc,new.tab)` où `LDA.vc` est le résultat de la validation croisée (étape 1). La fonction renvoie pour chaque ligne de `new.tab` le groupe prédit (colonne `Group`) et la probabilité de cette prédiction (colonne `Proba`).

Remarque : tout l'intérêt de générer un grand nombre de sous-modèles au moment de la validation croisée est de pouvoir associer à chaque prédiction une probabilité, car chaque sous-modèle va en fait servir à faire sa propre prédiction, ce qui permet d'estimer la fiabilité de la prédiction « moyenne ».


Groupe 3
Groupe 2
Groupe 1
Axe 2
Axe 1


V2
V4
V1
V3
Axe 2
Axe 1


Groupe 1
Groupe 2
Probability density
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
Axe 1


V4
V3
V2
V1
−1.0
−0.5
0.0
0.5
1.0
Axe 1

105. La régression PLS discriminante

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`RVAideMemoire`, [2]`pls`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Régression PLS discriminante
Régression des moindres carrés partiels discriminante

Anglais – *Partial least squares discriminant analysis* (*PLS-DA*)
Projection to latent structures discriminant analysis (*PLS-DA*)

Préparation des données

Il est fortement recommandé de standardiser les variables quantitatives (voir fiche **88**). Une transformation préalable est également souvent bénéfique, pour rendre plus linéaires les relations entre ces variables. Dans tous les cas, le tableau de variables quantitatives doit être transformé en matrice, par exemple *via* `tableau<-as.matrix(tableau)` où `tableau` est le tableau de variables quantitatives.

Remarque : si les variables ont été standardisées (voir fiche **88**), le résultat de la standardisation est déjà une matrice.

Le facteur doit également être transformé en *variables indicatrices*. Pour cela : `var.ind<-dummy(facteur)`[1] où `facteur` est le facteur définissant les groupes.

Réalisation de l'analyse

Pour réaliser la PLS-DA : `PLSDA<-cppls(var.ind~tableau)`[2].

Remarque : par défaut autant d'axes sont créés qu'il y a de variables quantitatives, ce qui peut parfois générer une erreur. Si tel est le cas, réduire le nombre d'axes en ajoutant l'argument `ncomp=10` (10 axes sont largement plus que nécessaire pour la suite).

Qualité de l'analyse

Pourcentage d'erreur de classification

Une des fonctions de toute analyse discriminante est de faire de la *prédiction*, *i.e.* de prédire le groupe auquel appartient un individu dont on connaît seulement la valeur pour les variables quantitatives. Une manière d'estimer la qualité d'une analyse discriminante est donc de tester à quel point elle permet de classer un individu de groupe inconnu sans erreur. On utilise pour cela une méthode de *validation croisée*, qui va générer plusieurs sous-modèles chacun sur une partie du jeu de données (définie aléatoirement) et prédire le groupe des individus non pris en compte dans le modèle. Pour plus de fiabilité l'ensemble de la procédure peut être répété plusieurs fois (avec à chaque fois un découpage aléatoire du jeu de données). Dans le cas d'une PLS-DA les choses sont toutefois assez complexes, et une procédure de *double validation croisée* (*cross model validation* ou 2CV) doit être utilisée, qui consiste en une validation croisée dont à chaque étape on réalise une seconde validation croisée interne (la validation croisée « interne » étant appelée *inner loop*, tandis que la validation croisée « externe » est appelée *outer loop*).

Pour réaliser l'opération : `MVA.cmv(tableau,facteur,model="PPLS-DA",crit.inn="NMC")`[1]. Par défaut le découpage du jeu de données se fait en 7 parties au niveau de la boucle externe (*7-fold cross-validation*) et en 6 parties au niveau de la boucle interne (*6-fold cross-validation*). Ces chiffres peuvent être modifiés grâce aux arguments `kout` et `kinn`. La fonction peut également ajuster ces chiffres automatiquement si au moins un groupe contient moins de 7 individus. Par défaut l'ensemble de la procédure est répété 10 fois (argument `repet`), ce qui au final génère $7 \times 10 = 70$ sous-modèles.

Remarque 1 : si g est le nombre de groupes, on utilise souvent $g-1$ axes pour tester et interpréter une PLS-DA, car c'est théoriquement suffisant. Pour suivre ce principe, ajouter l'argument `ncomp=nb` où `nb` $= g-1$.

Remarque 2 : il existe en fait plusieurs versions de la PLS-DA. La PPLS-DA (*Powered PLS-DA*) en est la plus efficace, c'est celle qui est recommandée et utilisée dans cette fiche.

Si l'on souhaite utiliser la PLS-DA dans un but prédictif, stocker le résultat de la fonction `MVA.cmv()`[1] dans un objet.

Test(s)

Le test à réaliser est un test par permutation basé sur la double validation croisée. Pour le réaliser : `MVA.test(tableau,facteur,model="PPLS-DA",cmv=TRUE)`[1]. Les arguments `kout`, `kinn` et surtout `ncomp` sont disponibles. Le temps de calcul nécessaire pour ce test est relativement long.

Si le facteur a un effet significatif et qu'il y a plus de deux groupes, il est nécessaire de réaliser des comparaisons multiples pour identifier les modalités qui diffèrent. Pour cela : `pairwise.MVA.test(tableau,facteur,model="PPLS-DA",cmv=TRUE)`[1] (avec toujours les arguments `kout`, `kinn` et surtout `ncomp`). Le temps de calcul nécessaire pour réaliser toutes les comparaisons deux-à-deux est très long.

Remarque : avec une PLS-DA, il est *absolument indispensable* de tester si le facteur a un effet significatif avant toute interprétation. En effet, les représentations graphiques auront toujours tendance à montrer des groupes séparés *même si cette séparation est totalement aléatoire* (le phénomène est d'autant plus probable que les variables quantitatives sont bien plus nombreuses que les individus). On ne peut donc *pas* se fier à ces représentations graphiques si les groupes ne sont pas significativement différents.

Représentations graphiques

En PLS-DA on a deux représentations possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Graphe des individus

Pour tracer le graphe : `MVA.plot(PLSDA,fac=facteur)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`.

Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser la représentation. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(PLSDA,"corr")`[2].

Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

On part du principe que les groupes sont significativement différents. Dans le cas contraire l'interprétation *doit être évitée*.

Le graphe des individus permet d'identifier comment les groupes se structurent, *i.e.* si tout ou partie des groupes se séparent ou au contraire se chevauchent.

Dans un second temps, le cercle des corrélations permet d'identifier les variables qui différencient ces groupes. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe des individus (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions sur le cercle des corrélations (voir fiche **89**).

Prédiction

L'un des intérêts des analyses discriminantes est à la fois de comprendre comment les groupes se différencient, mais également de pouvoir prédire le groupe d'un individu pour des valeurs *connues* des variables quantitatives. Prédire un groupe nécessite donc de fixer la valeur de *toutes* les variables quantitatives.

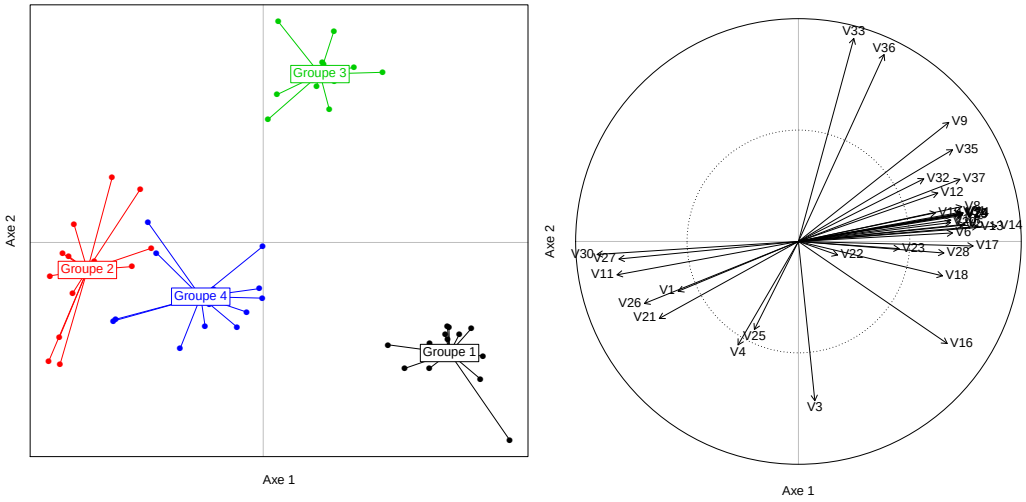
Cette prédiction se fait en trois étapes :

1. Générer une série de sous-modèles à partir du jeu de données initial, par double validation croisée.
2. Créer un tableau contenant une colonne par variable quantitative (les noms de colonnes doivent être *strictement identiques* aux noms des variables du jeu de données initial), et remplir chaque ligne en fonction des valeurs pour lesquelles la prédiction doit être faite (il y a donc une prédiction

par ligne de ce tableau). Si le jeu de données initial avait été standardisé, il faut également standardiser le jeu de données servant à la prédiction. Pour cela : `new.tab<-stand(new.tab,tableau)`[1] où `new.tab` est le tableau des individus à classer.

3. Réaliser la prédiction : `predict(PLSDA.vc,new.tab)` où `PLSDA.vc` est le résultat de la doube validation croisée (étape 1). La fonction renvoie pour chaque ligne de `new.tab` le groupe prédit (colonne `Group`) et la probabilité de cette prédiction (colonne `Proba`).

Remarque : tout l'intérêt de générer un grand nombre de sous-modèles au moment de la double validation croisée est de pouvoir associer à chaque prédiction une probabilité, car chaque sous-modèle va en fait servir à faire sa propre prédiction, ce qui permet d'estimer la fiabilité de la prédiction « moyenne ».



106. L'analyse canonique des correspondances

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`vegan`, [2]`RVAideMemoire`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse canonique des correspondances
Analyse factorielle des correspondances sur variables instrumentales (AFCVI)

Anglais – *Canonical correspondence analysis* (*CCA*)
Constrained correspondence analysis (*CCA*)
Correspondence analysis with respect to instrumental variables (*CAIV*)

Préparation des données

Dans un tableau de contingence ou un tableau de présence-absence, les lignes et les colonnes ont généralement un rôle symétrique (*i.e.* il n'y a pas d'« individus » et de « variables »). Cependant la CCA impose que les entités pour lesquelles les variables explicatives sont définies soient placées en *lignes*, comme dans un tableau classique. On appellera donc les lignes « individus ».

Réalisation de l'analyse

Pour réaliser la CCA : `CCA<-cca(formule,data=tab.explicatif)`[1] où `tab.explicatif` est le tableau contenant les variables explicatives. Voir fiche **40** pour une explication détaillée de la construction d'une formule. Dans cette formule, la réponse est le tableau à expliquer.

Capacité explicative globale

La CCA consiste en fait en deux étapes :

1. Séparer la variation (du tableau à expliquer) due aux variables explicatives (appelée variation *contrainte*) de la variation non expliquée (dite *résiduelle* ou *non contrainte*). La CCA travaille sur une certaine information qui est la *correspondance* entre les lignes et les colonnes du jeu de données, que l'on appellera plus généralement *inertie* (l'inertie est en fait un terme générique, le fait qu'elle représente une correspondance est un cas particulier). À ce stade le tableau à expliquer est considéré asymétriquement puisque les lignes jouent le rôle d'individus.
2. Réaliser deux AFC séparées, l'une sur la variation contrainte (« AFC contrainte ») et l'autre sur la variation non contrainte (« AFC non contrainte »). Comme en AFC classique (voir fiche **97**), lignes et colonnes jouent cette fois un rôle symétrique.

On peut estimer la capacité explicative globale de la CCA grâce au pourcentage d'inertie contrainte de l'analyse (*i.e.* d'inertie du tableau à expliquer expliquée par les variables explicatives). Plus ce pourcentage est élevé et plus la variation observée dans le tableau à expliquer est liée aux variables explicatives. Ce pourcentage est obtenu *via* `MVA.synt(CCA)`[2], dans le premier tableau renvoyé par la fonction.

Qualité de l'analyse

Test(s)

L'effet des variables explicatives est testé *via* `MVA.anova(CCA)`[2]. Un test F par permutation est réalisé.

Si au moins une variable explicative a un effet significatif, on peut se baser sur les résultats de l'AFC *contrainte* pour l'interprétation. Si aucune variable explicative n'a d'effet significatif, interpréter les résultats de cette AFC n'a pas beaucoup d'intérêt puisqu'aucun effet n'est montré.

En lien avec cette AFC contrainte, on peut réaliser des comparaisons multiples entre modalités d'un facteur (ou combinaisons de modalités d'une interaction entre facteurs) à effet significatif. Pour réaliser

le test : `pairwise.factorfit(CCA,facteur)`[2] où `facteur` est le facteur d'intérêt.

Synthèse

Si au moins une variable explicative a un effet significatif, on s'intéresse à l'AFC contrainte. Comme pour une AFC classique (voir fiche **97**), on estime la qualité de cette analyse par le pourcentage d'inertie expliqué par chaque axe. Ces pourcentages sont obtenus *via* `MVA.synt(CCA)`[2], dans le deuxième tableau renvoyé par la fonction.

Remarque 1 : il s'agit ici de pourcentages d'inertie *contrainte*, pas totale comme en AFC classique.

Remarque 2 : les pourcentages d'inertie sont toujours en ordre décroissant (*i.e.* l'axe 1 explique plus d'inertie que l'axe 2, qui en explique lui-même plus que l'axe 3...).

Remarque 3 : il n'y a pas de règle absolue sur le nombre d'axes à retenir pour l'interprétation. Il s'agit toujours d'un compromis entre une bonne synthèse de l'information (qui augmente avec le nombre d'axes) et une facilité à interpréter (qui diminue avec le nombre d'axes).

Représentations graphiques

À partir du moment où il y a au moins une variable explicative quantitative, deux représentations sont possibles : le « graphe d'association » et le *cercle des corrélations* (voir fiche **89**). S'il n'y a aucune variable explicative quantitative le seul graphe possible est celui d'association.

Graphe d'association

Sur ce graphe les lignes et les colonnes sont représentées chacune par un point. Il est cependant impossible de représenter à la fois les distances interlignes et les distances intercolonnes sans biais sur le même graphe. Il faut donc choisir entre représenter sans biais les premières (échelle de type 1) ou les secondes (échelle de type 2).

Pour représenter les distances interlignes sans biais : `MVA.plot(CCA,points=FALSE,scaling=1)`[2]. Pour représenter les distances intercolonnes sans biais : `MVA.plot(CCA,points=FALSE,scaling=2)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Par défaut c'est l'AFC contrainte qui est représentée. Pour représenter l'AFC non contrainte, ajouter l'argument `space=2`. Les arguments `col`, `pch` et `points` permettent de personnaliser une telle représentation. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Ajouter des groupes sur le graphe est possible mais la procédure est plus complexe :

```
> assoc <- MVA.plot(CCA,points=FALSE,col=couleurs)[2]
```

où `couleurs` est un vecteur à deux valeurs, la première pour les *colonnes* et la seconde pour les *lignes*. S'il l'on souhaite afficher des groupes de colonnes, la première couleur doit être `"white"` ; pour des groupes de lignes la seconde couleur doit être `"white"`.

```
> par(new=TRUE)
> MVA.plot(CCA,points=FALSE,xlim=assoc$xlim,ylim=assoc$ylim,set=nb,fac=facteur)[2]
```

où `nb` vaut 1 pour des groupes de lignes ou 2 pour des groupes de colonnes, et `facteur` est le facteur définissant le groupe de chaque ligne/colonne. Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser une telle représentation. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(CCA,"corr")`[2]. Comme pour le graphe des individus c'est l'AFC contrainte qui est représentée. L'AFC non contrainte n'a pas de sens puisqu'elle ne concerne pas les variables explicatives.

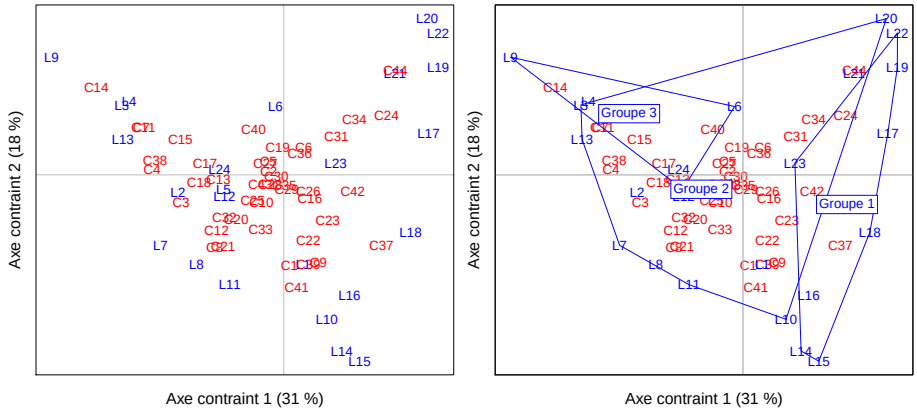
Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

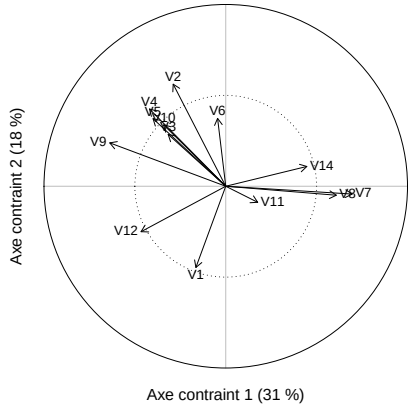
Interprétation

On ne considère que l'AFC contrainte puisque par définition c'est la seule qui permette d'interpréter les résultats en lien avec les variables explicatives.

Le graphe des individus permet (i) d'identifier les associations entre lignes et colonnes du tableau à expliquer (de la même façon qu'en AFC, voir fiche **97**), (ii) d'identifier comment les modalités d'un facteur à effet significatif se répartiessent et (iii) d'identifier des gradients linéaires.

Dans un second temps, le cercle des corrélations permet d'identifier les variables explicatives quantitatives (i) qui différencient d'éventuels groupes et/ou (ii) qui expliquent d'éventuels gradients. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe d'association (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions sur le cercle des corrélations (voir fiche **89**).





107. L'analyse des correspondances discriminante

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]ade4, [2]RVAideMemoire

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse des correspondances discriminante
Analyse discriminante des correspondances

Anglais – *Correspondence discriminant analysis* (*CDA*)
Discriminant correspondence analysis (*DCA*)

Préparation des données

Dans un tableau de contingence ou un tableau de présence-absence, les lignes et les colonnes ont généralement un rôle symétrique (*i.e.* il n'y a pas d'« individus » et de « variables »). Cependant la CDA impose que les entités pour lesquelles les groupes sont définis soient placées en *lignes*, comme dans un tableau classique. On appellera donc les lignes « individus » et les colonnes « variables ».

Remarque : la CDA consiste en fait en deux étapes. La première est de réaliser une AFC sur le tableau de contingence (ou de présence-absence) ; le tableau est bien considéré symétriquement à ce moment là (voir fiche **97**). Dans un second temps, les coordonnées des *lignes* du tableau de contingence sur les axes de l'AFC sont utilisées pour réaliser une LDA (voir fiche **104**), dans laquelle les *colonnes* du tableau de contingence vont servir de variables pour l'interprétation.

Réalisation de l'analyse

Pour réaliser la CDA : `CDA<-discrimin.coa(tableau,facteur,scannf=FALSE,nf=nlevels(facteur)-1)`[1] où `tableau` est le tableau de contingence (ou de présence-absence) et `facteur` le facteur définissant les groupes.

Qualité de l'analyse

Pourcentage d'erreur de classification

Une des fonctions de toute analyse discriminante est de faire de la *prédiction*, *i.e.* de prédire le groupe auquel appartient un individu dont on connaît seulement la valeur pour les variables quantitatives (*i.e.* les colonnes du tableau de contingence). Une manière d'estimer la qualité d'une analyse discriminante est donc de tester à quel point elle permet de classer un individu de groupe inconnu sans erreur. On utilise pour cela une méthode de *validation croisée*, qui va générer plusieurs sous-modèles chacun sur une partie du jeu de données (définie aléatoirement) et prédire le groupe des individus non pris en compte dans le modèle. Pour plus de fiabilité l'ensemble de la procédure peut être répété plusieurs fois (avec à chaque fois un découpage aléatoire du jeu de données).

Pour réaliser la validation croisée : `CDA.cv(tableau,facteur)`[2]. Par défaut la fonction découpe le jeu de données en 7 parties (*7-fold cross-validation*), mais ce chiffre peut être modifié *via* l'argument `k=nb` où `nb` est le nombre de sous-jeux de données à générer. La fonction peut également ajuster ce chiffre automatiquement si au moins un groupe contient moins de 7 individus. Par défaut l'ensemble de la procédure est répété 10 fois (argument `repet`), ce qui au final génère $7 \times 10 = 70$ sous-modèles.

Si l'on souhaite utiliser la CDA dans un but prédictif, stocker le résultat de la fonction `CDA.cv()`[2] dans un objet.

Test(s)

Le test à réaliser repose sur les coordonnées des individus sur les axes de l'AFC intermédiaire. Pour le réaliser : `CDA.test(tableau,facteur)`. Si g est le nombre de groupes, par défaut $g-1$ axes de l'AFC sont retenus pour le test. Ce nombre d'axes peut être modifié grâce à l'argument `ncomp=nb` où `nb` est le nombre d'axes souhaité.

Si un seul axe est retenu (par choix ou parce qu'il n'y a que deux groupes), le test est une ANOVA. Si au moins deux axes sont retenus, le test est une MANOVA (*Multivariate ANalysis Of VAriance*), une extenstion de l'ANOVA au cas multivarié.

Si le facteur a un effet significatif et qu'il y a plus de deux groupes, il est nécessaire de réaliser des comparaisons multiples pour identifier les modalités qui diffèrent. Pour cela : `pairwise.CDA.test(tableau, facteur)` (avec toujours l'argument `ncomp`).

Remarque : si le facteur n'a pas d'effet significatif, l'interprétation des résultats de la CDA n'a pas vraiment de sens et toute tentative de prédiction est inutile.

Représentations graphiques

En CDA on a deux représentations possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Remarque : s'il n'y a que deux groupes, la CDA ne produit qu'un seul axe. Le graphe des individus et le cercle des corrélations se réduisent donc à une seule dimension.

Graphe des individus

Pour tracer le graphe : `MVA.plot(CDA,fac=facteur)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut (s'il y a au moins deux axes), ils peuvent être changés grâce aux arguments `xax` et `yax`.

Dans le cas d'un graphe à deux dimensions, les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser la représentation. Dans le cas d'un graphe à une seule dimension, les arguments `col`, `legend` et `legend.lab` sont intéressants. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(CDA,"corr")`[2].

Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

Le graphe des individus permet d'identifier comment les groupes se structurent, *i.e.* si tout ou partie des groupes se séparent ou au contraire se chevauchent.

Dans un second temps, le cercle des corrélations permet d'identifier les variables (*i.e.* les colonnes du tableau de contingence) qui différencient ces groupes. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe des individus (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions sur le cercle des corrélations (voir fiche **89**).

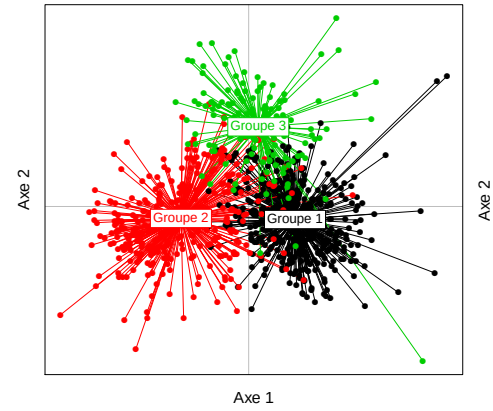
Prédiction

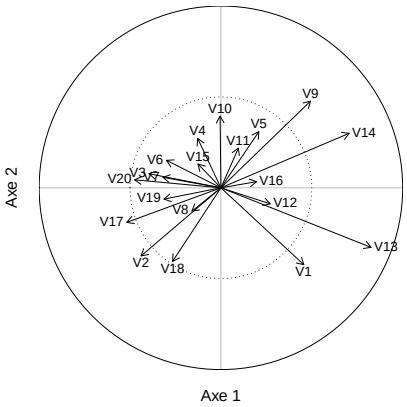
L'un des intérêts des analyses discriminantes est à la fois de comprendre comment les groupes se différencient, mais également de pouvoir prédire le groupe d'un individu pour des valeurs *connues* des variables quantitatives. Prédire un groupe nécessite donc de fixer la valeur de *toutes* les variables quantitatives.

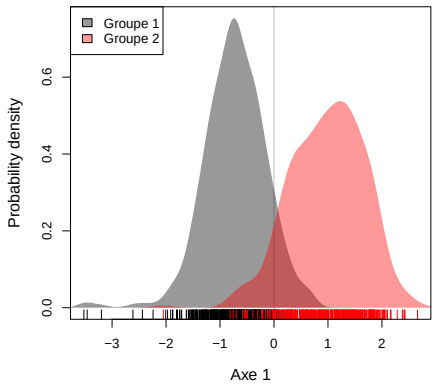
Cette prédiction se fait en trois étapes :

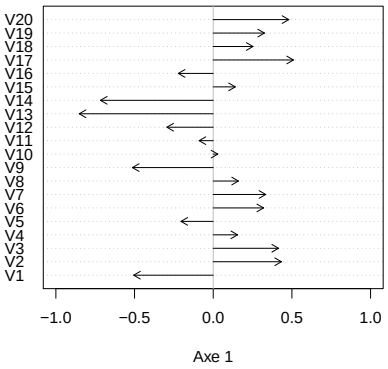
1. Générer une série de sous-modèles à partir du jeu de données initial, par validation croisée.
2. Créer un tableau contenant une colonne par colonne du tableau initial (les noms de colonnes doivent être *strictement identiques* à ceux du jeu de données initial), et remplir chaque ligne en fonction des valeurs pour lesquelles la prédiction doit être faite (il y a donc une prédiction par ligne de ce tableau).
3. Réaliser la prédiction : `predict(CDA.vc,new.tab)` où `CDA.vc` est le résultat de la validation croisée (étape 1) et `new.tab` le tableau des individus à classer. La fonction renvoie pour chaque ligne de `new.tab` le groupe prédit (colonne `Group`) et la probabilité de cette prédiction (colonne `Proba`).

Remarque : tout l'intérêt de générer un grand nombre de sous-modèles au moment de la validation croisée est de pouvoir associer à chaque prédiction une probabilité, car chaque sous-modèle va en fait servir à faire sa propre prédiction, ce qui permet d'estimer la fiabilité de la prédiction « moyenne ».









108. Analyser un ensemble de variables quantitatives

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`vegan`, [2]`RVAideMemoire`, [3]`car`

⚠ L'analyse présentée dans cette fiche est basée sur un modèle. Il est indispensable de maîtriser les notions des fiches **39** à **42**. ⚠

Réponse

Le modèle qui va être utilisé fait l'hypothèse que les matrices de variance-covariance (l'équivalent multivarié de la variance) sont homogènes entre les différentes modalités des variables explicatives qualitatives. Pour le tester : `anova(betadisper(dist(tableau),facteur)`[1]`)` où `tableau` est le tableau *à expliquer* et `facteur` le facteur définissant les groupes. Le modèle est toutefois assez robuste à un non respect (modéré) de cette condition. Si elle n'est pas du tout respectée, une transformation préalable du tableau peut grandement aider à améliorer la situation (voir fiche **88**).

Dans tous les cas, les variables à expliquer doivent être bien moins nombreuses que les individus et elles ne doivent pas être trop corrélées entre elles. Si cette double condition n'est pas respectée, transformer le tableau à expliquer en matrice de distance (voir fiche **100**) et utiliser cette matrice comme réponse dans une analyse dédiée (voir fiche **110**).

Après avoir fait ces vérifications et possiblement transformé le tableau à expliquer, celui-ci doit être converti en matrice. Pour cela : `tableau<-as.matrix(tableau)`.

Modèle utilisé

Le modèle utilisé est un Modèle Linéaire Multivarié (*Multivariate Linear Model* ou MLM), une extension du Modèle Linéaire classique du cadre univarié (voir fiche **76**). À la différence du Modèle Linéaire classique, il y a plusieurs variables à expliquer dans le MLM.

Construction du modèle

Pour créer le modèle : `modele<-lm(formule)`. Voir fiche **40** pour une explication détaillée de la construction d'une formule. Dans cette formule, la réponse est `tableau` (*i.e.* le tableau à expliquer). De manière générale, on peut dire que :

— inclure un facteur permet de tester si la position des individus dans l'espace multidimensionnel des variables à expliquer diffère entre les niveaux de ce facteur.
— inclure une covariable permet de tester s'il existe une relation entre cette covariable et les variables à expliquer.
— inclure une interaction entre deux variables explicatives permet de tester si l'effet de l'une dépend de l'autre (le raisonnement est le même pour une interaction incluant plus de deux termes). Le cas particulier d'une interaction entre une covariable et un facteur permet de tester si la relation entre la covariable et les variables à expliquer est différente selon la modalité du facteur.

Remarque : bien que la gestion des covariables ne pose pas de problème théorique avec les MLMs, il est en pratique difficile de se représenter leur effet dans le cadre multivarié (sauf à l'interpréter séparément pour chaque variable à expliquer). Ce type de modèle se prête par contre très bien à l'analyse de l'effet de facteurs.

Vérification de la validité du modèle

Avant d'aller plus loin, il est indispensable de vérifier que le modèle s'ajuste bien aux données. Cette étape est fondamentale, et ce pour tout modèle, car un test basé sur un modèle mal ajusté n'est simplement pas fiable (ni valide). L'ajustement d'un MLM est cependant plus difficile à vérifier que celui d'un modèle à réponse univariée. Le seul critère réellement observable est la distribution des *résidus* du modèle (*i.e.* les écarts entre les valeurs réellement observées et celles prédites par le modèle). Pour que le

modèle soit validé, les résidus doivent suivre une distribution normale multivariée (une extension de la distribution normale classique).

Pour faire cette vérification : `plotresid(modele)`[2]. L'hypothèse de normalité est acceptée lorsque les points sont à peu près alignés sur une droite. Ces points peuvent ne pas être parfaitement alignés sur la droite, mais s'ils restent à peu près dans l'intervalle de confiance de celle-ci (représenté en pointillés), l'ajustement est considéré comme correct.

Si l'hypothèse de normalité est intenable, l'alternative au MLM est de transformer l'ensemble des variables à expliquer en une matrice de distance (voir fiche **100**) et d'utiliser cette matrice comme réponse dans une analyse dédiée (voir fiche **110**).

Test

L'effet des variables explicatives est testé par une MANOVA (*Multivariate ANalysis Of VAriance*), une extension de l'ANOVA au cas multivarié. Pour réaliser le test : `Manova(modele)`[3]. Le test réalisé est de type II (voir fiche **42** pour une explication détaillée des hypothèses testées).

Si un facteur (ou une interaction entre plusieurs facteurs) a un effet significatif, il est nécessaire de réaliser des comparaisons multiples pour identifier les modalités (ou combinaisons de modalités) qui différent. Voir fiche **43** pour réaliser ces comparaisons.

Si une covariable a une effet significatif, il n'y a pas vraiment d'autre solution pour estimer la direction de son effet que de l'étudier séparément sur chaque variable à expliquer. Cette direction est donnée par le *signe* du paramètre associé. Les valeurs de tous les paramètres des différents modèles (un par variable à expliquer) sont obtenues *via* `summary(modele)`. Elles sont appelées `Estimate` et se trouvent dans les tableaux `Coefficients`. Si le coefficient portant le nom de la covariable est négatif, la valeur de la variable à expliquer correspondante diminue quand la valeur de la covariable augmente ; s'il est positif, la valeur de la variable à expliquer augmente quand la valeur de la covariable augmente.

Prédiction à partir du modèle

L'intérêt des modèles est à la fois de comprendre l'importance des différentes variables explicatives sur les variables à expliquer, mais également de prédire les valeurs que prendraient ces variables à expliquer pour des valeurs *connues* des variables explicatives. Faire une prédiction (une par variable à expliquer) nécessite donc de fixer la valeur de *toutes* les variables explicatives.

Deux méthodes peuvent être utilisées pour la prédiction, les deux étant basées sur la fonction `predict()` :

— donner la valeur (ou les valeurs pour faire plusieurs prédictions) de chacune des variables explicatives directement dans la fonction, sous la forme d'une liste : `predict(modele,newdata=list(variables))`, où `variables` est un enchaînement de `variable1=valeur,variable2=valeur...`

— créer un tableau contenant une colonne par variable explicative (les noms de colonnes doivent être *strictement identiques* aux noms des variables du modèle), et remplir chaque ligne en fonction des valeurs pour lesquelles la prédiction doit être faite (il y a donc une prédiction par ligne de ce tableau). Puis : `predict(modele,newdata=tab.pred)`.

EXEMPLE(S)

Avec un modèle contenant un facteur à deux niveaux (A et B), une covariable variant de 0 à 30, et leur interaction :

```
> modele <- lm(reponse~facteur*covariable)
```

On peut prédire la valeur de chaque variable à expliquer de cette façon :

```
> predict(modele,newdata=list(facteur="A",covariable=10))
```

Ou, pour plusieurs prédictions :

```
> predict(modele,newdata=list(facteur=c("A","B"),covariable=c(10,10)))
```

Ou encore créer un tableau de ce type :

```
> tab.pred
  facteur covariable
1       A         10
2       B         10
```

Puis :

```
> predict(modele,newdata=tab.pred)
```

109. L'analyse de redondance sur matrice de distance

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`vegan`, [2]`RVAideMemoire`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse de redondance sur matrice de distance
Anglais – *Distance-based redundancy analysis* (*db-RDA*)
Canonical analysis of principal coordinates (*CAP*)
Constrained analysis of principal coordinates (*CAP*)

Préparation des données

Il est indispensable de vérifier si la matrice de distance est euclidienne ou non (voir fiche **100**).

Réalisation de l'analyse

Pour réaliser la db-RDA : `dbRDA<-dbrda(formule,data=tab.explicatif)`[1] où `tab.explicatif` est le tableau contenant les variables explicatives. Voir fiche **40** pour une explication détaillée de la construction d'une formule. Dans cette formule, la réponse est `mat.dist` (*i.e.* la matrice de distance). Si celle-ci n'est pas euclidienne, il est nécessaire d'appliquer une correction. Ajouter pour cela l'argument `add=TRUE`.

Capacité explicative globale

La db-RDA consiste en fait en deux étapes :

1. Réaliser une PCoA sur la matrice de distance (voir fiche **101**).
2. Réaliser une RDA (voir fiche **103**) sur les résultats de la PCoA. Le résultat de la RDA consiste en deux ACP (voir fiche **96**) :
   — Une sur la variation (de la matrice de distance) due aux variables explicatives, appelée variation *contrainte*. C'est l'« ACP contrainte ».
   — Une sur la variation non expliquée, appelée variation *résiduelle* ou *non contrainte*. C'est l'« ACP non contrainte ».

On peut estimer la capacité explicative globale de la db-RDA grâce au pourcentage de variance contrainte de l'analyse (*i.e.* de variance de la matrice de distance expliquée par les variables explicatives). Plus ce pourcentage est élevé et plus la variation observée dans la matrice de distance est liée aux variables explicatives. Ce pourcentage est obtenu *via* `MVA.synt(dbRDA)`[2], dans le premier tableau renvoyé par la fonction.

Qualité de l'analyse

Test(s)

L'effet des variables explicatives est testé *via* `MVA.anova(dbRDA)`[2]. Un test F par permutation est réalisé.

Si au moins une variable explicative a un effet significatif, on peut se baser sur les résultats de l'ACP *contrainte* pour l'interprétation. Si aucune variable explicative n'a d'effet significatif, interpréter les résultats de cette ACP n'a pas beaucoup d'intérêt puisqu'aucun effet n'est montré.

En lien avec cette ACP contrainte, on peut réaliser des comparaisons multiples entre modalités d'un facteur (ou combinaisons de modalités d'une interaction entre facteurs) à effet significatif. Pour réaliser le test : `pairwise.factorfit(dbRDA,facteur)`[2] où `facteur` est le facteur d'intérêt.

Synthèse

Si au moins une variable explicative a un effet significatif, on s'intéresse à l'ACP contrainte. Comme pour une ACP classique (voir fiche **96**), on estime la qualité de cette analyse par le pourcentage de variance expliqué par chaque axe. Ces pourcentages sont obtenus *via* `MVA.synt(dbRDA)`[2], dans le deuxième tableau renvoyé par la fonction.

Remarque 1 : il s'agit ici de pourcentages de variance *contrainte*, pas totale comme en ACP classique.

Remarque 2 : les pourcentages de variance sont toujours en ordre décroissant (*i.e.* l'axe 1 explique plus de variance que l'axe 2, qui en explique lui-même plus que l'axe 3...).

Remarque 3 : il n'y a pas de règle absolue sur le nombre d'axes à retenir pour l'interprétation. Il s'agit toujours d'un compromis entre une bonne synthèse de l'information (qui augmente avec le nombre d'axes) et une facilité à interpréter (qui diminue avec le nombre d'axes).

Représentations graphiques

À partir du moment où il y a au moins une variable explicative quantitative, deux représentations sont possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**). S'il n'y a aucune variable explicative quantitative le seul graphe possible est celui des individus.

Graphe des individus

Pour tracer le graphe : `MVA.plot(dbRDA)`[2]. Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Par défaut c'est l'ACP contrainte qui est représentée. Pour représenter l'ACP non contrainte, ajouter l'argument `space=2`.

Pour ajouter des groupes sur le graphe, utiliser l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe de chaque individu. Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser une telle représentation.

Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(RDA,"corr")`[2]. Comme pour le graphe des individus c'est l'ACP contrainte qui est représentée. L'ACP non contrainte n'a pas de sens puisqu'elle ne concerne pas les variables explicatives.

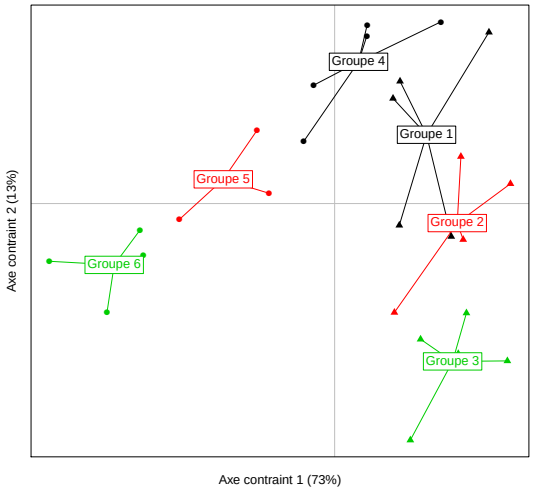
Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

On ne considère que l'ACP contrainte puisque par définition c'est la seule qui permette d'interpréter les résultats en lien avec les variables explicatives.

Le graphe des individus permet d'identifier la structuration des données de la matrice de distance qui est due aux variables explicatives. On y repère comment les modalités d'un facteur à effet significatif se répartissent, ou des gradients linéaires.

Dans un second temps, le cercle des corrélations permet d'identifier les variables explicatives quantitatives qui expliquent d'éventuels gradients. Pour cela, on repère quelles sont les directions pertinentes pour l'interprétation biologique sur le graphe des individus (ce peuvent être des axes ou n'importe quelles diagonales), et on identifie les variables qui corrèlent le plus avec ces directions sur le cercle des corrélations (voir fiche **89**).


Groupe 4
Groupe 1
Groupe 5
Groupe 2
Groupe 6
Groupe 3
Axe contraint 2 (13%)
Axe contraint 1 (73%)

110. Analyser une matrice de distance

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`RVAideMemoire`

Le test à réaliser est non paramétrique et souvent appelé « MANOVA par permutation ». Il est basé sur un modèle (assez analogue au MLM, voir fiche **108**), ce qui permet d'étudier l'effet de plusieurs variables explicatives à la fois (quantitatives et/ou qualitatives), possiblement en interactions. Les *p-value*s sont cependant obtenues par permutation, donc non paramétriquement.

Pour réaliser le test : `adonis.II(formule)`[1]. Voir fiche **40** pour une explication détaillée de la construction d'une formule. Dans cette formule, la réponse est la matrice de distance. Cette matrice n'a pas à être euclidienne.

Si un facteur (ou une interaction entre plusieurs facteurs) a un effet significatif, il est nécessaire de réaliser des comparaisons multiples pour identifier les modalités (ou combinaisons de modalités) qui diffèrent. Pour réaliser ces comparaisons : `pairwise.perm.manova(mat.dist,facteur)`[1] où `mat.dist` est la matrice de distance et `facteur` la facteur d'intérêt.

Si une covariable a un effet significatif, l'interprétation n'est pas si facile. Le plus simple est soit de réaliser une PCoA sur la matrice de distance (voir fiche **101**) puis de corréler ses résultats à la covariable (voir fiche **91**), soit de réaliser une db-RDA (voir fiche **109**). La première approche est préférable si la covariable n'est pas contrôlée, la seconde si elle l'est.

111. L'analyse PLS à deux blocs

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`mixOmics`, [2]`RVAideMemoire`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse PLS à deux blocs
Anglais – *2-block partial least squares* (*2B-PLS*)
2-block projection to latent structures (*2B-PLS*)

Préparation des données

La 2B-PLS fonctionne d'autant mieux que les variables de chaque tableau ont une distribution à peu près normale (au moins symétrique) et qu'elles sont reliées entre elles par des relations à peu près linéaires. Une transformation préalable des données peut grandement aider à améliorer la situation (voir fiche **88**).

Il est également recommandé, la plupart du temps, de standardiser les variables de chaque tableau avant l'analyse (voir fiche **88**). Cela permet de donner le même poids à toutes les variables.

Réalisation de l'analyse

Pour réaliser la 2B-PLS : `PLS2B<-pls(tableau1,tableau2,mode="canonical")`[1] où `tableau1` et `tableau2` sont les deux tableaux. Si l'on souhaite standardiser les variables mais que l'on n'a pas effectué l'opération au préalable, la fonction `pls()`[1] le fait par défaut. Deux paires d'axes sont construits par défaut, pour changer cette valeur ajouter l'argument `ncomp=nb` où `nb` est le nombre de paires d'axes souhaité.

Remarque : on parle de *paires* d'axes en 2B-PLS car l'analyse créé deux ordinations séparées, l'une pour le premier tableau et l'autre pour le second. On a donc un axe 1 pour le premier tableau, et un axe 1 pour le second tableau (et ainsi de suite).

Qualité de l'analyse

Test

La 2B-PLS a pour objectif de synthétiser au mieux une certaine information qui est la *covariance* entre les deux jeux de données. Mais pour que son interprétation soit pertinente, encore faut-il qu'il y ait une covariance (*i.e.* une relation) significative. La significativité de la covariance est testée *via* `cov.test(tableau1,tableau2)`[2]. Un test par permutation est réalisé. Si sa *p-value* est significative, les résultats de la 2B-PLS peuvent être interprétés. Dans le cas contraire l'interprétation n'a pas de sens puisqu'il n'y a pas d'association significative entre les deux tableaux.

Remarque : la covariance est intimement liée à la notion de corrélation (voir fiche **84**). Elle peut donc être négative ou positive, indiquant une relation négative ou positive entre les deux tableaux (ou parties de ces tableaux). Cependant la covariance n'est pas bornée, donc sa valeur absolue n'est pas directement interprétable (d'où l'intérêt de la corrélation, qui est une version standardisée de la covariance).

Synthèse

Si la covariance entre les deux tableaux est significative, on estime la qualité de l'analyse par le pourcentage de covariance expliqué par chaque paire d'axes. Ces pourcentages sont obtenus *via* `MVA.synt(PLS-2B)`[2], dans le premier tableau renvoyé par la fonction.

Remarque 1 : les pourcentages renvoyés par la fonction `MVA.synt()`[2] sont en fait des pourcentages de covariance *au carré*. Peu importe pour l'interprétation : plus le pourcentage est élevé et plus la paire d'axes en question explique une part importante de la covariance entre les jeux de données.

Remarque 2 : les pourcentages de covariance sont toujours en ordre décroissant (*i.e.* la paire d'axes 1 explique plus de covariance que la paire d'axes 2, qui en explique elle-même plus que la paire d'axes 3...).

Remarque 3 : il n'y a pas de règle absolue sur le nombre d'axes à retenir pour l'interprétation. Il s'agit toujours d'un compromis entre une bonne synthèse de l'information des jeux de données (qui augmente avec le nombre d'axes) et une facilité à interpréter (qui diminue avec le nombre d'axes).

Représentations graphiques

En 2B-PLS on a deux représentations possibles *pour chaque tableau* : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Graphe des individus

Pour tracer le graphe du premier tableau : `MVA.plot(PLS2B,space=1)`[2]. Pour le second tableau : `MVA.plot(PLS2B,space=2)`[2].

Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe du premier tableau : `MVA.plot(PLS2B,"corr",space=1)`[2]. Pour le second tableau : `MVA.plot(PLS2B,"corr",space=2)`[2].

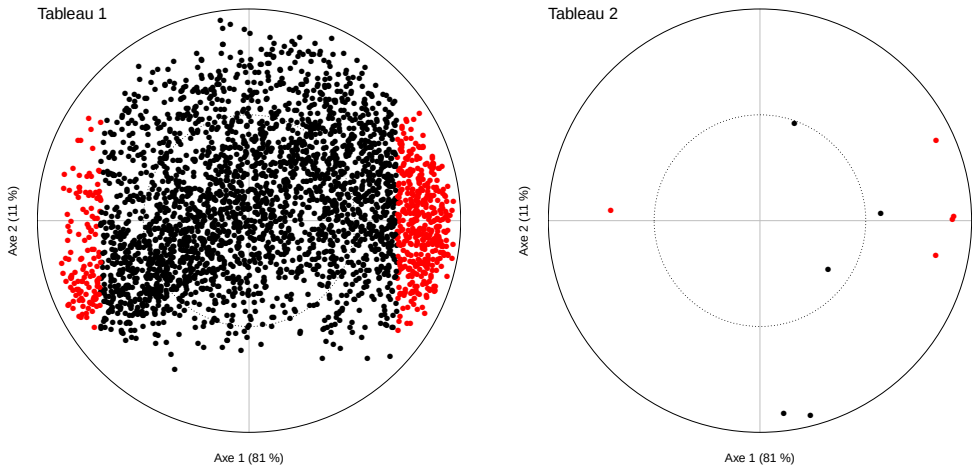
Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

L'objectif de la 2B-PLS est le plus souvent d'identifier les variables de chaque tableau qui sont associées entre elles, mais aussi (et surtout) les associations entre variables de tableaux différents. À ce titre l'interprétation se base essentiellement sur les cercles des corrélations.

Les pourcentages de covariance expliqués indiquent sur quels axes se concentrer pour l'interprétation. On se place dans le cas le plus fréquent, où une seule paire d'axes est suffisante pour expliquer une grande part de la covariance entre les deux tableaux. Dans un tel cas, on repère pour chaque tableau les variables les plus corrélées au premier axe (voir fiche **89**). Ce sont ces variables qui sont les plus associées.

Au sein d'un tableau, l'interprétation est simple : les variables aux extrémités opposées d'un axe sont corrélées négativement (voir fiche **89**). Pour la relation entre variables de tableaux différents, il manque cependant une information essentielle : dans quel sens sont corrélées les paires d'axes. Ces corrélations sont données par la fonction `MVA.synt()`[2], dans le second tableau qu'elle renvoie. Si la corrélation pour la première paire d'axes est *positive*, les variables dirigées vers la *gauche* de l'axe 1 pour le premier tableau sont positivement reliées aux variables dirigées vers la *gauche* de l'axe 1 pour le second tabeau. Si la corrélation pour la première paire d'axes est *négative*, les variables dirigées vers la *gauche* de l'axe 1 pour le premier tableau sont positivement reliées aux variables dirigées vers la *droite* de l'axe 1 pour le second tabeau.



112. L'analyse procustéenne

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`vegan`, [2]`RVAideMemoire`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse procustéenne
Analyse Procuste
Rotation procustéenne
Rotation Procuste

Anglais – *Procrustes analysis* (*Proc*)
Procrustes rotation (*Proc*)
Procrustean superimposition analysis (*Proc*)

Préparation des données

L'analyse procustéenne est intéressante quand elle est réalisée sur deux tableaux à deux colonnes chacun (voir fiche **114** pour le cas où au moins un tableau contient plus de deux colonnes). Ce peuvent être deux tableaux à deux variables chacun, ou deux axes d'une précédente ordination (voir fiche **90** pour récupérer les coordonnées des individus sur ces axes). Une des applications courantes de l'analyse procustéenne est ainsi de comparer les résultats de deux ordinations réalisées sur une matrice de distance, en particulier deux nMDS (voir fiche **102**).

Réalisation de l'analyse

Pour réaliser l'analyse procustéenne : `Proc<-procrustes(tableau1,tableau2)`[1] où `tableau1` et `tableau2` sont les deux tableaux. Le premier est appelé X et le second Y.

Remarque : l'un et/ou l'autre des deux tableaux peut être directement une ordination si elle a été réalisée avec le package `vegan`.

Qualité de l'analyse

On peut à la fois estimer la « corrélation » entre les deux tableaux et la tester *via* `protest(tableau1,tableau2)`[1]. La fonction renvoie ce qui s'apparente à un coefficient de corrélation (`Correlation in a symmetric Procrustes rotation`) et le résultat d'un test non paramétrique appelé PROTEST (basé sur le coefficient de corrélation). Une *p-value* significative indique qu'il y a une concordance significative entre les deux tableaux.

Représentation graphique

La seule représentation possible en analyse procustéenne est le *graphe des individus*, qui montre la position des individus à la fois pour le tableau X et le tableau Y, sur un *plan factoriel* composé de deux axes. Ces deux axes correspondent aux variables du tableau X.

Pour tracer le graphe : `MVA.plot(Proc,"pairs")`[2]. Chaque individu est représenté par une flèche. Celle-ci démarre aux coordonnées de l'individu pour le tableau X, et s'arrête aux coordonnées de ce même individu pour le tableau Y (celui-ci ayant subi rotation, translation et/ou homothétie pour s'ajuster au mieux au tableau X).

Voir `?MVA.pairplot` pour bien d'autres options graphiques.

Interprétation

Plus les flèches du graphe des individus sont courtes et plus cela indique une concordance forte entre les deux tableaux (*i.e.* les individus sont à peu près positionnés au même endroit quel que soit le tableau). À l'inverse, plus les flèches sont longues et partent dans tous les sens, et plus cela indique une

concordance faible entre les deux tableaux. Il y aura toujours une cohérence entre le graphe et le résultat du test statistique.

La longueur des flèches n'est jamais identique pour tous les individus ; on peut ainsi identifier lesquels sont « semblables » dans les deux tableaux ou au contraire très différents.

113. L'analyse de co-inertie

PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE

[1]RVAideMemoire, [2]ade4

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse de co-inertie
Anglais – *Co-inertia analysis* (*CoIA* ou *CIA*)

Préparation des données

La CIA est une méthode très générale permettant de coupler deux tableaux. Ceux-ci peuvent être de types différents (un tableau de variables classique et un tableau de contingence par exemple), ou contenir des variables de nature différente (quantitatives et/ou qualitatives).

La première étape de l'analyse est de réaliser une ordination séparée sur les deux tableaux. Selon la nature des variables, on utilise le plus souvent l'ACP (voir fiche **96**), l'AFC (voir fiche **97**), l'ACM (voir fiche **98**) ou l'analyse mixte (voir fiche **99**). les deux analyses n'ont pas à être identiques.

Remarque 1 : la CIA exige que les poids des *lignes* des deux tableaux soient égaux entre ces tableaux. Avec l'ACP, l'ACM et l'analyse mixte c'est bien le cas puisque, sauf volonté contraire explicite, toutes les lignes ont le même poids. Avec l'AFC la situation est différente, car dans cette analyse le poids d'une ligne est fonction de la somme des valeurs de cette ligne. Pour coupler une AFC à une autre analyse dans une CIA, il est donc nécessaire de recréer cette autre analyse en spécifiant comme poids ceux utilisés par l'AFC. Cette opération peut se faire automatiquement *via* `analyse.bis<-ord.rw(analyse,AFC)`[1] où `analyse` est l'ordination à coupler avec l'AFC, et `AFC` l'AFC. L'ordination `analyse.bis` est celle qui sera utilisée dans la CIA.

Remarque 2 : pour la raison expliquée dans la remarque précédente, il n'est pas possible de coupler deux AFC dans une CIA.

Réalisation de l'analyse

Pour réaliser la CIA : `CIA<-coinertia(analyse1,analyse2,scannf=FALSE,nf=10)`[2] où `analyse1` et `analyse2` sont les ordinations réalisées séparément sur chaque tableau.

Remarque : la fonction `coinertia()`[2] n'accepte que des ordinations réalisées grâce à des fonctions du package `ade4`. Cela est bien le cas pour l'ACM (voir fiche **98**) et l'analyse mixte (voir fiche **99**), mais pas pour l'ACP (voir fiche **96**) et l'AFC (voir fiche **97**) qui sont réalisées grâce au package `vegan`. Pour convertir le résultat d'une de ces ordinations en objet utilisable par la fonction `coinertia()`[2] : `analyse.bis<-to.dudi(analyse)`[1] où `analyse` est le nom de l'ACP ou l'AFC. Si la fonction `ord.rw()`[1] est utilisée en amont de la CIA, l'ordination qu'elle renvoie est directement compatible avec `coinertia()`[2].

EXEMPLE(S)

On veut coupler dans une CIA une ACP (nommée `ACP`) et une AFC (nommée `AFC`), les deux ayant été réalisées grâce au package `vegan` comme expliqué dans les fiches **96** et **97**. On commence par recréer l'ACP en donnant aux lignes les poids utilisés par l'AFC :

```
> ACP.bis <- ord.rw(ACP,AFC)[1]
```

L'objet `ACP.bis` est directement compatible avec la fonction `coinertia()`[2]. Cependant ce n'est pas le cas de l'objet `AFC`. On transforme celui-ci au bont format :

```
> AFC.bis <- to.dudi(AFC)[1]
```

La CIA peut maintenant être réalisée :

```
> CIA <- coinertia(ACP.bis,AFC.bis,scannf=FALSE,nf=10)[2]
```

Qualité de l'analyse

Test

La CIA a pour objectif de synthétiser au mieux une certaine information qui est la *co-inertie* entre les deux tableaux, *i.e.* la partie de leurs informations respectives qui varie en commun. Mais pour que l'interprétation de la CIA soit pertinente, encore faut-il qu'il y ait une co-inertie (*i.e.* une relation) significative. La significativité de la co-inertie est testée par un test par permutation basé sur le coefficient RV (aucun rapport avec l'auteur de cet aide-mémoire). Ce test est réalisé *via* `randtest(CIA)`[2]. La valeur du coefficient est appelée `Observation` et vaut entre 0 (aucune co-inertie) et 1 (co-inertie parfaite). Une *p-value* significative indique qu'il y a une concordance significative entre les deux tableaux, et donc que les résultats de la CIA peuvent être interprétés. Dans le cas contraire l'interprétation n'a pas de sens puisqu'il n'y a pas de relation significative entre les deux tableaux.

Remarque : si l'une des deux ordinations couplées par la CIA est une AFC, il est nécessaire d'ajouter l'argument `fixed=nb` où `nb` est le numéro de l'AFC (1 si c'est le premier argument de `coinertia()`[2], 2 si c'est le deuxième).

Synthèse

Si la co-inertie entre les deux tableaux est significative, on estime la qualité de l'analyse par le pourcentage de co-inertie expliqué par chaque paire d'axes. Ces pourcentages sont obtenus *via* `MVA.synt(CIA)`[1], dans le premier tableau renvoyé par la fonction.

Remarque 1 : on parle de *paires* d'axes en CIA car l'analyse créé deux ordinations séparées (sur la base de celles fournies), l'une pour le premier tableau et l'autre pour le second. On a donc un axe 1 pour le premier tableau, et un axe 1 pour le second tableau (et ainsi de suite).

Remarque 2 : les pourcentages de co-inertie sont toujours en ordre décroissant (*i.e.* la paire d'axes 1 explique plus de co-inertie que la paire d'axes 2, qui en explique elle-même plus que la paire d'axes 3...).

Remarque 3 : il n'y a pas de règle absolue sur le nombre d'axes à retenir pour l'interprétation. Il s'agit toujours d'un compromis entre une bonne synthèse de l'information des jeux de données (qui augmente avec le nombre d'axes) et une facilité à interpréter (qui diminue avec le nombre d'axes).

Représentations graphiques

En CIA on a deux représentations possibles *pour chaque tableau* : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Remarque : seul le graphe des individus est disponible pour une ACM (voie fiche **98**).

Graphe des individus

Pour tracer le graphe du premier tableau : `MVA.plot(CIA,space=1)`[1]. Pour le second tableau : `MVA.plot(CIA,space=2)`[1].

Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`.

Pour ajouter des groupes sur le graphe, utiliser l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe de chaque individu. Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser une telle représentation.

Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe du premier tableau : `MVA.plot(CIA,"corr",space=1)`[2]. Pour le second tableau : `MVA.plot(CIA,"corr",space=2)`[2].

Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

L'objectif de la CIA est d'identifier les associations entre variables de tableaux différents.

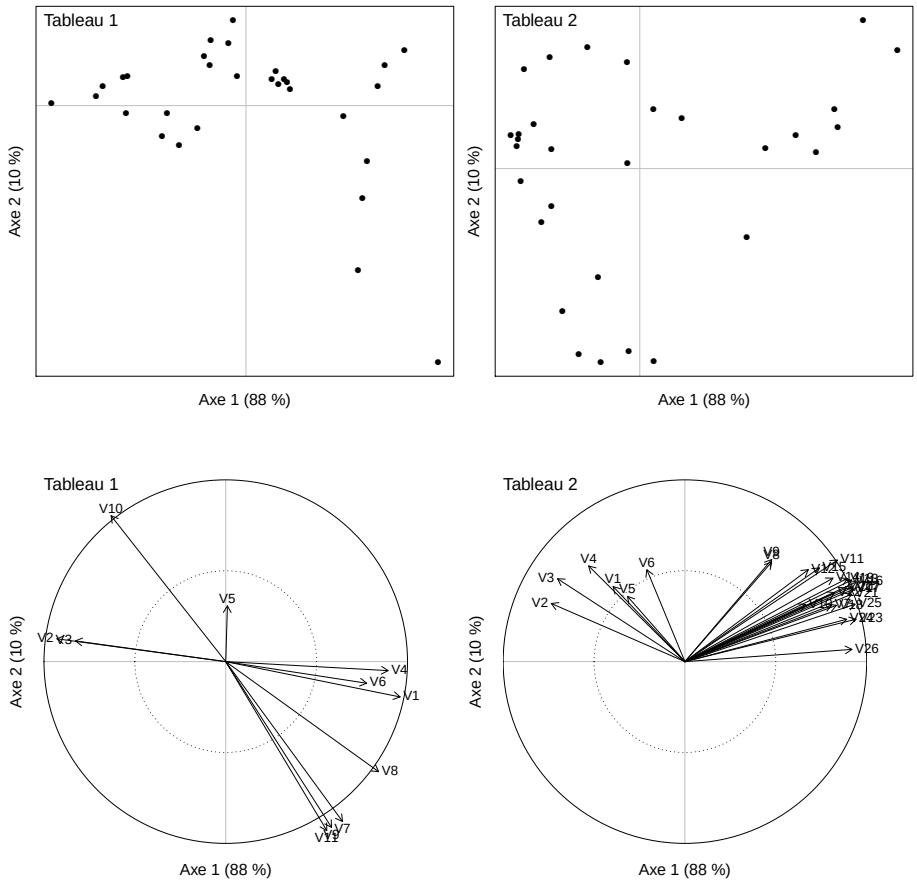
Pour les variables quantitatives, l'interprétation est basée sur la comparaison des deux cercles des corrélations : deux flèches pointant dans la même direction ou dans des directions diamétralement opposées indiquent deux variables corrélées (voir fiche **89**). Le sens de la corrélation est indiqué par le

signe de la corrélation entre les axes d'une même paire. Ces corrélations entre axes sont données par la fonction `MVA.synt()`[1], dans le second tableau qu'elle renvoie. Si la corrélation pour une paire d'axes est *positive*, les variables dirigées vers la *gauche* de cet axe pour le premier tableau sont positivement reliées aux variables dirigées vers la *gauche* du même axe pour le second tabeau. Si la corrélation pour la paire d'axes est *négative*, les variables dirigées vers la *gauche* de cet axe pour le premier tableau sont positivement reliées aux variables dirigées vers la *droite* du même axe pour le second tabeau.

Pour les variables qualitatives l'interprétation est similaire à l'ACM : plus deux modalités sont proches et plus elles sont associées (voir fiche **98**). Là encore, le sens de l'association dépend de la corrélation entre paires d'axes.

Les relations entre variables quantitatives et qualitatives s'interprètent comme en analyse mixte : plus le barycentre d'une modalité est situé vers l'extrémité d'une flèche (quand on le projette perpendiculairement à cette flèche) et plus la valeur moyenne des individus de cette modalité est extrême pour la variable quantitative représentée par la fèche (le sens de la relation étant donné par la corrélation entre paires d'axes).

Remarque : les relations entre variables d'un même tableau s'interprètent comme en ACP (voir fiche **96**), ACM (voir fiche **98**) et analyse mixte (voir fiche **99**).



114. L'analyse de de co-inertie procustéenne

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]ade4, [2]RVAideMemoire

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse de co-inertie procustéenne
Anglais – *Procustean co-inertia analysis* (*PCIA*)

Préparation des données

La PCIA généralise l'analyse procustéenne (voir fiche **112**) au cas où au moins un tableau contient plus de deux colonnes. Les deux tableaux peuvent être des tableaux de variables, ou des axes d'une précédente ordination (voir fiche **90** pour récupérer les coordonnées des individus sur ces axes). Toutes les variables doivent être quantitatives, et une standardisation de ces variables est souvent bénéfique (voir fiche **88**).

Réalisation de l'analyse

Pour réaliser la PCIA : `PCIA<-procuste(tableau1,tableau2)`[1] où `tableau1` et `tableau2` sont les deux tableaux. Le premier est appelé X et le second Y.

Qualité de l'analyse

On peut à la fois estimer la « concordance » entre les deux tableaux et la tester *via* `randtest(PCIA)`[1]. La fonction renvoie une statistique appelée m^2 (`Observation`) qui varie entre 0 (concordance parfaite) et 1 (aucune concordance), et le résultat d'un test non paramétrique appelé PROTEST (basé sur le m^2). Une *p-value* significative indique qu'il y a une concordance significative entre les deux tableaux.

Représentations graphiques

En PCIA on a deux représentations possibles : le *graphe des individus*, qui montre la position des individus à la fois pour le tableau X et le tableau Y sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Remarque : en PCIA chaque individu a deux coordonnées dans le même plan factoriel, l'une pour le tableau X et l'autre pour le tableau Y. Il y a donc deux nuages de points dans le même espace.

Graphe des individus

Pour tracer le graphe : `MVA.plot(PCIA,"pairs")`[2]. Chaque individu est représenté par une flèche. Celle-ci démarre aux coordonnées de l'individu pour le tableau X, et s'arrête aux coordonnées de ce même individu pour le tableau Y.

Voir `?MVA.pairplot` pour bien d'autres options graphiques.

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(PCIA,"corr")`[2]. Par défaut les variables des deux tableaux sont représentées. Pour ne représenter que les variables du tableau X ajouter l'argument `set=1`, pour les variables du tableau Y `set=2`.

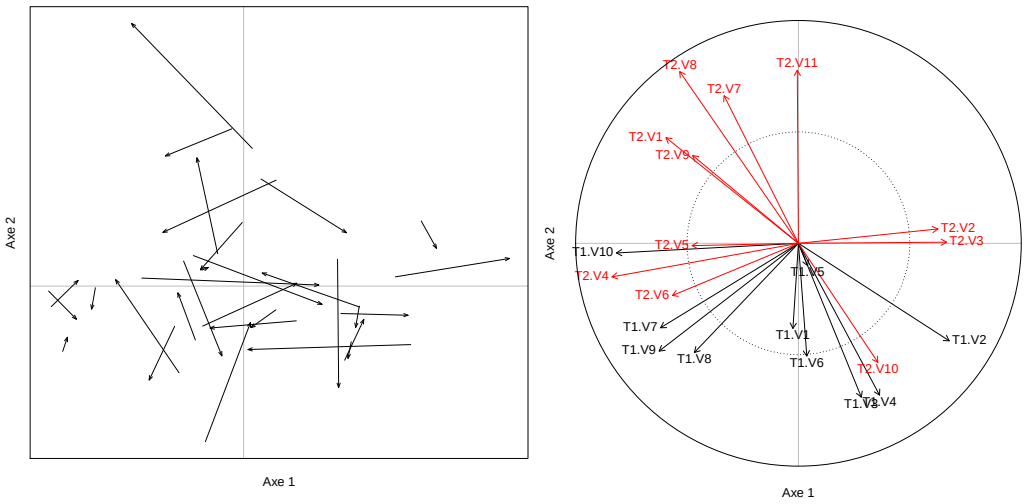
Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Pour représenter différemment les variables des différents tableaux, ajouter l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe (*i.e.* le tableau) de chaque variable. Les arguments `col`, `pch` et `lwd` permettent de personnaliser une telle représentation. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

Plus les flèches du graphe des individus sont courtes et plus cela indique une concordance forte entre les deux tableaux (*i.e.* les individus sont à peu près positionnés au même endroit quel que soit le tableau). À l'inverse, plus les flèches sont longues et partent dans tous les sens, et plus cela indique une concordance faible entre les deux tableaux. Il y aura toujours une cohérence entre le graphe et le résultat du test statistique.

Remarque : la longueur des flèches n'est jamais identique pour tous les individus ; on peut ainsi identifier lesquels sont « semblables » dans les deux tableaux ou au contraire très différents.

À la condition que la concordance entre les deux tableaux soit significative, le cercle des corrélations permet d'identifier les variables qui sont associées entre ces tableaux. Son interprétation est identique à un cercle des corrélations classique (voir fiche **89**).



115. L'analyse canonique des corrélations régularisée généralisée

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`RVAideMemoire`, [2]`mixOmics`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse canonique des corrélations régularisée généralisée
Anglais – *Regularized generalized canonical correlation analysis* (*RGCCA*)

Préparation des données

La RGCCA exige que toutes les variables soient quantitatives. Ce peuvent être des variables mesurées telles quelles et/ou des axes d'une précédente ordination (voir fiche **90** pour récupérer les coordonnées des individus dans ce dernier cas). Les différents tableaux (appelés « blocs ») n'ont pas à contenir le même nombre de variables.

L'analyse fonctionne d'autant mieux que les variables ont une distribution à peu près normale (au moins symétrique) et que dans chaque tableau elles sont reliées entre elles par des relations linéaires. Une transformation préalable des données peut grandement aider à améliorer la situation (voir fiche **88**). Une standardisation des variables de chaque tableau est également souvent bénéfique, car elle permet de donner le même poids à toutes les variables (voir fiche **88**).

Remarque : on peut intégrer un facteur dans l'analyse en le transformant en *variables indicatrices*, *via* `var.ind<-dummy(facteur)`[1] où `facteur` est un vecteur définissant le groupe de chaque individu. Le tableau `var.ind` est ensuite traité comme les autres, excepté pour la transformation qui n'est jamais nécessaire. Cette façon de procéder permet d'utiliser la RGCCA comme analyse discriminante.

Tous les tableaux doivent être rassemblés dans une liste. Pour cela : `blocs<-list(bloc1=tableau1, bloc2=tableau2...)` où `tableau1`, `tableau 2` et `...` sont les différents tableaux (qui seront nommés « bloc1 », « bloc2 »… dans l'analyse). Les compartiments de la liste doivent obligatoirement avoir un nom.

Pour finir, il est nécessaire de définir les relations interblocs à analyser (*i.e.* les tableaux à relier). Rien n'oblige en effet à ce que l'analyse cherche l'information commune à toutes les paires de tableaux deux-à-deux. Au contraire, il vaut mieux définir les relations qui sont les plus pertinentes vis-à-vis des questions biologiques. Ces relations sont définies dans une matrice symétrique (*design matrix*) à autant de lignes et de colonnes que de blocs, remplie de 0 (pas de relation) et de 1 (relation). La diagonale est nécessairement constituée uniquement de 0.

— EXEMPLE(S) —

Les blocs 1 et 2 sont à relier chacun au bloc 3, mais pas entre eux :

```
> matrice <- matrix(c(0,0,1,0,0,1,1,1,0),nrow=3)
> matrice
     [,1] [,2] [,3]
[1,]    0    0    1
[2,]    0    0    1
[3,]    1    1    0
```

Réalisation de l'analyse

Pour réaliser la RGCCA : `RGCCA<-wrapper.rgcca(blocs,C=matrice,tau="optimal",ncomp=nb)`[2] où `matrice` est la *design matrix* et `nb` le nombre d'axes à construire.

Remarque 1 : si l'argument `C` n'est pas fourni toutes les paires de tableaux deux-à-deux sont considérées.

Remarque 2 : par défaut la fonction standardise tous les jeux de données, ce qui est souvent intéressant car cela permet de donner le même poids à toutes les variables (voir fiche **88**). Pour ne pas standardiser, ajouter l'argument `scale=FALSE`.

Remarque 3 : la RGCCA créé autant d'ordinations séparées qu'il y a de tableaux. On a donc un axe 1 pour chaque bloc, un axe 2 pour chaque bloc et ainsi de suite. Si l'argument `ncomp` n'est pas précisé un seul axe est construit par bloc.

Remarque 4 : la RGCCA a pour objectif de maximiser une certaine information qui se situe sur un continuum à deux extrêmes : (i) la *variance totale intrabloc* expliquée et (ii) la *corrélation* entre les axes de même niveau (*i.e.* les axes 1, les axes 2…). L'argument `tau` permet de préciser où l'on doit se situer sur ce continuum. On peut ainsi choisir de maximiser plutôt l'information de chaque bloc expliquée par l'analyse (extrême « variance », *via* `tau=rep(1,nb)` où `nb` est le nombre de blocs), plutôt la corrélation entre les axes de même niveau (extrême « corrélation », *via* `tau=rep(0,nb)`) ou de trouver le meilleur compromis entre ces deux extrêmes (`tau="optimal"`, intéressant par défaut).

Qualité de l'analyse

On estime la qualité de l'analyse par le pourcentage de variance totale intrabloc expliqué par chaque axe (de chaque bloc), et par la corrélation entre les axes qui doivent être reliés selon la *design matrix*. Toutes ces valeurs sont obtenues *via* `MVA.synt(RGCCA)`[1], dans les différents tableaux renvoyés par la fonction (un par critère et par bloc).

Remarque : contrairement à la plupart des analyses multivariées, les pourcentages de variance expliqués ne sont pas nécessairement en ordre décroissant (même si c'est le cas le plus souvent). Les corrélations entre axes sont elles toujours de moins en moins fortes.

Représentations graphiques

En RGCCA on a deux représentations possibles *pour chaque bloc* : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et le *cercle des corrélations* (voir fiche **89**) qui permet d'interpréter la répartition des individus.

Graphe des individus

Pour tracer le graphe du premier bloc : `MVA.plot(RGCCA,space=1)`[1]. Pour le bloc suivant : `MVA.plot(RGCCA,space=2)`[1]. Et ainsi de suite.

Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

Cercle des corrélations

Pour tracer le graphe du premier bloc : `MVA.plot(RGCCA,"corr",space=1)`[1]. Pour le bloc suivant : `MVA.plot(RGCCA,"corr",space=2)`[1]. Et ainsi de suite.

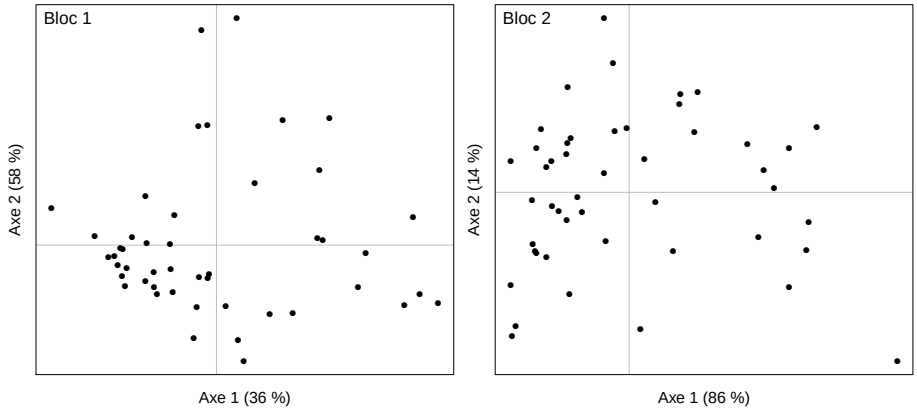
Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

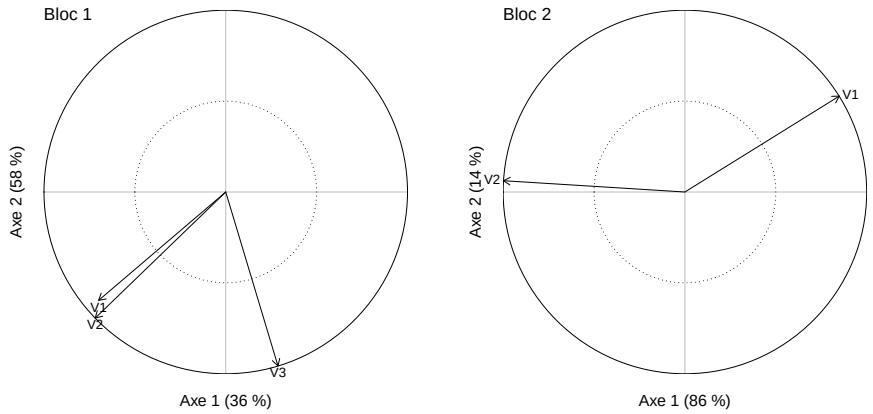
Interprétation

L'objectif de la RGCCA est le plus souvent d'identifier les variables de chaque tableau qui sont associées entre elles, mais aussi (et surtout) les associations entre variables de tableaux différents. À ce titre l'interprétation se base essentiellement sur les cercles des corrélations.

Les corrélations interblocs (*i.e.* entre axes de même niveau) indiquent sur quels axes se concentrer pour l'interprétation. On ne retient que les axes très corrélés entre eux, car c'est cette corrélation (*i.e.* cette information commune à plusieurs tableaux) qui est la cible de l'analyse et donc la base de l'interprétation. Si par exemple la corrélation est forte pour les axes 1, on repère pour chaque tableau les variables les plus corrélées à ce premier axe (voir fiche **89**). Ce sont ces variables qui sont les plus associées.

Au sein d'un tableau, l'interprétation est simple : les variables aux extrémités opposées d'un axe sont corrélées négativement (voir fiche **89**). Pour la relation entre variables de deux tableaux différents, si la corrélation pour la première paire d'axes est *positive*, les variables dirigées vers la *gauche* de l'axe 1 pour le premier tableau sont positivement reliées aux variables dirigées vers la *gauche* de l'axe 1 pour le second tabeau. Si la corrélation pour la première paire d'axes est *négative*, les variables dirigées vers la *gauche* de l'axe 1 pour le premier tableau sont positivement reliées aux variables dirigées vers la *droite* de l'axe 1 pour le second tabeau.


Bloc 1
Axe 2 (58 %)
Axe 1 (36 %)
Bloc 2
Axe 2 (14 %)
Axe 1 (86 %)


Bloc 1
Axe 2 (58 %)
V1
V2
V3
Axe 1 (36 %)
Bloc 2
Axe 2 (14 %)
V1
V2
Axe 1 (86 %)

116. L'analyse procustéenne généralisée

— PACKAGE(S) ADDITIONNEL(S) UTILISÉ(S) DANS CETTE FICHE —

[1]`FactoMineR`, [2]`RVAideMemoire`

SYNONYMES, TRADUCTIONS ET ABRÉVIATIONS

Français – Analyse procustéenne généralisée
Anglais – *Generalized Procrustes analysis* (*GPA*)
Generalized Procrustes superimposition

Préparation des données

La GPA exige que toutes les variables soient quantitatives. Ce peuvent être des variables mesurées telles quelles et/ou des axes d'une précédente ordination (voir fiche **90** pour récupérer les coordonnées des individus dans ce dernier cas). Les différents tableaux n'ont pas à contenir le même nombre de variables.

L'analyse fonctionne d'autant mieux que les variables ont une distribution à peu près normale (au moins symétrique) et que dans chaque tableau elles sont reliées entre elles par des relations linéaires. Une transformation préalable des données peut grandement aider à améliorer la situation (voir fiche **88**).

Pour finir, tous les tableaux doivent être rassemblés dans un unique tableau. Pour cela : `tableau<-data.frame(tableau1,tableau2...)` où `tableau1`, `tableau 2` et `...` sont les différents tableaux (de gauche à droite dans le tableau final).

Réalisation de l'analyse

Pour réaliser la GPA : `GPA<-GPA(tableau,group=colonnes,plot=FALSE)`[1] où `colonnes` est un vecteur donnant le nombre de colonnes de chaque tableau (de la gauche vers la droite), afin que la fonction puisse les séparer en interne. Pour donner un nom à chaque tableau (ce qui peut faciliter l'interprétation graphique ultérieure), ajouter l'argument `name.group=noms` où `noms` est un vecteur contenant le nom de chaque tableau (de la gauche vers la droite). Par défaut ils sont nommés `group.1`, `group.2`...

Remarque : par défaut la fonction standardise tous les jeux de données, ce qui est souvent intéressant car cela permet de donner le même poids à toutes les variables (voir fiche **88**). Pour ne pas standardiser, ajouter l'argument `scale=FALSE`.

Qualité de l'analyse

Test

L'objectif de la GPA est de trouver une *configuration consensus*, *i.e.* une « moyenne » des ordinations respectives de chaque tableau, après que celles-ci aient subi les étapes de translation, rotation et homothétie qui les ajustent au mieux les unes par rapport aux autres. C'est souvent cette configuration consensus qui est l'objectif de l'analyse, afin d'être interprétée ou utilisée dans une analyse ultérieure (voir fiche **90** pour en récupérer les coordonnées).

Remarque : la première étape de la GPA est en fait de réaliser une ACP sur chaque tableau, d'où l'obligation que les variables soient toutes quantitatives (voir fiche **96**). Ce sont ces ACP qui sont ensuite ajustées les unes sur les autres.

On peut tester si la configuration consensus est significative, *i.e.* s'il y a réellement concordance entre les différents tableaux. Pour réaliser le test : `GPA.test(tableau,group=colonnes)`[2]. Le temps de calcul nécessaire pour ce test est relativement long.

Synthèse

Si la concordance entre les différents tableaux est significative, un certain nombre d'indicateurs permettent d'estimer la qualité de l'analyse. La manière dont la GPA identifie la configuration consensus est de capter un maximum de la *variance totale* de l'ensemble des tableaux. On s'intéresse donc au

pourcentage de variance totale globalement capté, et plus précisément pour chaque axe de cette ordination consensus. Ces pourcentages sont obtenus *via* `MVA.synt(GPA)`[2], dans les deux premiers tableaux renvoyés par la fonction. Le troisième tableau renvoyé diffère du deuxième car il donne le pourcentage de *variance consensus* expliqué par chaque axe. Traditionnellement on utilise plutôt les pourcentages de variance totale.

Une autre manière d'estimer la qualité de l'analyse est d'identifier si tous les tableaux sont aussi bien représentés par la configuration consensus. Pour cela on examine la *variance résiduelle, i.e.* celle qui n'est pas captée par la configuration consensus. Si la synthèse est efficace pour tous les tableaux, cette variance résiduelle devrait se répartir équitablement entre eux. Si un ou quelques tableaux ont un pourcentage de variance résiduelle bien plus élevé que les autres tableaux, c'est que la configuration consensus n'est pas à même de bien les représenter (cependant il est important de prendre en compte le pourcentage de variance consensus global, car si celui-ci est très élevé on peut considérer que les tableaux sont tous plutôt bien représentés, même si certains le sont moins bien que d'autres). La répartition de la variance résiduelle est fournie dans le dernier tableau renvoyé par la fonction `MVA.synt()`[2].

Représentations graphiques

En GPA on a deux représentations graphiques possibles : le *graphe des individus* qui montre la position de ces individus sur un *plan factoriel* composé de deux axes, et éventuellement le *cercle des corrélations* (voir fiche **89**) si l'interprétation en termes de variables originales est l'objectif de l'analyse.

Graphe des individus

Pour tracer le graphe : `MVA.plot(GPA)`[2]. C'est la configuration consensus qui est représentée.

Les axes 1 (horizontal) et 2 (vertical) sont représentés par défaut, ils peuvent être changés grâce aux arguments `xax` et `yax`. Pour ajouter des groupes sur le graphe, utiliser l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe de chaque individu. Les arguments `col`, `pch`, `fac.lab`, `contours`, `stars` et `barycenters` permettent de personnaliser une telle représentation. Voir `?MVA.scoreplot` pour bien d'autres d'options graphiques.

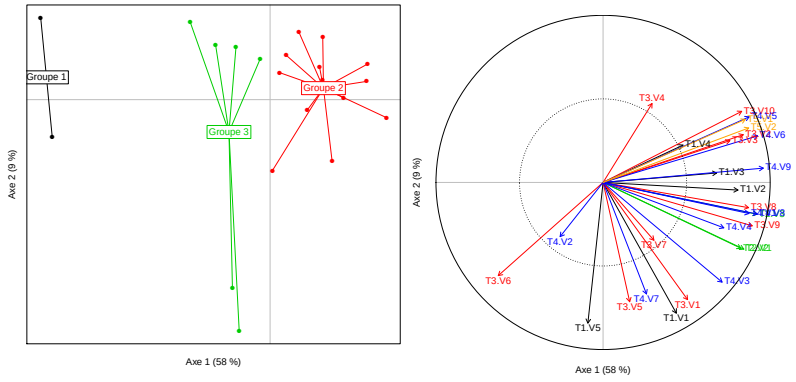
Pour représenter les individus dans toutes les ordinations : `plot(GPA)`. Les points d'un même individu sont reliés au point de l'individu dans la configuration consensus. On peut ainsi identifier les individus qui sont « semblables » dans les différents tableaux, ou au contraire très différents (donc pour lesquels le consensus est probablement moins pertinent).

Cercle des corrélations

Pour tracer le graphe : `MVA.plot(GPA,"corr")`[2]. Pour supprimer les flèches, ajouter l'argument `arrows=FALSE`. Pour représenter différemment les variables des différents tableaux, ajouter l'argument `fac=facteur` où `facteur` est le facteur définissant le groupe (*i.e.* le tableau) de chaque variable. Les arguments `col`, `pch` et `lwd` permettent de personnaliser une telle représentation. Voir `?MVA.corplot` pour bien d'autres d'options graphiques.

Interprétation

Si l'interprétation de la configuration consensus est l'objectif de l'analyse, celle-ci est identique à l'interprétation d'une ACP (voir fiche **96**).



Bibliographie et ouvrages/documents/liens recommandés

Livres

Crawley M.J. (2007) The R Book. John Wiley & Sons, inc.

Dagnelie P. (2006a) Statistique théorique et appliquée. 1. Statistique descriptive et base de l'inférence statistique. 3ème édition. De Boeck.

Dagnelie P. (2006b) Statistique théorique et appliquée. 2. Inférence statistique à une et à deux dimensions. 3ème édition. De Boeck.

Legendre P. and **Legendre L.** (2012) Numerical ecology. 3rd edition. Elsevier.

Borcard D., **Gillet F.** and **Legendre P.** (2012) Numerical ecology with R. Springer.

Millot G. (2008) Comprendre et réaliser les tests statistiques à l'aide de R. 2ème édition. De Boeck.

Documents en ligne

Bates D. (2010) lme4 :Mixed-effects modeling with R. [*http://lme4.r-forge.r-project.org*]

Champely S. (2005) Introduction à l'analyse multivariée (factorielle) sous R. [*http://pierre.jeandenand.free.fr/Cours/Statistiques%20avec%20Rgui.pdf*]

Dagnelie P. (2003) Principes d'expérimentation : planification des expériences et analyse de leurs résultats. Les presses agronomiques de Gembloux. [*http://www.dagnelie.be*]

Fox J. (2002) Cox proportional-hazards regression for survival data. [*http://cran.r-project.org*, rubrique *Contributed* : *Web Appendix to the book "An R and S-PLUS Companion to Applied Regression"*]

Genolini C. (2002) R, bonnes pratiques. [*http://cran.r-project.org*, rubrique *Contributed*]

Paradis E. (2002) R pour les débutants. [*http://cran.r-project.org*, rubrique *Contributed*]

Poinsot D. (2004) Statistiques pour statophobes. [*http://perso.univ-rennes1.fr/denis.poinsot*]

Poinsot D. (2005) R pour les statophobes. [*http://perso.univ-rennes1.fr/denis.poinsot*]

Les nombreux cours en ligne de **D. Chessel**, **A.B. Dufour**, **J.R. Lobry** et **M. Royer** : *http://pbil.univ-lyon1.fr/R/enseignement.html*

Les cours en ligne de **M-L. Delignette-Muller** : *http://www2.vet-lyon.fr/ens/biostat/accueil.html*

Un site très complet et très bien fait sur les analyses multivariées en écologie des communautés : *https://sites.google.com/site/mb3gustame*

Le site de R : *http://www.R-project.org*.

Groupes d'utilisateurs

Le forum du groupe des utilisateurs du logiciel R : *http://forums.cirad.fr/logiciel-R/index.php*

Semin-R, un autre groupe d'utilisateurs de R : *http://rug.mnhn.fr/semin-r*