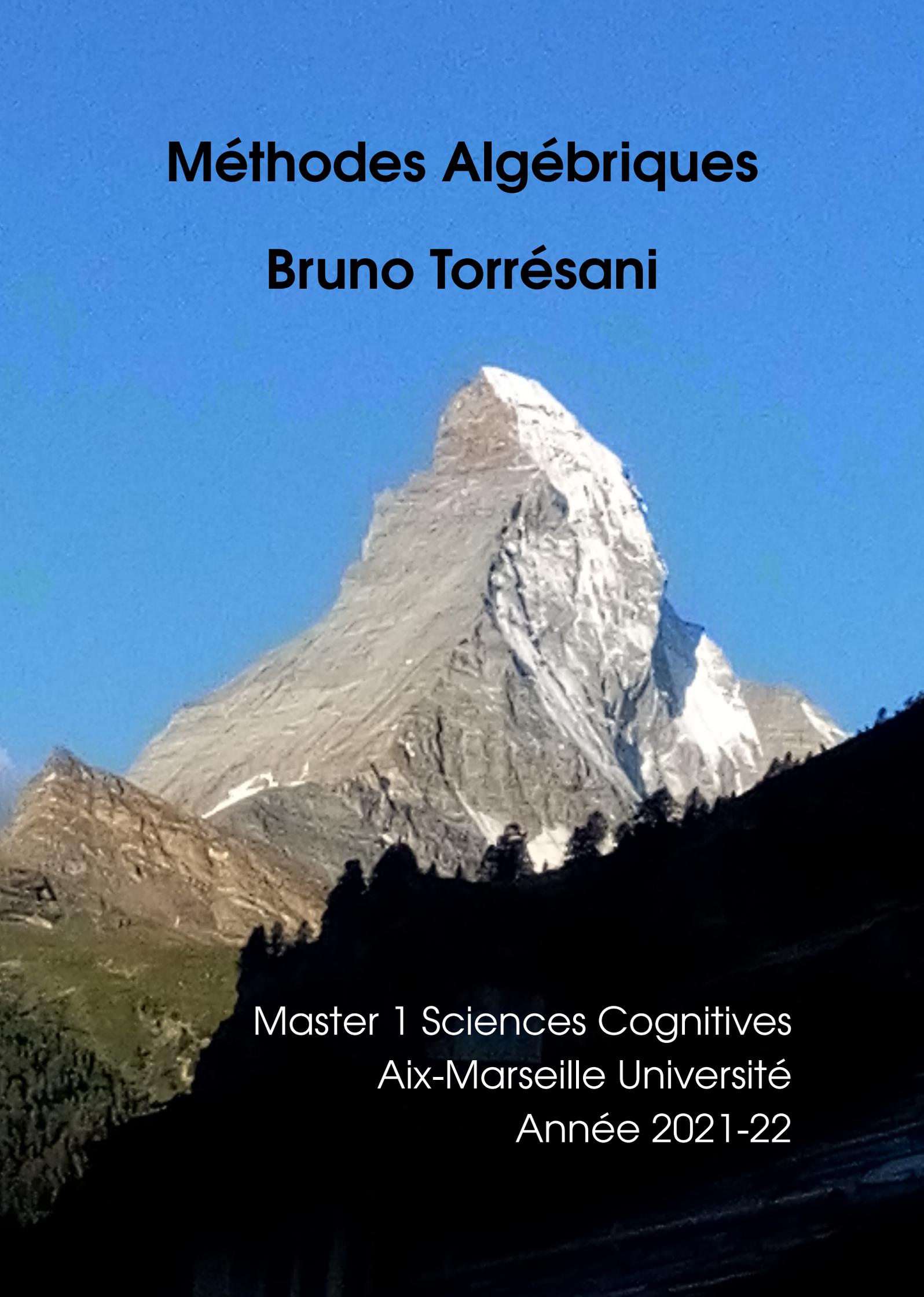


# Méthodes Algébriques

**Bruno Torrèsani**



Master 1 Sciences Cognitives  
Aix-Marseille Université  
Année 2021-22

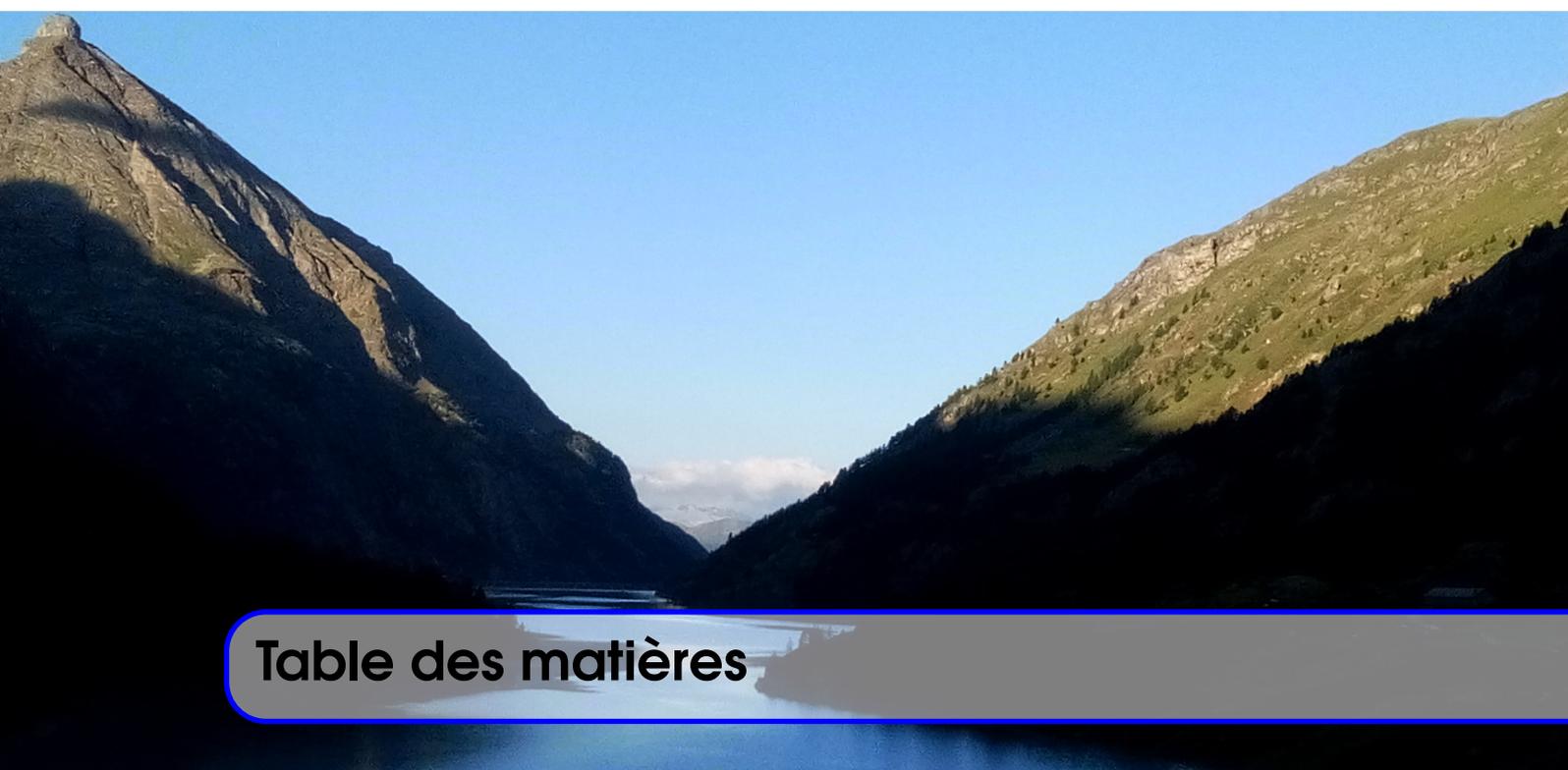
Copyright © 2021 Bruno Torrèsani

Publié par Bruno Torrèsani  
Institut de Mathématiques de Marseille  
Aix-Marseille Université

<https://www.i2m.univ-amu.fr/perso/bruno.torresani>

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*9 avril 2022*



# Table des matières

I	Cours	
<b>1</b>	<b>Éléments d'algèbre linéaire</b>	<b>15</b>
1.1	Tableaux, nuages de points dans un espace de grande dimension,... et géométrie vectorielle	18
1.2	Espace vectoriel, vecteur, dimension, base	19
1.2.1	Premières définitions	19
1.2.2	Base d'un espace vectoriel	21
1.3	Sous-espace vectoriel	23
1.4	Davantage d'outils : espace Euclidien	26
1.4.1	Norme, produit scalaire	26
1.4.2	Déterminant	27
1.4.3	Orthogonalité	28
1.4.4	Espaces Hermitiens	30
1.5	Exemples de vecteurs "structurés" : signaux	31
1.5.1	Bases d'espaces de signaux	32
1.5.2	Projection sur des sous-espaces de signaux	35
1.5.3	Parenthèse : spectrogramme	36
1.6	Résumé	38
<b>2</b>	<b>Transformations linéaires, calcul matriciel</b>	<b>39</b>
2.1	Matrices, calcul matriciel	39
2.1.1	Addition, transposition	39
2.1.2	Produit d'une matrice carrée par un vecteur	40
2.1.3	Produit matriciel : produit de matrices carrées, inversion	43
2.1.4	Produit de matrices rectangulaires	47

<b>2.2</b>	<b>Matrices, familles de vecteurs, changement de base et projection orthogonale</b>	<b>49</b>
2.2.1	Matrice d'une famille de vecteurs, matrice d'une base	49
2.2.2	Matrices des coordonnées d'un vecteur dans une base	50
2.2.3	Matrices et projection orthogonale	53
<b>2.3</b>	<b>Résumé</b>	<b>54</b>
<b>3</b>	<b>Réduction de matrices et analyse multivariée</b>	<b>57</b>
<b>3.1</b>	<b>Généralités</b>	<b>57</b>
3.1.1	Tableaux de données, variables, individus	57
3.1.2	Changements de base dans l'espace des variables et dans l'espace des individus	58
3.1.3	Mesures de position : moyenne	58
3.1.4	Mesures de variabilité : moment d'ordre deux, variance, covariance	59
<b>3.2</b>	<b>Réduction des matrices</b>	<b>62</b>
3.2.1	Diagonalisation des matrices carrées	63
3.2.2	Quid des matrices rectangulaires ?	64
<b>3.3</b>	<b>Analyse en composantes principales</b>	<b>65</b>
3.3.1	Décomposition en valeurs singulières	65
3.3.2	Pourcentage de variance expliquée	66
3.3.3	Représentation des individus : <i>scores</i>	66
3.3.4	Représentation des variables : <i>loadings</i>	67
3.3.5	Un premier exemple : analyse chimique d'échantillons de vins en Italie	67
3.3.6	Un second exemple : photos d'identité	71
3.3.7	Un troisième exemple : signaux EEG	72
<b>3.4</b>	<b>Résumé</b>	<b>75</b>
<b>4</b>	<b>Optimisation, analyse discriminante</b>	<b>77</b>
<b>4.1</b>	<b>Rappels d'analyse</b>	<b>77</b>
4.1.1	Fonctions	77
4.1.2	Dérivation, gradient	80
4.1.3	Dérivées d'ordre supérieur	83
4.1.4	Calcul différentiel pour les nuls	84
<b>4.2</b>	<b>Optimisation</b>	<b>84</b>
4.2.1	Optimisation d'une fonction de plusieurs variables	84
4.2.2	Un exemple d'application : neuroimagerie cérébrale	86
<b>4.3</b>	<b>Analyse discriminante</b>	<b>88</b>
4.3.1	Position du problème	88
4.3.2	Discriminant de Fisher	89
4.3.3	Optimisation du critère de Fisher	90
4.3.4	Une session typique d'analyse discriminante linéaire	92

## II

## Exercices

<b>5</b>	<b>Exercices sur le chapitre 1</b>	<b>97</b>
<b>5.1</b>	<b>Vecteurs</b>	<b>97</b>
<b>5.2</b>	<b>Sous-espaces vectoriels, projection orthogonale</b>	<b>98</b>
<b>5.3</b>	<b>Calcul matriciel</b>	<b>98</b>
<b>5.4</b>	<b>Changements de base, projection orthogonale</b>	<b>99</b>

<b>6</b>	<b>Exercices sur le chapitre 3</b>	<b>101</b>
6.1	Généralités	101
6.2	Diagonalisation	101
6.3	Analyse en composantes principales	102
6.4	Régression linéaire	102
6.5	Analyse discriminante de Fisher	103

### III

## Annexes

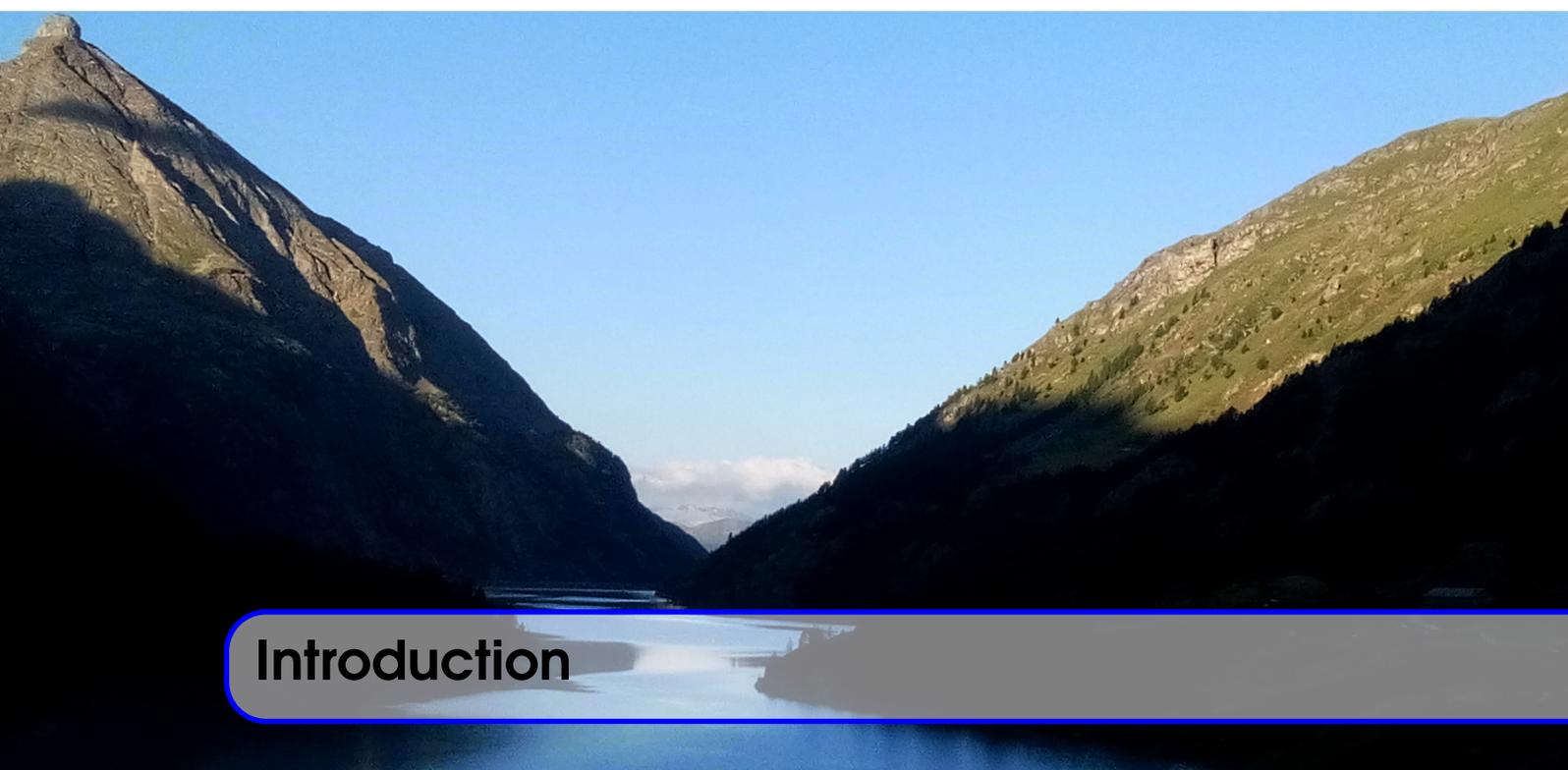
<b>7</b>	<b>Quelques éléments sur R</b>	<b>107</b>
7.1	Introduction	107
7.2	Utiliser R comme une calculatrice scientifique	107
7.3	L'environnement sous R	109
7.3.1	Répertoire de travail	109
7.3.2	Espace de travail	109
7.4	Variables, vecteurs, matrices, data.frames	109
7.4.1	Mode des objets R	109
7.4.2	Listes	110
7.4.3	Vecteurs	110
7.4.4	Matrices, tableaux	112
7.4.5	data.frames	115
7.5	Graphiques	116
7.6	Programmation, scripts et fonctions	117
7.6.1	Structures de contrôle	117
7.6.2	Utilisation de scripts	119
7.6.3	Utilisation de fonctions	119
7.7	Installer et charger des packages	120
7.8	Pour en savoir plus	120

### IV

## Compléments

<b>Bibliographie</b>	<b>123</b>
<b>Index</b>	<b>125</b>





## Introduction

L'objectif de ce cours est de revisiter et approfondir des éléments d'algèbre linéaire déjà abordés par les étudiants dans leur cursus antérieur, et les mettre en situation dans un contexte de modélisation et analyse de tableaux de données. Il s'agit notamment de donner aux étudiants les bases nécessaires pour comprendre et utiliser des méthodes simples d'analyse multivariée.

Comme on le verra, la majorité des notions d'algèbre linéaire sont difficilement dissociables de notions empruntant davantage à la géométrie. On mettra donc naturellement l'accent à chaque fois que cela sera possible sur la géométrie vectorielle. Vers la fin du cours, on s'intéressera également à des problèmes d'optimisation, qui nous amèneront aussi à voir ou revisiter des notions d'analyse (notamment la dérivation et la notion de gradient).

L'algèbre linéaire repose sur deux opérations simples, l'addition et la multiplication, opérations bien connues lorsque l'on les applique à des nombres, et que l'on généralise ensuite à des objets de plus en plus complexes (vecteurs, matrices, tenseurs,...). Nous allons voir deux notions fondamentales de l'algèbre linéaire. D'une part, la notion d'*espace vectoriel*. Un espace vectoriel est un ensemble d'objets (appelés *vecteurs*) sur lesquels on peut définir l'addition et la multiplication par un nombre (appelé *scalaire*). D'autre part, la notion de *base d'un espace vectoriel* : une base de l'espace vectoriel est un ensemble spécifique d'éléments de cet espace vectoriel, tel que l'idée est que tout autre vecteur de l'espace peut s'écrire comme somme pondérée (on parle de *combinaison linéaire*) des vecteurs de la base. C'est une notion importante, que nous utiliserons fortement.

Il existe de nombreux ouvrages introductifs et avancés consacrés à l'algèbre linéaire. On peut notamment citer le livre de J. Grifone [1], ou le livre de G. Strang [4], qui est un texte de référence dans un grand nombre d'universités américaines. Cependant, peu d'entre eux repartent des bases comme nous allons le faire ici, où nous nous attacherons à faire le lien avec la géométrie à chaque fois que cela sera possible <sup>1</sup>.

Les principaux sujets qui seront abordés dans le cours sont :

1. Éléments de base : espaces vectoriels, norme produit scalaire, orthogonalité ; projection orthogonale ; dimension et bases ; liens avec la géométrie vectorielle.
2. Matrices, opérations sur les matrices, représentation matricielle des opérations de base. Systèmes linéaires, et leur formulation matricielle.
3. Diagonalisation des matrices carrées ; décomposition en valeurs singulières ; application à l'analyse multivariée (réduction de dimension, analyse exploratoire) ;

---

1. Un point de départ peut être la page Wikipedia consacrée aux vecteurs [5], qui décrit simplement le point de vue géométrique et le point de vue algébrique dans le cas du plan et de l'espace.

4. Optimisation de formes quadratiques ; application à des problèmes d'analyse de données (régression linéaire multiple, classification,...)
5. Modèles de données structurées (signaux, graphes,...), bases pour des espaces de signaux ou données définies sur des graphes.
6. Études de cas sur des données en lien avec les autres cours du master (analyse de signaux de parole, analyse de signaux EEG et interfaces cerveau machine, neuroimagerie électromagnétique,...)

Ces aspects seront illustrés sur des travaux pratiques informatiques, basés sur l'utilisation du logiciel R, dans l'environnement RStudio.

## Quelques exemples de mise en situation

Avant d'entrer dans le vif du sujet, voici quelques exemples, correspondant à différentes situations, qui motivent le recours à une modélisation mathématique plus poussée.

### Analyse multivariée

L'analyse multivariée recouvre un ensemble de méthodes visant à synthétiser l'information issue de plusieurs variables, et en particulier les liens entre ces variables.

Pour prendre un exemple simple, supposons que l'on ait mesuré conjointement 3 variables, et qu'on dispose de 5000 mesures. On peut par exemple imaginer trois capteurs EEG (électro-encéphalographiques), fournissant 5000 échantillons temporels.

On peut représenter ces mesures sous la forme d'un nuage de 5000 points dans l'espace 3D, voir la FIGURE 1. Différentes visualisations sont nécessaires pour appréhender la forme de ce nuage, et dégager les directions les plus pertinentes. En multipliant les visualisations, on peut notamment détecter que le nuage est relativement "plat" (donc que la variabilité est plus faible dans la direction perpendiculaire à ce plateau), et les données pourraient alors être bien décrites, ou approximées, en se limitant à un plan de l'espace.

Ceci étant la visualisation ne fait pas tout, et pour aller plus loin, il faudra recourir à des techniques mathématiques pour estimer ces directions particulières, caractériser la forme du nuage et en déduire les informations pertinentes. On pourra alors caractériser le plan évoqué ci-dessus, déterminer son équation,...

Si maintenant les données sont de dimension plus élevée (imaginons 64 capteurs EEG), il sera difficile de trouver les bonnes visualisations, et il deviendra obligatoire de recourir à des techniques automatiques permettant d'extraire l'information pertinente. De la même manière que plus haut, on peut imaginer des données comme des points dans un espace de dimension 64 (difficile à visualiser).

La décomposition en valeurs singulières qu'on verra au chapitre 1 permet de déterminer les directions les plus pertinentes dans cet espace de dimension 64, et de leur accorder une mesure d'importance. On représente dans la FIGURE 2 les mesures d'importance de ces 64 directions triées par ordre décroissant (calculées sur un jeu de données EEG réelles). On peut en particulier voir que seules les premières sont réellement significatives.

Si on décide d'ignorer celles qui ne sont pas considérées significatives, ceci revient à se limiter à une partie de l'espace de dimension 64 (on parlera de *sous-espace*). Ici, il n'est pas envisageable de penser aux visualisations graphiques, les mathématiques permettent de donner une caractérisation de ce sous-espace, exploitable pratiquement.

Le domaine des mathématiques qui s'intéresse à ces notions d'espace, sous-espace, dimension, réduction de dimension,... et bien d'autres choses est la géométrie vectorielle, qui fait l'objet du chapitre 1.

### Signaux

La notion de signal est intimement liée à la notion de transmission d'information. Pour ne prendre que quelques exemples, le signal de parole convoie une information transmise par un locuteur, des signaux acoustiques plus généraux nous informent sur notre environnement (bruit de fond, chants d'oiseau, bruit de voitures en approche,...), les dauphins et autres mammifères marins communiquent en utilisant des signaux SONAR, les signaux GSM (ondes électromagnétiques) permettent des communications à longue distances...

Vu comme des données, les signaux ont généralement une forme particulière. On peut voir un signal (numérique) comme une ensemble de mesures d'une variable, les mesures étant naturellement ordonnées. On peut évoquer par exemple

- des résultats de mesures dépendant du temps (courant électrique, son, onde électromagnétique...)
- des spectres donnant la décomposition d'une grandeur physique sur une échelle d'énergie, ou toute autre grandeur se ramenant à une énergie (fréquence, longueur d'onde, etc.)

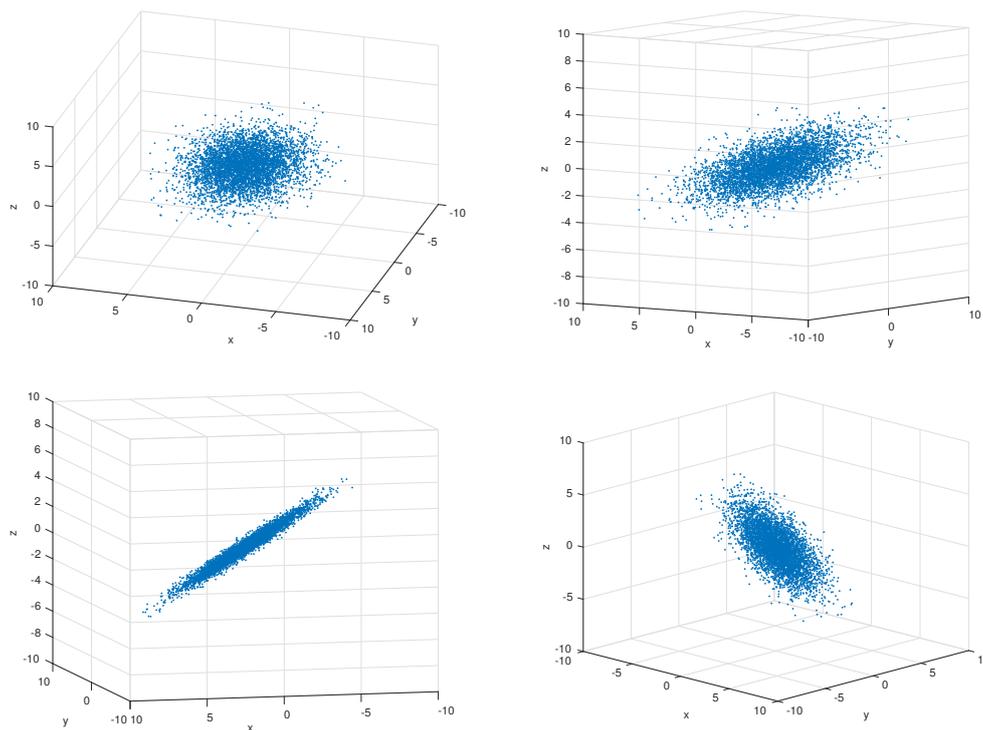


FIGURE 1 – Quatre vues différentes d'un nuage de points

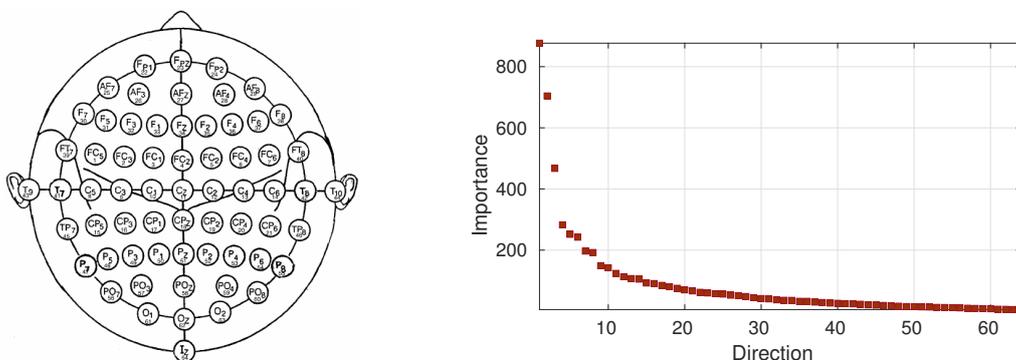


FIGURE 2 – Jeu de données EEG. Gauche : position des capteurs sur le scalp (64 capteurs). Droite : importance des 64 dimensions (par ordre décroissant) dans la décomposition en valeurs singulières d'un jeu de données correspondant (64 capteurs et 240 mesures).

— ...

Le traitement du signal inclut une grande variété de tâches, incluant l'analyse et l'extraction d'information, la transformation, la restauration (dé-bruitage), la reconnaissance, le codage, ... et bien d'autres.

On représente en FIGURE 3 quatre tracés de signal de parole. Il s'agit du même segment de phrase prononcé par quatre locuteurs différents. Même si ces graphes présentent des similarités, il n'est pas facile de voir que les quatre personnes ont dit la même chose.

On représente en FIGURE 4 une autre représentation de ces quatre signaux. La similarité est cette fois plus grande que dans la FIGURE 3, même si les quatre images ne sont pas identiques. C'est ce type de représentation qui est utilisée par la majorité des systèmes de reconnaissance vocale.

Quel rapport y a-t-il avec la discussion précédente? En fait, le passage des graphes des signaux aux

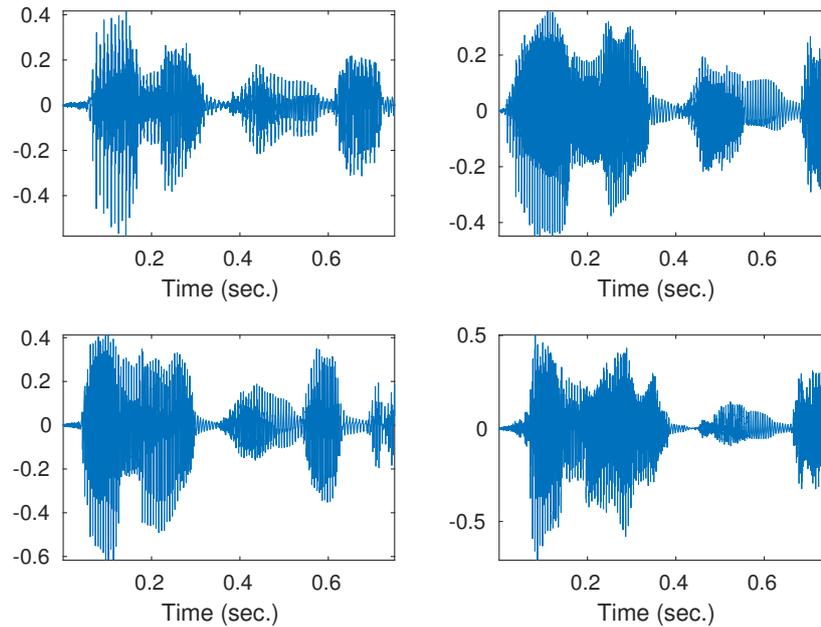


FIGURE 3 – Signal de parole : graphes de quatre signaux acoustiques émis par quatre locuteurs prononçant la même phrase.

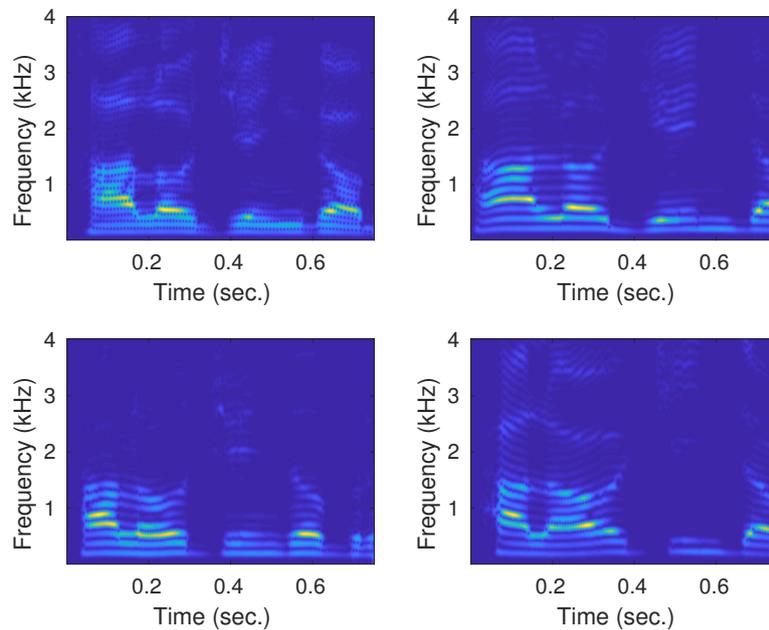


FIGURE 4 – Signal de parole : spectrogrammes des signaux de la FIGURE 3.

images est très similaire mathématiquement aux réductions de dimension évoquées plus haut, il s'agit de se limiter à un sous-espace de l'espace des données dans lequel est concentrée l'information pertinente. Ceci peut être fait de façon adaptative, comme le fait l'approche basée sur la décomposition en valeurs singulières évoquée plus haut, ou de façon fixe, en choisissant un sous-espace dont on sait à l'avance qu'il a de grandes

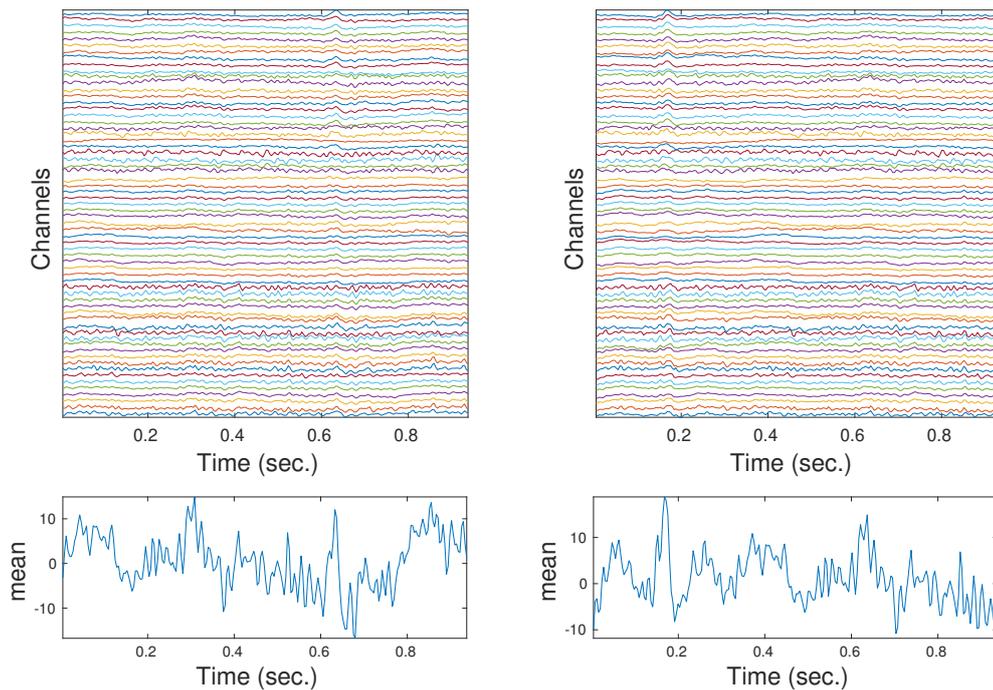


FIGURE 5 – Un exemple de signaux multi-capteurs (haut), et la moyenne sur les capteurs (bas)

chances de contenir l'information pertinente, comme c'est le cas ici.

Une représentation particulièrement utile en traitement du signal est la représentation de Fourier, aussi appelée représentation spectrale (les images de spectrogrammes de la FIGURE 4 sont des versions modifiées de représentations fréquentielles), qui décompose les signaux en sommes de sinusoides. On verra comment ces méthodes peuvent être vues comme des instances des méthodes algébriques développées au Chapitre 1.

### Signaux multi-capteurs

Dans certaines situations, on n'enregistre pas un seul signal, mais plusieurs signaux en parallèle. C'est le cas du son, qui est généralement enregistré en stéréo (deux capteurs), voire sur davantage de canaux (6 canaux dans le système *5.1 surround* couramment utilisé dans le *home cinema*). C'est aussi le cas de nombreux signaux physiologiques (EEG et MEG par exemple, enregistrés sur quelques dizaines à centaines de capteurs).

La FIGURE 5 représente en haut les tracés des signaux de 64 capteurs EEG, dans deux situations. Un évènement est censé être absent dans ceux de gauche, et présent dans ceux de droite. Le moins qu'on puisse dire est que ça ne saute pas aux yeux. Pour simplifier on peut calculer la moyenne des 64 signaux, les deux moyennes sont représentées dans les deux tracés du bas. Là encore, les moyennes sont différentes, mais il est difficile d'identifier une caractéristique qui serait présente à droite mais pas à gauche. La moyenne a beaucoup trop simplifié les choses, et gommé des caractéristiques pertinentes.

Pour aller plus loin, on doit recourir à des techniques plus élaborées, qui font appel à la fois au traitement du signal et à l'analyse multivariée. La problématique est principalement de résumer l'information contenue dans les données, sans trop la simplifier. Le tracé de droite dans la Figure 2 peut être utilisé à cet effet, dans la mesure où il donne des indications sur le nombre de dimensions importantes parmi les 64 dimensions de l'espace des capteurs.

### Localisation de sources en imagerie cérébrale

Il existe maintenant de nombreux dispositifs expérimentaux permettant de représenter ou *imager* l'activité cérébrale (EEG, MEG, IRM,...). Si on fait abstraction des techniques invasives, ces dispositifs fournissent des mesures extérieures sur des capteurs localisés à l'extérieur du crâne. La reconstruction de l'activité cérébrale

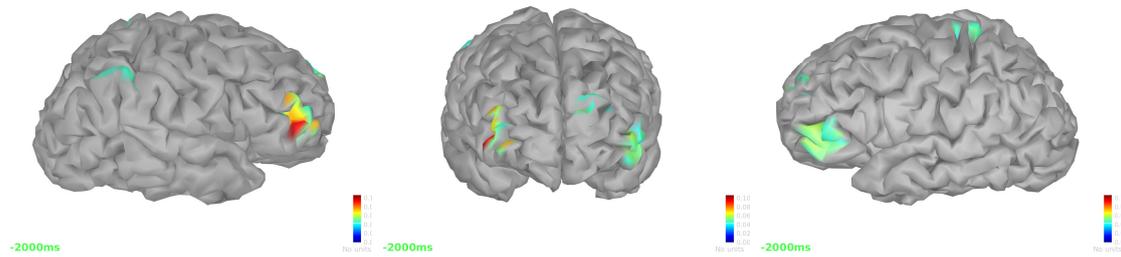


FIGURE 6 – Reconstruction de sources à partir de données MEG : ondes lentes du sommeil profond

à partir de ces mesures est un problème difficile. La modélisation de la physique de ces dispositifs permet de modéliser la propagation de l'activité cérébrale (par exemple une densité de courant électrique sur le cortex) vers les capteurs (potentiels électriques pour l'EEG, champs magnétiques pour la MEG,...). Il s'agit de ce que l'on nomme le problème direct, qui est donc assez bien compris et résolu.

La reconstruction de l'activité cérébrale à partir de mesures extérieures (problème inverse, ou localisation de sources) est par contre bien plus difficile, et constitue encore aujourd'hui un problème largement ouvert. Ceci constitue un exemple de système linéaire, très mal déterminé en l'occurrence, pour la résolution duquel il existe de nombreux algorithmes, généralement peu satisfaisants.

Un exemple d'une telle reconstruction se trouve en Figure 6, il faut noter qu'il est excessivement difficile de quantifier la précision d'une telle reconstruction, en l'absence de *vérité terrain*, qui demanderait des mesures invasives très précises.

### Connectivité, graphes

Comme on l'a mentionné plus haut, on peut voir un signal comme une suite ordonnée (par exemple dans le temps) de mesure d'une variable, ou de plusieurs variables dans le cas multicapteur. Cet ordonnancement donne une structure particulière aux signaux, qui donne lieu à des techniques spécifiques.

Dans certaines situations, cet ordonnancement est remplacé par des relations entre certaines mesures, que l'on représente graphiquement par des liens. On obtient ainsi des graphes, tels que ceux qui sont représentés dans la FIGURE ci-dessous..

Un graphe est constitué de sommets et d'arêtes, qui connectent certains sommets. Ils sont très utilisés dans différents domaines applicatifs. En particulier, l'identification de graphes de connectivité cérébrale est un défi majeur en neurosciences actuellement.

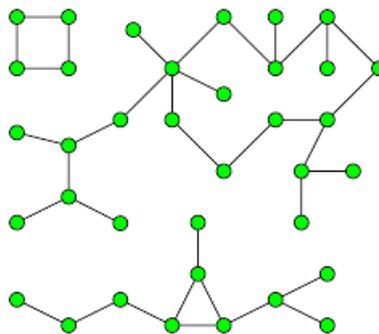


FIGURE 7 – Quelques exemples de graphes

<b>1</b>	<b>Éléments d'algèbre linéaire</b> . . . . .	<b>15</b>
1.1	Tableaux, nuages de points dans un espace de grande dimension, . . . et géométrie vectorielle	
1.2	Espace vectoriel, vecteur, dimension, base	
1.3	Sous-espace vectoriel	
1.4	Davantage d'outils : espace Euclidien	
1.5	Exemples de vecteurs "structurés" : signaux	
1.6	Résumé	
<b>2</b>	<b>Transformations linéaires, calcul matriciel</b>	<b>39</b>
2.1	Matrices, calcul matriciel	
2.2	Matrices, familles de vecteurs, changement de base et projection orthogonale	
2.3	Résumé	
<b>3</b>	<b>Réduction de matrices et analyse multivariée</b> . . . . .	<b>57</b>
3.1	Généralités	
3.2	Réduction des matrices	
3.3	Analyse en composantes principales	
3.4	Résumé	
<b>4</b>	<b>Optimisation, analyse discriminante</b> . . .	<b>77</b>
4.1	Rappels d'analyse	
4.2	Optimisation	
4.3	Analyse discriminante	





# 1. Éléments d'algèbre linéaire

L'objectif principal de ce chapitre est d'aborder (et pour certains revisiter) des concepts centraux d'algèbre linéaire (et de géométrie vectorielle), et les illustrer dans le cadre de problèmes d'analyse de données.

Les données numériques se présentent souvent sous forme de tableaux de nombres, de dimension 1 ou 2, parfois plus, et leur interprétation fait appel à des outils mathématiques, souvent de l'algèbre et la géométrie, et pour d'autres aspects des probabilités et statistiques (traités dans un autre cours). La représentation et la visualisation de telles données est un premier aspect important de leur modélisation et leur exploitation. Une difficulté majeure est souvent liée à la taille de telles données, qui rend la visualisation extrêmement ardue.

Les techniques algébriques et géométriques permettent de faciliter la visualisation, et de donner sens à des notions telles que les projections sur des espaces de dimension inférieure, qu'il s'agisse d'espaces fixes à l'avance ou d'espace où la pertinence des données est la plus grande.

On va tout d'abord revoir dans ce chapitre les notions de vecteur, espace, base, dimension, puis on introduira les matrices, dont on verra qu'elles peuvent représenter des transformations (projections, rotations, symétries,...) qui seront utiles en pratique.

Les développements théoriques seront illustrés par quelques exercices simples, ainsi que des manipulations pratiques utilisant le logiciel R (dans l'environnement Rstudio).

## Rappels sur les vecteurs du plan et de l'espace

Chez les Romains, le mot *vector* désignait le passager ou le conducteur d'un bateau ou d'un chariot. Au Moyen-âge et jusqu'à la Renaissance, le mot *vecteur* désigne le conducteur d'un bateau ou d'un véhicule. Il faut attendre 1844 pour que le mathématicien anglais William Hamilton utilise le mot vecteur dans son sens actuel.

On rappelle dans cette courte section les éléments essentiels de géométrie vectorielle du plan. Cette section peut facilement être ignorée pour ceux qui se sentent suffisamment à l'aise avec ces notions. Les étudiants intéressés à approfondir peuvent se référer aux nombreux manuels de lycée, ou ressources en ligne, par exemple [3].

### Vecteurs du plan

Commençons par le cadre bidimensionnel. Imaginons deux points  $A$  et  $B$  dans le plan. Le couple  $(A, B)$  est appelé *bipoint*. On peut lui associer deux caractéristiques :

- Sa *norme*, qui est la longueur du segment  $AB$ .

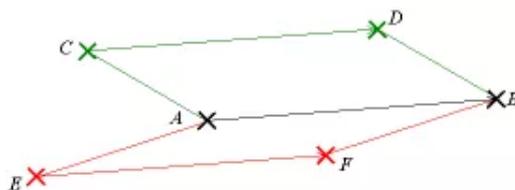


FIGURE 1.1 – Exemples de bipoints équipollents dans le plan.

— Son *orientation*, qui comprend sa direction (commune à toutes les droites de plan parallèles à la droite  $(AB)$ ) et son sens.

On associe à un bipoint  $(A, B)$  un objet appelé vecteur, noté  $\overrightarrow{AB}$ , déterminé précisément par ces deux caractéristiques.

Notons qu'étant donné un bipoint  $(A, B)$ , il existe une infinité de bipoints  $(C, D)$  tels que  $ABDC$  soit un parallélogramme. Un tel bipoint a la même norme, la même orientation, et le même sens que  $(A, B)$ . On dit qu'il est *équipollent* à  $(A, B)$ . On lui associe le vecteur  $\overrightarrow{CD}$ , qui est égal au vecteur  $\overrightarrow{AB}$  :

$$\overrightarrow{CD} = \overrightarrow{AB}.$$

Des exemples de bipoints équipollents se trouvent en FIGURE 1.1.

Comme il existe une infinité de bipoints associés à un même vecteur, on omet la dépendance dans les points  $A$  et  $B$ , et on utilise une notation générique  $\vec{U} = \overrightarrow{AB}$  (dans le reste de ce cours, on supprimera aussi la flèche au dessus des vecteurs). On dit que le bipoint  $(A, B)$  est un représentant du vecteur  $\vec{U}$  (tous les bipoints équipollents à  $(A, B)$  sont également des représentants du vecteur  $\vec{U}$ ).

Une propriété cruciale est que *pour tout vecteur  $\vec{U}$  et tout point  $A$  du plan, il existe un unique point  $B$  du plan tel que  $\vec{U} = \overrightarrow{AB}$ .*

Il n'existe pas de notion naturelle d'addition ou de multiplication associée à des points du plan ou de l'espace. Par contre il est possible de définir de telles notions sur les vecteurs. Par exemple, étant donnés deux vecteurs  $\overrightarrow{AB}$  et  $\overrightarrow{BC}$ , la relation de Chasles donne

$$\overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}.$$

Venons en aux deux opérations fondamentales sur les vecteurs :

- Comment additionner deux vecteurs quelconques  $\vec{U}$  et  $\vec{V}$  ? D'après la propriété précédente, on sait qu'il existe une infinité de points  $A, B$  tels que  $\vec{U} = \overrightarrow{AB}$ . De plus, il existe un (unique) point  $D$  tel que  $\vec{V} = \overrightarrow{BD}$ . De là, par la relation de Chasles, on en déduit que  $\vec{U} + \vec{V} = \overrightarrow{AD}$ . Notons que dans ce cas, en choisissant un point  $C$  tel que  $\overrightarrow{AC} = \vec{V}$ , on a aussi construit un parallélogramme  $ABDC$ . On a donc bien réussi à additionner deux vecteurs "abstraites". La construction est illustrée en FIGURE 1.2.
- Comment multiplier un vecteur par un nombre ? on sait qu'un vecteur  $\vec{U}$  est caractérisé par sa norme, sa direction et son sens. Étant donné un nombre réel (aussi appelé *scalaire*)  $\lambda$ , le vecteur  $\vec{V} = \lambda \vec{U}$  est l'unique vecteur défini par
  - Sa norme égale à  $\|\vec{V}\| = |\lambda| \|\vec{U}\|$ , où  $|\lambda|$  est la valeur absolue de  $\lambda$ .
  - Sa direction identique à la direction de  $\vec{U}$ .
  - Son sens, identique au sens de  $\vec{U}$  si  $\lambda$  est positif, et inverse si  $\lambda$  est négatif.

### Base et coordonnées

Nous n'avons jusqu'à présent parlé que de géométrie. L'approche algébrique arrive naturellement dès que l'on introduit les notions de base et coordonnées. L'utilisation des coordonnées cartésiennes, permet d'identifier un vecteur du plan à un couple de nombres réels. On a besoin pour cela de choisir deux vecteurs, qu'on notera ici  $\vec{E}_1$  et  $\vec{E}_2$ . La famille constituée de ces deux vecteurs  $\{\vec{E}_1, \vec{E}_2\}$  est appelée *base canonique du plan*. Tout vecteur du plan peut alors s'écrire sous la forme

$$\vec{U} = x_1 \vec{E}_1 + x_2 \vec{E}_2,$$

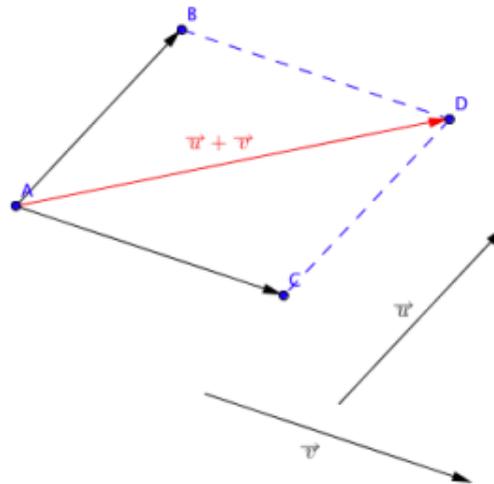


FIGURE 1.2 – Addition de deux vecteurs et règle du parallélogramme

les nombres  $x_1, x_2$  étant les coordonnées de  $\vec{U}$  dans la base. On note alors le vecteur sous la forme d'une expression faisant intervenir ses coordonnées :

$$\vec{U} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} .$$

Dans ce cadre, les vecteurs de la base canonique s'écrivent alors

$$\vec{E}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \quad \vec{E}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} .$$

Les opérations s'écrivent elles aussi de façon tout aussi simples

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \end{pmatrix} , \quad \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \end{pmatrix} .$$

Il y a bien d'autres notions de géométrie du plan, telles que les notions métriques (angle et produit scalaire, aire d'un parallélogramme ou d'un triangle,...) que nous aborderons dans un cadre général, mais pas dans ce résumé.

### Notions métriques : norme, produit scalaire, angle,...

Les coordonnées d'un vecteur du plan permettent d'exprimer de façon différente certaines notions métriques, telles que la norme. La norme d'un vecteur  $\vec{X}$  de coordonnées  $(x_1, x_2)$  se calcule en utilisant le théorème de Pythagore, et vaut  $\|\vec{X}\| = \sqrt{x_1^2 + x_2^2}$ . On introduit aussi le produit scalaire de deux vecteurs  $\vec{X}$  et  $\vec{Y}$ , défini par

$$\vec{X} \cdot \vec{Y} = \|\vec{X}\| \|\vec{Y}\| \cos(\theta) ,$$

où  $\theta$  est l'angle (orienté) que font les deux vecteurs. Le produit scalaire mesure en quelque sorte une ressemblance entre les deux vecteurs.

Si on note  $(x_1, x_2)$  et  $(y_1, y_2)$  les coordonnées respectives de  $\vec{X}$  et  $\vec{Y}$ , on montre alors que

$$\vec{X} \cdot \vec{Y} = x_1 y_1 + x_2 y_2 .$$

On verra aussi dans ce cours d'autres notions métriques comme le déterminant, qui jouera un rôle fondamental.

## Géométrie et vecteurs dans l'espace

La majorité des notions de géométrie du plan que nous venons de revoir se transposent sans difficulté au cas de géométrie de l'espace. Les propriétés sont même identiques pour la plupart d'entre elles (en particulier les relations entre bipoints et vecteurs, la relation de Chasles, les additions et multiplications par un scalaire...). La seule différence réside dans les coordonnées des vecteurs. Dans ce cas, la base canonique est constituée de trois vecteurs  $\{\vec{E}_1, \vec{E}_2, \vec{E}_3\}$ . Tout vecteur de l'espace peut alors s'écrire sous la forme

$$\vec{U} = x_1 \vec{E}_1 + x_2 \vec{E}_2 + x_3 \vec{E}_3,$$

les nombres  $x_1, x_2, x_3$  étant les coordonnées de  $\vec{U}$  dans la base. On note alors le vecteur sous la forme d'une expression faisant intervenir ses coordonnées :

$$\vec{U} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

et les vecteurs de la base canonique s'écrivent

$$\vec{E}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{E}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{E}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Les opérations s'écrivent elles aussi de façon tout aussi simples

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{pmatrix}, \quad \lambda \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \lambda x_3 \end{pmatrix}.$$

On pourra par exemple se référer à [2] pour davantage de propriétés.

**Remarque 1.1 — Notation.** Dans cette introduction nous avons utilisé une notation représentant les vecteurs par un symbole surmonté d'une flèche. A partir de maintenant, nous omettons la flèche et représenterons les vecteurs par des lettres majuscules.

### 1.1 Tableaux, nuages de points dans un espace de grande dimension,... et géométrie vectorielle

Pour justifier la nécessité de recourir à l'algèbre et la géométrie dans des problèmes pratiques, commençons par un petit exemple simple.

■ **Exemple 1.1 — Un tout petit tableau.** On considère le tableau de données ci-dessous, qui compte trois lignes et deux colonnes.

	$c_1$	$c_2$
$\ell_1$	1	3
$\ell_2$	2	-2
$\ell_3$	3	1

Géométriquement, on peut concevoir deux façons différentes de le représenter :

- on peut interpréter ses lignes comme les coordonnées de trois points dans un espace bidimensionnel,
- on peut interpréter ses colonnes comme les coordonnées de deux points dans un espace tridimensionnel.

Ces deux représentations sont explicitées dans la Figure 1.3, sur lesquelles on a également porté des lignes reliant l'origine aux points considérés, ornant ainsi des vecteurs. Ceci nous permettra d'appliquer des techniques d'algèbre linéaire et de géométrie pour analyser des tableaux de données plus grands. ■

■ **Exemple 1.2 — Un tableau un peu plus grand.** Considérons maintenant le tableau

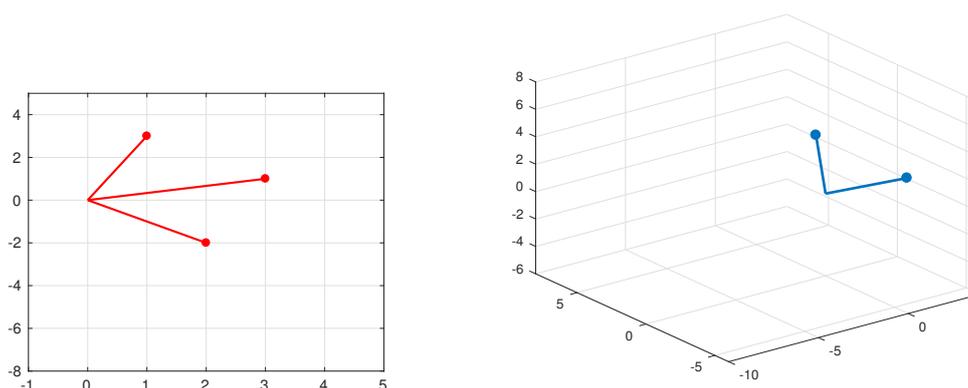


FIGURE 1.3 – Représentation d'un tableau  $3 \times 2$  sous forme de 3 points dans le plan (gauche), ou deux points dans l'espace tridimensionnel (droite)

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
$\ell_1$	0.5377	-0.2050	0.8884	-0.0301	-1.0891	-1.4023	-1.1480
$\ell_2$	1.8339	-0.1241	-1.1471	-0.1649	0.0326	-1.4224	0.1049
$\ell_3$	-2.2588	1.4897	-1.0689	0.6277	0.5525	0.4882	0.7223
$\ell_4$	0.8622	1.4090	-0.8095	1.0933	1.1006	-0.1774	2.5855
$\ell_5$	0.3188	1.4172	-2.9443	1.1093	1.5442	-0.1961	-0.6669
$\ell_6$	-1.3077	0.6715	1.4384	-0.8637	0.0859	1.4193	0.1873
$\ell_7$	-0.4336	-1.2075	0.3252	0.0774	-1.4916	0.2916	-0.0825
$\ell_8$	0.3426	0.7172	-0.7549	-1.2141	-0.7423	0.1978	-1.9330
$\ell_9$	3.5784	1.6302	1.3703	-1.1135	-1.0616	1.5877	-0.4390
$\ell_{10}$	2.7694	0.4889	-1.7115	-0.0068	2.3505	-0.8045	-1.7947
$\ell_{11}$	-1.3499	1.0347	-0.1022	1.5326	-0.6156	0.6966	0.8404
$\ell_{12}$	3.0349	0.7269	-0.2414	-0.7697	0.7481	0.8351	-0.8880
$\ell_{13}$	0.7254	-0.3034	0.3192	0.3714	-0.1924	-0.2437	0.1001
$\ell_{14}$	-0.0631	0.2939	0.3129	-0.2256	0.8886	0.2157	-0.5445
$\ell_{15}$	0.7147	-0.7873	-0.8649	1.1174	-0.7648	-1.1658	0.3035

Il peut lui aussi s'interpréter comme un nuage de 15 points dans un espace de dimension 7 (chaque ligne étant un point à 7 coordonnées), ou un nuage de 7 points dans un espace de dimension 15 (chaque colonne étant un point à 15 coordonnées).

Il est difficile ici d'avoir une intuition claire de la situation et de ce que peut représenter un tel tableau. Que peut on visualiser ? ou se trouve l'information importante ? Les outils algébriques vont nous permettre d'aller plus loin. ■

Dans un premier temps, nous allons privilégier l'interprétation "colonnes", les vecteurs seront donc représentés par des colonnes ; nous reviendrons sur les lignes plus loin.

## 1.2 Espace vectoriel, vecteur, dimension, base

### 1.2.1 Premières définitions

On considère un tableau à  $N$  lignes et  $M$  colonnes. Comme précédemment, on peut faire les interprétations géométriques suivantes :

- Chaque colonne représente les  $N$  coordonnées d'un point dans un espace à  $N$  dimensions, par rapport à la base canonique.
- Alternativement, chaque ligne représente les  $M$  coordonnées d'un point dans un espace à  $M$  dimensions.

Qu'entend on par coordonnées ? Étant donné un point  $P$  d'un espace à  $N$  dimensions, et en notant  $O$  une origine de l'espace<sup>1</sup>, ses coordonnées sont les poids de la décomposition du vecteur  $V = \vec{OP}$

$$\vec{OP} = x_1 E_1 + x_2 E_2 + \dots + x_N E_N = V. \quad (1.1)$$

1. En géométrie vectorielle, on représente tous les vecteurs partant de l'origine de l'espace.

Une telle expression est appelée *combinaison linéaire* des vecteurs  $E_1, \dots, E_N$ .

Dans la suite, on notera généralement les vecteurs sous forme de colonnes, par exemple

$$V = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}, \quad E_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots \quad E_N = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix},$$

On utilisera aussi la notion de *transposition*, qui transforme les colonnes en lignes et vice-versa (voir plus loin pour une définition précise), sous la forme

$$V = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = (x_1, x_2, \dots, x_N)^T.$$

On dit que les  $N$  vecteurs  $E_1, \dots, E_N$  forment la *base canonique* de l'espace  $N$ -dimensionnel  $\mathbb{R}^N$ .

■ **Exemple 1.3 — Suite de l'exemple 1.1.** Dans notre petit exemple, les trois lignes du tableau constituent les vecteurs du plan

$$L_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \quad L_3 = \begin{pmatrix} 3 \\ 1 \end{pmatrix},$$

alors que les deux colonnes du tableau forment les vecteurs de l'espace

$$C_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 3 \\ -2 \\ 1 \end{pmatrix}.$$

On dispose donc de deux représentations possibles pour ce tableau. ■

Commençons par rappeler des notations standard. On note  $\mathbb{Z}$  l'ensemble des nombres entiers relatifs (positifs ou négatifs), et  $\mathbb{R}$  l'ensemble des nombres réels. Ces deux ensembles sont munis d'opérations bien connues (addition et multiplication) sur lesquelles on ne revient pas.

**Définition 1.1 — Espace vectoriel.** On note  $\mathbb{R}^N$  l'ensemble des vecteurs à  $N$  coordonnées, que l'on munit de deux opérations

1. Addition des vecteurs : à toute paire de vecteurs  $X = (x_1, \dots, x_N)^T$  et  $(Y = (y_1, \dots, y_N)^T$  on associe le vecteur  $X + Y$  défini par

$$X + Y = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_N + y_N \end{pmatrix} \quad (1.2)$$

2. Multiplication d'un vecteur par un scalaire : à tout vecteur  $X = (x_1, \dots, x_N)^T$  et tout nombre  $\lambda$  on associe le vecteur  $\lambda X$  défini par

$$\lambda X = \lambda \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_N \end{pmatrix} \quad (1.3)$$

On dit que  $\mathbb{R}^N$  muni de ces deux opérations est un *espace vectoriel*.

Notons que  $\mathbb{R}^N$  contient notamment le *vecteur nul*  $(0, 0, \dots, 0)^T$ , que l'on note  $0$  par abus de notation, et que tout vecteur  $V = (x_1, x_2, \dots, x_N)^T$  admet un *vecteur opposé*, noté  $-V = (-x_1, -x_2, \dots, -x_N)$ .

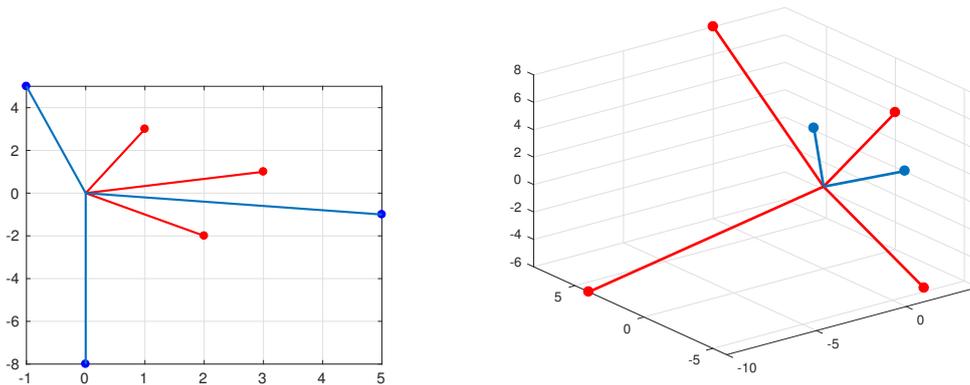


FIGURE 1.4 – Construction de nouveaux points dans un espace bidimensionnel (gauche) ou tridimensionnel (droite)

- **Exemple 1.4 — Quelle est l'utilité d'additionner les vecteurs ?.** — Les combinaisons linéaires de vecteurs permettent de construire de nouveaux points dans le plan (ou l'espace). Par exemple, dans la figure 1.4 (gauche), on a représenté en plus des vecteurs initiaux les vecteurs

$$L_4 = L_1 - L_2, \quad L_5 = L_3 - 3L_1 \quad \text{et} \quad L_6 = L_1 + 2L_2$$

De même, dans le tracé de droite, on a représenté en plus les points correspondant aux vecteurs

$$C_3 = C_1 + C_2, \quad C_4 = 3C_1 - C_2, \quad C_5 = -2C_1 + C_2 \quad \text{et} \quad C_6 = -C_1 - 2C_2.$$

- L'addition et la multiplication scalaire permettent aussi de manipuler les vecteurs comme des nombres (dans une certaine mesure), et par exemple de résoudre des équations : la solution  $X \in \mathbb{R}^3$  de l'équation

$$3X + \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 0$$

est

$$X = -\frac{1}{3} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -1/3 \\ -2/3 \\ -1 \end{pmatrix}.$$

■

### 1.2.2 Base d'un espace vectoriel

Un vecteur est un objet relativement abstrait. Il est souvent utile de le représenter par ses coordonnées dans une base de l'espace.

**Définition 1.2 — Bases.** Une base de  $\mathbb{R}^N$  est une famille  $\mathcal{U} = \{U_1, \dots, U_N\}$  constituée exactement de  $N$  vecteurs  $U_1, \dots, U_N \in \mathbb{R}^N$ , telle que tout vecteur de  $\mathbb{R}^N$  puisse s'écrire sous la forme

$$X = \alpha_1 U_1 + \alpha_2 U_2 + \dots + \alpha_N U_N.$$

Cette décomposition, appelée combinaison linéaire des vecteurs de  $\mathcal{U}$  est unique au sens suivant : pour tout vecteur  $X$  de  $\mathbb{R}^N$ , il existe un unique  $N$ -uplet de nombres réels  $(\alpha_1, \dots, \alpha_N)$ , appelés coordonnées (ou parfois coefficients) de  $X$  dans la base  $\mathcal{U}$  tel que l'équation ci-dessus soit satisfaite.

Par exemple, dans la base canonique, on décompose tout vecteur  $V = (x_1, \dots, x_N)^T$  sous la forme

$$V = x_1 E_1 + x_2 E_2 + \dots + x_N E_N.$$

Étant donnée une base de  $\mathbb{R}^N$ , la détermination des coordonnées de ce vecteur par rapport à la base  $\mathcal{U}$  demande de résoudre un système linéaire. On illustre cela sur deux exemples en dimension 2 et 3

■ **Exemple 1.5 — Décomposition d'un vecteur sur une base dans  $\mathbb{R}^2$ .** On considère les vecteurs suivants de  $\mathbb{R}^2$  :  $U = (1, 1)^T$  et  $V = (1, 2)^T$ .

1. On considère le vecteur  $A = (1, -3)^T$ . Vérifions que le vecteur  $A$  peut se décomposer de façon unique comme combinaison linéaire de  $U$  et  $V$ , c'est à dire qu'il existe un unique couple de réels  $(\lambda, \mu)$  tel que  $A = \lambda U + \mu V$ . Cette égalité s'écrit

$$\begin{pmatrix} 1 \\ -3 \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \lambda + \mu \\ \lambda + 2\mu \end{pmatrix},$$

et conduit à un système linéaire

$$\begin{cases} \lambda + \mu = 1 \\ \lambda + 2\mu = -3 \end{cases} \iff \begin{cases} \lambda + \mu = 1 \\ \mu = -4 \end{cases}$$

(où on a remplacé la dernière ligne par la différence de la dernière avec la première, pour éliminer le terme en  $\lambda$ ) qui a pour solution  $\lambda = 5$  et  $\mu = -4$ .

2. Même question pour un vecteur  $B = (x, y)^T \in \mathbb{R}^2$  quelconque. On obtient le *système linéaire paramétrique*, au sens où les inconnues  $\lambda, \mu$  doivent être exprimées en fonction des paramètres  $x, y$

$$\begin{cases} \lambda + \mu = x \\ \lambda + 2\mu = y \end{cases} \iff \begin{cases} \lambda + \mu = x \\ \mu = y - x \end{cases}.$$

Ce système a pour solution  $\lambda = 2x - y$  et  $\mu = y - x$ . ■

Le système linéaire de l'exemple 1.5 peut en fait s'écrire en faisant intervenir ce que l'on appellera plus loin des *matrices*. On forme la matrice à deux lignes et deux colonnes à partir des coefficients intervenant dans le système linéaire, c'est à dire le tableau à deux lignes et deux colonnes

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

(qui est aussi la concaténation des deux vecteurs). Le système s'écrit alors sous la forme d'une équation matricielle

$$M \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \end{pmatrix},$$

qui fait intervenir un produit entre la matrice  $M$  et le vecteur inconnu  $(\lambda, \mu)^T$ , nous définirons ce produit un peu plus loin.



### (Résolution numérique du système linéaire)

Il existe des techniques de calcul permettant de résoudre ce système de façon analytique (à la main). Dans ce cours, on se contentera d'utiliser les algorithmes de résolution numérique fournis par le logiciel R. Dans ce cas particulier, il suffit d'utiliser une fonction appelée `solve`, la syntaxe est de la forme

```
> U1 <- c(1, 1)
> U2 <- c(1, 2)
> A <- matrix(c(U1, U2), ncol=2)
> b <- c(1, -3)
> x <- solve(A, b)
```

Les deux premières lignes définissent les vecteurs  $U_1$  et  $U_2$ , et la troisième les concatène pour former la matrice. La dernière ligne résout le système linéaire et l'affecte dans le vecteur  $x$ .

Voyons maintenant ceci d'un point de vue plus mathématique. De la même façon que l'équation usuelle  $2x = 3$  admet une solution unique que l'on peut écrire (un peu lourdement)  $x = 2^{-1} \times 3$ , l'équation matricielle ci-dessus admet une solution unique

$$\begin{pmatrix} \lambda \\ \mu \end{pmatrix} = M^{-1} \begin{pmatrix} 1 \\ -3 \end{pmatrix},$$

où  $M^{-1}$  est une autre matrice, appelée *matrice inverse* de  $M$ . La matrice inverse  $M^{-1}$  existe dans ce cas particulier, on dit que la matrice  $M$  possède la propriété d'inversibilité, autre notion qui sera introduite plus loin.

■ **Exemple 1.6 — Décomposition d'un vecteur sur une base dans  $\mathbb{R}^3$ .** On considère la famille  $\mathcal{U}$  constituée des trois vecteurs de  $\mathbb{R}^3$

$$U_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad U_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad U_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Il est possible de démontrer que ces trois vecteurs forment une base de  $\mathbb{R}^3$ .

1. Soit  $A = (1, 2, 3)^T$ . Décomposons  $A$  sur la base  $\mathcal{U} = \{U_1, U_2, U_3\}$ . Il faut pour cela trouver trois réels  $\lambda_1, \lambda_2$  et  $\lambda_3$  tels que

$$\lambda_1 U_1 + \lambda_2 U_2 + \lambda_3 U_3 = A.$$

On peut voir que cette équation vectorielle est équivalente au système linéaire

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ \lambda_1 - \lambda_2 + \lambda_3 = 2 \\ \lambda_3 = 3 \end{cases}$$

Comme précédemment il est possible d'écrire ce système sous la forme d'une équation matricielle

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

que l'on peut résoudre numériquement, ou analytiquement. Notons que la matrice intervenant dans cette équation s'obtient simplement en concaténant les trois vecteurs  $U_1, U_2$  et  $U_3$ .

2. Soit  $B = (x, y, z)^T \in \mathbb{R}^3$  quelconque. On peut de la même manière décomposer  $B$  sur la base  $\mathcal{U}$ . Ceci passe par la résolution d'un système paramétrique, qui peut se résoudre analytiquement, ou en utilisant un logiciel de calcul formel.

**Remarque 1.2 — Changement de base.** On a déjà vu comment trouver les coordonnées d'un vecteur dans une base. Étant données deux bases  $\mathcal{U} = \{U_1, \dots, U_N\}$  et  $\mathcal{V} = \{V_1, \dots, V_N\}$ , tout vecteur  $X \in \mathbb{R}^N$  admet deux décompositions correspondantes

$$X = \sum_{n=1}^N \alpha_n U_n = \sum_{n=1}^N \beta_n V_n,$$

et il existe des méthodes permettant de passer des coefficients  $\alpha_n$  aux coefficients  $\beta_n$  et inversement. On va voir que le calcul matriciel permet de faire de tels calculs simplement.

## 1.3 Sous-espace vectoriel

La notion de sous-espace vectoriel est une notion fondamentale pour les applications visées dans ce cours. Un sous-espace vectoriel est un sous-ensemble d'un espace vectoriel dont les éléments vérifient des propriétés particulières, appelées propriétés de stabilité : stabilité par addition et multiplication par un scalaire. Plus précisément :

**Définition 1.3 — Sous-espace vectoriel.** Un sous-espace vectoriel de  $\mathbb{R}^N$  est un sous-ensemble  $E$  de  $\mathbb{R}^N$  stable par addition et multiplication scalaire :

1. Quels que soient les vecteurs  $U, V$  de  $E$ , leur somme  $U + V$  est aussi un vecteur de  $E$ .
2. Quel que soit le vecteur  $U$  de  $E$  et le réel  $\alpha$ , le vecteur  $\alpha U$  appartient à  $E$ .

Notons en particulier qu'un sous-espace vectoriel contient nécessairement le vecteur nul  $(0, 0, \dots, 0)^T$ .

Pour bien appréhender cette notion, il est utile de considérer quelques exemples.

■ **Exemple 1.7** — Dans  $\mathbb{R}^2$ .  $\mathbb{R}^2$  est un espace vectoriel de dimension 2 (toute base de  $\mathbb{R}^2$  est constituée de 2 vecteurs). Il existe trois catégories de sous-espaces vectoriels de  $\mathbb{R}^2$  :

- $\mathbb{R}^2$  lui-même, qui est donc de dimension 2
- L'ensemble réduit au seul vecteur nul  $\{0\}$ , qui est de dimension 0
- Toutes les droites contenant le vecteur nul sont des sous-espaces vectoriels de dimension 1. Rappelons qu'une droite vectorielle de  $\mathbb{R}^2$  est caractérisée par une équation cartésienne, de la forme  $ax + by = 0$  où  $a$  et  $b$  sont deux réels, ce que l'on écrit

$$D = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \text{ tels que } ax + by = 0 \right\}.$$

Rappelons que la dimension de l'espace ambiant  $\mathbb{R}^2$  vaut 2. Dans le premier cas, la dimension du sous-espace vaut aussi 2, et aucune équation cartésienne n'est nécessaire. Dans le second, la dimension du sous-espace vaut 0, et il est caractérisé par deux équations cartésiennes :  $x = 0$  et  $y = 0$ . Dans le troisième cas, la dimension du sous-espace vaut 1, et il est caractérisé par une équation cartésienne. Dans tous les cas, le nombre d'équations cartésiennes est égal à la dimension de l'espace ambiant dont on soustrait la dimension du sous-espace. ■

■ **Exemple 1.8** — Dans  $\mathbb{R}^3$ . Il existe cette fois quatre types de sous-espaces vectoriels de  $\mathbb{R}^3$ .

1.  $\mathbb{R}^3$  lui-même, de dimension trois, aucune équation cartésienne n'est nécessaire.
2. L'ensemble réduit au vecteur nul  $\{0\}$ , qui est de dimension 0. Trois équations cartésiennes sont nécessaires, à savoir  $x = 0$ ,  $y = 0$  et  $z = 0$ .
3. *Plans vectoriels* : Les plans de  $\mathbb{R}^3$  sont de dimension 2, et sont caractérisés par une équation cartésienne.
4. *Droites vectorielles* : Les droites vectorielles sont de dimension 1 et peuvent être définies comme des intersections de deux plans vectoriels. Les vecteurs de cette intersection doivent satisfaire les deux équations cartésiennes des deux plans. Une droite vectorielle est caractérisée par deux équations cartésiennes non-équivalentes (c'est à dire qui n'ont pas le même ensemble de solutions ; par exemple les équations  $x + 2y + 3z = 0$  et  $2x + 4y + 6z = 0$  sont équivalentes.).

Là encore, le nombre d'équations cartésiennes non-équivalentes est toujours égal à la dimension de l'espace ambiant dont on soustrait la dimension du sous-espace.

La Figure 1.5 représente trois plans vectoriels de  $\mathbb{R}^3$ , ainsi qu'une droite qui est l'intersection de deux de ces plans. ■

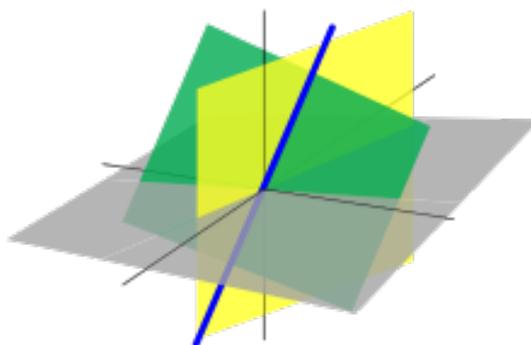


FIGURE 1.5 – Quelques plans vectoriels dans l'espace  $\mathbb{R}^3$  (en gris, vert et jaune). La droite en bleu est l'intersection des plans jaune et verts, et est de dimension 1.

Similairement, l'espace  $\mathbb{R}^4$ , de dimension quatre, admet cinq catégories de sous-espaces : outre  $\mathbb{R}^4$  lui-même et le sous-espace réduit au vecteur nul, il y a les droites vectorielles (dimension 1), les plans vectoriels (dimension 2) ainsi que les hyperplans (dimension 3).

De façon générique,  $\mathbb{R}^N$  admet des sous-espaces vectoriels de dimensions allant de 0 à  $N$ , un sous-espace vectoriel de dimension  $M \leq N$  de  $\mathbb{R}^N$  peut être caractérisé par un système de  $N - M$  équations cartésiennes.

Une notion importante est la notion de base d'un sous-espace vectoriel, et de sous-espace vectoriel engendré par une famille de vecteurs. Les bases des sous-espaces sont définies de façon similaires aux bases de  $\mathbb{R}^N$ .

**Définition 1.4 — Base d'un sous-espace vectoriel.** Étant donné un sous-espace vectoriel  $E$  de  $\mathbb{R}^N$ , une base de  $E$  est une famille de vecteurs  $\mathcal{F} = \{F_1, \dots, F_M\}$  de  $E$  telle que tout vecteur  $V \in E$  s'écrit de façon unique comme combinaison linéaire des vecteurs de  $\mathcal{F}$ . En d'autres termes, pour tout  $V \in E$ , il existe un unique  $M$ -uplet de réels  $\alpha_1, \dots, \alpha_M$  tels que

$$V = \alpha_1 F_1 + \alpha_2 F_2 + \dots + \alpha_M F_M .$$

$E$  est un sous-espace de dimension  $M$ .

Il est important de se souvenir que toutes les bases d'un même sous-espace vectoriel de  $\mathbb{R}^N$  ont le même nombre d'éléments, égal à la dimension du sous-espace.

■ **Exemple 1.9 — Base d'un sous-espace de  $\mathbb{R}^3$ .** On considère le plan vectoriel de  $\mathbb{R}^3$  d'équation cartésienne  $x - 2y + z = 0$ . Comment trouver une base de  $E$ ? Comme  $E$  est un plan (il est caractérisé par une seule équation cartésienne) il est de dimension 2, il faut donc trouver deux vecteurs de  $E$ . De plus, ces deux vecteurs ne doivent pas être colinéaires (sinon ils engendrent une droite, pas un plan). Deux vecteurs non colinéaires engendrent automatiquement un plan.

Pour les trouver, il suffit de prendre des cas particuliers, les plus simples possibles. Si on choisit  $z = 0$ , l'équation cartésienne devient  $x = 2y$ , ce qui conduit naturellement à choisir le vecteur  $F_1 = (2, 1, 0)^T$ . Si on choisit maintenant  $x = 0$ , l'équation cartésienne devient  $z = 2y$ , ce qui conduit naturellement à choisir le vecteur  $F_2 = (0, 1, 2)^T$ .

La famille

$$\mathcal{F} = \{(2, 1, 0)^T, (0, 1, 2)^T\}$$

est donc une base du plan  $E \subset \mathbb{R}^3$ . ■

**Définition 1.5 — Sous-espace engendré par une famille de vecteurs.** Étant donnés  $M$  vecteurs  $U_1, \dots, U_M$  de  $\mathbb{R}^N$ , le sous-espace de  $\mathbb{R}^N$  engendré par ces vecteurs est l'ensemble de toutes les combinaisons linéaires de ces vecteurs de la forme

$$\alpha_1 U_1 + \alpha_2 U_2 + \dots + \alpha_M U_M ,$$

où  $\alpha_1, \dots, \alpha_M$  sont des réels quelconques. Ce sous-espace vectoriel est noté  $\text{Vect}\{U_1, \dots, U_M\}$ .

La dimension de  $\text{Vect}\{U_1, \dots, U_M\}$ , qui est inférieure ou égale à  $N$  puisqu'il s'agit d'un sous-espace de  $\mathbb{R}^N$ , est aussi inférieure ou égale à  $M$ . Elle est appelée *rang* de la famille de vecteurs, et notée

$$\text{rg}\{U_1, \dots, U_M\} = \dim(\text{Vect}\{U_1, \dots, U_M\}) .$$

**Remarque 1.3 — Rang et cardinal d'une famille de vecteurs.** Notons qu'une famille de vecteurs est une base du sous-espace qu'elle engendre si son rang est égal au nombre de vecteurs de la famille (qu'on appelle aussi *cardinal* de la famille).

■ **Exemple 1.10 — Calcul de rang.** On considère la famille de vecteurs de  $\mathbb{R}^3$  constituée des vecteurs  $U_1 = (1, 2, 3)^T$ ,  $U_2 = (4, 5, 6)^T$  et  $U_3 = (7, 8, 9)^T$ . Le calcul montre que ces vecteurs engendrent un sous-espace de  $\mathbb{R}^3$  qui est de dimension 2 (plus précisément, le plan d'équation cartésienne  $x - 2y + z = 0$ ). Donc le rang de cette famille est égal à 2. Il est différent du cardinal de la famille, cette dernière n'est donc pas une base du sous-espace qu'elle engendre.

Ceci signifie aussi que ces trois vecteurs sont coplanaires, il existe une relation entre eux. Le calcul montre que cette relation est de la forme  $U_1 - 2U_2 + U_3 = 0$ , soit par exemple  $U_3 = 2U_2 - U_1$ . On vérifie facilement que  $U_1$  et  $U_2$  ne sont pas colinéaires (i.e. proportionnels), donc la famille  $\{U_1, U_2\}$  est une base du plan  $\text{Vect}\{U_1, U_2, U_3\}$ . On aurait aussi faire d'autres choix, par exemple  $\{U_1, U_3\}$  est une autre base du plan. ■



### (Évaluation numérique du rang d'une famille de vecteurs)

Le paquet `pracma` implémente une fonction `Rank` qui permet d'évaluer le rang d'une famille de vecteurs. Il suffit de concaténer ces vecteurs en une matrice, et d'utiliser cette fonction. Dans le cas de l'Exemple 1.10, on peut faire

```
> require("pracma")
> U1 <- 1:3
> U2 <- 4:6
> U3 <- 7:9
> M <- matrix(c(U1,U2,U3),ncol=3)
> Rank(M)
[1] 2
```

La première ligne charge le paquet `pracma` (et l'installe s'il n'est pas déjà installé), les trois lignes suivantes définissent les vecteurs  $U_1$ ,  $U_2$  et  $U_3$ , la ligne suivante constitue la matrice, et la dernière calcule son rang.

Attention : ne pas confondre la fonction `Rank` utilisée ici avec la fonction `rank` (avec un `r` minuscule), qui n'a rien à voir.

## 1.4 Davantage d'outils : espace Euclidien

### 1.4.1 Norme, produit scalaire

Pour aller plus loin, il est nécessaire de continuer à équiper  $\mathbb{R}^N$  d'outils. On aura en particulier besoin de ce que l'on appelle des outils métriques, qui permettent de donner un sens à la notion de longueur d'un vecteur, ou encore d'angle entre deux vecteurs, et par la suite de définir des notions d'aire, de volume, hypervolume,...

**Définition 1.6 — Espace Euclidien.** Un espace Euclidien est un espace vectoriel muni de deux autres opérations :

1. Le produit scalaire associe à tout couple de vecteurs  $X, Y \in \mathbb{R}^N$  un nombre noté  $X \cdot Y$  défini par

$$X \cdot Y = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N \quad (1.4)$$

2. La norme d'un vecteur  $X \in \mathbb{R}^N$  est le nombre positif ou nul

$$\|X\| = \sqrt{X \cdot X} = \sqrt{x_1^2 + x_2^2 + \cdots + x_N^2} . \quad (1.5)$$

La norme d'un vecteur représente une mesure de sa longueur. Quant au produit scalaire, il peut s'interpréter comme suit. L'inégalité de Cauchy-Schwarz stipule que pour tous vecteurs  $X, Y \in \mathbb{R}^N$ , on a

$$X \cdot Y \leq \|X\| \|Y\| . \quad (1.6)$$

Par conséquent, le quotient de  $X \cdot Y$  par le produit des normes  $\|X\| \|Y\|$  est un nombre compris entre -1 et 1, ... comme un cosinus. On introduit donc un angle  $\theta$  tel que

$$X \cdot Y = \|X\| \|Y\| \cos(\theta) , \quad (1.7)$$

et  $\theta$  n'est autre que l'angle entre les vecteurs  $X$  et  $Y$ . Si le produit scalaire de deux vecteurs (différents du vecteur nul) est nul, alors le cosinus de l'angle est égal à 0, ce qui correspond à un angle égal à  $\pm\pi/2$ , les deux vecteurs sont donc orthogonaux. On va revenir plus loin sur cette importante notion.



**(Norme et produit scalaire)** Le paquet `pracma` fournit les fonctions `dot` et `Norm` (attention, la majuscule est importante, la fonction `norm` fait autre chose) qui évaluent respectivement le produit scalaire de deux vecteurs et la norme d'un vecteur. La syntaxe est ci-dessous.

```
> require("pracma")
> U1 <- 1:3
> U2 <- 4:6
> Norm(U1)
[1] 3.741657
> dot(U1,U2)
[1] 32
```

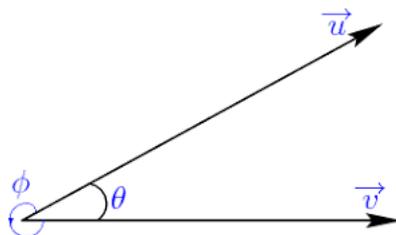


FIGURE 1.6 – Représentation de l'angle entre deux vecteurs

### 1.4.2 Déterminant

On ne donnera pas ici la définition formelle du déterminant, ni ses propriétés mathématiques, on se contentera d'en donner une vision géométrique.

- **Exemple 1.11 — Les cas de  $\mathbb{R}^2$  et  $\mathbb{R}^3$ .** 1. Dans  $\mathbb{R}^2$  le déterminant de deux vecteurs  $U = (x, y)^T$  et  $V = (x', y')^T$  est égal à

$$\det(U, V) = xy' - yx' . \quad (1.8)$$

On peut montrer par une construction géométrique simple que la valeur absolue du déterminant coïncide avec l'aire du parallélogramme engendré par ces deux vecteurs, alors que son signe est égal au signe de l'angle orienté entre ces deux vecteurs.

2. Dans  $\mathbb{R}^3$ , le déterminant de trois vecteurs  $U = (x, y, z)^T$ ,  $V = (x', y', z')^T$  et  $W = (x'', y'', z'')^T$  est donné par

$$\det(U, V, W) = x(y'z'' - z'y'') - y(x'z'' - z'x'') + z(x'y'' - y'z'') . \quad (1.9)$$

Il est inutile de retenir la formule par cœur. Par contre il est important de savoir que la valeur absolue du déterminant est égale au volume du parallélépipède engendré par ces trois vecteurs, alors que le signe du déterminant donne des informations sur leur orientation.

■

Plus généralement, dans  $\mathbb{R}^N$ , le déterminant d'une famille de  $N$  vecteurs<sup>2</sup> donne des informations sur l'hyper-parallélépipède engendré par ces  $N$  vecteurs :

- la valeur absolue du déterminant est égale à l'hyper-volume de l'hyper-parallélépipède engendré par ces  $N$  vecteurs,
- le signe du déterminant donne des informations sur leur orientation relative.

Prenons l'exemple de trois vecteurs de  $\mathbb{R}^3$ . Si leur déterminant est nul, cela veut dire que les trois vecteurs sont coplanaires, et n'engendrent donc pas  $\mathbb{R}^3$  : les vecteurs n'appartenant pas au plan engendré ne peuvent s'écrire comme combinaison linéaire de ces trois vecteurs. Ils ne peuvent donc pas former une base de  $\mathbb{R}^3$ . Par contre si le déterminant est non nul, ils forment bien une base de  $\mathbb{R}^3$ . Ceci reste vrai en toute dimension :

**Corollaire 1.4** Une famille de  $N$  vecteurs de  $\mathbb{R}^N$  est une base de  $\mathbb{R}^N$  si son déterminant est non-nul.



La fonction `det` permet le calcul numérique du déterminant. Elle prend en entrée la matrice construite à partir des vecteurs. L'exemple ci-dessous évalue le volume d'un hyper-parallélépipède aléatoire de dimension quatre

```
> V1 <- runif(4)
> V2 <- runif(4)
> V3 <- runif(4)
> V4 <- runif(4)
> M <- matrix(c(V1, V2, V3, V4), ncol=4)
> det(M)
```

2. Le déterminant n'est bien défini que pour un nombre de vecteurs égal à la dimension de l'espace.

Les quatre premières lignes génèrent les vecteurs (coordonnées distribuées suivant une loi uniforme entre 0 et 1), la cinquième forme la matrice. On aurait pu plus simplement remplacer les cinq premières lignes par

```
> V <- runif(16)
> M <- matrix(V,ncol=4)
```

### 1.4.3 Orthogonalité

La notion d'orthogonalité va être centrale dans ce qui suit. On va voir l'orthogonalité des vecteurs, mais aussi des sous-espaces.

**Définition 1.7 — Orthogonalité.** 1. Deux vecteurs  $X, Y$  de  $\mathbb{R}^N$  sont orthogonaux (ce que l'on note  $X \perp Y$ ) si leur produit scalaire est nul (ils font donc entre eux un angle égal à  $\pm\pi/2$ ) :

$$X \cdot Y = 0 .$$

2. Étant donné un sous-espace  $E$  de  $\mathbb{R}^N$  et un vecteur  $X \in \mathbb{R}^N$ ,  $X$  est orthogonal à  $E$  s'il est orthogonal à tout vecteur de  $E$ .
3. Deux sous-espaces  $E, F$  de  $\mathbb{R}^N$  sont orthogonaux si tout vecteur de  $E$  est orthogonal à tout vecteur de  $F$ .

■ **Exemple 1.12 — Vecteur normal à un plan dans  $\mathbb{R}^3$ .** Considérons le plan de  $\mathbb{R}^3$  d'équation Cartésienne

$$x + 2y + 3z = 0$$

On vérifie facilement que le vecteur  $U = (1, 2, 3)$  est orthogonal à ce sous-espace. En effet, pour tout vecteur  $V$  appartenant au sous-espace, le produit scalaire  $U \cdot V$  vaut

$$U \cdot V = x + 2y + 3z = 0$$

car les coordonnées de  $V$  satisfont l'équation cartésienne. ■

**Remarque 1.5** Les définitions d'un vecteur orthogonal à un sous-espace et de deux sous-espaces orthogonaux ne sont pas opérationnelles, car il n'est par exemple pas possible de tester tous les vecteurs d'un sous-espace vectoriel (il y en a une infinité). On peut toutefois simplifier ces conditions en utilisant des bases des sous-espaces. On peut remplacer ces conditions par

- Un vecteur  $X \in \mathbb{R}^N$  est orthogonal à un sous-espace  $E \subset \mathbb{R}^N$  de  $\mathbb{R}^N$  si étant donnée une base  $\{V_1, \dots, V_M\}$  de  $E$ ,  $X \perp V_m$  pour tout  $m = 1, \dots, M$ .
- Deux sous-espaces  $E$  et  $F$  de  $\mathbb{R}^N$  sont orthogonaux si étant données deux bases de  $E$  et  $F$  respectivement, tous les vecteurs de la base de  $E$  sont orthogonaux à tous les vecteurs de la base de  $F$ .

**Définition 1.8 — Base orthonormée.** On considère l'espace Euclidien  $\mathbb{R}^N$ .

1. Une base  $\mathcal{U} = \{U_1, \dots, U_N\}$  de  $\mathbb{R}^N$  est orthonormée si et seulement si tous ses vecteurs sont de norme égale à 1 et s'ils sont orthogonaux deux à deux :

$$\|U_n\| = 1 \quad \forall n = 1, \dots, N, \quad U_n \cdot U_m = 0 \quad \forall n, m = 1, \dots, N, \quad n \neq m .$$

2. Étant donné un sous-espace  $E$  de  $\mathbb{R}^N$ , on définit de la même façon une base orthonormée de  $E$ . Toute famille  $\mathcal{V} = \{V_1, \dots, V_M\}$  de vecteurs de  $E$  (où  $M = \dim(E)$ ), de norme unité et orthogonaux deux à deux.

■ **Exemple 1.13 — Base canonique et variantes.** — Dans  $\mathbb{R}^3$ , la base canonique que nous avons notée  $\{E_1, E_2, E_3\}$  est orthonormée. De même la base canonique de  $\mathbb{R}^N$  est orthonormée.

- La famille  $\{E_1, E_2, -E_3\}$  est également orthonormée, comme toutes les variantes obtenues en changeant des signes (ce qui n'est pas une opération très fondamentale...).
- Autre variante, si on fait tourner deux vecteurs dans le plan qu'ils engendrent, on ne change ni les normes ni les produits scalaires, donc on obtient encore une base orthonormée. En tournant d'un angle

$\pi/6$  par exemple dans le plan engendré par  $E_1$  et  $E_2$ , on obtient les vecteurs

$$E'_1 = \begin{pmatrix} \sqrt{3}/2 \\ 1/2 \\ 0 \end{pmatrix}, \quad E'_2 = \begin{pmatrix} -1/2 \\ \sqrt{3}/2 \\ 0 \end{pmatrix},$$

et la famille  $\{E'_1, E'_2, E_3\}$  est encore une base orthonormée de  $\mathbb{R}^3$ . ■

Les bases orthonormées sont des outils de choix, bien plus simple à manier que des bases quelconques. On a par exemple

**Proposition 1.6** 1. Une famille orthonormée  $\mathcal{U} = \{U_1, \dots, U_N\}$  de  $N$  vecteurs de  $\mathbb{R}^N$  est automatiquement une base orthonormée de  $\mathbb{R}^N$ .

2. Soit  $M \leq N$  un entier. Une famille orthonormée  $\mathcal{V} = \{V_1, \dots, V_M\}$  de  $M$  vecteurs de  $\mathbb{R}^N$  est automatiquement une base orthonormée du sous-espace  $E = \text{Vect}\{\mathcal{V}\}$  qu'elle engendre.

Les bases orthonormées permettent de simplifier bien des opérations, par exemple le calcul des coordonnées d'un vecteur d'un espace vectoriel (qui demande dans le cas général de résoudre un système linéaire)

**Théorème 1.7** Soit  $\mathcal{U} = \{U_1, \dots, U_N\}$  une base orthonormée de  $\mathbb{R}^N$ .

1. Tout vecteur  $X \in \mathbb{R}^N$  s'écrit

$$X = (X \cdot U_1)U_1 + (X \cdot U_2)U_2 + \dots + (X \cdot U_N)U_N. \quad (1.10)$$

2. Pour tout vecteur  $X \in \mathbb{R}^N$  on a

$$\|X\|^2 = (X \cdot U_1)^2 + (X \cdot U_2)^2 + \dots + (X \cdot U_N)^2. \quad (1.11)$$

C'est la formule de Pythagore-Parseval.

3. Pour tous vecteurs  $X, Y$  de  $\mathbb{R}^N$ ,

$$X \cdot Y = (X \cdot U_1)(Y \cdot U_1) + (X \cdot U_2)(Y \cdot U_2) + \dots + (X \cdot U_N)(Y \cdot U_N). \quad (1.12)$$

Une notion que nous allons beaucoup utiliser par la suite est la notion de *projection orthogonale* sur un sous-espace donné  $E \subset \mathbb{R}^N$ .

**Définition 1.9 — Projection orthogonale.** Soit  $E$  un sous-espace vectoriel de  $\mathbb{R}^N$ . La projection orthogonale de  $\mathbb{R}^N$  sur  $E$  est l'opération notée  $\Pi_E$  qui associe à tout vecteur  $X$  de  $\mathbb{R}^N$  le vecteur  $Y \in E$  de  $E$  qui est le plus proche de  $X$ , c'est à dire tel que  $\|X - Y\|$  est le plus petit possible.

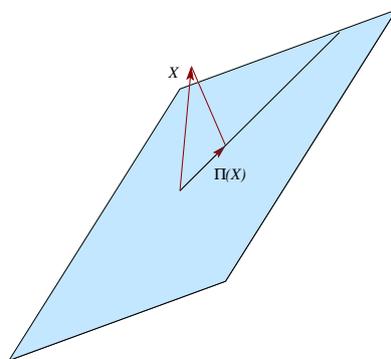


FIGURE 1.7 – Exemple de projection orthogonale de l'espace sur un plan. Le segment reliant  $X$  à son projeté  $\Pi(X)$  est orthogonal au plan.

Il est possible de démontrer qu'étant donné un vecteur  $X$  de  $\mathbb{R}^N$ , son projeté orthogonal  $Y = \Pi_E(X)$  sur un sous-espace vectoriel  $E \subset \mathbb{R}^N$  est l'unique vecteur de  $\mathbb{R}^N$  tel que  $X - \Pi_E(X)$  soit orthogonal à tout vecteur de  $E$  (d'où le nom de projection orthogonale) :

$$\forall Y \in E, X - \Pi_E(X) \perp Y, \quad \text{soit donc} \quad \forall Y \in E, (X - \Pi_E(X)) \cdot Y = 0.$$

Un exemple de projection orthogonale d'un vecteur de l'espace sur un plan se trouve en Figure 1.7.

Si une base  $\mathcal{V} = \{V_1, \dots, V_M\}$  de  $E$  est connue, le projeté orthogonal de  $X \in \mathbb{R}^N$  sur  $E$  s'écrit donc comme combinaison linéaire de cette base, sous la forme

$$\Pi_E(X) = \alpha_1 V_1 + \dots + \alpha_M V_M,$$

où les nombres  $\alpha_1, \dots, \alpha_M$  sont à déterminer. Ceci conduit généralement à un système linéaire, qu'on verra plus loin, sauf si la base  $\mathcal{V}$  est orthonormée. Dans ce cas les coefficients  $\alpha_1, \dots, \alpha_M$  sont facile à calculer, comme le montre le résultat suivant, qui est similaire au théorème 1.7.

**Théorème 1.8** Soit  $\mathcal{V} = \{V_1, \dots, V_M\}$  une base orthonormée d'un sous-espace vectoriel  $E$  de  $\mathbb{R}^N$ .

1. Le projeté orthogonal sur  $E$  de tout vecteur  $X \in \mathbb{R}^N$  s'écrit

$$\Pi_E(X) = (X \cdot V_1)V_1 + (X \cdot V_2)V_2 + \dots + (X \cdot V_M)V_M. \quad (1.13)$$

2. Pour tout vecteur  $X \in \mathbb{R}^N$  on a une formule de Pythagore-Parseval pour son projeté orthogonal sur  $E$

$$\|\Pi_E(X)\|^2 = (X \cdot V_1)^2 + (X \cdot V_2)^2 + \dots + (X \cdot V_M)^2. \quad (1.14)$$

Notons que l'orthonormalité de la base est importante dans cet énoncé. Si on a une base non orthonormée, il existe aussi une expression explicite pour la projection orthogonale, mais elle est plus complexe et demande de résoudre un système linéaire.

■ **Exemple 1.14 — Un plan particulier dans l'espace.** On considère les deux vecteurs de l'espace

$$F_1 = \left( \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T, \quad F_2 = \left( -\frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T.$$

Le calcul montre que la famille de deux vecteurs  $\mathcal{F} = \{F_1, F_2\}$  est orthonormée. Ces deux vecteurs constituent donc une base orthonormée du sous-espace qu'ils engendrent (qui est en fait le plan d'équation cartésienne  $y = z$ ).

Étant donné un vecteur  $X = (x, y, z)^T$  de l'espace, ses coordonnées dans la base  $\mathcal{F}$  de ce plan sont

$$\alpha_1 = X \cdot F_1 = \frac{x\sqrt{2} + y + z}{2}, \quad \text{et} \quad \alpha_2 = X \cdot F_2 = \frac{-x\sqrt{2} + y + z}{2}.$$

Le projeté orthogonal de  $X$  sur ce plan s'écrit quant à lui

$$\begin{aligned} \Pi(X) &= \alpha_1 F_1 + \alpha_2 F_2 \\ &= \frac{x\sqrt{2} + y + z}{2} \left( \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T + \frac{-x\sqrt{2} + y + z}{2} \left( -\frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T \\ &= \left( x, \frac{y+z}{2}, \frac{y+z}{2} \right)^T. \end{aligned}$$

■

#### 1.4.4 Espaces Hermitiens

Dans certaines situations il peut être plus simple de remplacer les nombres réels par des nombres complexes. Les espaces Hermitiens sont les équivalents des espaces Euclidiens, dans le cas où on se permet d'utiliser des nombres complexes.

**Nombres complexes**

Un nombre complexe s'écrit sous la forme

$$z = a + ib, \quad (1.15)$$

où  $a$  et  $b$  sont des nombres réels appelés respectivement partie réelle et partie imaginaire, et  $i$  est un nouveau nombre, dont la particularité est que son carré est négatif :

$$i^2 = -1. \quad (1.16)$$

Les règles de calcul usuelles sont modifiées pour tenir compte de la présence de  $i$ . L'addition de deux nombres complexes est obtenue par l'addition de leurs deux parties réelles et de leurs deux parties imaginaires

$$(a + ib) + (c + id) = (a + c) + i(b + d), \quad (1.17)$$

et leur multiplication est un peu plus compliquée : en développant

$$(a + ib)(c + id) = ac + iad + iab + i^2bd = (ac - bd) + i(ad + bc), \quad (1.18)$$

ce qui donne la règle de multiplication des complexes.

Le conjugué du nombre  $z = a + ib$  est le nombre complexe noté  $\bar{z}$  défini par

$$\bar{z} = a - ib. \quad (1.19)$$

Compte tenu de la nature du nombre  $i$ , le carré d'un nombre complexe n'est plus nécessairement un nombre positif ou nul. Par contre, le module carré d'un nombre complexe est quant à lui toujours positif ou nul, et joue donc le rôle du carré d'un nombre réel. Il est défini comme le produit du nombre  $z$  par son conjugué  $\bar{z}$ , et vaut donc

$$\text{si } z = a + ib, \quad |z|^2 = z\bar{z} = (a + ib)(a - ib) = a^2 + b^2,$$

qui est bien positif ou nul puisque  $a$  et  $b$  sont réels.

**Espaces Hermitiens**

Un espace Hermitien est l'espace vectoriel complexe  $\mathbb{C}^N$  des  $N$ -uplets de nombres complexes  $(z_1, \dots, z_N)^T$ , que l'on munit d'un produit scalaire adapté, défini par

$$(z_1, \dots, z_N)^T \cdot (\zeta_1, \dots, \zeta_N)^T = z_1 \bar{\zeta}_1 + z_2 \bar{\zeta}_2 + \dots + z_N \bar{\zeta}_N \quad (1.20)$$

(noter le complexe conjugué sur toutes les coordonnées  $\zeta_i$ ). Ceci garantit que la norme au carré d'un vecteur  $X = (x_1, \dots, x_N)^T \in \mathbb{C}^N$ , définie par son produit scalaire par lui-même, est bien un nombre positif ou nul

$$\|X\|^2 = x_1 \bar{x}_1 + x_2 \bar{x}_2 + \dots + x_N \bar{x}_N = |x_1|^2 + |x_2|^2 + \dots + |x_N|^2 \quad (1.21)$$

Toutes les notions vues plus haut (bases, orthogonalité, projections orthogonales,...) restent valables dans les espaces vectoriels complexes et les espaces Hermitiens, à condition d'utiliser le produit scalaire adapté.

**1.5 Exemples de vecteurs "structurés" : signaux**

Le terme "signal" est utilisé génériquement pour représenter un objet qui transmet une information. Par exemple, un son numérique est un signal, qui représente généralement une mesure de pression acoustique au cours du temps, donc mesurée en un certain nombre d'instants, généralement régulièrement espacés. Il s'agit donc d'un vecteur, dont la dimension est le nombre d'instants de mesure. Ce vecteur est dit structuré au sens où ses coordonnées dans la base canonique sont ordonnées (instants consécutifs), et une permutation de ces coordonnées produit un son qui n'a plus rien à voir, et a donc perdu son sens.

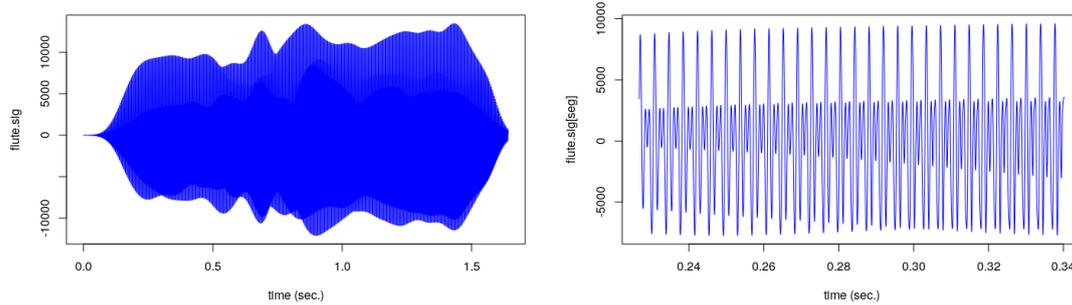


FIGURE 1.8 – Signal audio d'une note de flûte (gauche) et un segment plus court (droite)

### 1.5.1 Bases d'espaces de signaux

En utilisant les approches vues dans ce chapitre, on est naturellement amené à modéliser les signaux (de longueur fixée  $L$ ) comme des vecteurs d'un espace Euclidien (ou Hermitien) de dimension  $L$ , et les représenter par leurs coordonnées dans une base bien choisie. L'exemple qui suit aborde un cas très classique et important en traitement du signal, qui utilise la base de Fourier.

■ **Exemple 1.15 — Exemple d'un son d'instrument.** On considère ici un son provenant d'un enregistrement d'un son de flûte jouant une simple note. Le signal est enregistré à une fréquence d'échantillonnage de 44100 Hz (44100 échantillons par seconde), et on représente ici la voie gauche. Il s'agit d'un signal qui présente des oscillations rapides au cours du temps. Les quelques lignes de code qui suivent donnent les instructions permettant de charger un fichier son, et de tracer le signal correspondant. Les graphes se trouvent en Figure 1.8.



(Représentation d'un son par ses coordonnées dans la base de Fourier) On va utiliser le paquet `tuneR`, qui permet de manipuler et jouer des sons sous R.

```
require('tuneR')
flute <- readWave('sounds/flute.wav')
flute.sig <- flute@left
flute.fs <- flute@samp.rate
flute.siglength <- length(flute.sig)
t <- seq(0,flute.siglength-1)/flute.fs
plot(t,flute.sig,col='blue',type='l',xlab=('time (sec.)'))
```

Pour analyser de tels signaux on utilise traditionnellement des bases constituée de vecteurs présentant eux aussi des oscillations. Le prototype de ces bases est la base de Fourier. En notant  $X = (x_0, \dots, x_{L-1})^T$  le vecteur des échantillons<sup>3</sup>, on écrit

$$X = c_0 \varepsilon^{(0)} + c_1 \varepsilon^{(1)} + \dots + c_{L-1} \varepsilon^{(L-1)},$$

où les  $\varepsilon^{(k)} = (\varepsilon_0^{(k)}, \varepsilon_1^{(k)}, \dots, \varepsilon_{L-1}^{(k)})^T$  sont des sinusoïdes (complexes) de longueur  $L$

$$\varepsilon_\ell^{(k)} = \frac{1}{\sqrt{L}} e^{2i\pi k \ell / L} = \frac{1}{\sqrt{L}} \left[ \cos\left(\frac{2\pi k \ell}{L}\right) + i \sin\left(\frac{2\pi k \ell}{L}\right) \right]. \quad (1.22)$$

Le nombre  $k$  est appelé fréquence, et règle la vitesse des oscillations. Les nombres (complexes)  $c_k$  sont appelés coefficients de Fourier, ils fournissent une mesure de l'importance des sinusoïdes correspondantes dans le signal. Quelques exemples de sinusoïdes  $\varepsilon^{(k)}$  de différentes fréquences sont tracées en Figure 1.9.

3. Dans certaines applications et/ou langages informatiques, on numérote à partir de 0 plutôt que 1, ça n'est rien d'autre qu'une convention.

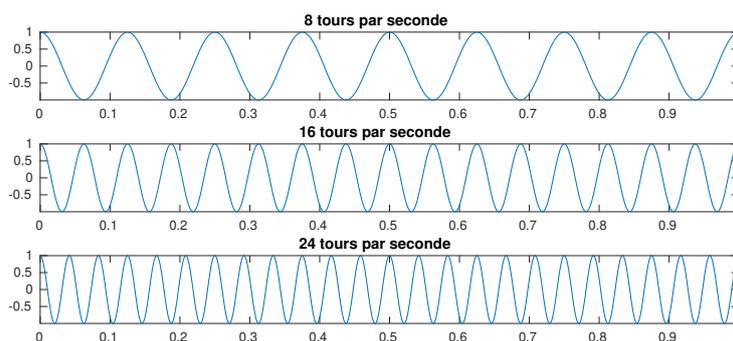


FIGURE 1.9 – Quelques exemples de sinusôides  $\varepsilon^{(k)}$  de différentes fréquences (seule la partie réelle est tracée)

**Théorème 1.9 — Base de Fourier.** La famille de vecteurs  $\{\varepsilon^{(k)}, k = 0, \dots, L-1\}$  est une base orthonormée de l'espace vectoriel complexe  $\mathbb{C}^L$ . En d'autres termes, tout  $x \in \mathbb{C}^L$  s'écrit comme

$$x_\ell = \frac{1}{\sqrt{L}} \sum_{k=0}^{L-1} \hat{x}_k e^{2i\pi k\ell/L},$$

où les nombres (complexes)  $\hat{x}_k, k = 0 \dots L-1$  constituent la transformée de Fourier de  $x$  et sont donnés par

$$\hat{x}_k = \frac{1}{\sqrt{L}} \sum_{\ell=0}^{L-1} x_\ell e^{-2i\pi k\ell/L}.$$

De plus, le théorème de Pythagore-Parseval s'écrit

$$\sum_{k=0}^{L-1} |\hat{x}_k|^2 = \sum_{\ell=0}^{L-1} |x_\ell|^2.$$

Le calcul effectif des coefficients de Fourier d'un vecteur utilise un algorithme rapide appelé Transformation de Fourier Rapide (*Fast Fourier Transform*, ou FFT en anglais). L'exemple qui suit donne une illustration sous R.

■ **Exemple 1.16 — Transformation de Fourier d'un son.** Le code R qui suit donne les détails du calcul de la transformée de Fourier du signal sonore tracé dans l'exemple 1.15. La fonction `fft` permet de calculer la transformée de Fourier, c'est à dire les coordonnées du vecteur dans la base de Fourier (la transformation de Fourier inverse utilise la fonction `ifft`).

La transformée de Fourier est un vecteur complexe, on ne trace ici que son module, en fonction de la fréquence représentée en abscisse dans les trois graphes de la Figure 1.10. On a utilisé ici la fréquence d'échantillonnage pour représenter les vraies valeurs de la fréquence, qui ont une signification physique. Par défaut la transformée de Fourier est un vecteur qui correspond aux fréquences comprises entre 0 et la fréquence d'échantillonnage `fs`.

Il s'avère que dans le cas d'un signal réel (comme le son de flûte, ou plus généralement tous les sons), la transformée de Fourier possède une symétrie particulière, appelée symétrie Hermitienne, par laquelle les valeurs de la transformée de Fourier entre `fs/2` et `fs` sont identiques, à une conjugaison complexe près (qui ne se voit pas sur le module) aux valeurs dans l'intervalle de fréquences comprises entre `-fs/2` et 0. La fonction `fftshift` permet d'exploiter cette symétrie et de représenter la transformée de Fourier de façon symétrique par rapport à la fréquence nulle, ce qui est plus naturel (graphe en haut à droite). La symétrie fait aussi que les fréquences négatives peuvent être ignorées car elles n'apportent pas d'information supplémentaire. On peut ainsi ne représenter que les fréquences positives, ce qui est fait dans le graphe du bas de la Figure 1.10, ou on a de plus représenté le logarithme de  $|\hat{x}|$  plutôt que  $|\hat{x}|$ , ce qui est souvent

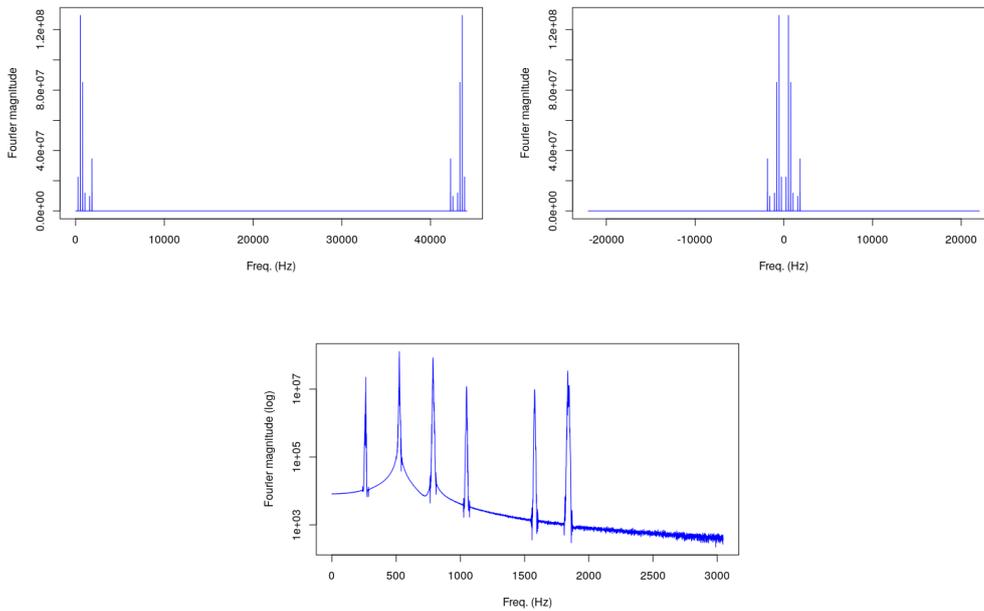


FIGURE 1.10 – Modules des coefficients de Fourier du signal de flûte : fréquences positives (haut gauche), centré sur la fréquence nulle (haut droite), zoom sur les basses fréquences, en représentation log (bas).

plus pertinent, et on s'est restreint à une plage de fréquences réduite pour mieux visualiser les pics de la transformée de Fourier.

On y voit notamment que les pics apparaissent à des valeurs de la fréquence qui sont multiples d'une fréquence fondamentale, ce qui est une propriété caractéristique de tels signaux (note tenue d'instrument de musique).



#### (Calcul et représentation des coefficients de Fourier)

Ces quelques lignes utilisent des fonctions fournies par le paquet `pracma`, qu'il faut donc charger avant de les exécuter.

```
require('pracma')
flute.fourier <- fft(flute.sig)
frequencies <- (0:(flute.siglength-1))/flute.siglength*flute.fs
plot(frequencies,abs(flute.fourier),type='l',
     xlab='Freq. (Hz)',ylab='Fourier magnitude', col='blue')
centered.frequencies <- frequencies-flute.fs/2
plot(centered.frequencies,fftshift(abs(flute.fourier)),type='l',
     xlab='Freq. (Hz)',ylab='Fourier magnitude', col='blue')
plot(frequencies[1:5000],abs(flute.fourier[1:5000]),type='l',
     col='blue',log='y',xlab='Freq. (Hz)',ylab='Fourier magnitude (log)')
```

■

Il existe de nombreux choix possibles de bases pour des espaces de signaux. On peut en particulier citer des bases appelées MDCT (*modified discrete cosine transform*), qui au lieu de sinusoides font intervenir des "petits morceaux de sinusoides", qui portent à la fois une information fréquentielle et une information temporelle. Ces bases sont notamment utilisées dans les standards de compression audio tels que MP3, OGG, ...

Les images constituent un autre exemple de données structurées. Le standard JPEG de compression d'images utilise soit des bases DCT (*discrete cosine transform*) bidimensionnelles, soit des bases appelées bases d'ondelettes, qui conduisent à des résultats un peu meilleurs en termes de compression d'images.

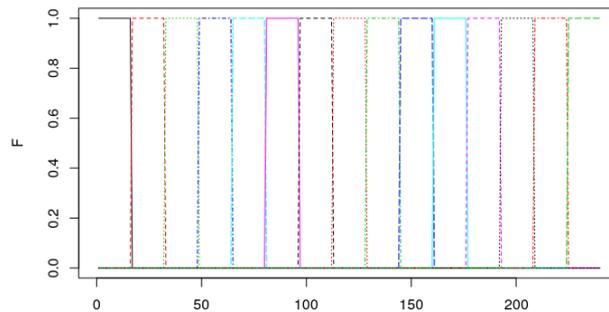


FIGURE 1.11 – Signaux constituant une base d'un sous-espace de signaux constants par morceaux.

### 1.5.2 Projection sur des sous-espaces de signaux

Les techniques de projection orthogonale que nous avons vues précédemment peuvent être utilisées dans le contexte des signaux pour en construire des approximations, plus simples. Par exemple, étant donné un signal, on peut l'approximer en le remplaçant par un signal proche, et plus simple. Un exemple d'approximation par un signal plus simple est celui d'une approximation par un signal qui est constant sur des intervalles de longueur constante. L'ensemble de tels signaux forme un sous-espace de  $\mathbb{R}^L$ , qui peut être engendré par une base comme le montre l'exemple ci-dessous.

■ **Exemple 1.17 — Projection sur des espaces de signaux constants par morceaux.** Ici on définit un sous-espace d'un espace de signaux  $\mathbb{R}^L$  par une base constituée de signaux égaux à 1 sur un intervalle et 0 en dehors. Ces signaux sont représentés (avec des couleurs différentes) en Figure 1.11.

Le code ci-dessous construit les vecteurs de base et les assemble dans une matrice.



```
N <- 240
M <- 15
K <- N/M
F <- matrix(numeric(M*N),nrow=N)

for (m in 1:M) {
  goodinds <- (m-1) * K + (0:(K-1)) + 1
  F[goodinds,m] <- 1
}
```

Les détails du programme ont peu d'importance.  $N$  est la longueur des signaux (la dimension de l'espace vectoriel considéré),  $M$  est le nombre de vecteurs de base (la dimension du sous-espace), et  $K$  est donc la longueur des intervalles.

Ensuite, étant donné un signal à approximer (ici un signal EEG dans un protocole impliquant des potentiels évoqués), noté  $x$ , de même dimension que les vecteurs de base, on calcule son projeté sur le sous-espace.



```
y = F %*% solve((t(F)%*%F), t(F)%*%x)
y <- as.vector(y)
plot(x,type='l',col='blue')
lines(y,col='red')
```

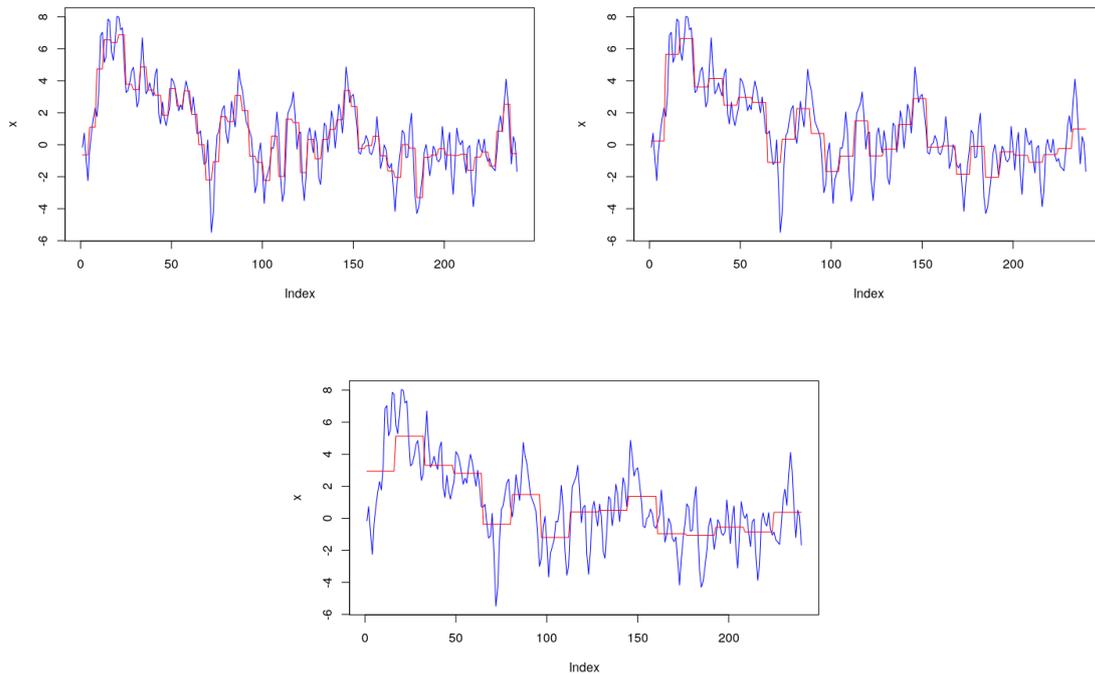


FIGURE 1.12 – Approximations constantes par morceaux d'un signal EEG, à des précisions décroissantes (de gauche à droite et haut en bas :  $M=60$ ,  $M=30$  et  $M=15$ )

En augmentant la dimension  $M$  du sous-espace d'approximation (et donc en diminuant  $K$ ), on améliore la précision de l'approximation. ■

### 1.5.3 Parenthèse : spectrogramme

Dans certaines applications, notamment en analyse et traitement du signal de parole, la base de Fourier n'est pas suffisante pour donner une représentation pertinente des signaux d'intérêt. En effet, les vecteurs de base sont des sinusoides de même durée que le signal lui-même. Si un signal est constitué par exemple de deux notes de musique jouées l'une après l'autre, il sera possible à partir des coefficients de Fourier de déterminer quelles sont les notes qui ont été jouées, mais pas à quel moment. L'information temporelle a été perdue en passant dans le domaine des fréquences.

Pour pallier cette difficulté, on introduit alors une variante de l'analyse de Fourier, basée sur des vecteurs de base qui sont des versions "fenêtrées" des sinusoides : des sinusoides multipliées par un vecteur localisé au voisinage d'un instant donné (que l'on fait ensuite varier). On appelle ces vecteurs atomes temps-fréquence, ou atomes de Gabor, en référence à D. Gabor<sup>4</sup>.

Un atome temps-fréquence  $g^{(m,n)} \in \mathbb{C}^L$  de longueur  $L$  prend donc la forme

$$g_\ell^{(m,n)} = g_{\ell-na} e^{2i\pi m b \ell / L},$$

où  $g \in \mathbb{R}^L$  est la fenêtre, et  $a$  et  $b$  sont respectivement des pas d'échantillonnage temporel et fréquentiel. L'indice  $m$  est un indice fréquentiel, et l'indice  $n$  est un indice temporel.

Ceci conduit à la transformée de Fourier à fenêtre glissante, ou transformée de Gabor. Étant donné un signal  $x \in \mathbb{R}^L$ , sa transformée de Gabor est un tableau à deux entrées (une matrice donc)  $V_g x \in \mathbb{C}^{M \times N}$ ,

4. Récipiendaire du prix Nobel de Physique pour l'invention de l'holographie, et l'un des fondateurs de la théorie de la communication.

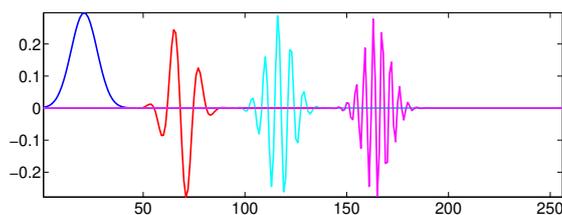


FIGURE 1.13 – Quelques exemples d'atomes de Gabor. La fenêtre (à gauche), et quelques atomes temps-fréquence obtenus en décalant la fenêtre le long de l'axe temporel, et en la multipliant par des sinusôides de différentes fréquences.

complexe, défini par produits scalaires de  $x$  avec les atomes de Gabor

$$(V_g x)_{m,n} = x \cdot g^{(m,n)} = \sum_{\ell=0}^{L-1} x_\ell g_{\ell-na} e^{-2i\pi m b \ell / L}.$$

A l'instar de la transformée de Fourier, la transformée de Gabor est à valeurs complexes, on préfère en représenter le module, appelé spectrogramme  $|V_g x|$ , sous forme d'une image dans laquelle les valeurs de  $|V_g x|$  sont encodées par des niveaux de gris ou un code couleur.

Un exemple est représenté en Figure 1.14, où apparaissent les représentations temporelle et fréquentielle d'un signal de parole de 5 secondes (haut) et le spectrogramme. La représentation fréquentielle est peu explicite, alors que la représentation temporelle donne une information partielle sur les instants d'apparition des phonèmes. Le spectrogramme donne une représentation différente, qui montre le déploiement des fréquences au cours du temps. Cette information est généralement plus interprétable par des spécialistes, mais elle est aussi utilisable par des systèmes experts tels que des algorithmes de reconnaissance de parole ou d'identification de locuteur.

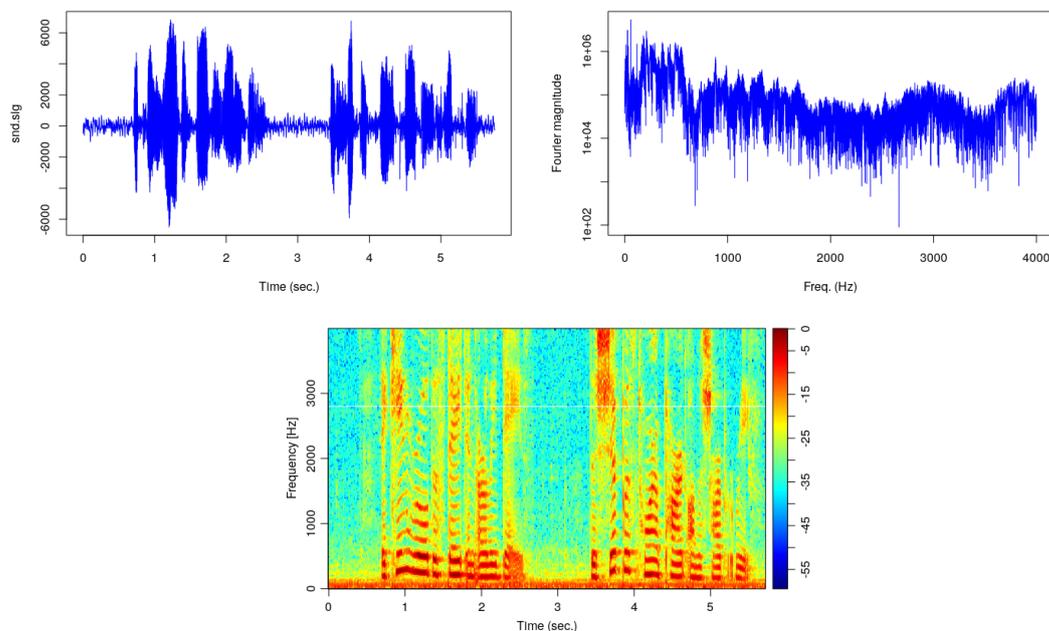


FIGURE 1.14 – Signal de parole : représentation temporelle (haut gauche), représentation fréquentielle (spectre, haut droite) et représentation temps-fréquence (spectrogramme, bas)

## 1.6 Résumé

Qu'avons nous vu dans ce chapitre ? Ci-dessous un résumé, qui reprend aussi l'essentiel des aspects à retenir.

- **Interprétation géométrique d'un tableau de nombres :** chaque colonne (ou chaque ligne) peut être interprétée comme les coordonnées d'un point dans un espace de grande dimension.
- *Lien entre géométrie et algèbre :* en passant de la notion de point à la notion de vecteur (ce qui n'est pas difficile, car on peut associer à chaque point  $P$  le vecteur  $\overrightarrow{OP}$ , dès qu'on a fait le choix d'une origine  $O$ ), on se donne la possibilité d'utiliser des notions algébriques : addition de vecteurs, multiplication d'un vecteur par un nombre,...
- **Autres notions métriques :**
  - La norme d'un vecteur décrit sa longueur : en reprenant la terminologie ci-dessus, si  $\overrightarrow{OP}$  est un représentant du vecteur, sa norme est la distance entre l'origine  $O$  et le point  $P$ .
  - Le produit scalaire de deux vecteurs dépend de leurs normes respectives, et de l'angle entre ces vecteurs.
  - Dans un espace de dimension  $N$ , la valeur absolue du déterminant de  $N$  vecteurs donne l'hyper-volume de l'hyper-parallélépipède engendré par ces  $N$  vecteurs.
- **Base de  $\mathbb{R}^N$  :**
  - une base de  $\mathbb{R}^N$  est constituée de  $N$  vecteurs exactement. Tout vecteur de  $\mathbb{R}^N$  admet un et un seul développement comme combinaison linéaire de la base.
  - Une famille de  $N$  vecteurs de  $\mathbb{R}^N$  est une base de  $\mathbb{R}^N$  si et seulement si son déterminant est différent de zéro.
- **Sous-espace vectoriel :**
  - Un sous-ensemble  $E$  de  $\mathbb{R}^N$  est un sous-espace vectoriel de  $\mathbb{R}^N$  si il est stable par addition et multiplication par un nombre.
  - Une famille de  $M$  vecteurs d'un sous-espace  $E$  de  $\mathbb{R}^N$  est une base de  $E$  si tout vecteur de  $E$  admet un et un seul développement comme combinaison linéaire de cette famille.  $E$  est alors de dimension  $M$  (et nécessairement  $M \leq N$ ).
  - Étant donnée une famille de  $K$  vecteurs de  $\mathbb{R}^N$ , elle engendre (par combinaisons linéaire) un sous-espace de  $\mathbb{R}^N$ . La dimension  $M$  de ce sous-espace est appelée rang de la famille de vecteurs. On a nécessairement  $M \leq K$ . Si  $M = K$ , la famille est de rang plein.
- **Orthogonalité, projection orthogonale :**
  - Deux vecteurs  $U, V$  de  $\mathbb{R}^N$  sont orthogonaux si leur produit scalaire est nul :  $U \cdot V = 0$ .
  - Un vecteur  $V$  est orthogonal à un sous-espace  $E$  si il est orthogonal à tous vecteurs de  $E$ . De façon équivalente,  $V$  est orthogonal à  $E$  si il est orthogonal à tous les vecteurs d'une base de  $E$ , quelle que soit celle-ci.
  - Étant donné un sous-espace  $E$  de  $\mathbb{R}^N$ , et un vecteur  $V \in \mathbb{R}^N$ , le projeté orthogonal de  $V$  sur le sous-espace  $E$  est le vecteur  $W \in E$  qui est à distance minimale de  $V$  : il minimise la norme  $\|V - W\|$ .
  - $V - W$  est alors orthogonal à  $E$  (de façon équivalente,  $V - W$  est orthogonal à tout vecteur d'une base de  $E$ , quelle que soit celle-ci).

## 2. Transformations linéaires, calcul matriciel

Nous avons déjà rencontré des exemples de matrices, sans les définir vraiment. Par exemple, en concaténant des vecteurs nous avons obtenu des matrices dont on a dit qu'elles pouvaient permettre de résoudre des systèmes linéaires. On va voir que les matrices sont un outil fondamental en algèbre linéaire, et qu'elles permettent de simplifier et systématiser un certain nombre d'opérations.

Nous allons aussi voir que les matrices peuvent, dans certains cas, être considérées comme des généralisations des nombres, au sens où on peut définir des opérations similaires aux opérations que nous effectuons avec des nombres.

### 2.1 Matrices, calcul matriciel

**Définition 2.1 — Matrices.** Étant donnés deux entiers positifs  $M, N$ , une matrice à  $M$  lignes et  $N$  colonnes est un tableau de nombres que l'on écrit

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix},$$

où les nombres  $a_{mn} \in \mathbb{R}$ , avec  $m = 1, \dots, M$  et  $n = 1, \dots, N$  sont appelés éléments de matrice. On note  $\mathcal{M}_{MN}(\mathbb{R})$ , l'ensemble des matrices à  $M$  lignes et  $N$  colonnes à coefficients réels, et plus simplement  $\mathcal{M}_N(\mathbb{R})$  l'ensemble des matrices "carrées"  $N \times N$  à coefficients réels.

■ **Exemple 2.1 — Matrice d'une famille de vecteurs.** Étant donnée une famille de vecteurs  $\{F_1, \dots, F_M\}$  de  $\mathbb{R}^N$ , la concaténation de ces vecteurs forme une matrice  $F$  à  $N$  lignes et  $M$  colonnes, appelée *Matrice de la famille de vecteurs*. On va voir que les propriétés de cette matrice permettent de caractériser certaines propriétés géométriques de la famille de vecteurs. ■

#### 2.1.1 Addition, transposition

Comme on l'a fait pour les vecteurs, il est possible de définir dans  $\mathcal{M}_{M,N}(\mathbb{R})$  les opérations usuelles, à savoir addition et multiplication scalaire, de la façon suivante : si  $A, B \in \mathcal{M}_{M,N}(\mathbb{R})$ , la matrice  $A + B$  est

définie par

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1N} \\ b_{21} & b_{22} & \dots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M1} & b_{M2} & \dots & b_{MN} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1N} + b_{1N} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2N} + b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} + b_{M1} & a_{M2} + b_{M2} & \dots & a_{MN} + b_{MN} \end{pmatrix},$$

et pour tout  $\lambda \in \mathbb{R}$ , on définit la matrice  $\lambda A$  par

$$\lambda \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix} = \begin{pmatrix} \lambda a_{11} & \lambda a_{12} & \dots & \lambda a_{1N} \\ \lambda a_{21} & \lambda a_{22} & \dots & \lambda a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda a_{M1} & \lambda a_{M2} & \dots & \lambda a_{MN} \end{pmatrix}.$$

**Définition 2.2 — Transposition des matrices.** Soit  $A \in \mathcal{M}_{M,N}(\mathbb{R})$  une matrice. La transposée de  $A$  est la matrice  $A^T \in \mathcal{M}_{N,M}(\mathbb{R})$  définie en permutant les rôles des lignes et des colonnes :

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix}^T = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{M1} \\ a_{12} & a_{22} & \dots & a_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1N} & a_{2N} & \dots & a_{MN} \end{pmatrix}.$$

La matrice  $A$  est dite *symétrique* si elle est égale à sa transposée :  $A^T = A$ . Une matrice symétrique est nécessairement carrée.

### 2.1.2 Produit d'une matrice carrée par un vecteur

L'ingrédient supplémentaire est le produit matriciel, que l'on voit tout d'abord sous forme de produit matrice  $\times$  vecteur, dans  $\mathbb{R}^2$ . Étant donné  $V = \begin{pmatrix} x \\ y \end{pmatrix}$  et la matrice carrée  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ , le produit  $AV$  est le vecteur

$$AV = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}.$$

■ **Exemple 2.2 — Transformations classiques dans le plan et matrices.** — La plus simple et la plus importante est la *matrice identité*, ici

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

On voit facilement que pour tout vecteur  $V = (x, y) \in \mathbb{R}^2$ , on a  $I_2 V = V$ . La matrice identité est associée à une transformation... qui ne transforme rien.

— *Projections orthogonales sur les deux axes* : on considère les deux matrices  $\Pi_1$  et  $\Pi_2$  suivantes

$$\Pi_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \Pi_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Le calcul montre que pour tout vecteur  $V = (x, y)^T$ , on a

$$\Pi_1 V = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix}, \quad \Pi_2 V = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix}.$$

Ces deux matrices associent donc à un vecteur du plan ses deux projetés orthogonaux sur les axes de coordonnées. On peut construire de même la matrice  $2 \times 2$  effectuant la projection orthogonale du plan sur n'importe quelle droite vectorielle.

— *Permutation* : La matrice

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

permuté les coordonnées :

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix} .$$

— *Rotations du plan* : Il est possible d'utiliser des matrices pour effectuer des rotations dans le plan. Par exemple, pour tout vecteur  $(x, y)^T$  du plan, le calcul donne

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -y \\ x \end{pmatrix} .$$

En représentant les vecteurs  $(x, y)^T$  et  $(-y, x)^T$  dans le plan, on peut voir facilement que le second n'est autre que le premier que l'on a fait tourner de 90 degrés ( $\pi/2$ ) dans le sens trigonométrique (le sens inverse des aiguilles d'une montre).

Plus généralement, étant donné un nombre réel  $\theta$ , la matrice

$$R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

effectue une rotation d'angle  $\theta$  : pour tout vecteur  $V = (x, y)^T$

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \end{pmatrix} .$$

On voit en FIGURE 2.1 l'action de la matrice de rotation sur un vecteur, qui fait lui-même un angle  $\alpha$  avec l'axe des abscisses.

— *Matrices diagonales* : une *matrice diagonale* est une matrice dont seuls les éléments situés sur la diagonale sont non nuls. Le produit d'une matrice diagonale par un vecteur a pour effet de multiplier les coordonnées du vecteur par les éléments diagonaux de la matrice. Par exemple,

$$\begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2x \\ 3y \end{pmatrix}$$

■

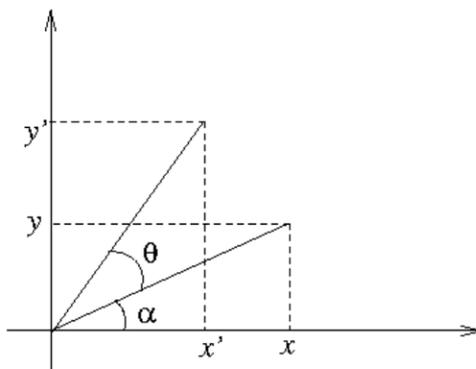


FIGURE 2.1 – Rotation d'angle  $\theta$  autour de l'origine du plan.

Cette notion s'étend au cas de matrices carrées et vecteurs d'autres dimensions

**Définition 2.3 — Produit matrice  $\times$  vecteur.** Soit  $A \in \mathcal{M}_N(\mathbb{R})$  une matrice carrée et  $X = (x_1, \dots, x_N)^T \in \mathbb{R}^N$  un vecteur. Le produit  $AX$  est le vecteur de  $\mathbb{R}^N$  défini par

$$AX = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N \\ \vdots \\ a_{N1}x_1 + a_{N2}x_2 + \dots + a_{NN}x_N \end{pmatrix}.$$

Les coordonnées de  $AX$  sont donc les produits scalaires des lignes de  $A$  par les colonnes de  $X$ . Ce sera la règle à retenir : **produit ligne par colonne**.

■ **Exemple 2.3 — Transformations dans l'espace.** Les transformations de l'espace sont décrites par des matrices  $3 \times 3$ .

— *Matrice identité* : La matrice

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

est appelée matrice identité du plan, elle joue le même rôle dans  $\mathbb{R}^3$  que le nombre 1 dans  $\mathbb{R}$  : elle est l'élément neutre pour la multiplication. Pour tout vecteur  $X \in \mathbb{R}^3$  on voit facilement que  $I_3X = X$ . Plus généralement les matrices de la forme

$$\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix},$$

où  $a, b, c$  sont des nombres quelconques, sont appelées *matrices diagonales*.

— *Projections orthogonales* : comme dans le plan, on peut construire des matrices de projection orthogonale. Par exemple les matrices définies ci-dessous effectuent respectivement des projections orthogonales de l'espace sur le premier axe, le second axe, et le plan engendré par les deux premiers axes :

$$\Pi_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \Pi_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \Pi_{1,2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Par exemple,

$$\Pi_{1,2} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}.$$

Là encore, on peut associer une matrice  $3 \times 3$  à tout projecteur sur un sous-espace vectoriel de  $\mathbb{R}^3$ . Ces matrices sont appelées *projecteurs*. ■

**Remarque 2.1 — Formulation matricielle d'un système linéaire.** D'après la définition du produit ci-dessus, on voit qu'un système linéaire à  $N$  équations et  $N$  inconnues peut s'écrire sous forme d'une équation matricielle. Par exemple le système de 3 équations à 3 inconnues

$$\begin{cases} x + 2y + 3z = a \\ 4x + 5y + 6z = b \\ 7x + 8y + 9z = c \end{cases}$$

où  $a, b, c$  sont trois paramètres, peut s'écrire sous la forme d'une équation matricielle

$$AX = V,$$

où  $X = (x, y, z)^T$  est le vecteur inconnu,  $V = (a, b, c)^T$  est le second membre supposé connu, et  $A$  est la

matrice

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

Si ces quantités étaient des nombres on écrirait la solution  $X = V/A$ , malheureusement cette expression n'a pas de sens pour le produit matriciel. Une meilleure solution serait de multiplier (à gauche) les deux membres de cette équation matricielle par l'inverse  $A^{-1}$  de  $A$  s'il existe, ce qui donnerait la solution  $X = A^{-1}V$ . Ce sera la démarche à suivre, qui malheureusement ne fonctionne pas ici car  $A^{-1}$  n'existe pas. On verra que ceci équivaut au fait que ce système n'a pas de solution.

### 2.1.3 Produit matriciel : produit de matrices carrées, inversion

Nous savons maintenant comment multiplier une matrice par un vecteur. Supposons maintenant que nous ayons deux matrices  $A, A' \in \mathcal{M}_2(\mathbb{R})$ ; prenons un vecteur  $X \in \mathbb{R}^2$ , et calculons

$$\begin{aligned} A'[AX] &= \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \left[ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right] = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix} \\ &= \begin{pmatrix} a'(ax+by) + b'(cx+dy) \\ c'(ax+by) + d'(cx+dy) \end{pmatrix} \\ &= \begin{pmatrix} (a'a + b'c)x + (a'b + b'd)y \\ (c'a + d'c)x + (c'b + d'd)y \end{pmatrix} \\ &= \begin{pmatrix} a'a + b'c & a'b + b'd \\ c'a + d'c & c'b + d'd \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \end{aligned}$$

Ceci suggère de définir une matrice  $B = A'A$ , produit des matrices  $A'$  et  $A$ , de la façon suivante :

$$A'A = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a'a + b'c & a'b + b'd \\ c'a + d'c & c'b + d'd \end{pmatrix}.$$

On observe que chacun des éléments de la matrice produit est obtenu en prenant le produit coefficient d'une ligne de  $A'$  par une colonne de  $A$ , et en sommant ensuite les nombres ainsi obtenus. On retrouve encore le **produit ligne par colonne** déjà rencontré plus haut.

■ **Exemple 2.4** Calculons le produit

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix} = \begin{pmatrix} 19 & 21 \\ 43 & 47 \end{pmatrix}.$$

On définit de la même façon le produit de deux matrices  $3 \times 3$ . Soient  $A$  et  $B$  les matrices d'éléments de matrice respectifs  $a_{ij}$  et  $b_{ij}$  :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}.$$

Le produit de  $A$  par  $B$  est la matrice  $C = AB$ , dont l'élément de matrice  $c_{ij}$  est défini par

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j} = \sum_{k=1}^3 a_{ik}b_{kj}.$$

Quelle est l'interprétation de cette formule : on voit que la sommation s'effectue sur le second indice de la matrice  $A$ , donc l'indice des colonnes de  $A$ , et le premier indice de  $B$ , donc l'indice de ligne. Ainsi, dans le calcul de  $c_{ij}$ , la sommation s'effectue en parcourant la  $i$ -ème ligne de  $A$  et la  $j$ -ème colonne de  $B$ . Bien entendu, cette sommation n'est possible que parce que les lignes de  $A$  et les colonnes de  $B$  ont le même nombre d'éléments.

Plus généralement, le produit de deux matrices carrées de même taille (quelconque) est défini de même :

**Définition 2.4 — Produit de matrices carrées.** Soient  $A, B \in \mathcal{M}_N(\mathbb{R})$  deux matrices carrées de même dimension. Le produit  $C = AB$  est bien défini, et donne encore une matrice carrée  $N \times N$  définie par

$$\begin{cases} c_{11} &= a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \cdots + a_{1N}b_{N1}, \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} + \cdots + a_{1N}b_{N2}, \\ &\dots \\ c_{1N} &= a_{11}b_{1N} + a_{12}b_{2N} + a_{13}b_{3N} + \cdots + a_{1N}b_{NN}, \\ c_{21} &= a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} + \cdots + a_{2N}b_{N1}, \\ c_{22} &= a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} + \cdots + a_{2N}b_{N2}, \\ &\dots \\ c_{NN} &= a_{N1}b_{1N} + a_{N2}b_{2N} + a_{N3}b_{3N} + \cdots + a_{NN}b_{NN}, \end{cases}$$

ce que l'on synthétise sous la forme suivante : pour tous  $m, n = 1, \dots, N$ ,

$$c_{mn} = a_{m1}b_{1n} + a_{m2}b_{2n} + \cdots + a_{mN}b_{Nn} = \sum_{k=1}^N a_{mk}b_{kn}.$$



### (Produit matriciel et produit de Hadamard)

Sous R, le produit  $*$  appliqué à des matrices de même taille représente le produit point par point (aussi appelé *produit de Hadamard*), qui n'est donc pas la même chose que le produit matriciel. Le produit matriciel utilise la notation  $\%*\%$ . Par exemple, la multiplication ci-dessus s'écrit

```
> A <- matrix(1:4, ncol=2, byrow=T)
> B <- matrix(5:8, ncol=2, byrow=T)
> C <- A %*% B
```

■ **Exemple 2.5 — Rotation du plan.** On a déjà vu que la matrice associée à une rotation d'angle  $\theta$  dans le plan est la matrice

$$R_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Si on fait deux rotations consécutives d'angles  $\theta$  et  $\theta'$ , la matrice correspondant à cette composition est donc le produit de deux matrices de rotation

$$\begin{aligned} R_\theta R_{\theta'} &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \cos(\theta') & -\sin(\theta') \\ \sin(\theta') & \cos(\theta') \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta)\cos(\theta') - \sin(\theta)\sin(\theta') & -\cos(\theta)\sin(\theta') - \sin(\theta)\cos(\theta') \\ \cos(\theta)\sin(\theta') + \sin(\theta)\cos(\theta') & \cos(\theta)\cos(\theta') - \sin(\theta)\sin(\theta') \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta + \theta') & -\sin(\theta + \theta') \\ \sin(\theta + \theta') & \cos(\theta + \theta') \end{pmatrix} \\ &= R_{\theta + \theta'}. \end{aligned}$$

Il n'est pas surprenant de retrouver la matrice de rotation d'angle  $\theta + \theta'$ . ■

**Remarque 2.2** Il faut noter que le produit matriciel ainsi défini n'est pas commutatif. Par exemple,

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{alors que} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

■ **Exemple 2.6 — Rotation de l'espace et coordonnées sphériques/angulaires.** On est parfois amené à représenter des points de l'espace par leurs coordonnées sphériques : une coordonnée radiale et deux coordonnées angulaires. Ces dernières correspondent en fait à des rotations. Plus précisément, notons

$$R_\theta^{(z)} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R_\varphi^{(x)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix}.$$

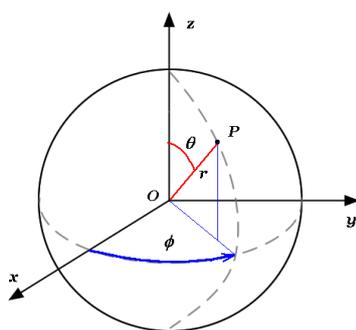


FIGURE 2.2 – Coordonnées sphériques

les matrices de rotation de l'espace correspondant respectivement à une rotation d'angle  $\theta$  dans le plan  $xOy$  (perpendiculaire à l'axe  $Oz$ ) et une rotation d'angle  $\varphi$  dans le plan  $yOz$  (perpendiculaire à l'axe  $Ox$ ). Si on applique ces deux matrices au vecteur (colonne) de coordonnées  $(r, 0, 0)$ , on obtient

$$R_{\varphi}^{(x)} R_{\theta}^{(z)} \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix} = R_{\varphi}^{(x)} \begin{pmatrix} r \cos(\theta) \\ r \sin(\theta) \\ 0 \end{pmatrix} = \begin{pmatrix} r \cos(\theta) \\ r \sin(\theta) \cos(\varphi) \\ r \sin(\theta) \sin(\varphi) \end{pmatrix},$$

qui redonne donc les coordonnées sphériques. ■

On voit donc que les matrices carrées ont un statut particulier. On peut déjà noter que deux matrices carrées  $A$  et  $B$  de même taille sont toujours compatibles, et donc multipliables, quel que soit l'ordre :  $AB$  et  $BA$  existent et ont également la même taille que  $A$  et  $B$ . Les matrices carrées sont donc des généralisations naturelles des nombres que nous utilisons quotidiennement :

- On peut les ajouter,... et le soustraire
- On peut les multiplier par un nombre
- On peut les multiplier entre elles

La matrice identité joue un rôle fondamental :

**Définition 2.5 — Matrice identité.** En dimension  $N$ , la matrice identité, notée  $I_N$ , est la matrice carrée

$$I_N = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Un calcul simple montre en effet que pour toute matrice  $A \in \mathcal{M}_N(\mathbb{R})$ , on a

$$I_N A = A I_N = A.$$

La matrice identité  $I_N$  est donc l'**élément neutre** pour la multiplication matricielle dans  $\mathcal{M}_N(\mathbb{R})$ . Dans l'analogie avec les nombres, elle est donc l'équivalent du nombre 1.

De même, l'inverse d'un nombre  $x \neq 0$  est le nombre  $y$  tel que  $xy = 1$ . On va définir de même l'inverse  $A^{-1}$  d'une matrice carrée  $A$ . L'inverse d'une matrice carrée doit être une matrice carrée de même taille. Mais il n'existe pas toujours, ce qui oblige à introduire la notion d'*inversibilité d'une matrice*.

**Définition 2.6 — Inversibilité.** Une matrice carrée  $A \in \mathcal{M}_N(\mathbb{R})$  est inversible si il existe une matrice  $B \in \mathcal{M}_N(\mathbb{R})$  telle que  $AB = I_N = BA$ . On note alors  $B = A^{-1}$ .

Il est en fait possible de démontrer qu'une seule de ces deux égalités est nécessaire, l'autre en étant une conséquence directe.

**Théorème 2.3** Soit  $A \in \mathcal{M}_N(\mathbb{R})$ . Si il existe une matrice  $B \in \mathcal{M}_N(\mathbb{R})$  telle que  $BA = I_N$  (ou  $AB = I_N$ ), alors  $A$  est inversible, et  $B = A^{-1}$ .

**Remarque 2.4 — Interdiction formelle!** . La notation

$$A^{-1} = \frac{1}{A}$$

n'a aucun sens pour les matrices, et est donc interdite 

Il existe des techniques permettant de calculer à la main l'inverse d'une matrice, on ne s'en occupera pas ici et on privilégiera le calcul numérique.

■ **Exemple 2.7** On considère les matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}.$$

On a déjà vu que  $A$  n'est pas inversible, mais on va voir grâce à R que  $B$  l'est, et que son inverse est donné par

$$B^{-1} = \frac{1}{3} \begin{pmatrix} -2 & -2 & 3 \\ -4 & 11 & -6 \\ 3 & -6 & 3 \end{pmatrix}.$$



### (Inversion des matrices carrées)

La fonction `solve`, déjà vue pour la résolution de systèmes linéaires, peut aussi être utilisée pour inverser les matrices.

```
> A <- matrix(1:9,ncol=3)
> det(A)
[1] 0
> B <- matrix(c(1,2,3,4,5,6,7,8,10),ncol=3)
> det(B)
[1] -3
> solve(B)
      [,1]      [,2] [,3]
[1,] -0.6666667 -0.6666667  1
[2,] -1.3333333  3.6666667 -2
[3,]  1.0000000 -2.0000000  1
> 3*solve(B)
      [,1] [,2] [,3]
[1,]   -2   -2   3
[2,]   -4   11  -6
[3,]    3   -6   3
```

Dans l'exemple précédent, on a aussi calculé les  $\det(A)$  et  $\det(B)$ , les déterminants des matrices  $A$  et  $B$ . On a vu que  $A$  n'est pas inversible, et que son déterminant est nul, alors que  $B$  est inversible et son déterminant est non-nul. C'est tout sauf une coïncidence!

Commençons par définir le déterminant d'une matrice carrée (nous n'avons vu jusqu'à présent que le déterminant de  $N$  vecteurs de  $\mathbb{R}^N$ ): le déterminant  $\det(A)$  d'une matrice carrée  $A \in \mathcal{M}_N$  n'est autre que le déterminant des  $N$  vecteurs constitués par ses colonnes<sup>1</sup>. Le résultat qui suit est fondamental... et très utile!

1. Notons que c'est ainsi que la fonction `det` de R est définie, sur des matrices et pas des vecteurs).

**Théorème 2.5 — Déterminant, inversibilité et base.** 1. Une matrice carrée  $A \in \mathcal{M}_N(\mathbb{R})$  est inversible si et seulement si son déterminant est non-nul.  
 2. Une matrice carrée  $A \in \mathcal{M}_N(\mathbb{R})$  est inversible si et seulement si ses colonnes forment une base de  $\mathbb{R}^N$ .

Comme de plus on montre que

$$\det(A^T) = \det(A)$$

$\det(A) \neq 0$  implique aussi que les lignes de  $A$  forment une base de  $\mathbb{R}^N$ .

**Remarque 2.6** 1. On peut considérer que les matrices sont en quelque sorte une généralisation des nombres, puisque l'on peut les additionner, les soustraire et les multiplier. La première différence de marque entre nombres et matrices tient dans le fait que le produit matriciel n'est pas commutatif en général, contrairement au produit des nombres. Une seconde différence fondamentale est que toute matrice n'admet pas nécessairement une matrice inverse (parmi les nombres, seul le nombre zéro n'admet pas d'inverse). On verra par exemple que des matrices très simples, telles que

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix},$$

et bien d'autres encore, n'admettent pas d'inverse. C'est ce qui fait une des difficultés,... mais aussi le piment du calcul matriciel...

2. La définition de la matrice inverse est identique si on travaille avec des matrices à coefficients complexes plutôt que réels. Les propriétés générales sont aussi identiques.

### 2.1.4 Produit de matrices rectangulaires

Ce produit peut se généraliser à des matrices quelconques, ... ou presque. Pour être multipliables, deux matrices doivent avoir des tailles compatibles. Dans ce cas, on définit le produit de matrices compatibles de la façon suivante

**Définition 2.7 — Produit de matrices compatibles.** Soient  $A \in \mathcal{M}_{MK}(\mathbb{R})$  et  $B \in \mathcal{M}_{KN}(\mathbb{R})$  deux matrices. Le produit  $C = AB$  est bien défini, et donne une matrice  $M \times N$  définie par

$$\begin{cases} c_{11} &= a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \cdots + a_{1K}b_{K1}, \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} + \cdots + a_{1K}b_{K2}, \\ \cdots & \cdots \\ c_{1N} &= a_{11}b_{1N} + a_{12}b_{2N} + a_{13}b_{3N} + \cdots + a_{1K}b_{KN}, \\ c_{21} &= a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} + \cdots + a_{2K}b_{K1}, \\ c_{22} &= a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} + \cdots + a_{2K}b_{K2}, \\ \cdots & \cdots \\ c_{MN} &= a_{M1}b_{1N} + a_{M2}b_{2N} + a_{M3}b_{3N} + \cdots + a_{MK}b_{KN}, \end{cases}$$

ce que l'on synthétise sous la forme suivante : pour tous  $m = 1, \dots, M$  et  $n = 1, \dots, N$ ,

$$c_{mn} = a_{m1}b_{1n} + a_{m2}b_{2n} + \cdots + a_{mK}b_{Kn} = \sum_{k=1}^K a_{mk}b_{kn}.$$

Comme on l'a déjà signalé, on voit bien la contrainte de compatibilité posée sur les matrices  $A$  et  $B$  : les lignes de  $A$  et les colonnes de  $B$  doivent être de même taille, égale ici à  $K$ .

■ **Exemple 2.8 — Matrices compatibles... ou pas.** Considérons

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}.$$

Le produit  $AB$  existe, et donne une matrice  $4 \times 3$ , mais le produit  $BA$  n'est pas défini. ■

**(Matrices compatibles... ou pas)**

Le code R associé à ces produits, ainsi que le résultat se trouvent ci-dessous

```
> A <- matrix(1:8,ncol=2,byrow=T)
> B <- matrix(1:6,ncol=3,byrow=T)
> A%*%B
      [,1] [,2] [,3]
[1,]    9  12  15
[2,]   19  26  33
[3,]   29  40  51
[4,]   39  54  69
> B%*%A
Error in B %*% A : arguments inadéquats
```

Comme on l'a vu, le **produit matriciel est (en général) non commutatif**. Par contre, quelques calculs permettent de montrer que

**Proposition 2.7** 1. Le produit des matrices, quand il est bien défini, est associatif : étant données trois matrices  $A \in \mathcal{M}_{MK}$ ,  $B \in \mathcal{M}_{KL}$  et  $C \in \mathcal{M}_{LN}$ , on a

$$(AB)C = A(BC) .$$

2. Le produit des matrices est distributif par rapport à l'addition à droite : étant données trois matrices  $A \in \mathcal{M}_{MK}$  et  $B, C \in \mathcal{M}_{KN}$ , on a

$$A(B+C) = AB + AC .$$

De façon similaire, on a aussi distributivité à gauche : étant données trois matrices  $A, B \in \mathcal{M}_{MK}$  et  $C \in \mathcal{M}_{KN}$ , on a

$$(A+B)C = AC + BC .$$

Dernière propriété simple, le produit matriciel se comporte de façon simple vis à vis de la transposition des matrices : la transposée du produit est égal au produit des transposées, pris en ordre inverse. C'est une propriété à rapprocher de l'inverse du produit de deux matrices carrées que nous verrons plus loin.

**Proposition 2.8** Soient  $A \in \mathcal{M}_{MK}$  et  $B \in \mathcal{M}_{KN}$ . Alors

$$(AB)^T = B^T A^T .$$

**Remarque 2.9 — Produit matriciel et produit scalaire.** Le produit scalaire de deux vecteurs peut aussi s'écrire sous forme d'un produit matriciel. Ainsi on a par exemple en dimension 2, pour tous  $U = (x, y)^T$  et  $V = (z, t)^T$ , on voit facilement que  $U \cdot V = xz + yt = U^T V$ .

Ceci reste vrai dans  $\mathbb{R}^N$  : pour tous  $U, V \in \mathbb{R}^N$ , on a bien

$$U \cdot V = U^T V$$

le produit de gauche étant le produit scalaire des vecteurs, et le produit de droite étant le produit matriciel.

Notons qu'on pourrait aussi écrire  $U \cdot V = V^T U$ , on se limitera dans ce cours à la première expression.

Ce qui est vrai pour le produit scalaire de deux vecteurs l'est aussi pour l'ensemble des produits scalaires de deux familles de vecteurs :

**Remarque 2.10 — Produit matriciel et produit scalaire (suite).** Supposons maintenant que l'on dispose de deux familles de deux vecteurs,  $\mathcal{U} = \{U_1, U_2\}$  et  $\mathcal{V} = \{V_1, V_2\}$ . On peut calculer quatre produits scalaires croisés,  $U_1 \cdot V_1$ ,  $U_1 \cdot V_2$ ,  $U_2 \cdot V_1$  et  $U_2 \cdot V_2$ , et les arranger dans une nouvelle matrices  $2 \times 2$

$$\begin{pmatrix} U_1 \cdot V_1 & U_1 \cdot V_2 \\ U_2 \cdot V_1 & U_2 \cdot V_2 \end{pmatrix} = U^T V .$$

Plus généralement, étant données deux familles de vecteurs  $\mathcal{U} = \{U_1, \dots, U_M\}$  et  $\mathcal{V} = \{V_1, \dots, V_N\}$  et les matrices associées  $U$  et  $V$ , la matrice des produits scalaires est la matrice  $M \times N$  définie par

$$\begin{pmatrix} U_1 \cdot V_1 & U_1 \cdot V_2 & \cdots & U_1 \cdot V_N \\ U_2 \cdot V_1 & U_2 \cdot V_2 & \cdots & U_2 \cdot V_N \\ \vdots & \vdots & \ddots & \vdots \\ U_M \cdot V_1 & U_M \cdot V_2 & \cdots & U_M \cdot V_N \end{pmatrix} = U^T V.$$

## 2.2 Matrices, familles de vecteurs, changement de base et projection orthogonale

### 2.2.1 Matrice d'une famille de vecteurs, matrice d'une base

Étant donnée une famille de  $M$  vecteurs  $\{F_1, \dots, F_M\}$  de  $\mathbb{R}^N$ , nous avons déjà vu comment lui associer une matrice, en concaténant horizontalement les  $M$  vecteurs. Ceci produit une matrice  $N \times M$  notée  $F \in \mathcal{M}_{N,M}(\mathbb{R})$ .

Le calcul matriciel offre une manière simple de vérifier qu'une famille de vecteurs est une famille orthonormée.

**Théorème 2.11 — Famille orthonormée, base orthonormée.** 1. Soit  $\mathcal{F} = \{F_1, \dots, F_M\}$  une famille de  $M$  vecteurs de  $\mathbb{R}^N$  (avec  $M \leq N$ ), soit  $F \in \mathcal{M}_{N,M}$  la matrice associée. Alors  $\mathcal{F}$  est une famille orthonormée si et seulement si sa matrice de Gram  $G = F^T F$  est égale à la matrice identité :

$$F^T F = I_M.$$

2. Une famille  $\mathcal{F} = \{F_1, \dots, F_N\}$  de  $N$  vecteurs de  $\mathbb{R}^N$  est une base orthonormée de  $\mathbb{R}^N$  si et seulement si

$$F^T F = F F^T = I_N,$$

où  $F \in \mathcal{M}_N$  est la matrice de la famille  $\mathcal{F}$ . Ceci équivaut à dire que  $F$  est inversible, et que  $F^{-1} = F^T$ .



**(Retour sur l'exemple 1.14)** Revenons sur la famille de cet exemple, et vérifions que c'est une famille orthonormée.

```
> F1 <- c(sqrt(2),1,1)/2
> F2 <- c(-sqrt(2),1,1)/2
> F.mat <- matrix(c(F1,F2),ncol=2)
> prod.scal.mat <- t(F.mat)%*% F.mat
> print(prod.scal.mat)
      [,1] [,2]
[1,] 1.000000e+00 -1.110223e-16
[2,] -1.110223e-16 1.000000e+00
```

La matrice des produits scalaires ( $2 \times 2$ ) est égale à la matrice identité (à la précision numérique près), on a donc une famille orthonormée.

Notons que dans ce cas, la matrice  $F F^T$  n'est pas égale à la matrice identité

```
> print(F.mat %*% t(F.mat))
      [,1] [,2] [,3]
[1,] 1 0.0 0.0
[2,] 0 0.5 0.5
[3,] 0 0.5 0.5
```

**Remarque 2.12** Une famille orthonormée est automatiquement base orthonormée du sous-espace de  $\mathbb{R}^N$  qu'elle engendre. Si elle est constituée de  $N$  vecteurs c'est une base orthonormée de  $\mathbb{R}^N$ .

**Théorème 2.13** Soit  $\mathcal{F} = \{F_1, \dots, F_N\}$  une famille de  $N$  vecteurs de  $\mathbb{R}^N$ , soit  $F \in \mathcal{M}_N$  la matrice associée. Les assertions suivantes sont équivalentes

1.  $\mathcal{F}$  est une base de  $\mathbb{R}^N$ .
2. La matrice  $F$  est inversible.
3.  $\det(F) \neq 0$  (on rappelle que  $\det(F)$  est le déterminant de  $\mathcal{F}$ ).

### 2.2.2 Matrices des coordonnées d'un vecteur dans une base

On a déjà vu que les matrices peuvent être associées à des transformations (rotations, projections,...). On va maintenant voir qu'elles fournissent un outil extrêmement puissant pour effectuer des changements de base.

Pour cela, considérons un vecteur  $X = (x_1, x_2, \dots, x_N)^T$ . On a vu que  $x_1, \dots, x_N$  sont les coordonnées de  $X$  dans la base canonique, notée

$$\mathcal{B}_0 = \{E_1, E_2, \dots, E_N\}.$$

On peut associer à cela une matrice à  $N$  lignes et 1 colonne notée

$$M(X)_{\mathcal{B}_0} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

qui coïncide en fait avec  $X$ .

Plus généralement, étant donnée une base quelconque

$$\mathcal{B} = \{F_1, F_2, \dots, F_N\}$$

tout vecteur  $X \in \mathbb{R}^N$  se décompose de façon unique sur cette base,

$$X = y_1 F_1 + y_2 F_2 + \dots + y_N F_N,$$

et on note

$$M(X)_{\mathcal{B}} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

la matrice de  $X$  dans la base  $\mathcal{B}$ . La question est : comment passe-t-on des coordonnées dans la base canonique, c'est à dire de  $M(X)_{\mathcal{B}_0}$  (ou  $X$ ) aux coordonnées de  $X$  dans la base  $\mathcal{B}$ , c'est à dire de  $M(X)_{\mathcal{B}}$ ? Commençons par un petit exemple.

■ **Exemple 2.9 — Changement de base dans  $\mathbb{R}^2$ .** On considère les deux vecteurs

$$U_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad U_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

Comme

$$\det(U_1, U_2) = 1 \times 4 - 2 \times 3 = -2 \neq 0,$$

la famille  $\mathcal{B} = \{U_1, U_2\}$  est une base de  $\mathbb{R}^2$ . Soit maintenant  $V = (1, 3)^T$ , quelles sont les coordonnées de  $V$  dans la base  $\mathcal{B}$ ? Il s'agit de trouver deux nombres  $\alpha_1, \alpha_2$  tels que  $V = \alpha_1 U_1 + \alpha_2 U_2$ , ce qui s'écrit sous forme d'un système linéaire

$$\begin{cases} \alpha_1 + 3\alpha_2 = 1 \\ 2\alpha_1 + 4\alpha_2 = 3 \end{cases}$$

On observe que ceci est aussi équivalent à l'équation matricielle

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

et que la matrice intervenant à gauche n'est autre que la matrice obtenue par concaténation des deux vecteurs de la base. Résoudre le système équivaut à trouver la matrice inverse, par

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 3 \end{pmatrix}.$$

Le calcul donne

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}^{-1} = \begin{pmatrix} -2 & 3/2 \\ 1 & -1/2 \end{pmatrix}, \quad \text{soit} \quad \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 5/2 \\ -1/2 \end{pmatrix}.$$

Le résultat de l'exemple ci-dessus n'a pas grande importance, la chose importante est de bien comprendre le rôle fondamental de la matrice associée à la base. Comme on l'a déjà vu, une famille de  $N$  vecteurs est une base de  $\mathbb{R}^N$  si et seulement si sa matrice est inversible. Cette matrice inverse est précisément celle qui est utilisée pour calculer les coordonnées d'un vecteur dans une nouvelle base, comme le spécifie le théorème qui suit.

**Théorème 2.14 — Changement de base.** Soit  $\mathcal{B} = \{F_1, F_2, \dots, F_N\}$  une base de  $\mathbb{R}^N$ , et soit  $F$  la matrice de la base  $\mathcal{B}$ . Alors  $F$  est inversible, et les coordonnées d'un vecteur  $X$  quelconque dans la base  $\mathcal{B}$  sont données par

$$M(X)_{\mathcal{B}} = F^{-1}M(X)_{\mathcal{B}_0} = F^{-1}X.$$

**Remarque 2.15 — Le cas des bases orthonormées.** Les bases orthonormées forment un cas particulier important. Comme on l'a déjà vu, la matrice  $F$  d'une base orthonormée est *orthogonale* dans le sens suivant :

$$FF^T = F^T F = I_N.$$

Ainsi, on a alors

$$F^{-1} = F^T,$$

et il n'est pas nécessaire de calculer l'inverse de  $F$ .

■ **Exemple 2.10 — Base orthonormée.** On considère les trois vecteurs de l'espace

$$F_1 = \left( \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T, \quad F_2 = \left( -\frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T, \quad F_3 = \left( 0, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right)^T.$$

On peut vérifier que ces trois vecteurs forment une base orthonormée  $\mathcal{B}$  de  $\mathbb{R}^3$ . Par conséquent, étant donnée la matrice  $F$  de cette base, son inverse  $F^{-1}$  est égale à sa transposée  $F^T$  :

$$F = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ 1/2 & 1/2 & \sqrt{2}/2 \\ 1/2 & 1/2 & -\sqrt{2}/2 \end{pmatrix}, \quad \text{et} \quad F^{-1} = \begin{pmatrix} \sqrt{2}/2 & 1/2 & 1/2 \\ -\sqrt{2}/2 & 1/2 & 1/2 \\ 0 & \sqrt{2}/2 & -\sqrt{2}/2 \end{pmatrix}$$

De là, les coordonnées du vecteur  $X = (x, y, z)^T$  dans cette base  $\mathcal{B}$  peuvent s'écrire sous la forme du vecteur

$$M(X)_{\mathcal{B}} = F^{-1}X = \begin{pmatrix} \sqrt{2}/2 & 1/2 & 1/2 \\ -\sqrt{2}/2 & 1/2 & 1/2 \\ 0 & \sqrt{2}/2 & -\sqrt{2}/2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{2} \begin{pmatrix} x\sqrt{2} + y + z \\ -x\sqrt{2} + y + z \\ (y - z)\sqrt{2} \end{pmatrix}$$

■



Etudions cet exemple sous R :

```
> F1 <- c(sqrt(2),1,1)/2
> F2 <- c(-sqrt(2),1,1)/2
> F3 <- c(0,sqrt(2),-sqrt(2))/2
> F.mat <- matrix(c(F1,F2,F3),ncol=3)
> print(t(F.mat)%*% F.mat)
      [,1]      [,2]      [,3]
[1,] 1.000000e+00 -1.110223e-16 0
[2,] -1.110223e-16 1.000000e+00 0
[3,] 0.000000e+00 0.000000e+00 1
> print(F.mat %*% t(F.mat))
      [,1]      [,2]      [,3]
[1,] 1 0.000000e+00 0.000000e+00
[2,] 0 1.000000e+00 -1.110223e-16
[3,] 0 -1.110223e-16 1.000000e+00
```

Ces lignes montrent que  $FF^T = F^TF = I_3$  (à la précision des calculs près), et que  $F$  est donc la matrice d'une base orthonormée.

Calculons le vecteur  $Y = M(X)_{\mathcal{B}}$  des coordonnées d'un vecteur, disons  $X = (1, 2, 3)^T$  dans cette base :

```
> X <- 1:3
> Y <- t(F.mat) %*% X
> print(Y)
      [,1]
[1,] 3.2071068
[2,] 1.7928932
[3,] -0.7071068
```

■ **Exemple 2.11 — Base non-orthonormée.** On considère les trois vecteurs de l'espace

$$F_1 = (1, 2, 3)^T, \quad F_2 = (4, 5, 6)^T, \quad F_3 = (7, 8, 10)^T.$$

On peut vérifier que ces trois vecteurs forment une base  $\mathcal{B}$  de  $\mathbb{R}^3$ , qui n'est pas orthonormée. Par conséquent, la matrice  $F$  de cette base est inversible. Le calcul donne (voir plus loin)

$$F^{-1} = \frac{1}{3} \begin{pmatrix} -2 & -2 & 3 \\ -4 & 11 & -6 \\ 3 & -6 & 3 \end{pmatrix}$$

De là, les coordonnées du vecteur  $X = (x, y, z)^T$  dans cette base  $\mathcal{B}$  peuvent s'écrire sous la forme du vecteur

$$M(X)_{\mathcal{B}} = F^{-1}X = \frac{1}{3} \begin{pmatrix} -2 & -2 & 3 \\ -4 & 11 & -6 \\ 3 & -6 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{3} \begin{pmatrix} x\sqrt{2} + y + z \\ -x\sqrt{2} + y + z \\ (y - z)\sqrt{2} \end{pmatrix}$$

■



Etudions cet exemple sous R :

```
> F1 <- 1:3
> F2 <- 4:6
> F3 <- c(7,8,10)
> F.mat <- matrix(c(F1,F2,F3),ncol=3)
> det(F.mat)
[1] -3
> F.inv <- solve(F.mat)
> print(F.inv)
      [,1]      [,2]      [,3]
[1,] -0.6666667 -0.6666667 1
[2,] -1.3333333 3.6666667 -2
[3,] 1.0000000 -2.0000000 1
```

Les quatre premières lignes génèrent les trois vecteurs puis la matrice associée, la suivante vérifie que c'est bien la matrice d'une base de  $\mathbb{R}^3$ , et qu'elle est donc inversible. On calcule ensuite la matrice inverse, où on reconnaît la matrice  $F^{-1}$  donnée plus haut.

La base n'étant pas orthonormée, on ne peut plus calculer les coordonnées d'un vecteur dans cette base comme dans l'exemple précédent, en utilisant la transposée. On peut néanmoins utiliser l'instruction `solve`, avec une syntaxe différente. Calculons le vecteur  $Y = M(X)_{\mathcal{B}}$  des coordonnées d'un vecteur, disons  $X = (1, 2, 3)^T$  dans cette base :

```
> X <- 1:3
> Y <- solve(F.mat,X)
> print(Y)
[1] 1 0 0
```

résultat qu'on aurait pu deviner tout de suite...

### 2.2.3 Matrices et projection orthogonale

Pour finir, revenons sur le problème de la projection orthogonale sur une sous-espace vectoriel de  $\mathbb{R}^N$  engendré par une famille de vecteurs. On notera  $\mathcal{F} = \{F_1, \dots, F_M\}$  cette famille, et  $F \in \mathcal{M}_{N,M}(\mathbb{R})$  la matrice correspondante. On a vu que le projeté orthogonal d'un vecteur quelconque  $X \in \mathbb{R}^N$  sur le sous-espace engendré par cette famille de vecteurs s'écrit sous la forme

$$\Pi(X) = a_1 F_1 + \dots + a_M F_M,$$

et les nombres  $a_1, \dots, a_M$  forment un vecteur  $A = (a_1, \dots, a_M)^T$  à  $M$  composantes. Comment calculer ces nombres (ou ce vecteur)? la réponse se trouve dans le théorème 2.17 ci-dessous, dans le cas où la famille  $\mathcal{F}$  est une base du sous-espace de  $\mathbb{R}^N$  qu'elle engendre (c'est à dire si son rang est égal à  $M$ ).

**Proposition 2.16** Avec les notations ci-dessus, la famille  $\mathcal{F}$  est une base du sous-espace  $\text{Vect}\{\mathcal{F}\}$  qu'elle engendre si et seulement si la matrice de Gram  $F^T F$  est inversible.

On peut alors énoncer le résultat principal

**Théorème 2.17 — Projection orthogonale sur un sous espace engendré.** Soit  $\mathcal{F} = \{F_1, \dots, F_M\}$  une famille de vecteurs de  $\mathbb{R}^N$ , base du sous-espace qu'elle engendre, et soit  $F \in \mathcal{M}_{N,M}(\mathbb{R})$  la matrice correspondante. Pour tout vecteur  $X$  de  $\mathbb{R}^N$ , le vecteur  $A \in \mathbb{R}^M$  des coefficients de la projection orthogonale  $Y = \Pi(X)$  sur le sous-espace de  $\mathbb{R}^N$  engendré par  $\mathcal{F}$  est donné par

$$A = (F^T F)^{-1} F^T X,$$

et le projeté orthogonal  $Y = \Pi(X)$  est quant à lui donné par

$$\Pi(X) = FA = F(F^T F)^{-1} F^T X.$$

**Remarque 2.18 — Cas de familles orthonormées.** Dans le cas où la famille de vecteurs  $\mathcal{F}$  est orthonormée, on a déjà vu que  $F^T F = I_M$ , de sorte que les résultats du théorème 2.17 se simplifient, on a ainsi avec les mêmes notations

$$A = F^T X, \quad Y = FF^T X.$$

■ **Exemple 2.12 — Retour sur l'exemple 1.14.** On considère les deux vecteurs de l'espace

$$F_1 = \left( \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T, \quad F_2 = \left( -\frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2} \right)^T.$$

on a déjà vu qu'ils constituent une famille orthonormée, donc une base orthonormée du sous-espace qu'ils engendrent. Les coordonnées dans cette base du projeté orthogonal  $Y = \Pi(X)$  d'un vecteur  $X$  quelconque sont obtenues par

$$A = F^T X = \frac{1}{2} \begin{pmatrix} \sqrt{2} & 1 & 1 \\ -\sqrt{2} & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{2} x\sqrt{2} + y + z \quad -\frac{1}{2} x\sqrt{2} + y + z.$$

De plus, on a

$$FF^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}$$

de sorte que le projeté orthogonal  $Y = \Pi(X)$  s'écrit

$$Y = FF^T X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ (y+z)/2 \\ (y+z)/2 \end{pmatrix}.$$

■ **Exemple 2.13 — Cas d'une famille non-orthonormée.** On considère les deux vecteurs de l'espace

$$F_1 = (1, 2, 3)^T, \quad F_2 = (4, 5, 6)^T.$$

Ils ne sont manifestement pas colinéaires, et forment une base du plan qu'ils engendrent. Cette base n'est pas orthonormée, on vérifie en effet que  $F^T F \neq I_2$ . Le calcul donne

$$F^T F = \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}, \quad (F^T F)^{-1} = \frac{1}{54} \begin{pmatrix} 77 & -32 \\ -32 & 14 \end{pmatrix}$$

Par conséquent, les coordonnées du projeté  $Y = \Pi(X)$  de  $X = (x, y, z)^T$  sur le sous-espace dans sa base  $\mathcal{F}$  sont données par

$$\begin{aligned} A = (F^T F)^{-1} F^T X &= \frac{1}{54} \begin{pmatrix} 77 & -32 \\ -32 & 14 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{54} \begin{pmatrix} -51 & -6 & 39 \\ 24 & 6 & -12 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\ &= \frac{1}{54} \begin{pmatrix} -51x - 6y + 39z \\ 24x + 6y - 12z \end{pmatrix} = \frac{1}{18} \begin{pmatrix} -17x - 2y + 13z \\ 8x + 2y - 4z \end{pmatrix} \end{aligned}$$

(les calculs peuvent se faire à la main... mais ça va plus vite avec un bon logiciel...). Pour ce qui est du projeté orthogonal  $Y = \Pi(X)$  lui-même, il est donné par

$$Y = FA = \frac{1}{18} \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \begin{pmatrix} -17x - 2y + 13z \\ 8x + 2y - 4z \end{pmatrix} = \frac{1}{18} \begin{pmatrix} 15x + 6y - 3z \\ 6x + 6y + 6z \\ -3x + 6y + 15z \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 5x + 2y - z \\ 2x + 2y + 2z \\ -x + 2y + 5z \end{pmatrix}.$$

Le résultat en lui-même n'a strictement aucun intérêt (sauf si on aime cet exemple...), mais il a le mérite de montrer que tout peut être calculé en utilisant du calcul matriciel, qui peut lui-même être effectué numériquement (par exemple sous R, ou matlab ou python ou tout autre logiciel de calcul numérique), ou exactement avec un logiciel de calcul formel (par exemple maxima, ou encore maple ou mathematica). ■

## 2.3 Résumé

Qu'avons nous vu dans ce chapitre ? Ci-dessous un résumé, qui reprend aussi l'essentiel des aspects à retenir.

- Un tableau de nombres (données) peut être interprété géométriquement :** si le tableau a  $N$  lignes et  $P$  colonnes, on peut le voir soit comme le tableau des coordonnées de  $P$  points dans un espace de dimension  $N$  (il y a  $N$  coordonnées), soit comme le tableau des coordonnées de  $N$  points dans un espace de dimension  $P$  (il y a alors  $P$  coordonnées).
- On adopte ici le premier point de vue.** Le tableau constitue une matrice  $N \times P$ , que l'on peut donc voir comme la concaténation de  $P$  vecteurs appartenant à  $\mathbb{R}^N$ . On notera  $X$  cette matrice. Chaque colonne est donc constituée des coordonnées du vecteur correspondant dans la base canonique.
- Supposons que l'on dispose d'une famille  $\mathcal{F} = \{F_1, F_2, \dots, F_N\}$  constituée de  $N$  vecteurs de  $\mathbb{R}^N$ .** On note  $F \in \mathcal{M}_N$  la matrice associée à cette famille, et  $F^T \in \mathcal{M}_N$  la matrice transposée.
  - La famille de vecteurs  $\mathcal{F}$  est une base de  $\mathbb{R}^N$  si et seulement si  $\det(F) \neq 0$ .

- Dans ce cas, les coordonnées d'un vecteur  $V \in \mathbb{R}^N$  dans cette base forment un autre vecteur à  $N$  coordonnées  $V' \in \mathbb{R}^N$ , qui se calcule via le produit matriciel  $V' = F^{-1}V$ .
- La base est orthonormée si et seulement si  $F^T F = I_N$ . Alors  $F^{-1} = F^T$ , et les coordonnées d'un vecteur  $X \in \mathbb{R}^N$  dans cette base forment un autre vecteur à  $N$  coordonnées  $V'$ , qui se calcule via le produit matriciel  $V' = F^T V$ .
- Étant donnée maintenant une matrice  $X \in \mathcal{M}_{N,P}$  à  $N$  lignes et  $P$  colonnes, chacune de ses colonnes est un vecteur de  $\mathbb{R}^N$ . Les coordonnées de ces colonnes dans la base  $\mathcal{F}$  forment une autre matrice  $X' \in \mathcal{M}_{NP}$ , qui se calcule elle-aussi via le produit matriciel  $X' = F^{-1}X$ .

4. **Supposons que l'on dispose d'une famille  $\mathcal{F} = \{F_1, F_2, \dots, F_M\}$  constituée de  $M$  vecteurs de  $\mathbb{R}^N$ , avec  $M \leq N$ . On note  $F \in \mathcal{M}_{N,M}$  la matrice associée à cette famille, et  $F^T \in \mathcal{M}_{M,N}$  la matrice transposée.**

- Ces vecteurs engendrent un sous-espace de  $\mathbb{R}^N$  de dimension inférieure ou égale à  $M$ . Pour vérifier que cette dimension est bien égale à  $m$ , il faut vérifier que  $\det(F^T F) \neq 0$ .
- Dans ce cas, le projeté orthogonal d'un vecteur quelconque  $V \in \mathbb{R}^N$  sur ce sous-espace, noté  $\Pi(V)$ , se calcule directement via

$$\Pi(V) = F(F^T F)^{-1} F^T V \in \mathbb{R}^N .$$

On procède de même pour obtenir d'un seul coup les projetés orthogonaux des colonnes d'une matrice  $X \in \mathcal{M}_{N,P}$  sur le sous-espace. Ceux-ci forment une matrice notée  $\Pi(X) \in \mathcal{M}_{N,P}$ , que l'on peut obtenir de façon similaire

$$\Pi(X) = F(F^T F)^{-1} F^T X \in \mathcal{M}_{N,P} .$$

- Ci-dessus,  $\Pi(V)$  a la même taille que  $V$ , mais s'écrit aussi comme combinaison linéaire de  $F_1, \dots, F_M$  :

$$\Pi(V) = a_1 F_1 + \dots + a_M F_M .$$

En notant  $A = (a_1, \dots, a_M)^T \in \mathbb{R}^M$  on a

$$A = (F^T F)^{-1} F^T V \in \mathbb{R}^M , \quad \text{d'où} \quad \Pi(V) = FA .$$

De même,  $\Pi(X)$  a la même taille que la matrice  $X$  de départ, ses colonnes donnent les coordonnées des vecteurs projetés correspondants dans la base canonique. Par contre, chacun de ces vecteurs projetés s'écrit comme combinaison linéaire de  $F_1, \dots, F_M$ . Les coefficients correspondants forment une matrice  $A \in \mathcal{M}_{N,M}$  qui s'écrit

$$A = (F^T F)^{-1} F^T X \in \mathcal{M}_{N,M} , \quad \text{d'où cette fois encore} \quad \Pi(X) = FA .$$

- Si la famille  $\mathcal{F}$  est orthonormée, alors  $F^T F = I_M$ , ces expressions deviennent

$$A = F^T V \in \mathcal{M}_{N,M} , \quad \text{et} \quad \Pi(X) = FA = FF^T V .$$



## 3. Réduction de matrices et analyse multivariée

Beaucoup de méthodes d'analyse multivariée et de réduction de dimension visent à réduire le nombre de variable pour simplifier les données à analyser. L'objectif est d'effectuer une telle réduction en perdant le moins d'information possible.

L'information apportée par une variable peut être, dans une certaine mesure, décrite par sa variabilité, ou dispersion. Une variable qui prend une valeur quasi constante sur tous les individus n'apporte pas ou très peu d'information sur ces individus. Au contraire une variable prenant des valeurs significativement différentes d'un individu à l'autre apporte davantage d'informations sur les individus. On utilise donc souvent des mesures de variabilité pour quantifier l'information. Pour effectuer sur des données une transformation tout en préservant l'essentiel de l'information, on se fixe souvent comme objectif de conserver le plus possible de variabilité totale, dans un sens que nous allons définir précisément un peu plus loin.

Nous allons particulièrement nous focaliser ici sur l'analyse en composante principale (ACP). Le principe de l'ACP est d'effectuer un changement de base dans l'espace des variables, en faisant en sorte que

1. Les nouvelles variables soient décorréélées c'est à dire portent le moins possible d'information commune
2. Elle soient ordonnées par ordre décroissant de variabilité.

L'ACP repose sur des techniques matricielles, nous allons étudier dans ce chapitre le lien entre ces deux aspects, ainsi que des applications pratiques.

### 3.1 Généralités

#### 3.1.1 Tableaux de données, variables, individus

En analyse de données, les données recueillies sont généralement mises sous forme d'un tableau, que l'on considérera comme une matrice notée  $X$  de taille  $n \times p$ . Conventionnellement, l'élément de matrice  $x_{ij}$  est la valeur prise par la  $j$ -ème variable sur le  $i$ -ème individu :

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{pmatrix}$$

Dans cette matrice :

- Les colonnes représentent les variables : la  $j$ -ième colonne  $x_{\bullet j} = (x_{1j}, x_{2j}, \dots, x_{nj})^T$  représente les valeurs prises par la  $j$ -ième variable sur les  $n$  individus. Chaque variable/colonne est un vecteur de  $\mathbb{R}^n$ .
- Les lignes représentent les individus : la  $i$ -ième ligne  $x_{i\bullet} = (x_{i1}, \dots, x_{ip})$  donne les valeurs des  $p$  variables pour le  $i$ -ième individu. Chaque ligne/individu est un vecteur de  $\mathbb{R}^p$ .

### 3.1.2 Changements de base dans l'espace des variables et dans l'espace des individus

- Une variable est donc représentée par un vecteur  $x$  de  $\mathbb{R}^n$ , qui est une matrice colonne. On a vu au chapitre précédent comment effectuer un changement de base (plus précisément, comment calculer le vecteur des coordonnées de  $V$  dans une nouvelle base). On va ici se limiter au cas des bases orthonormées. Soit  $U$  la matrice d'une base orthonormée  $\mathcal{U}$  de  $\mathbb{R}^n$  :  $U$  est une matrice orthogonale, c'est à dire

$$UU^T = U^T U = I_n \quad (3.1)$$

(en d'autres termes,  $U^{-1} = U^T$ ). Comme on l'a vu au chapitre précédent, le vecteur des coordonnées de  $x$  dans cette base est donné par

$$M(x)_{\mathcal{U}} = U^T x. \quad (3.2)$$

Similairement, l'action du changement de base sur la matrice de données  $X$  prend la forme

$$X \mapsto U^T X. \quad (3.3)$$

- Un individu est quant à lui représenté par les valeurs qu'il prend sur les  $p$  variables, ce qui fait un vecteur de  $\mathbb{R}^p$ . Attention, ce vecteur est cette fois représenté par une matrice ligne. Supposons maintenant qu'on veuille changer de base dans  $\mathbb{R}^p$ , l'espace des individus : soit  $V \in \mathcal{M}_p$  la matrice d'une base  $\mathcal{V}$  de  $\mathbb{R}^p$ , c'est une matrice orthogonale à  $p$  lignes et  $p$  colonnes ( $VV^T = V^T V = I_p$ ). En utilisant des transpositions de matrice, on peut montrer que le changement de base dans l'espace des individus se met sous la forme

$$X \mapsto XV. \quad (3.4)$$

### 3.1.3 Mesures de position : moyenne

La moyenne de la  $j$ -ème variable est définie par

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}. \quad (3.5)$$

Les moyennes forment une matrice ligne (à  $p$  colonnes), qu'on notera  $\bar{x} \in \mathbb{R}^p$ . Cette matrice peut être obtenue par calcul matriciel sous la forme

$$\bar{x} = \frac{1}{n} \mathbf{1}_n^T X, \quad (3.6)$$

où  $\mathbf{1}_n = (1, 1, \dots, 1)^T$  est un vecteur de longueur  $n$  ne contenant que des 1.

On organise aussi parfois les moyennes sous forme d'une matrice de même taille que  $X$ ,

$$\bar{X} = \begin{pmatrix} \bar{x}_1 & \bar{x}_2 & \bar{x}_3 & \cdots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \bar{x}_3 & \cdots & \bar{x}_p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \bar{x}_3 & \cdots & \bar{x}_p \end{pmatrix} = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T X. \quad (3.7)$$

Centrer les données revient à soustraire à chaque variable sa moyenne, c'est à dire effectuer l'opération

$$X^c = X - \bar{X}. \quad (3.8)$$

**Remarque 3.1** Par définition, les valeurs prises par une variable pour différents individus sont comparables, ça a donc du sens de calculer leur moyenne. On pourrait aussi imaginer calculer des moyennes sur les variables, et pas sur des individus, mais encore faut-il que cela ait du sens. Prenons deux exemples

- Supposons que les variables soient la taille, le poids, l'âge et la pointure. Cela a du sens de calculer les valeurs moyennes de ces variables. Par contre, calculer pour chaque individu la moyenne de son poids, son âge, sa traille et sa pointure n'a pas de sens.
- Supposons qu'on dispose de données d'électroencéphalographie pour un sujet : disons des mesures de potentiel électrique sur 50 capteurs, à 1000 instants différents. Il peut être tout aussi intéressant de calculer une moyenne sur les capteurs qu'une moyenne sur les instants, ces deux moyennes portent des informations différentes. Dans le premier cas, les variables sont les valeurs du potentiel à un instant donné et les individus sont les capteurs, et vice-versa dans l'autre situation.



**(Calcul de moyennes)** Sous R, la fonction `mean` permet de calculer la moyenne. La moyenne d'un vecteur se calcule avec une syntaxe simple

```
> x <- rnorm(100,1,2)
> x.mean <- mean(x)
> print(x.mean)
[1] 0.9797099
```

La première ligne génère 100 nombres pseudo-aléatoires, distribués suivant une loi normale de moyenne 1 et écart-type 2, la suivante calcule la moyenne empirique, dont on voit qu'elle est très proche de la valeur théorique.

Dans le cas d'une matrice de données, il faut être attentif à la syntaxe. La fonction `mean` retournera un nombre, qui est la moyenne de  $X$  sur les lignes et les colonnes. Pour obtenir les moyennes sur les lignes ou les colonnes, utiliser les fonctions `colMeans` et `rowMeans`.

```
> X <- matrix(rnorm(1000,1,2),nrow=10)
> mean(X)
[1] 1.062034
> length(colMeans(X))
[1] 10
> length(rowMeans(X))
[1] 100
```

La première ligne génère un tableau  $10 \times 10$  et la seconde évalue la moyenne empirique du tableau (et retourne donc une unique valeur). La fonction `colMeans` va retourner les moyennes des 10 colonnes (moyenne sur les lignes), la fonction `rowMeans` va retourner les moyennes des 100 lignes (moyenne sur les colonnes).

### 3.1.4 Mesures de variabilité : moment d'ordre deux, variance, covariance

La mesure la plus classique de variabilité d'une variable est sa variance empirique : la variance de la  $j$ -ème variable est définie par

$$s_j^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 \quad (3.9)$$

(on préfère parfois utiliser  $n$  au dénominateur). En introduisant le moment d'ordre deux

$$\overline{x_j^2} = \frac{1}{n} \sum_{i=1}^n x_{ij}^2 \quad (3.10)$$

on a aussi

$$s_j^2 = \overline{x_j^2} - \bar{x}_j^2. \quad (3.11)$$

Les racines carrées  $s_j$  des variances sont appelées écart-types.

**Remarque 3.2 — Réduction.** Dans certaines situations, il peut être utile de réduire les données après centrage. Si tous les écarts-types  $s_j$  sont strictement positifs, les données centrées et réduites prennent la forme

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}. \quad (3.12)$$

Matriciellement, en formant la matrice diagonale  $S_D$  définie par les écarts-type, on écrit

$$Z = (X - \bar{X})S_D^{-1}, \quad \text{où } S_D \text{ est la matrice diagonale} \quad S_D = \begin{pmatrix} s_1 & 0 & 0 & \cdots & 0 \\ 0 & s_2 & 0 & \cdots & 0 \\ 0 & 0 & s_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & s_p \end{pmatrix} \quad (3.13)$$

Il est généralement important de s'intéresser aux co-variations des variables. Étant données deux variables  $j$  et  $j'$ , on calcule leur covariance empirique

$$c_{jj'} = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ij'} - \bar{x}_{j'}) \quad (3.14)$$

Ces nombres forment une matrice, appelée *matrice de covariance empirique* (ou parfois matrice de variance-covariance). En utilisant le calcul matriciel, cette matrice s'écrit comme suit

**Définition 3.1 — Matrice de covariance empirique.** La matrice de covariance empirique associée à une matrice de données  $X \in \mathcal{M}_{n,p}(\mathbb{R})$  est la matrice  $C_X \in \mathcal{M}_p(\mathbb{R})$  d'éléments  $c_{jj'}$  définis ci-dessus, et prend la forme

$$C_X = \frac{1}{n-1} (X - \bar{X})^T (X - \bar{X}). \quad (3.15)$$

La variabilité totale (ou inertie) de  $X$  est la trace de  $C_X$ , c'est à dire la somme de ses éléments diagonaux

$$V_T = \text{Tr}(C_X) = \sum_{j=1}^p c_{jj}. \quad (3.16)$$

**Remarque 3.3 — Propriétés de la matrice de covariance.** 1. Les termes diagonaux de la matrice de covariance sont les variances des variables :  $c_{jj} = s_j^2$ .  
2. Les covariances des variables peuvent aussi se calculer sous la forme

$$c_{jj'} = \frac{1}{n-1} \sum_{i=1}^n x_{ij}x_{ij'} - \bar{x}_j\bar{x}_{j'}.$$

3. La matrice de covariance  $C_X$  est une matrice carrée symétrique :  $C_X^T = C_X$ , autrement dit pour tous  $j, j'$ , on a l'égalité  $c_{jj'} = c_{j'j}$ .



#### (Variance, covariance)

Les fonctions `var` et `cov` estiment les variances et covariances empiriques.

```
> X <- rnorm(100)
> Y <- 2*X + rnorm(100)/2
> X.var <- var(X)
> Y.var <- var(Y)
> print(paste('La variance de X vaut',X.var,'et celle de Y vaut',Y.var))
[1] "La variance de X vaut 1.2733876652014 et celle de Y vaut 5.3488526566343"
> XY.cov <- cov(X,Y)
```

```
> print(paste('La covariance de X et Y vaut',XY.cov))
[1] "La covariance de X et Y vaut 2.55128526835193"
```

Dans ces quelques lignes, on génère un vecteur  $X$  de 100 valeurs pseudo-aléatoires distribuées suivant une loi normale centrée réduite, puis un vecteur  $Y$  qui est une version “bruitée” de  $2 * X$ . On calcule et affiche ensuite les variances de  $X$  et  $Y$ , puis leur covariance. Cette dernière est difficile à interpréter, la corrélation est plus pertinente dans ce cas (et montrera clairement la relation presque linéaire entre  $X$  et  $Y$ ).

Dans le cas d’une matrice de données, la fonction `cov` retourne la matrice de covariance. Pour obtenir les variances des variables, on peut utiliser la fonction `diag`, qui extrait la diagonale d’une matrice.

```
> X <- matrix(rnorm(1000,1,2),ncol=10)
> X.cov <- cov(X)
> dim(X.cov)
[1] 10 10
> X.var <- diag(X.cov)
```

Toutes ces fonctions admettent soit des vecteurs ou matrices, soit des `data.frames` comme arguments.

**Remarque 3.4 — Matrice de covariance et changement de base.** La matrice de covariance d’un tableau de données se transforme de façon simple par changement de base.

1. Étant donnée la matrice  $V$  d’une base orthogonale de l’espace des individus  $\mathbb{R}^p$ , en notant  $Y = XV$  la matrice des données dans la nouvelle base, on a aussi  $\bar{Y} = \bar{X}V$ , de sorte que  $(Y - \bar{Y}) = (X - \bar{X})V$ . La matrice de covariance de  $Y$ , qu’on notera  $C_Y$  est alors donnée par

$$C_Y = V^T C_X V . \quad (3.17)$$

En effet, le calcul donne

$$C_Y = \frac{1}{n-1} V^T (X - \bar{X})^T (X - \bar{X}) V ,$$

ce qui est le résultat attendu.

2. Étant donnée la matrice  $U$  d’une base orthogonale de l’espace des variables  $\mathbb{R}^n$ , en notant  $Y = U^T X$  la matrice des données dans la nouvelle base, on a aussi  $\bar{Y} = U^T \bar{X}$ , de sorte que  $(Y - \bar{Y}) = U^T (X - \bar{X})$ . La matrice de covariance de  $Y$ , notée  $C_Y$  est alors donnée par

$$C_Y = C_X . \quad (3.18)$$

En effet, le calcul donne

$$C_Y = \frac{1}{n-1} (X - \bar{X})^T U U^T (X - \bar{X}) = C_X ,$$

car  $U$  est orthogonale.

**Définition 3.2 — Variabilité totale.** La variabilité totale (ou inertie) de d’un tableau de données  $X$  est la trace de sa matrice de covariance empirique  $C$ , c’est à dire la somme de ses éléments diagonaux

$$I_{\text{tot}} = \text{Tr}(C_X) = \sum_{j=1}^p c_{jj} . \quad (3.19)$$

Ici, l’expression  $\text{Tr}(C_X)$  représente la trace de  $C_X$ , qui est simplement la somme de ses éléments diagonaux.

Il faut noter que la variabilité totale de  $X$  est invariante par changement de base, qu’il s’agisse de l’espace des variables ou de l’espace des individus :  $\text{Tr}(C_{XV}) = \text{Tr}(C_{U^T X}) = \text{Tr}(C_X)$  pour toutes matrices orthogonales  $U \in \mathcal{M}_n$  ou  $V \in \mathcal{M}_p$ . C’est une conséquence des propriétés mathématiques de la trace.

Lorsque l’on cherche à quantifier le degré de dépendance de deux vecteurs  $x$  et  $y$ , le coefficient de corrélation

$$r(x,y) = \frac{\text{cov}(x,y)}{\sqrt{\text{Var}\{x\}\text{Var}\{y\}}} \quad (3.20)$$

est souvent plus pertinent que la covariance de  $x$  et  $y$ . On peut de même construire à partir de la matrice de covariance une matrice de corrélations.

**Définition 3.3 — Matrice de corrélations.** La matrice de corrélations empirique d'un tableau de données  $X \in \mathcal{M}_{n,p}(\mathbb{R})$  est la matrice de covariance empirique des données centrées réduites

$$R_X = \frac{1}{n-1} Z^T Z. \quad (3.21)$$

Compte tenu de la forme Sous forme matricielle de la matrice des données centrées réduites vue à la Remarque 3.2, on peut aussi écrire la matrice de corrélations sous la forme

$$R_X = \frac{1}{n-1} S^{-1} (X - \bar{X})^T (X - \bar{X}) S^{-1} = S^{-1} C_X S^{-1}.$$

On peut noter que les éléments diagonaux de  $R_X$  sont tous égaux à 1 (puisque ce sont les variances des données centrées réduites), et que la variabilité totale associée est égale à  $p$ , le nombre de variables.



### (Matrice de corrélations)

La fonction `cor` permet de calculer la corrélation de deux vecteurs, ou la matrice de corrélations d'un tableau de données. Reprenons l'exemple précédent :

```
> X <- rnorm(100)
> Y <- 2*X + rnorm(100)/2
> XY.cor <- cor(X,Y)
> print(paste('Le coefficient de corrélation de X et Y vaut',XY.cor))
[1] "Le coefficient de corrélation de X et Y vaut 0.96345563656584"
```

La valeur très proche de 1 est normale au vu de la façon dont  $Y$  est construit à partir de  $X$ .

De façon similaire, la fonction `cor` appliquée à un tableau de données va retourner la matrice de corrélations associée.

## 3.2 Réduction des matrices

Un problème central de l'analyse d'un tableau de données  $X \in \mathcal{M}_{n,p}$  est celui de la sélection des variables pertinentes. Plus précisément

- Quelles sont les variables les plus pertinentes ?
- Sont-ce les variables elles mêmes qui sont les plus pertinentes, ou pourrait on trouver davantage de pertinence dans des combinaisons de variables ?

Creuser la seconde question amène à réfléchir à un changement de base dans l'espace des variables. En d'autres termes, peut-on trouver une matrice orthogonale  $V \in \mathcal{M}_p(\mathbb{R})$  telle qu'en posant  $Y = (X - \bar{X})V$ , on ait les deux propriétés suivantes

1. Les variances des "pseudo-variables"  $y_{\bullet j} = \{y_{1j}, y_{2j}, \dots, y_{pj}\}$  soient décroissantes : en notant  $\lambda_j^2 = \text{Var}\{y_{\bullet j}\}$

$$\lambda_1^2 \geq \lambda_2^2 \geq \dots \geq \lambda_p^2$$

2. Les "pseudo-variables"  $y_{\bullet j}$  sont décorrélées : en notant  $\lambda_j^2 = \text{Var}\{y_{\bullet j}\}$ ,

$$C_Y = \begin{pmatrix} \lambda_1^2 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2^2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_p^2 \end{pmatrix}$$

On voit donc que la question que nous avons posée revient à chercher une nouvelle base orthonormée dans  $\mathbb{R}^p$  telle que la matrice de covariance des données dans cette nouvelle base soit une matrice diagonale, dont les éléments diagonaux décroissent.

### 3.2.1 Diagonalisation des matrices carrées

En quoi les matrices diagonales sont-elles plus simples que les autres ?

Le produit de telles matrices par un vecteur est simple, car il revient à multiplier les coordonnées de ce vecteur par les éléments diagonaux de la matrice. Par exemple,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ 2y \\ 3z \end{pmatrix}.$$

Ainsi, étant donnée la base canonique  $\mathcal{B}_0 = \{E_1, E_2, E_3\}$  de  $\mathbb{R}^3$ , on voit que le produit par la matrice diagonale laisse  $E_1$  inchangé, multiplie  $E_2$  par 2 et  $E_3$  par 3. On dit dans ce cas que  $E_1, E_2$  et  $E_3$  sont des vecteurs propres de la matrice, et les valeurs propres correspondantes sont respectivement 1, 2 et 3.

**Définition 3.4 — Vecteurs propres, valeurs propres.** Soit  $A \in \mathcal{M}_p(\mathbb{R})$  une matrice carrée réelle. Un vecteur non nul  $X \in \mathbb{R}^p$  est un vecteur propre de  $A$  si il existe un nombre  $\lambda \in \mathbb{R}$  tel que

$$AX = \lambda X. \quad (3.22)$$

$\lambda$  est appelé valeur propre associée à  $X$ .

**Remarque 3.5 — Calcul des valeurs propres et vecteurs propres.** L'équation définissant les vecteurs et valeurs propres peut aussi se mettre sous la forme  $(A - \lambda I_p)X = 0$ . Ceci n'est possible que si la matrice  $A - \lambda I_p$  n'est pas inversible, donc si son déterminant est nul. On voit donc qu'un nombre  $\lambda$  ne peut être valeur propre de  $A$  que si l'équation

$$\det(A - \lambda I_p) = 0 \quad (3.23)$$

est vérifiée. Cette équation est en fait une équation polynômiale de degré  $p$ , qui peut facilement être résolue numériquement. On sait en tous cas qu'elle admet au plus  $p$  solutions. Donc une matrice carrée  $p \times p$  admet au plus  $p$  valeurs propres distinctes.

Par ailleurs, une fois une valeur propre  $\lambda$  trouvée, la détermination des vecteurs propres correspondants nécessite de résoudre un système linéaire, ce qui est simple également, tout du moins tant que  $p$  n'est pas trop grand.

**Définition 3.5 — Diagonalisabilité.** Une matrice carrée  $A \in \mathcal{M}_p(\mathbb{R})$  est diagonalisable s'il existe une matrice inversible  $F \in \mathcal{M}_p(\mathbb{R})$  et une matrice diagonale  $D \in \mathcal{M}_p(\mathbb{R})$  telles que

$$A = FDF^{-1}. \quad (3.24)$$

Les éléments diagonaux de  $D$  sont les valeurs propres de  $A$ , et les colonnes de  $F$  sont les vecteurs propres.

Nous savons que toute matrice inversible est la matrice d'une base de  $\mathbb{R}^p$  (ses colonnes sont les vecteurs de la base). Un cas particulier est le cas où la base est orthonormée. Dans ce cas, nous avons vu que la matrice  $F$  est orthogonale, c'est à dire  $F^{-1} = F^T$ .

**Théorème 3.6** Soit  $A \in \mathcal{M}_p(\mathbb{R})$  une matrice carrée réelle et symétrique (c'est à dire telle que  $A^T = A$ ). Alors  $A$  est diagonalisable, ses valeurs propres sont réelles, et la matrice  $F$  est orthogonale. En d'autres termes, il existe une base orthonormée de  $\mathbb{R}^p$  formée de vecteurs propres de  $A$ , et on peut écrire

$$A = FDF^T. \quad (3.25)$$

■ **Exemple 3.1 — Diagonalisation d'une matrice symétrique.** Considérons la matrice

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Elle est réelle et symétrique, elle est donc diagonalisable. Pour évaluer ses valeurs propres, calculons

$$\det(A - \lambda I_2) = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^2 - 1 = (3-\lambda)(1-\lambda).$$

On a donc deux valeurs propres distinctes, égales à 3 et 1.

Cherchons un vecteur propre  $V_1 = (x_1, y_1)^T$  associé à la valeur propre  $\lambda_1 = 3$  : ceci revient à dire que  $AV_1 = 3V_1$ , soit donc le système

$$\begin{cases} 2x_1 + y_1 = 3x_1 \\ x_1 + 2y_1 = 3y_1 \end{cases} \iff y_1 = x_1.$$

Donc tout vecteur de la forme  $V_1 = (x_1, x_1)^T$  convient. Si on impose la condition de normalisation  $\|V_1\| = 1$ , ceci donne  $x_1 = \pm\sqrt{2}/2$ .

Cherchons maintenant un vecteur propre  $V_2 = (x_2, y_2)^T$  associé à la valeur propre  $\lambda_2 = 1$  : ceci revient à dire que  $AV_2 = V_2$ , soit donc le système

$$\begin{cases} 2x_2 + y_2 = x_2 \\ x_2 + 2y_2 = y_2 \end{cases} \iff y_2 = -x_2.$$

Donc tout vecteur de la forme  $V_2 = (x_2, -x_2)^T$  convient. Si on impose la condition de normalisation  $\|V_2\| = 1$ , ceci donne  $x_2 = \pm\sqrt{2}/2$ .

Il est facile de vérifier que ces deux vecteurs sont orthogonaux.  $V_1$  et  $V_2$  forment une base orthonormée de  $\mathbb{R}^2$ . ■



**(Diagonalisation)** La fonction `eigen`, qui fait partie du paquet de base de R, effectue la diagonalisation des matrices lorsque c'est possible. Étant donnée une matrice carrée, l'appel à `eigen` retourne une liste contenant les valeurs propres (champ `values`, qui est un vecteur) et la base des vecteurs propres correspondants (champ `vectors`, sous forme de la matrice de la base). Par exemple :

```
> A <- matrix(c(2,1,1,2),ncol=2)
> tmp <- eigen(A)
> tmp$values
[1] 3 1
> tmp$vectors
      [,1]      [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

La matrice  $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  est diagonalisable (elle est réelle symétrique), on voit qu'elle admet 3 et 1 comme valeurs propres, et  $(\sqrt{2}/2, \sqrt{2}/2)^T$  et  $(-\sqrt{2}/2, \sqrt{2}/2)^T$  sont les vecteurs propres orthonormés correspondants.

**Remarque 3.7 — Valeurs propres multiples.** Il peut arriver que deux ou plusieurs valeurs propres soient égales. Dans ce cas, les vecteurs propres associés à une même valeur propre ne sont plus déterminés de façon unique (même en imposant que leur norme soit égale à 1).

Dans les applications à l'analyse de données que nous allons voir, ceci peut arriver, mais avec une probabilité nulle ! On ne se préoccupera donc pas de ce cas ici.

**Remarque 3.8 — Matrices non-symétriques.** Toutes les matrices carrées ne sont pas nécessairement diagonalisables dans le sens ci-dessus. dans le cas particulier d'une matrice carrée non symétrique, il peut arriver qu'elle admette des valeurs propres, mais que celles-ci soient complexes et non plus réelles.

### 3.2.2 Quid des matrices rectangulaires ?

Existe-t-il une diagonalisation pour les matrices qui ne sont pas carrées ? A strictement parler, la réponse est non. Cependant, il existe un substitut appelé décomposition en valeurs singulières, que nous allons voir plus loin. L'idée de base est la suivante.

Étant donnée une matrice rectangulaire  $X \in \mathcal{M}_{n,p}$ , les matrices  $XX^T \in \mathcal{M}_n$  et  $X^T X \in \mathcal{M}_p$  sont toutes deux des matrices carrées, symétriques de surcroît. Elles sont donc diagonalisables, et la discussion de la section précédente s'applique. On va maintenant voir cela sur un problème d'analyse de données.

### 3.3 Analyse en composantes principales

Dans toute cette section, on supposera que le nombre d'individus  $n$  est supérieur au nombre de variables :

$$n \geq p .$$

Avec les notations précédentes, soit  $C_X \in \mathcal{M}_p(\mathbb{R})$  la matrice de covariance empirique de  $X$ . Nous avons vu en (3.15) que

$$C_X = \frac{1}{n-1} (X - \bar{X})^T (X - \bar{X})$$

de sorte que  $C_X$  est symétrique :  $C_X^T = C_X$ . Le théorème 3.6 s'applique, et on en déduit l'existence d'une base orthonormée  $\mathcal{V}$  de  $\mathbb{R}^p$  constituée de vecteurs propres de  $C_X$ . Notons  $D$  la matrice diagonale formée des valeurs propres ; comme elles sont réelles, elles peuvent être ordonnées par ordre décroissant. En notant  $V$  la matrice (orthogonale) associée, on a donc  $C_X = V D V^T$ .

En posant

$$Y = (X - \bar{X})V \tag{3.26}$$

nous avons donc bien réalisé notre objectif : comme  $C_Y = V^T C_X V = V^T V D V V^T = D$ ,

- $C_Y$  est diagonale, et
- ses éléments diagonaux sont ordonnées par ordre décroissant.

Nous avons bien trouvé la base de  $\mathbb{R}^p$  qui simplifie les données. La question suivante est : qu'en faire ?

#### 3.3.1 Décomposition en valeurs singulières

On a donc obtenu de nouvelles combinaisons linéaires de variables  $Y$ , telles que  $C_Y$  est diagonale. Les éléments diagonaux de  $C_Y$  étant des variances, ils sont positifs ou nuls, on les note donc  $\lambda_k^2$ , et on peut les ordonner par ordre décroissant

$$\lambda_1^2 \geq \lambda_2^2 \geq \dots \geq \lambda_p^2 \geq 0 . \tag{3.27}$$

On note aussi  $\Lambda^2 \in \mathcal{M}_p(\mathbb{R})$  la matrice diagonale

$$\Lambda^2 = \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_p^2) = \begin{pmatrix} \lambda_1^2 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2^2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_q^2 \end{pmatrix} . \tag{3.28}$$

On a donc la factorisation

$$C_X = V \Lambda^2 V^T . \tag{3.29}$$

On va maintenant supposer, pour simplifier, que toutes les valeurs propres  $\lambda_p$  de  $C_X$  sont non nulles

$$\lambda_k > 0 \quad \forall k .$$

On peut alors, pour toute colonne  $V_k$  (vecteur à  $p$  composantes) de  $V$ , calculer

$$U_k = \frac{1}{\lambda_k \sqrt{n-1}} (X - \bar{X}) V_k \in \mathbb{R}^n . \tag{3.30}$$

En notation matricielle, ces  $p$  vecteurs forment une matrice  $n \times p$

$$U = \frac{1}{\sqrt{n-1}} (X - \bar{X}) V \Lambda^{-1} \in \mathcal{M}_{np}(\mathbb{R}) . \tag{3.31}$$

Les produits scalaires des colonnes de  $U$  sont donnés par

$$U^T U = \frac{1}{n-1} \Lambda^{-1} V^T (X - \bar{X})^T (X - \bar{X}) V \Lambda^{-1} = \Lambda^{-1} V^T C_X V \Lambda^{-1} = \Lambda^{-1} V^T C_X V \Lambda^{-1} = I_p .$$

Par conséquent, les  $p$  colonnes de  $U$  constituent une famille orthonormée dans  $\mathbb{R}^n$ . Notons que si  $p < n$ , cette famille n'est pas une base de  $\mathbb{R}^n$  car elle n'est pas constituée de  $n$  vecteurs.

**Théorème 3.9 — SVD.** Soit  $X \in \mathcal{M}_{np}(\mathbb{R})$  un tableau de données. On suppose  $n \geq p$ . Soit  $C_X$  la matrice de covariance de  $X$ , soient  $\lambda_1^2 \geq \dots \geq \lambda_p^2$  les valeurs propres de  $C_X$ , et soit  $V$  une matrice orthogonale de vecteurs propres correspondants. On suppose  $\lambda_k \neq 0$  pour tout  $k$ . Alors  $X$  peut s'écrire sous la forme

$$X^c = X - \bar{X} = \sqrt{n-1} U \Lambda V^T, \quad (3.32)$$

où

- $V \in \mathcal{M}_p(\mathbb{R})$  est la matrice d'une base orthonormée de  $\mathbb{R}^p$  (l'espace des individus)
- $\Lambda \in \mathcal{M}_p(\mathbb{R})$  est diagonale
- $U \in \mathcal{M}_{np}(\mathbb{R})$  est la matrice d'une famille orthonormée dans  $\mathbb{R}^n$  (l'espace des variables).

Une représentation graphique de la SVD d'une matrice  $X$  se trouve en FIGURE 3.1.

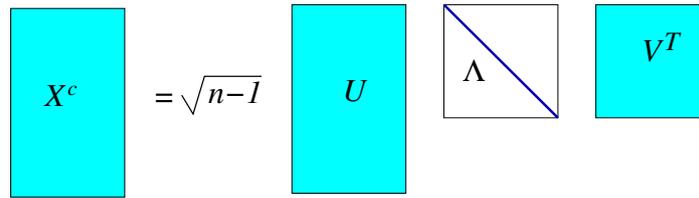


FIGURE 3.1 – Représentation graphique de la SVD d'une matrice  $n \times p$ , avec  $n \geq p$

### 3.3.2 Pourcentage de variance expliquée

La  $k$ -ème composante principale a une variance égale à  $\lambda_k^2$ . La fraction de variance expliquée par la  $k$ -ème composante principale est donc égale à

$$\rho_k = \frac{\lambda_k^2}{\lambda_1^2 + \lambda_2^2 + \dots + \lambda_p^2} = \frac{\lambda_k^2}{\sum_{i=1}^p \lambda_i^2}. \quad (3.33)$$

La fraction cumulée de variance expliquée par les  $k$  premières composantes principales est donnée par

$$\kappa_k = \frac{\lambda_1^2 + \lambda_2^2 + \dots + \lambda_k^2}{\lambda_1^2 + \lambda_2^2 + \dots + \lambda_p^2} = \frac{\sum_{i=1}^k \lambda_i^2}{\sum_{i=1}^p \lambda_i^2}. \quad (3.34)$$

Pour obtenir les pourcentages... multiplier par 100.

### 3.3.3 Représentation des individus : scores

Commençons par examiner l'espace des individus (de dimension  $p$ ). On dispose donc d'une base  $\mathcal{V} = \{V_1, \dots, V_p\}$  de cet espace, à laquelle on associe sa matrice  $V$ . Comme on l'a vu, les individus étant des vecteurs "ligne", le changement de base se fait par multiplication à droite par la matrice de la base, on pose donc comme en (3.26)

$$Y = X^c V.$$

L'interprétation de la matrice  $Y$  est la suivante :

- La première ligne de  $Y$  donne les  $p$  coordonnées du premier individu dans la base  $\mathcal{V}$ .
- Plus généralement, la  $k$ -ème ligne de  $Y$  donne les coordonnées du  $k$ -ème individu dans cette base.
- Ainsi, la première colonne de  $Y$  donne les premières coordonnées des  $n$  individus, la deuxième colonne donne les deuxièmes coordonnées, et de façon générale la  $\ell$ -ème colonne donne les  $\ell$ -èmes coordonnées des individus dans base  $\mathcal{V}$ .
- Les colonnes de  $X^c V$  sont appelés *scores* des individus.

Il est usuel de représenter ces scores sous forme de nuages de points dans le plan. Par exemple, en traçant la deuxième colonne en fonction de la première, on obtient ainsi une représentation des  $n$  individus par  $n$

points du plan engendré par les deux premières composantes principales (appelé premier plan factoriel). Cette représentation correspond en fait à la projection orthogonale des individus sur ce plan.

On peut de même tracer sous forme de nuages de points les projections sur les plans engendrés par les colonnes 1 et 3 de  $V$ , ou les colonnes 2 et 3, et ainsi de suite. Si on dispose d'outils de visualisation plus sophistiqués, on peut aussi représenter un nuage de points tridimensionnel représentant la projection sur le sous-espace engendré par les trois premières composantes.

Rappelons cependant que les composantes principales sont classées par ordre décroissant de variabilité (qui est ici un critère d'importance), on se limite donc généralement aux quelques premières.

**Remarque 3.10 — Dualité.** L'équation (3.32) du Théorème 3.9 donne aussi

$$X^c V = \sqrt{n-1} U \Lambda .$$

Ainsi, les scores (colonnes de  $Y = X^c V$ ) sont proportionnelles aux colonnes de  $U$ , c'est à dire aux vecteurs de la famille  $\{U_1, \dots, U_p\}$ . C'est ce que l'on appelle propriété de dualité de l'ACP.

### 3.3.4 Représentation des variables : *loadings*

Examinons maintenant l'espace des variables  $\mathbb{R}^n$ . On dispose d'une famille orthonormée  $\mathcal{U} = \{U_1, \dots, U_p\}$ , on note  $U \in \mathcal{M}_{n,p}$  la matrice correspondante (voir équation (3.31)). Rappelons que si  $n < p$  (inégalité stricte), ça n'est pas une base orthonormée de  $\mathbb{R}^n$ , elle engendre un sous-espace de  $\mathbb{R}^n$  de dimension au plus  $p$ .

Les coordonnées des projetés orthogonaux des colonnes de  $X^c$  sur ce sous-espace s'obtiennent en calculant

$$Z = U^T X^c ,$$

et on a une interprétation similaire à la précédente :

- La première colonne donne les  $p$  coordonnées du projeté orthogonal de la première variable sur l'axe engendré par  $U_1$ .
- De même, la  $k$ -ème colonne de  $Z$  donne les  $p$  coordonnées du projeté orthogonal de la première variable sur l'axe engendré par  $U_k$ .
- En raisonnant par ligne : la première ligne de  $Z$  donne les premières coordonnées des projetés orthogonaux des  $p$  variables sur l'axe engendré par  $U_1$ .
- De même, la  $\ell$ -ème ligne de  $Z$  donne les  $p$  coordonnées des projetés des  $p$  variables sur l'axe engendré par  $U_\ell$ .
- Les lignes de  $Z = U^T X$  sont appelés *loadings*.

On dispose ainsi des projections des variables sur les axes engendrés par les vecteurs  $U_k$  (qui sont eux aussi classés par ordre de variance  $\lambda_k^2$  décroissante). On peut donc comme précédemment représenter graphiquement ces projections sous forme de nuage de points. On verra ci-dessous quelques exemples de telles projections.

**Remarque 3.11 — Dualité.** L'équation (3.32) du Théorème 3.9 donne aussi

$$Z = U^T X^c = \sqrt{n-1} \Lambda V^T .$$

ce qui permet de relier les *loadings* aux lignes de  $V$ .

### 3.3.5 Un premier exemple : analyse chimique d'échantillons de vins en Italie

Le département de *machine learning* de l'Université d'Irvine (Californie) maintient un site web contenant un grand nombre de jeux de données "test" permettant de tester et comparer des algorithmes d'analyse de données, classification,...

On va s'intéresser ici à un jeu de données assez simple. Les données résultent d'analyses chimiques de vins cultivés dans la même région en Italie, avec trois cépages différents. Les analyses ont fourni des mesures de concentration (plus précisément, des coefficients de variation de ces concentrations) de 13 constituants dans chacun des échantillons. On va voir que ces données sont suffisamment riches pour que l'analyse en composantes principales permette de discriminer, sans information supplémentaire, les échantillons appartenant à chaque cépage.

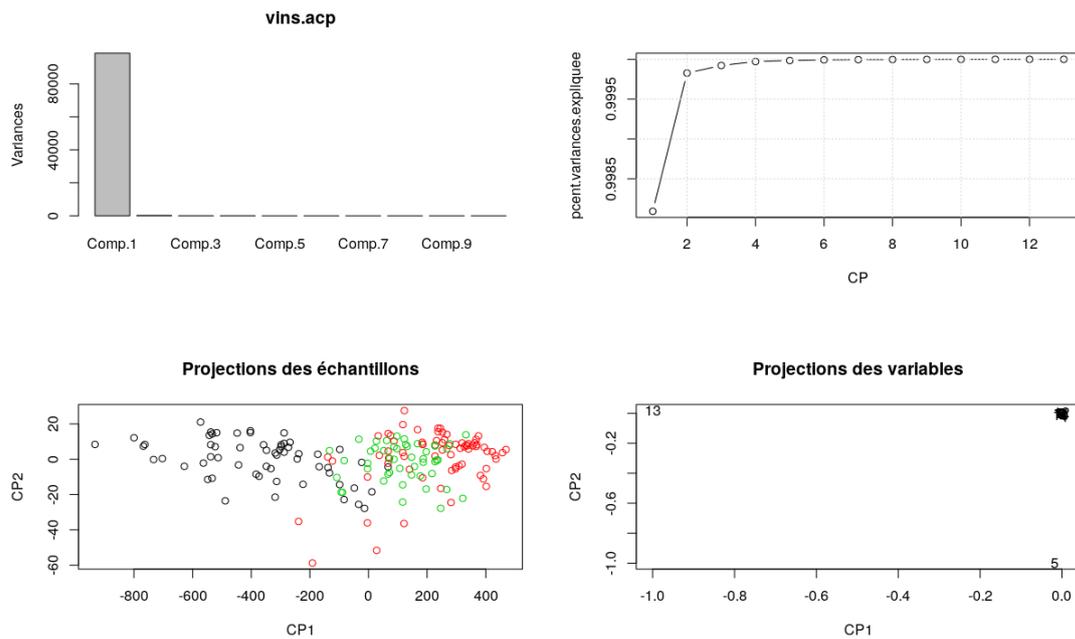


FIGURE 3.2 – ACP sur le jeu de données de vins italiens : variance expliquée individuelle et cumulée (haut), scores (bas gauche) et loadings (bas droite).

Les données se trouvent sur le site

<http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>

Nous allons maintenant examiner les étapes de l'analyse en composantes principales sur ce jeu de données, en utilisant des fonctions dédiées disponibles sous R (les plus simples, il existe d'autres fonctions permettant une analyse et des graphiques plus élaborés).



#### (Lecture et préparation des données)

On commence par télécharger les données et les informations complémentaires (noms des 13 variables) et les mettre en forme

```
> fname <- "http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"
> vins <- read.table(fname, sep=",")
> colnames(vins) <- c("Cvs", "Alcohol", "Malic acid", "Ash", "Alcalinity of ash",
  "Magnesium", "Total phenols", "Flavanoids", "Nonflavanoid phenols",
  "Proanthocyanins", "Color intensity", "Hue", "OD280/OD315 of diluted wines",
  "Proline")
> vins.classes <- factor(vins$Cvs)
> vins <- vins[,-1]
```

La dernière ligne enlève la première colonne du tableau de données, qui ne contient pas de variable mais seulement les numéros des échantillons. L'avant-dernière ligne transforme cette colonne en facteurs, ce qui sera utile plus loin.

Effectuons maintenant l'ACP. On va pour cela utiliser la plus simple des fonctions, la fonction `princomp`. Celle-ci prend au minimum comme argument le tableau de données, et retourne une liste contenant des quantités importantes, telles que `sdev`, `scores` et `loadings` que nous allons analyser ci-dessous, et d'autres quantités. La fonction `plot` appliquée à cette liste représente la première information à analyser, à savoir les variances des composantes principales.

**(ACP classique)**

```
> vins.acp <- princomp(vins)
> plot(vins.acp)
> pcent.variances.expliquee <- cumsum(vins.acp$sdev^2)/sum(vins.acp$sdev^2)
> plot(pcent.variances.expliquee,type='b')
```

La seconde ligne trace (sous forme de diagramme de barres) les variances des composantes principales (les nouvelles "pseudo-variables", combinaisons linéaires des colonnes). Les deux dernières calculent et représentent les pourcentages cumulés de variance expliquée..

Le résultat se trouve en FIGURE 3.2, tracé en haut à gauche). On voit que la première est très fortement dominante, et explique la plus grande part de la variabilité. Ceci est quantifié par le tracé en haut à droite, qui représente les pourcentages cumulés de variance expliquée par les composantes principales. On y voit en particulier que le pourcentage expliqué par la première composante est énorme, et celle-ci "écrase" toutes les autres. Ceci va poser problème, et on verra plus loin comment y remédier. Pour l'instant, continuons l'analyse.

On va maintenant s'intéresser à l'interprétation des composantes principales, en particulier la première qui est la plus significative. Pour cela, on représente les coefficients des projections des individus (les échantillons de vin) sur le plan engendré par les deux premiers vecteurs  $V_1$  et  $V_2$ .

**(Scores : représentation des individus)**

```
> plot(vins.acp$scores[,1:2],col=vins.classes,
      main='Projections des échantillons',xlab='CP1',ylab='CP2')
```

On a ici sélectionné les deux premières colonnes de `scores`, pour tracer la seconde en fonction de la première. La variable optionnelle `col` fixe les couleurs des différents points du graphe. On utilise pour cela les classes (1,2 ou 3), qui vont correspondre à des couleurs différentes.

Le résultat se trouve en FIGURE 3.2, bas gauche. On voit que le premier axe (l'axe des abscisses) permet dans une certaine mesure de séparer les trois cépages (enfin, pas vraiment séparer, mais on voit une gradation, les noirs plutôt à gauche et les rouges plutôt à droite...), alors que le second n'apporte guère d'information. L'analyse des projections sur les axes suivants n'apporte rien non plus. Ceci n'est guère surprenant, quand on voit que la première composante principale explique plus de 99% de la variabilité.

**(Loadings : représentations des variables)**

```
> plot(vins.acp$loadings[,1:2],main='Projections des variables',
      xlab='CP1',ylab='CP2',type='n')
> text(vins.acp$loadings[,1:2])
```

On a ici sélectionné les deux premières colonnes de `loadings`, pour tracer la seconde en fonction de la première. Les variables ne sont pas groupées en classes, donc ça n'a pas de sens de leur affecter une couleur différente. En revanche, il est utile de pouvoir les identifier sur le graphe. On a donc utilisé l'option `type='n'` de l'instruction `plot`, qui permet ... de ne rien représenter. En utilisant ensuite l'instruction `text` on représente les variables par leur numéro.

On voit (FIGURE 3.2, en bas à droite) que le premier axe (axe des abscisses) sépare très nettement la treizième variable des autres, alors que le second axe sépare la cinquième. En fait, un examen plus approfondi montre que ces deux variables ont une variance très grande par rapport aux autres, et elles ont donc tendance à "écraser" les autres. Ceci explique aussi la domination très forte de la première composante principale par rapport aux autres, en fait celle-ci est très fortement dominée par la treizième variable.

Pour s'abstraire des grandes différences de variabilité entre variables, il est utile pour cet exemple de réduire les variables, c'est à dire les normaliser en les divisant par leur écart-type. Il est possible de démontrer que dans ce cas, l'ACP sur données réduites est équivalente à une ACP classique, dans laquelle on diagonalise non pas la matrice de covariance mais la matrice de corrélations. C'est ce que nous allons maintenant faire.

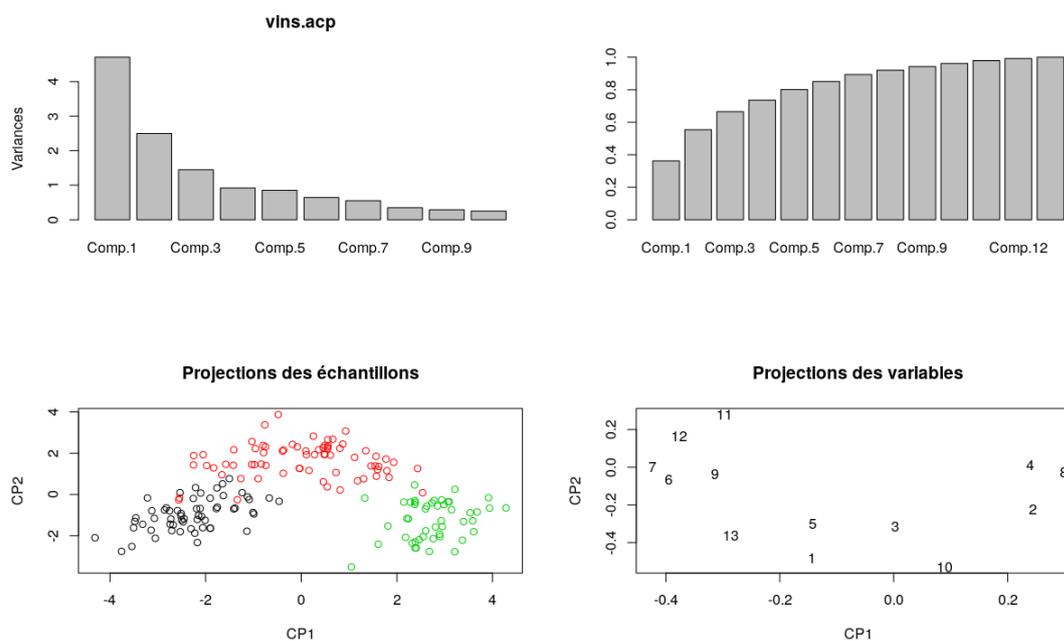


FIGURE 3.3 – ACP normée sur le jeu de données de vins italiens : variance expliquée individuelle et cumulée (haut), scores (bas gauche) et loadings (bas droite).



#### (ACP normée)

On donne ci-dessous les lignes de code équivalentes à l'analyse précédente.

```
> vins.acp <- princomp(vins,cor=T)
> plot(vins.acp)
> pcent.variances.expliquee <- cumsum(vins.acp$sdev^2)/sum(vins.acp$sdev^2)
> barplot(pcent.variances.expliquee)
> plot(vins.acp$scores[,1:2],col=vins.classes,
      main='Projections des échantillons',xlab='CP1',ylab='CP2')
> plot(vins.acp$loadings[,1:2],main='Projections des variables',
      xlab='CP1',ylab='CP2',type='n')
> text(vins.acp$loadings[,1:2])
```

Les résultats se trouvent en FIGURE 3.3. On liste ci-dessous les principales conclusions à tirer.

1. On remarque tout d'abord (haut gauche) que les variances des composantes principales sont beaucoup plus équilibrées que dans le cas précédent, la normalisation a donc pleinement joué son rôle.
2. Le tracé en haut à droite donne les pourcentages cumulés de variance expliquée, sous forme d'un diagramme en barres. Il faut environ 5 composantes principales pour expliquer 80% de la variabilité du jeu de données, et 7 composantes principales pour expliquer 90% environ. Il y a donc une différence saisissante par rapport à l'ACP classique dans laquelle la première écrasait toutes les autres... sans permettre de conclusion claire.
3. Le tracé des scores dans le premier plan factoriel (les deux premières composantes) est très intéressant à interpréter. On y voit en particulier que le premier axe (abscisses, i.e. horizontal) discrimine parfaitement la classe "noire" de la classe "verte", alors que le second axe discrimine assez bien la classe "rouge" des deux autres. Pour être complet, il est intéressant d'étudier aussi les autres plans factoriels (colonnes 1 et 3, 2 et 3,...), ce qu'on ne fait pas ici. R permet aussi de faire des scatter plots tridimensionnels, on peut alors représenter les trois premières composantes (ou d'autres) par un nuage 3D (par exemple avec le paquet `scatterplot3d`).
4. Faute de connaissances en chimie analytique, il est difficile d'interpréter finement le graphe des loadings. On peut toutefois remarquer que cette fois les variables sont relativement bien réparties dans le premier plan factoriel (aucune ne se singularise). On peut aussi remarquer que les projections

sur le premier axe des variables 2,4 et 8 s'opposent aux projections des variables 6, 7, 9, 11 et 13. On peut interpréter ceci comme un signe que cette opposition joue un rôle dans la séparation de la classe "noire" de la classe "verte".

5. Dernier point : un aspect important de l'ACP est cette *dualité* qu'elle introduit entre l'espace des variables et l'espace des individus (voir remarques 3.10 et 3.11). Ceci permet en fait de représenter variables et individus sur un même graphe, appelé *biplot* (fonction `biplot` sous R). On n'entrera pas dans les détails ici, l'interprétation de ce graphe étant souvent difficile.

### 3.3.6 Un second exemple : photos d'identité

On va s'intéresser ici à une application de l'ACP sur un jeu de données différent. Ici il n'est pas question de classification, on va principalement se focaliser sur les aspects de réduction de dimension.

On dispose d'une base de données constituée de photos d'identité de 40 individus, avec 10 photos par individu. Chacune de ces photos est une image de  $112 \times 92 = 10304$  pixels. Des exemples de photos de la base de données se trouvent dans la figure 3.4 ci-dessous



FIGURE 3.4 – Quelques exemples de photos d'identité de la base de données utilisée.

On commence par transformer ces photos en 400 vecteurs de dimension 10304 chacun, ce qui génère donc une matrice  $X \in \mathcal{M}_{10304,400}$  (on considère que les photos sont les variables, soit  $p = 400$ , et les pixels sont les individus, donc  $n = 10304$ ). La matrice de covariance  $C_X$  est de taille  $400 \times 400$ , sa diagonalisation produit une matrice  $V$  elle aussi  $400 \times 400$ , et les valeurs singulières  $\lambda_k$  représentées dans la figure 3.5. On peut y voir que ces variances décroissent relativement rapidement, sans que ce soit très marqué. Dix composantes sont suffisantes pour capturer environ 65% de la variabilité totale, alors qu'il en faut 21 pour capturer 08%.

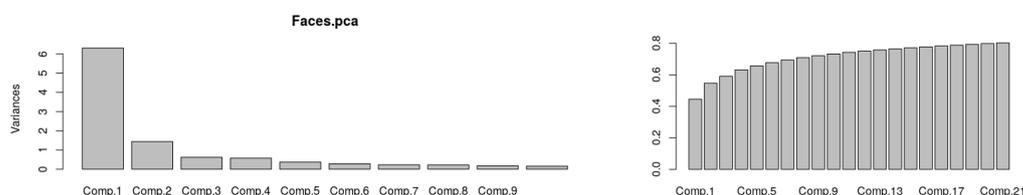


FIGURE 3.5 – Variances absolues (haut) et pourcentages de variance cumulée (bas).

En suivant le raisonnement ci-dessus, on en déduit une famille orthonormée de l'espace des images, constituée de 400 vecteurs  $U_k \in \mathbb{R}^{10304}$ , qui sont classés par ordre d'importance :

$$U_k = \frac{1}{\lambda_k \sqrt{n-1}} X^c V_k .$$

Il est intéressant d'examiner ces vecteurs (remis sous forme d'images), les premiers sont représentés en FIGURE 3.6. Ils représentent grossièrement les traits principaux des visages.



FIGURE 3.6 – Les 6 premiers "visages propres" de l'ACP, qui capturent environ 65% de la variabilité totale.



**(ACP sur une base de données de photos d'identité)** Ici le tableau  $X$  se nomme `Faces`. La fonction `princomp` effectue l'ACP sur la matrice de données, et retourne une liste contenant plusieurs éléments. L'un d'eux est `Faces$sdev`, qui contient les écarts-type. L'instruction `plot` appliquée au résultat de l'ACP représente les variances  $\lambda_k^2$  sous forme de diagramme de barres.

```
> Faces.pca <- princomp(Faces)
> plot(Faces.pca)
> pcent.cum.var <- cumsum(Faces.pca$sdev^2)/sum(Faces.pca$sdev^2)
> barplot(pcent.cum.var[1:21])
```

On représente souvent les coefficients des projections des variables (ici les 400 photos) et des individus (ici les 10304 pixels) sur les axes principaux. Dans le cas présent, ces deux graphiques ne donnent pas lieu à une interprétation simple, comme on peut le voir dans la FIGURE 3.7. On peut cependant noter une certaine régularité dans la représentation des pixels (liée au fait que des pixels proches ont tendance à prendre des valeurs proches), mais celle-ci est impossible à voir à une telle résolution.

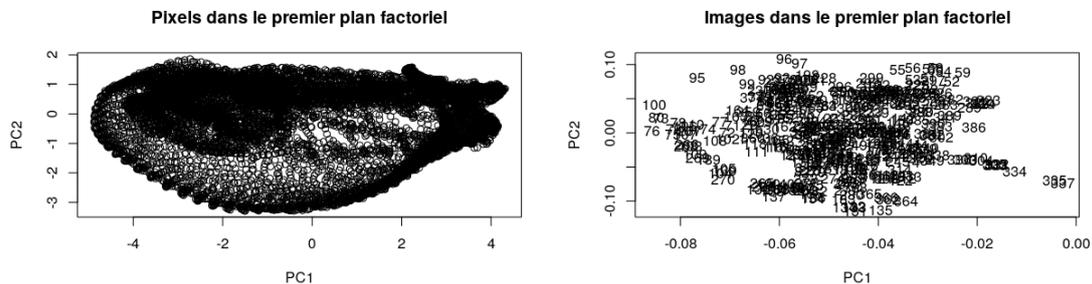


FIGURE 3.7 – Projection des individus (gauche) et des variables (droit) sur le premier plan factoriel.



**(Représentations des images et des pixels dans le premier plan factoriel)**

```
> plot(Faces.pca$scores[,1:2], xlab="PC1", ylab="PC2",
      main="Pixels dans le premier plan factoriel")
> plot(Faces.pca$loadings[,1:2], type='n', xlab="PC1", ylab="PC2",
      main="Images dans le premier plan factoriel")
> text(Faces.pca$loadings[,1:2])
```

### 3.3.7 Un troisième exemple : signaux EEG

On s'intéresse ici à un jeu de données EEG issu d'un protocole appelé *P300 speller* : on présente à l'écran un tableau de 36 caractères (26 lettres et 9 chiffres et un caractère spécial, organisé en 6 lignes et 6 colonnes). Les lignes et les colonnes sont illuminées tout à tour, dans un ordre aléatoire. Le sujet doit se concentrer sur un caractère précis (appelé caractère cible). Lorsque la ligne ou la colonne contenant le caractère cible est illuminée, un potentiel évoqué de type P300 est généré, qui est normalement mesurable sur des signaux EEG. Le problème posé par le P300 speller est de construire un algorithme permettant de détecter de façon optimale la présence ou l'absence de la P300, et donc d'identifier le caractère cible.

Les données se présentent sous la forme d'une matrice à 15300 lignes et 240 colonnes. En accord avec la convention utilisée en statistiques, les colonnes représentent les variables, et sont les valeurs des signaux sur les 240 instants consécutifs considérés. Les 15300 lignes correspondent à 15300 signaux différents : chaque signal représente la tension électrique mesurée sur le capteur Cz pendant 1 seconde après le stimulus. I y a 240 valeurs par seconde, on dit que la fréquence d'échantillonnage vaut 240 Hz (Hertz).

Dans une première analyse exploratoire, on peut s'intéresser à la moyenne de tous les signaux. Cette moyenne est représentée en FIGURE 3.8, on y retrouve une forme qui se répète de façon assez régulière, avec une période de 175 millisecondes environ. Il s'agit en fait de la trace de la réponse à chaque stimulus,

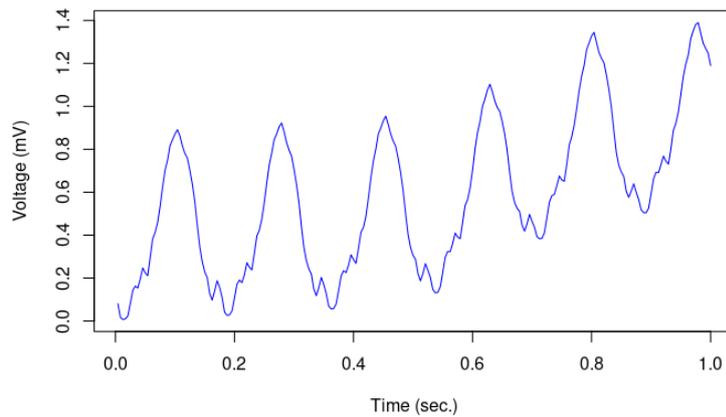


FIGURE 3.8 – Données de *P300 speller* : moyenne des 15300 signaux EEG sur le capteur Cz

apparaissant 100 millisecondes après le stimulus (qui sont présentés avec la même période de 175 msec). On ne voit par contre rien de spécifique aux alentours de 300 millisecondes après le stimulus, lieu où est attendue la P300. Ceci étant, seuls 1/5 des signaux sont censés contenir une P300, on peut donc s'attendre à ce que ces signaux soient "noyés" dans les autres.

L'analyse en composantes principales repose sur une estimation de la matrice de covariance des variables. Après diagonalisation de cette matrice, on obtient une base de vecteurs propres (des signaux de longueur 240), et les valeurs propres correspondantes. On numérote ces éléments propres par ordre décroissant de valeur propre. Les six premiers signaux sont tracés dans la FIGURE 3.9. On peut voir qu'ils présentent une allure quasiment sinusoïdale.

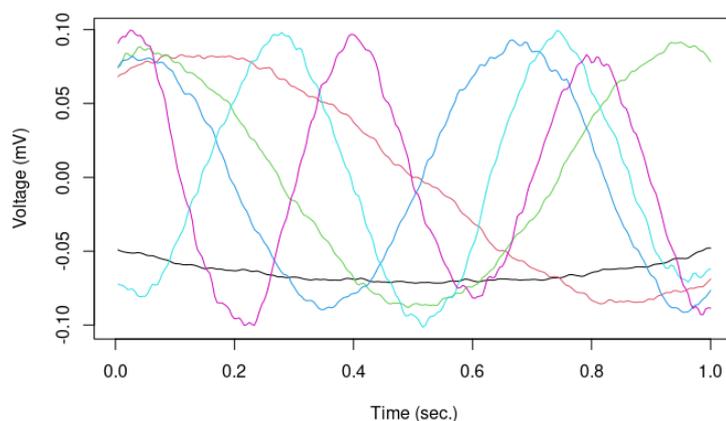


FIGURE 3.9 – Données de *P300 speller* : graphe des six premiers vecteurs propres de la matrice de covariance

Par construction, chacun des 15300 signaux EEG peut s'écrire comme combinaison linéaire de ces 240 vecteurs propres.

L'importance de chaque vecteur propre est quantifiée par les valeurs propres de la matrice de covariance. Celles-ci sont tracées en FIGURE 3.10. La figure de gauche donne les variances des composantes principales (classées par ordre décroissant), on peut voir que les deux ou trois premières sont fortement dominantes. La figure de droite donne les proportions de variance expliquées par les premières composantes cumulées. On peut voir par exemple que la première permet d'expliquer environ 40% de la variabilité, les deux premières environ 60%, et ainsi de suite.

On utilise souvent l'analyse en composantes principales comme outil de visualisation dans l'analyse exploratoire des données. Une pratique courante est de représenter les projections orthogonales des individus

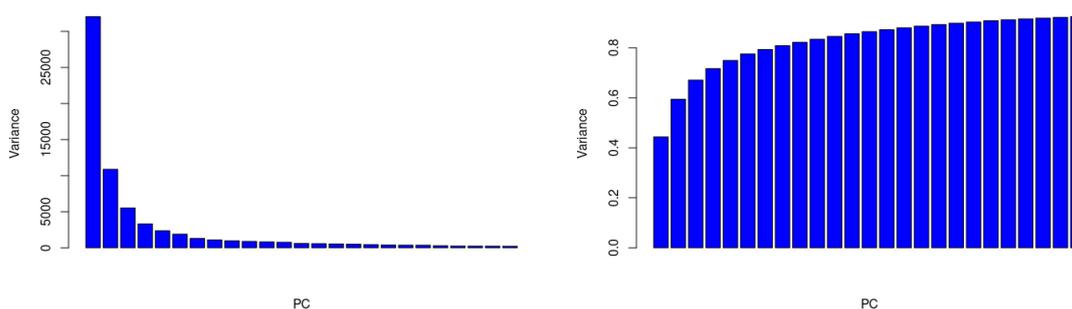


FIGURE 3.10 – Données de *P300 speller* : variances des composantes principales (gauche), et proportions de variance cumulée expliquée (droite).

(ici les 15300 signaux) sur le plan engendré par les deux premières composantes principales (puis la première et la troisième, la deuxième et la troisième... on s'arrête généralement là). Les projections orthogonales des signaux sur le premier plan sont tracées en FIGURE 3.11, tracé de gauche. Il ne semble pas y avoir de structure particulièrement spectaculaire.

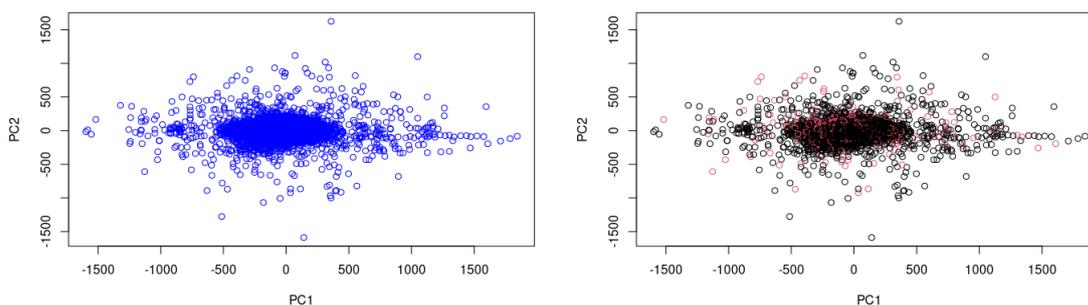


FIGURE 3.11 – Données de *P300 speller* : projections orthogonales des 15300 signaux sur le plan engendré par les deux premières composantes principales. Gauche : sans information de classe. Droite : avec information de classe (Noir : signaux non P300 ; rouge : signaux P300).

Dans la mesure où pour ce jeu de données, les classes des signaux sont connues (on sait lesquels sont censés contenir une P300 et quels sont ceux qui normalement n'en contiennent pas), il est possible de représenter leurs projections avec des couleurs différentes, ce qui est fait en FIGURE ??, tracé de droite. Comme on peut le constater, l'ACP ne fournit pas de séparation visible des deux classes. Des projections sur les plans engendrés par les composantes 1 et 3, ou 2 et 3, ou d'autres, ne permettent pas non plus de faire apparaître de séparation. L'utilisation de l'analyse en composante principale comme méthode de classification n'est donc pas adaptée à ce jeu de données.

On peut alors se demander si les signaux des deux classes sont suffisamment différents pour pouvoir être discriminés. Pour vérifier cela, on trace la moyenne des signaux contenant une P300 et la moyenne des signaux qui ne contiennent pas de P300. Ces graphes se trouvent en FIGURE 3.12. On voit que ces deux moyennes apparaissent en effet très différentes, il devrait donc être possible de distinguer ces signaux. On verra dans le chapitre qui suit une méthode spécifiquement destinée à effectuer de telles séparations. A l'inverse de l'ACP, qui est une méthode non supervisée (c'est à dire qui n'utilise pas d'information autre que les données), les méthodes du chapitre suivant seront des méthodes supervisées : elles reposent sur un jeu de données dit jeu d'apprentissage sur lequel sera construit l'algorithme, et un jeu de test qui servira à l'analyse des performances.

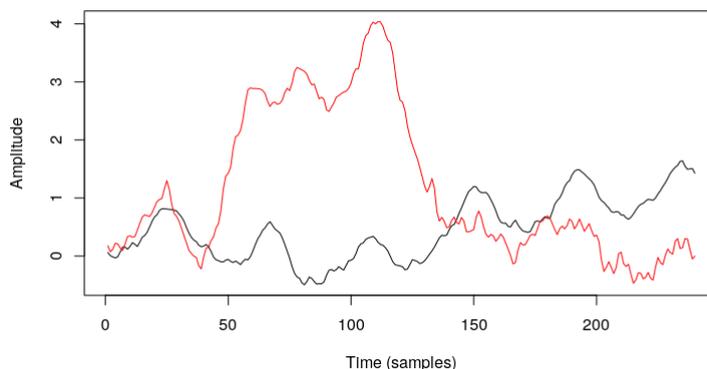


FIGURE 3.12 – Données de *P300 speller* : moyenne des 12750 signaux EEG non P300 (noir) et des 2550 signaux P300 (rouge) sur le capteur Cz.

### 3.4 Résumé

On reprend ici les principaux concepts et résultats décrits dans ce chapitre.

- Notre point de départ est un tableau de données, modélisé par une matrice  $X \in \mathcal{M}_{n,p}$  à  $n$  lignes et  $p$  colonnes. Les lignes correspondent conventionnellement à des individus (ou échantillons) et les colonnes correspondent à des variables. On a supposé ici  $n \geq p$ , qui est le cas de figure le plus courant.
- On associe aux variables des mesures de position (moyenne de chaque variable) et de dispersion, ou variabilité (écart-type). Les mesures classiques sont la moyenne (3.5) et la variance (3.9) (ou sa racine carrée, l'écart type) des variables. On peut parfois recourir à d'autres mesures : par exemple la médiane au lieu de la moyenne, ou la distance inter-quartiles au lieu de l'écart-type. Ces mesures sont moins sensibles aux valeur aberrantes qui peuvent parfois se trouver dans des tableaux de données. On s'est limité ici à moyenne et variance.
- Les covariances (3.14) fournissent des mesure de co-variation entre variables. Ces covariances sont organisées sous forme d'une matrice  $C_X$ , appelée *matrice de covariance* (3.15). Cette matrice  $C_X$  est symétrique, et semi-définie positive. Elle joue un rôle fondamental en analyse de données. Les éléments diagonaux de la matrice de covariance sont les variances des variables.
- Ces propriétés de la matrice de covariance assurent l'existence d'une décomposition en éléments propres :
  - Il existe une base orthonormée  $\mathcal{V} = \{V_1, \dots, V_p\}$  de l'espace des individus (c'est à dire l'espace  $\mathbb{R}^p$  des lignes) qui diagonalise  $C_X$  : on écrit

$$C_X = VC_YV^T,$$

où  $V \in \mathcal{M}_p$  est la matrice associée à la base  $\mathcal{V}$ , et  $C_Y = \Lambda^2 \in \mathcal{M}_p$  est une matrice diagonale (voir équation (3.28)), dont les éléments diagonaux sont positifs ou nuls<sup>1</sup>.

- Les éléments diagonaux de  $C_Y = \Lambda^2$  sont conventionnellement ordonnées par ordre décroissant (voir équation (3.27)).
- La base  $\mathcal{V}$  de l'espace des individus permet de construire une famille orthonormée  $\mathcal{U}$  de l'espace des variables, à laquelle on associe la matrice rectangulaire  $U \in \mathcal{M}_{n,p}$ .
- Ces différentes matrices sont reliées par la décomposition en valeurs singulières (3.32).
- Ces deux bases permettent de représenter et étudier les projetés orthogonaux des variables et des individus sur les sous-espaces qu'elles engendrent. On s'intéresse en général aux axes et les plans engendrés par les premiers vecteurs.
  - On appelle *scores* des individus les coordonnées de leurs projetés orthogonaux sur les axes définis par les vecteurs  $V_1, V_2, \dots$ . Les scores sont les colonnes de la matrice  $Y = X^cV$ , et peuvent

1. On rappelle que comme la base  $\mathcal{V}$  est une base orthonormée de  $\mathbb{R}^p$ , on a  $V^{-1} = V^T$ .

être interprétés comme des *pseudo-variables*, qui sont décorrélées (leur matrice de covariance  $C_Y = \Lambda^2$  est diagonale).

- Les *loadings* des variables sont les coordonnées de leurs projetés orthogonaux sur les axes définis par les vecteurs  $U_1, \dots, U_p$ . Ce sont aussi les lignes de la matrice  $U^T X^c$ .

## 4. Optimisation, analyse discriminante

Les mathématiques appliquées font très largement appel à la notion d'optimisation. Supposons par exemple qu'une méthode dépendant d'un paramètre  $\theta$  soit appliquée à des données. On peut alors se demander pour quelle valeur du paramètre la méthode donne de meilleurs résultats. Ceci sous-entend qu'on définit une quantité qui décrit la qualité du résultat. Cette quantité, qu'on appellera fonction objective associée à toute valeur de  $\theta$  la mesure de qualité correspondante, qu'on notera par exemple  $f(\theta)$ . On cherchera alors quelle est la valeur de  $\theta$  qui rend cette quantité maximale. On notera  $\theta_*$  cette valeur, et on écrira (lire  $\theta_*$  est la valeur de  $\theta$  qui maximise  $f(\theta)$ ).

$$\theta_* = \arg \max_{\theta} f(\theta) .$$

C'est ce que l'on appelle un problème de maximisation, ou plus généralement problème d'optimisation.

### 4.1 Rappels d'analyse

#### 4.1.1 Fonctions

On rappelle qu'une fonction d'une variable réelle est une application

$$f : x \in \mathcal{D}_f \mapsto f(x) \in \mathbb{R} ,$$

où  $\mathcal{D}_f \subset \mathbb{R}$  est l'ensemble de définition de  $f$ . Par exemple, la fonction  $f(x) = 1/x$  admet comme ensemble de définition  $\mathcal{D}_f = \mathbb{R}^*$ , l'axe réel privé de son origine. La fonction  $g(x) = \exp(x)$  est définie pour tout  $x \in \mathbb{R}$ , son domaine de définition est  $\mathcal{D}_g = \mathbb{R}$ . La fonction  $h(x) = \ln(x)$  est définie pour tout  $x$  réel strictement positif, son domaine de définition est  $\mathcal{D}_h = \mathbb{R}_+^*$ .

De même, une fonction de deux variables réelles est une application

$$f : (x, y) \in \mathcal{D}_f \mapsto f(x, y) \in \mathbb{R}$$

associant à tout couple  $(x, y) \in \mathcal{D}_f$  un réel  $f(x, y)$ .  $\mathcal{D}_f \subset \mathbb{R}^2$  est aussi appelé domaine de définition de  $f$ .

■ **Exemple 4.1** — Si on considère un rectangle de côtés  $x, y$ , on sait que son aire vaut  $xy$  et son périmètre vaut  $2(x+y)$ . Ces deux quantités définissent deux fonctions

$$a : (x, y) \mapsto a(x, y) = xy , \quad p : (x, y) \mapsto p(x, y) = 2(x+y) .$$

Les domaines de définition de ces deux fonctions sont identiques, il s'agit de l'ensemble des tous les couples de réels.

— La fonction

$$f : (x, y) \in \mathcal{D}_f \mapsto \ln(x) + \sqrt{1+y}$$

admet pour domaine de définition  $\mathcal{D}_f = \mathbb{R}_+^* \times [-1, +\infty[$ , où " $\times$ " représente le produit Cartésien. ■

**Définition 4.1 — Fonction de plusieurs variables réelles.** Soit  $\mathcal{D}$ , une partie de  $\mathbb{R}^n$ , c'est à dire un ensemble de vecteurs de réels  $(x_1, x_2, \dots, x_n)^T$ . On appelle fonction de plusieurs variables définie sur  $\mathcal{D}$ , le procédé qui consiste à associer à chaque couple  $(x_1, x_2, \dots, x_n)^T \in \mathcal{D}$  un réel unique. On note généralement :  $f(x_1, x_2, \dots, x_n) = z$ .

On associe à une fonction d'une variable réelle  $f$  définie sur  $\mathcal{D} \subset \mathbb{R}$  l'ensemble

$$\mathcal{C} = \{(x, f(x)), x \in \mathcal{D}\}$$

appelé graphe de  $f$ , ou courbe de  $f$ .

De même, étant donnée une fonction  $f$  définie sur  $\mathcal{D} \subset \mathbb{R}^2$ , l'ensemble

$$\mathcal{C} = \{(x, y, f(x, y)), (x, y) \in \mathcal{D}\}$$

est le graphe de  $f$ , aussi appelé la surface associée à  $f$ .

De façon générale, étant donnée une fonction de plusieurs variables réelles  $f$ , on lui associe son graphe défini par

$$\mathcal{C} = \{(x, x_2, \dots, x_n, f(x)), (x_1, x_2, \dots, x_n)^T \in \mathcal{D}\}. \quad (4.1)$$

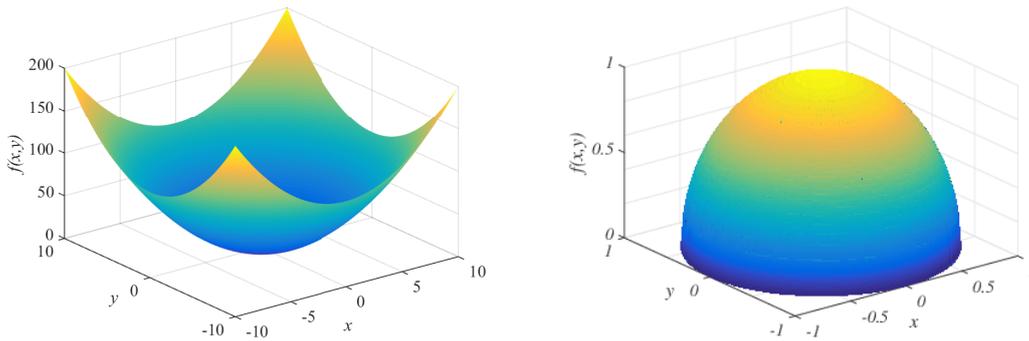


FIGURE 4.1 – Graphes de la fonction  $f(x, y) = x^2 + y^2$  (gauche) et la fonction  $\sqrt{1 - x^2 - y^2}$  (droite).

Quelques mots maintenant sur la notion de limite en un point, et la notion de continuité en un point.

Pour les fonctions d'une variable réelle, la notion de continuité est assez intuitive. On trouve dans certains manuels de lycée une définition littérale (mais tout à fait correcte) du type

Soit  $f$  une fonction d'une variable réelle, définie dans un voisinage de  $a$ . L'expression

$$\lim_{x \rightarrow a} f(x) = \ell$$

traduit le comportement suivant. Étant donné un réel positif quelconque, que l'on peut prendre aussi petit que l'on veut (et qu'on note traditionnellement  $\varepsilon$ ), il suffit que  $x$  s'approche suffisamment près de  $a$  pour assurer que  $|f(x) - \ell|$  soit plus petit que ce nombre  $\varepsilon$ .

On peut donner une explication similaire pour des fonctions qui tendent vers  $\pm\infty$  en un point. Par exemple

Soit  $f$  une fonction d'une variable réelle, définie dans un voisinage de  $a$ . L'expression

$$\lim_{x \rightarrow a} f(x) = +\infty$$

traduit le comportement suivant. Étant donné un réel positif quelconque  $A$ , que l'on peut prendre aussi grand que l'on veut, il suffit que  $x$  s'approche suffisamment près de  $a$  pour assurer que  $f(x)$  soit plus grand que ce nombre  $A$ .

Ces définitions littérales restent valables quand on les applique à des fonctions de plusieurs variables. La difficulté supplémentaire tient à la façon dont on définit l'expression "s'approcher près de a".

- Dans le cas de fonctions d'une variable, on peut en gros s'approcher de  $a$  par la droite ou par la gauche. On peut montrer que  $b$  est limite de  $f(x)$  quand  $x$  tend vers  $a$  si et seulement si  $b$  est limite de  $f(x)$  quand  $x$  tend vers  $a$  par valeurs supérieures (par la droite donc) et par valeurs inférieures. Autrement dit, il faut que les deux limites (à gauche et à droite) existent, et qu'elles soient égales.
- Dans le cas de fonctions de 2 variables ou plus, on peut s'approcher d'un point  $a \in \mathcal{D}_f$  d'une infinité de façons : par la gauche, par la droite, par le haut, en diagonale... la notion de limite est donc plus compliquée...

Pour être complet, on donne ici la définition formelle de la limite d'une fonction de plusieurs variables. **Les lecteurs qui ne sont pas passionnés par les formulations abstraites peuvent se passer de cette définition.** On y utilise la notation suivante : la boule  $\mathcal{B}(A, \delta) \subset \mathbb{R}^n$  centrée sur  $A$  de rayon  $\delta$  dans  $\mathbb{R}^n$  est définie par

$$\mathcal{B}(A, \delta) = \{x \in \mathbb{R}^n, \|x - A\| < \delta\} . \quad (4.2)$$

**Définition 4.2 — Limite.** Soient  $\mathcal{D}$  une partie ouverte de  $\mathbb{R}^n$ ,  $A \in \mathcal{D}$  et  $f$  une fonction définie sur  $\mathcal{D}$ , éventuellement non définie en  $A$ , à valeurs réelles.

1. On dit que  $f$  a pour limite  $\ell$  au point  $A$ , ce que l'on écrit

$$\lim_{x \rightarrow A} f(x) = \ell ,$$

si pour tout  $\varepsilon > 0$  il existe  $\delta > 0$  tel que  $x \in \mathcal{D} \setminus \{A\}$  et  $\|x - A\| < \delta$  impliquent  $|f(x) - \ell| < \varepsilon$ , autrement dit

$$x \in \mathcal{D} \cap \mathcal{B}(A, \delta) \implies |f(x) - \ell| < \varepsilon$$

2. On dit que  $f$  tend vers  $+\infty$  quand  $x$  tend vers  $A$ , ce que l'on écrit

$$\lim_{x \rightarrow A} f(x) = +\infty ,$$

si pour tout  $M > 0$  il existe  $\delta > 0$  tel que  $x \in \mathcal{D} \setminus \{A\}$  et  $\|x - A\| < \delta$  impliquent  $f(x) > M$ , autrement dit

$$x \in \mathcal{D} \cap \mathcal{B}(A, \delta) \implies |f(x) - \ell| > M .$$

3. On dit que  $f$  tend vers  $-\infty$  quand  $x$  tend vers  $A$ , ce que l'on écrit

$$\lim_{x \rightarrow A} f(x) = -\infty ,$$

si pour tout  $M < 0$  il existe  $\delta > 0$  tel que  $x \in \mathcal{D} \setminus \{A\}$  et  $\|x - A\| < \delta$  impliquent  $f(x) > M$ , autrement dit

$$x \in \mathcal{D} \cap \mathcal{B}(A, \delta) \implies |f(x) - \ell| < M .$$

La notion de limite permet de définir simplement la notion de continuité d'une fonction.

**Définition 4.3 — Continuité.** Soit  $f$  une fonction de  $\mathcal{D} \subset \mathbb{R}^n$  dans  $\mathbb{R}$ . On dit que  $f$  est continue en  $z = (x_1, x_2, \dots, x_n) \in \mathcal{D}$  si  $f$  possède en  $z$  une limite égale à  $f(z)$ . Si  $f$  est continue en chaque point de  $\mathcal{D}$ , on dit que  $f$  est continue sur  $\mathcal{D}$ .

**Remarque 4.1 — Comment savoir si une fonction est continue.** Il existe un certain nombre de cas simples pour lesquels il n'est pas nécessaire de revenir à la définition... heureusement !

- Les fonctions élémentaires telles que les polynômes, les fonctions exponentielles, logarithmiques et trigonométriques sont continues dans leurs domaines de définition respectifs.
- La continuité de certaines autres fonctions peut souvent s'établir en tant que somme, produit, composée, le quotient (lorsque le dénominateur ne s'annule pas) etc., de fonctions continues :
  1. Si  $f$  (définie sur  $\mathcal{D}_f$ ) et  $g$  (définie sur  $\mathcal{D}_g$ ) sont continues, alors  $f + g$  est continue sur  $\mathcal{D}_f \cap \mathcal{D}_g$ .

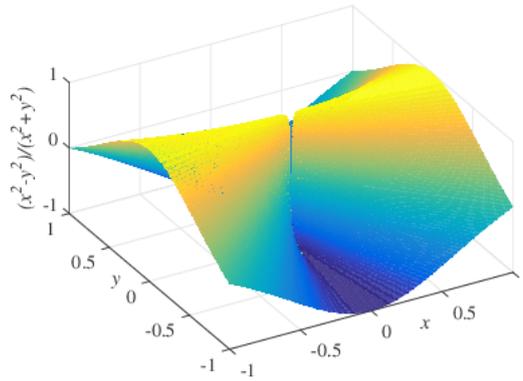


FIGURE 4.2 – Graphe d'un exemple de fonction discontinue :  $f(x,y) = (x^2 - y^2)/(x^2 + y^2)$ .

2. Si  $f$  (définie sur  $\mathcal{D}_f$ ) et  $g$  (définie sur  $\mathcal{D}_g$ ) sont continues, alors  $fg$  est continue sur  $\mathcal{D}_f \cap \mathcal{D}_g$ .
3. Si  $f$  (définie sur  $\mathcal{D}_f$ ) et  $g$  (définie sur  $\mathcal{D}_g$ ) sont continues, alors  $f/g$  est continue sur  $\mathcal{D}_f \cap \mathcal{D}_g^\circ$ , où  $\mathcal{D}_g^\circ$  est le sous-ensemble de  $\mathcal{D}_g$  dans lequel  $g(x,y) \neq 0$ .

- **Exemple 4.2**
1.  $f(x,y) = x^2 + y^2 - xy + y$  est continue dans  $\mathbb{R}^2$  (polynôme du second degré à deux variables).
  2.  $f(x,y) = e^y + xy^2$  est continue dans  $\mathbb{R}^2$  (somme d'une exponentielle et d'un polynôme).
  3.  $f(x,y) = \ln(x+y^2) - 3$  est continue dans  $\mathcal{D} = \{(x,y) \in \mathbb{R}^2 : x+y^2 > 0\}$  comme somme du logarithme d'un polynôme (fonction composée) et d'une constante.
  4. La fonction  $f(x,y) = 1/(x-y)$  est continue sur son domaine de définition  $\mathbb{R}^2 \setminus \{(x,x), x \in \mathbb{R}\}$ .

- **Exemple 4.3** Soit  $f$  définie sur  $\mathcal{D} = \mathbb{R}^2 \setminus \{(0,0)\}$  par

$$f(x,y) = \frac{x^2 - y^2}{x^2 + y^2}$$

Si on prend la limite  $y \rightarrow 0$  le long de l'axe  $x = 0$  on trouve -1, alors que la limite le long de l'axe  $y = 0$  vaut 1. Les deux limites ne coïncident pas, donc

$$\lim_{(x,y) \rightarrow (0,0)} f(x,y) \text{ n'existe pas.}$$

On peut visualiser ce phénomène sur le graphe donné en figure 4.2, qui présente un pincement à l'origine.

## 4.1.2 Dérivation, gradient

### Fonctions d'une variable réelle

Commençons par revoir ces notions dans le cas de fonctions d'une variable réelle. Une fonction  $f$  est dérivable en un point  $x_0 \in \mathcal{D}_f$  si on peut lui associer une tangente en ce point. Cette tangente est une droite d'équation  $y = ax + b$ . La pente de cette droite définit alors la dérivée de  $f$  en  $x_0$ , que l'on note  $f'(x_0)$ . Si  $f$  est différentiable sur un intervalle, ceci permet de définir une fonction dérivée  $f' : x \rightarrow f'(x)$ . Un exemple est donné en FIGURE 4.3, où sont représentées les tangentes en deux points.

Mathématiquement, la dérivée est définie, lorsqu'elle existe par un calcul de limite

$$f'(x_0) = \lim_{\varepsilon \rightarrow 0} \frac{f(x_0 + \varepsilon) - f(x_0)}{\varepsilon}. \quad (4.3)$$

Si une fonction différentiable atteint un extremum (maximum ou un minimum) en un point  $x_0$  (à l'exception des bords de son domaine de définition), la tangente au graphe en ce point est nécessairement horizontale :  $f'(x_0) = 0$ . On peut en voir un exemple en FIGURE 4.3. Le fait que ce point corresponde à un maximum est lié à la concavité du graphe autour du point.

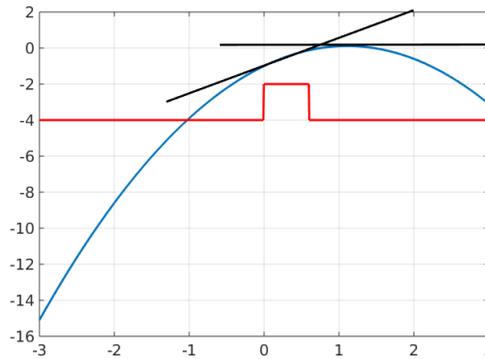


FIGURE 4.3 – Graphes de deux fonctions définies sur l'intervalle  $[-3, 3]$ . La fonction dont le graphe est rouge est discontinue en deux points, la fonction dont le graphe est bleu est continue et différentiable : il existe une tangente au graphe en chacun de ses points.

**Remarque 4.2** La condition ci-dessus est une condition nécessaire, pas suffisante. Il peut très bien exister des cas où  $f'(x_0) = 0$  sans que  $x_0$  soit un extremum de  $f$ . Par exemple, pour  $f(x) = x^3$ , on a  $f'(x) = 3x^2$ , qui s'annule en  $x = 0$ , mais ce point n'est pas un extremum (c'est un point d'inflexion).

#### Fonctions de deux variables réelles

Commençons par le cas de fonctions de deux variables, ceci va nous permettre de voir quelles sont les difficultés nouvelles. Le cas de fonctions de plus de deux variables se traitera de façon similaire.

Étant donnée une fonction de deux variables  $f : (x_1, x_2) \in \mathcal{D}_f \subset \mathbb{R}^2 \rightarrow f(x_1, x_2)$ , on appelle fonction partielle  $f_{\xi_2}$  une fonction de la forme

$$f_{\xi_2} : x_1 \rightarrow f_{\xi_2}(x) = f(x_1, \xi_2),$$

où  $\xi_2$  est fixé.  $f_{\xi_2}$  est donc une fonction d'une variable réelle. Cette fonction n'est définie que sur l'ensemble des  $x \in \mathbb{R}$  tels que  $(x, \xi_2) \in \mathcal{D}_f$ . On définira la dérivée partielle de  $f$  dans la direction  $x_1$  comme la dérivée (usuelle) de la fonction  $f_{\xi_2}$  par rapport à  $x_1$ . Géométriquement, ceci revient à faire une coupe sur le graphe de  $f$  (qui est ici une surface dans l'espace tri-dimensionnel) par un plan, comme représenté sur la FIGURE 4.4, où on a pris le plan d'équation  $x_2 = \xi_2 \approx 0.5$ . Cette intersection est une courbe, et on s'intéresse aux tangentes à cette courbe en différents points (différentes valeurs de  $x_1$ ). Les pentes de ces tangentes sont les valeurs de la dérivée partielle de  $f$  aux points  $(x_1, \xi_2)$ , que l'on peut définir formellement par passage à la limite

$$\frac{\partial f}{\partial x_1}(x_1, x_2) = \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2) - f(x_1, x_2)}{h}. \quad (4.4)$$

En d'autres termes, on garde fixée la valeur de la seconde coordonnée et on dérive par rapport à la première.

De même, la fonction partielle  $f_{\xi_1}$  est définie par

$$f_{\xi_1} : x_2 \rightarrow f_{\xi_1}(y) = f(\xi_1, x_2),$$

définie sur l'ensemble des  $y \in \mathbb{R}$  tels que  $(\xi_1, x_2) \in \mathcal{D}_f$ . On peut là encore voir le graphe de cette fonction partielle comme l'intersection du graphe de  $f$  (surface) avec le plan d'équation  $x = \xi_1 \approx 1$  (sur la figure 4.4). La dérivée partielle de  $f$  par rapport à  $x_2$  est la dérivée usuelle de  $f_{\xi_1}$  par rapport à  $x_2$ , et s'écrit

$$\frac{\partial f}{\partial x_2}(x_1, x_2) = \lim_{h \rightarrow 0} \frac{f(x_1, x_2 + h) - f(x_1, x_2)}{h}. \quad (4.5)$$

Ces deux dérivées partielles peuvent être assemblées dans un vecteur à deux coordonnées, appelé gradient de  $f$ , et noté

$$\nabla f(x_1, x_2) = \left( \begin{array}{c} \frac{\partial f}{\partial x_1}(x_1, x_2) \\ \frac{\partial f}{\partial x_2}(x_1, x_2) \end{array} \right). \quad (4.6)$$

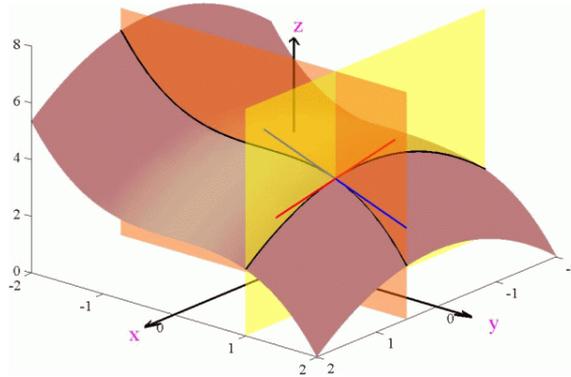


FIGURE 4.4 – Illustration des dérivées partielles d'une fonction de deux variables réelles.

On dit que le gradient d'une fonction est un champ de vecteurs, puisqu'il définit un vecteur (a priori différent) en chaque point  $(x_1, x_2) \in \mathbb{R}^2$ .

#### Fonctions de plusieurs variables réelles

On considère maintenant un cas plus général de fonctions

$$f : X = (x_1, \dots, x_n) \in \mathcal{D}_f \rightarrow f(X),$$

de domaine de définition  $\mathcal{D}_f \subset \mathbb{R}^n$ .

**Définition 4.4 — Dérivées partielles.** Les dérivées partielles de  $f : \mathcal{D}_f \rightarrow \mathbb{R}$  sont les dérivées (au sens usuel) des fonctions partielles de  $f$  :

$$\frac{\partial f}{\partial x_k}(x_1, \dots, x_n) = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_k + h, \dots, x_n) - f(x_1, \dots, x_k, \dots, x_n)}{h}$$

Si toutes les dérivées partielles premières de  $f$  en  $(x_1, \dots, x_n)$  existent, on dit que  $f$  est dérivable en  $(x_1, \dots, x_n)$ .  $f$  est dérivable dans  $\mathcal{D}_f$  si  $f$  est dérivable en  $(x_1, \dots, x_n)$ , pour tout  $(x_1, \dots, x_n) \in \mathcal{D}_f$ .

Ces dérivées partielles peuvent être assemblées dans un vecteur à  $n$  coordonnées, appelé gradient de  $f$ , et noté

$$\nabla f(x_1, \dots, x_n) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x_1, \dots, x_n) \\ \frac{\partial f}{\partial x_2}(x_1, \dots, x_n) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x_1, \dots, x_n) \end{pmatrix}. \quad (4.7)$$

Là encore, le gradient d'une fonction de  $n$  variables est un champ de vecteurs, il définit un vecteur (a priori différent) en chaque point  $(x_1, \dots, x_n) \in \mathbb{R}^n$ .

■ **Exemple 4.4** Soit  $f(x_1, x_2, x_3) = 3x_1^3 + x_1x_2 - 2x_2^4 + x_3^9$ . Alors  $\mathcal{D}_f = \mathbb{R}^3$ , et  $f$  est une fonction dérivable. On a

$$\frac{\partial f}{\partial x_1}(x_1, x_2, x_3) = 9x_1^2 + x_2, \quad \frac{\partial f}{\partial x_2}(x_1, x_2, x_3) = x_1 - 8x_2^3, \quad \frac{\partial f}{\partial x_3}(x_1, x_2, x_3) = 9x_3^8.$$

La dérivation partielle étant une opération de dérivation d'une fonction d'une variable réelle, les propriétés classiques se généralisent au cas multidimensionnel.

### 4.1.3 Dérivées d'ordre supérieur

Chaque dérivée partielle  $\partial f / \partial x_k$  d'une fonction dérivable définit une nouvelle fonction. Si celle-ci est encore dérivable, on peut en calculer les dérivées partielles, que l'on note

$$\frac{\partial^2 f}{\partial x_k \partial x_\ell} = \frac{\partial}{\partial x_k} \frac{\partial f}{\partial x_\ell}.$$

Si toutes les dérivées partielles d'ordre deux existent on dit que la fonction est deux fois dérivable.

Heureusement, les choses se simplifient, grâce au résultat suivant qui nous dit que si les dérivées partielles multiples existent, dériver par rapport à  $x_k$  puis à  $x_\ell$  équivaut à dériver par rapport à  $x_\ell$  puis  $x_k$ .

**Théorème 4.3 — Théorème de Schwartz.** Si  $f$  est deux fois dérivable en  $X \in \mathbb{R}^n$ , alors pour tous  $k, \ell = 1, \dots, n$  on a

$$\frac{\partial^2 f}{\partial x_k \partial x_\ell} = \frac{\partial^2 f}{\partial x_\ell \partial x_k}.$$

Les dérivées partielles secondes forment une matrice, appelée matrice Hessienne de  $f$ .

$$Hf(x_1, \dots, x_n) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \quad (4.8)$$

On dit que la matrice Hessienne (ou tout simplement Hessienne) est en fait un *champ de matrices*, une fonction deux fois dérivable admet une Hessienne en chaque point.

**Remarque 4.4** Le théorème de Schwartz montre que la matrice Hessienne est une matrice symétrique. Comme c'est une matrice à coefficients réels, on peut en déduire, par le Théorème 3.6 qu'elle est diagonalisable, et que ses valeurs propres sont réelles. Cette propriété joue un rôle important dans les problèmes d'optimisation que nous allons maintenant étudier.

**Remarque 4.5 — Notations.** La notation  $\nabla$  pour le gradient est une notation très pratique, si on la met en correspondance avec le calcul matriciel. On note généralement

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix}.$$

De là, en utilisant les notations du calcul matriciel, on peut calculer

$$\Delta = \nabla^T \nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$$

qui est appelé opérateur Laplacien, ou tout simplement Laplacien. Le Laplacien joue un rôle fondamental en mathématiques.

Similairement, on peut aussi calculer

$$\nabla \nabla^T = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} \\ \frac{\partial^2}{\partial x_2 \partial x_1} & \frac{\partial^2}{\partial x_2^2} & \cdots & \frac{\partial^2}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} & \frac{\partial^2}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2}{\partial x_n^2} \end{pmatrix}$$

qui est appelé Opérateur Hessien.

#### 4.1.4 Calcul différentiel pour les nuls

Comment fait-on pour calculer des dérivées partielles et des gradients ? Fort heureusement, on n'est pas obligés de revenir à la définition à chaque fois, il existe une grande quantité de fonctions classiques dont les dérivées sont connues (polynômes, fonctions trigonométriques, exponentielle, logarithme, fonctions spéciales,...). Il existe aussi (et c'est même le cas le plus général) énormément de fonctions dont la dérivée n'admet pas de forme simple, on doit alors avoir recours au calcul numérique pour évaluer les dérivées.

On va ici donner quelques éléments permettant de calculer des gradients dans les cas les plus simples, qui nous seront utiles par la suite. On va pour cela recourir au calcul matriciel.

■ **Exemple 4.5 — Fonctions linéaires et affines.** Plaçons nous dans le cas de fonctions de  $n$  variables réelles. Une fonction affine sur  $\mathbb{R}^n$  est une fonction de la forme

$$f(X) = V^T X + b,$$

où  $V \in \mathbb{R}^n$  et  $b \in \mathbb{R}$ . On voit facilement que  $f(X)$  est un nombre réel pour tout  $X \in \mathbb{R}^n$ . Si  $b = 0$ , on dit que la fonction  $f$  est une fonction linéaire. Le gradient d'une fonction affine prend une forme très simple. De même que la dérivée d'une fonction affine d'une variable réelle est une fonction constante, on a ici

$$\nabla f(X) = V,$$

le gradient d'une fonction affine est donc un vecteur constant. ■

■ **Exemple 4.6 — Fonctions quadratiques.** Les fonctions quadratiques de plusieurs variables réelles sont les généralisations de la fonction  $x \in \mathbb{R} \rightarrow ax^2$ . On peut les définir comme des fonctions de la forme

$$f : X \in \mathbb{R}^n \mapsto f(X) = X^T A X,$$

où  $A \in \mathcal{M}_n$  est une matrice carré. Dans ce cas, on peut montrer que

$$\nabla f(X) = (A + A^T)X,$$

qui est une fonction linéaire de  $X$ .

Un exemple particulier de fonction quadratique est la norme carrée d'un vecteur

$$\|X\|^2 = X^T X.$$

Dans ce cas la matrice  $A$  est la matrice identité  $A = I_n$ , de sorte que

$$\nabla \|X\|^2 = (I_n + I_n^T)X = 2X,$$

à mettre en correspondance avec le résultat bien connu en dimension 1 : si  $f(x) = x^2$  alors  $f'(x) = 2x$ . ■

## 4.2 Optimisation

### 4.2.1 Optimisation d'une fonction de plusieurs variables

En mathématiques, l'optimisation consiste à rechercher les minima et/ou maxima d'une fonction, d'une ou plusieurs variables. On parle génériquement d'extrema pour couvrir les deux cas. Il peut s'agir d'extrema locaux (points où la fonction prend une valeur plus petite ou plus grande que sur les points voisins) ou globaux.

**Définition 4.5** Soit  $f : (x_1, \dots, x_n) \in \mathcal{D}_f \rightarrow f(x_1, \dots, x_n) \in \mathbb{R}$  une fonction de plusieurs variables.

1.  $f$  admet un extremum local en un point  $\Xi = (\xi_1, \dots, \xi_n)$  de son domaine de définition s'il existe une boule  $\mathcal{B}$  de  $\mathbb{R}^n$  centrée en  $\Xi$  telle que pour tout  $X = (x_1, \dots, x_n) \in \mathcal{B}$ , on a  $f(X) \leq f(\Xi)$  (on dit alors que  $f$  est localement maximale en  $\Xi$ ) ou  $f(X) \geq f(\Xi)$  (on dit alors que  $f$  est localement minimale en  $\Xi$ ).
2.  $f$  admet un extremum global en un point  $\Xi = (\xi_1, \dots, \xi_n)$  de son domaine de définition si pour tout  $X = (x_1, \dots, x_n) \in \mathcal{D}_f$ , on a  $f(X) \leq f(\Xi)$  ( $f$  est globalement maximale en  $\Xi$ ) ou  $f(X) \geq f(\Xi)$  (on dit alors que  $f$  est globalement minimale en  $\Xi$ ).

Comme on l'a déjà vu, dans le cas de fonctions différentiables, la notion d'extremum local est étroitement liée à la notion d'annulation du gradient de la fonction. Sans entrer dans les détails, une condition nécessaire pour que  $X_* \in \mathcal{D}_f^\circ$  soit un extremum local de  $f$  est que  $X$  soit un *point critique* de  $f$ , c'est à dire vérifie l'équation

$$\nabla f(X_*) = 0. \quad (4.9)$$

Ici,  $\mathcal{D}_f^\circ$  est le domaine de définition de  $f$  privé de sa frontière, un point technique sur lequel on ne s'appesantira pas ici.

**Remarque 4.6 — Point de vue géométrique.** Étant donnée une fonction de  $n$  variables  $f : X = (x_1, \dots, x_n) \in \mathbb{R}^n \rightarrow f(X) \in \mathbb{R}$  : son graphe est une hypersurface dans un espace de dimension  $n + 1$ .

Si  $f$  est différentiable, le graphe admet un hyperplan tangent en tout point  $X$ , qui est un hyperplan de dimension  $n$ .

Le gradient de  $f$  au point  $X$  est le vecteur des pentes des droites tangentes

$$\nabla f(X) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(X) \\ \vdots \\ \frac{\partial f}{\partial x_n}(X) \end{pmatrix},$$

et  $f$  peut être extrémale en  $X_*$  si ces pentes sont toutes nulles, ce qui revient à l'équation des points critiques (4.9). Deux exemples en dimension 2 se trouvent à la FIGURE 4.1, où le point  $X_* = (0, 0)^T$  est un minimum (gauche) ou maximum (droite) de la fonction tracée. Dans les deux cas le plan tangent au graphe en  $X_*$  est horizontal.

Supposons maintenant connu un point critique  $X_*$  d'une fonction deux fois dérivable  $f$ . Comment savoir s'il s'agit d'un maximum, d'un minimum... ou ni l'un ni l'autre ? On peut pour cela interroger la matrice Hessienne de la fonction, définie en (4.8). Comme on l'a vu, cette matrice est réelle symétrique, elle est donc diagonalisable et admet des valeurs propres réelles. De plus il existe une base orthonormée de  $\mathbb{R}^n$  formée de vecteurs propres de cette matrice. L'étude des valeurs propres permet de répondre à la question.

**Proposition 4.7** Soit  $f$  une fonction deux fois différentiable de  $n$  variables réelles. Soit  $X_* \in \mathbb{R}^n$  un point critique de  $f$ , soit  $Hf(X_*)$  la Hessienne de  $f$  au point  $X_*$ .

1. Si toutes les valeurs propres de  $Hf(X_*)$  sont strictement négatives,  $X_*$  est un maximum local de  $f$ .
2. Si toutes les valeurs propres de  $Hf(X_*)$  sont strictement positives,  $X_*$  est un minimum local de  $f$ .

On voit donc qu'un point critique n'est pas toujours un extremum, loin de là. Par exemple, si l'une des valeurs propres est nulle, alors on ne peut pas conclure directement. Aussi, si il existe des valeurs propres positives et des valeurs propres négatives, on est en présence d'un *point selle*.

■ **Exemple 4.7 — Point selle.** Un exemple de point selle est représenté en FIGURE 4.5, pour la fonction  $f(x_1, x_2) = x_1^2 - x_2^2$ . La matrice Hessienne admet deux valeurs propres égales à  $\pm 2$ . Quand on atteint le point critique le long de la direction du vecteur propre correspondant à la valeur propre égale à 2 (ici le premier axe) on passe par un minimum. Si par contre on l'atteint le long de la direction du vecteur propre correspondant à la valeur propre égale à -2 (le second axe) on passe par un maximum. ■

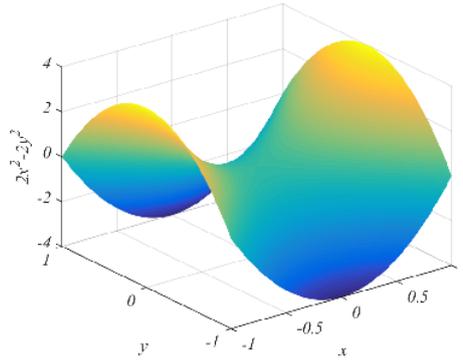


FIGURE 4.5 – Exemple de point selle. Le point de coordonnées  $(0,0)$  est un point critique de la fonction  $f(x_1, x_2) = x_1^2 - x_2^2$  mais n'est ni un maximum ni un minimum de  $f$ . C'est un minimum si on parcourt la droite d'équation  $x_2 = 0$ , et c'est un maximum si on parcourt la droite d'équation  $x_1 = 0$ .

#### 4.2.2 Un exemple d'application : neuroimagerie cérébrale

On utilise parfois l'électroencéphalographie (EEG) ou la magnétoencéphalographie (MEG) pour faire de l'imagerie cérébrale. En EEG ou en MEG, on dispose de mesures de potentiel électrique (EEG) ou champ magnétique (MEG) sur divers capteurs à l'extérieur du crâne, que l'on peut modéliser par un vecteur  $Z \in \mathbb{R}^J$  ( $J$  étant le nombre de capteurs).

Le champ électromagnétique trouve son origine dans l'activité électrique corticale, qui peut être décrite par un courant électrique échantillonné sur un certain nombre de points du cortex. On peut aussi modéliser cela par un vecteur  $X \in \mathbb{R}^K$  (où  $K \gg J$  est le nombre de points considérés sur le cortex).

La physique permet de décrire le lien entre l'activité électrique corticale  $X$  et les mesures  $Z$ . En première approximation (très réaliste cependant) la relation est linéaire et prend la forme

$$Z = GX + \text{bruit} , \quad (4.10)$$

où  $G \in \mathcal{M}_{JK}$  est une matrice appelée matrice de gain, qui résume la physique de la propagation du cortex jusqu'aux capteurs. Cette matrice peut être évaluée numériquement à partir de mesures IRM, on ne détaillera pas ce point ici.

En pratique,  $Z$  est mesuré et  $X$  est l'inconnue du problème (le bruit est lui aussi inconnu).

Pour reconstruire l'activité corticale à partir des mesures, il faut essayer de résoudre l'équation ci-dessus, au moins approximativement : trouver  $X$  tel que  $Z \approx GX$ , ou plus précisément tel que  $Z - GX$  soit petit. On formule souvent cela comme un problème d'optimisation

$$\min_{X \in \mathbb{R}^K} [\|Z - GX\|^2 + \lambda \|X\|^2] . \quad (4.11)$$

Autant le premier terme de la fonction à minimiser peut paraître naturel, autant on il est légitime de s'interroger sur la présence du second. On verra plus tard que ce second terme assure l'existence et l'unicité de la solution.

On peut donc rechercher les points critiques de la fonction  $f : X \rightarrow \|Z - GX\|^2 + \lambda \|X\|^2$ , soit résoudre

$$\nabla f(X) = 0 .$$

Il faut donc calculer ce gradient. On va pour cela utiliser les règles que nous avons déjà vues. Pour ce qui est du second terme on sait que

$$\nabla \lambda \|X\|^2 = 2\lambda X .$$

Par ailleurs, développons le premier terme avant d'en calculer le gradient. On a

$$\|Z - GX\|^2 = (Z - GX)^T (Z - GX) = Z^T Z - Z^T GX - (GX)^T Z + (GX)^T GX = \|Z\|^2 - 2Z^T GX + X^T G^T GX ,$$

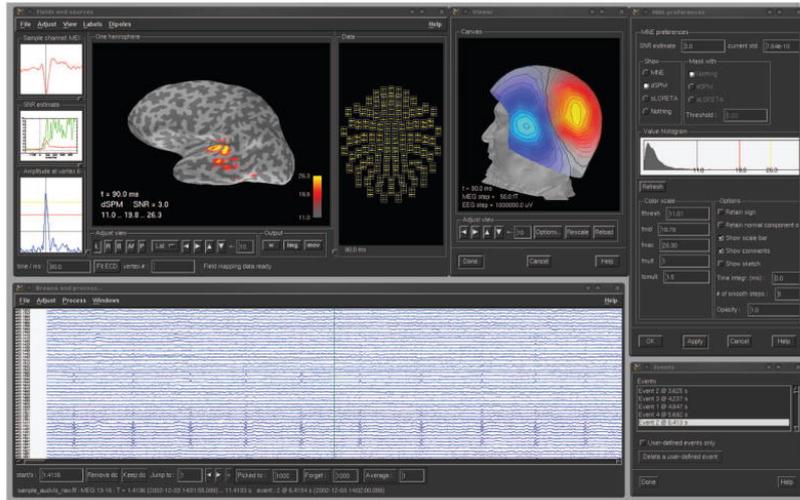


FIGURE 4.6 – Imagerie MEG. Des capteurs (haut, centre) enregistrent le champ magnétique (représentation spatiale en haut à droite, décours temporels en bas à gauche). La localisation de source consiste à essayer de reconstruire l'activité corticale (ou du moins sa localisation spatiale, haut à gauche) à partir des mesures externes. Référence : Gramfort et al, MNE software for processing MEG and EEG data, *Neuroimage*. 2014 Feb 1 ; 86 : 446–460.

où on a utilisé le fait que le produit scalaire est symétrique, donc  $(GX)^T Z = Z \cdot GX = GX \cdot Z = Z^T GX$ .

De là, on remarque que  $\nabla \|Z\|^2 = 0$  car  $\|Z\|^2$  ne dépend pas de  $X$  (c'est donc une constante par rapport à  $X$ ), on a donc

$$\nabla \|Z - GX\|^2 = -2G^T Z + [(G^T G) + (G^T G)^T] X = 2[-G^T Z + G^T GX]$$

car  $G^T G$  est une matrice symétrique (en effet :  $(G^T G)^T = G^T G$ ).  $\nabla f(X) = 0$  équivaut donc à

$$2[G^T Z + G^T GX + \lambda X] = 0 \quad \iff \quad [G^T G + \lambda I_K] X = G^T Z,$$

qui conduit à la solution explicite

$$X = [G^T G + \lambda I_K]^{-1} G^T Z, \quad (4.12)$$

à condition que la matrice  $G^T G + \lambda I_K$  soit inversible. Cette contrainte est en fait ce qui justifie l'introduction du second terme dans la fonction  $f$  optimisée.

- Si  $\lambda = 0$ , la solution est  $Z = [G^T G]^{-1} G^T Z$ ... mais  $G^T G$  n'est pas nécessairement inversible, et même quand elle l'est, on peut réaliser qu'elle est souvent instable numériquement à cause de divisions par de très petits nombres, proches de la précision du calcul numérique.
- L'introduction de  $\lambda > 0$  permet de régler le problème. Dans ce cas  $G^T G + \lambda I_K$  est toujours inversible. Le prix à payer est l'introduction de ce paramètre, dont il faut choisir la valeur, ni trop petite (dans ce cas on risque de retrouver les instabilités numériques) ni trop grande (on s'éloigne trop du problème de départ).

**Remarque 4.8 — Extension au cadre spatio-temporel.** Dans ce calcul, on n'a considéré qu'une seule mesure sur  $J$  capteurs, à un seul instant. Il se trouve que le problème peut être formulé de la même façon dans un cadre spatio-temporel, en considérant cette fois une matrice de mesures  $Z \in \mathcal{M}_{JL}$ ,  $L$  étant le nombre d'échantillons temporels : chaque colonne de  $Z$  représente un instant particulier. On recherche alors une matrice de sources corticales  $X \in \mathcal{M}_{KL}$  avec le même nombre de pas de temps.

De façon remarquable, le problème prend exactement la même forme  $Z \approx GX$ , mais fait intervenir des objets mathématiques différents (matrices plutôt que vecteurs). Le problème d'optimisation peut lui aussi s'écrire de façon identique, comme en (4.11), la seule petite différence est qu'on doit utiliser ici une norme définie sur des matrices, appelée norme de Frobenius, donnée par la racine carrée de la somme

des carrés des éléments de matrice :  $\|A\|_F^2 = \sum_{i,j} a_{ij}^2$ . La solution prend elle aussi exactement la même forme (4.12).

### 4.3 Analyse discriminante

Revenons maintenant à une problématique plus proche d'analyse des données, sur laquelle va se poser un problème d'optimisation tel que mentionné plus haut.

#### 4.3.1 Position du problème

Supposons que l'on dispose d'un jeu de données (disons, une matrice  $n \times p$ ), dans laquelle les données proviennent de classes différentes (par exemple, des échantillons de mesures chimiques de différentes quantités, sur des vins provenant de différents vignobles, comme nous l'avons vu plus haut).

La classification a pour objectif de déterminer automatiquement les classes à partir des données : déterminer quels individus appartiennent à quelle classe. On distingue généralement deux approches :

- *Classification supervisée* : on dispose d'un jeu de données d'apprentissage, à partir duquel une méthode de classification peut être construite. Les performances de la méthode peuvent être évaluées sur un autre jeu de données, appelé jeu de données de test.
- *Classification non supervisée* : dans ce cas on ne dispose pas de jeu de données d'apprentissage, les classes doivent être déterminées à partir d'un unique jeu de données.

Dans ce qui suit, on se focalisera sur le cas supervisé : supposons qu'on dispose d'échantillons de la forme de couples  $(x_i, c_i)$ ,  $i = 1, \dots, n$ , où  $x_i \in \mathbb{R}^p$ ,  $c_i = 1, 2$  (ou  $c_i = 1, 2, \dots, C$ ). Les  $c_i$  sont les étiquettes des vecteurs  $x_i$ .

L'objectif de l'analyse discriminante est de prédire une étiquette  $c$  pour chaque observations  $x$ .

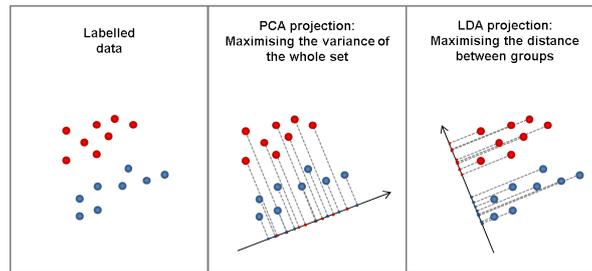


FIGURE 4.7 – Exemples de projections d'un nuage de points sur des axes

Dans certains cas, l'analyse en composante principale (qui est une approche non supervisée) peut fournir une bonne prédiction, mais tel n'est pas le cas en général, comme on peut le voir en Figure 4.7, où la projection sur le premier axe principal (milieu) mélange totalement les données des deux classes, alors que la projection sur un axe plus adapté (droite) conduit à une bonne séparation.

Entrons maintenant davantage dans les détails. Supposons qu'on dispose d'un jeu de données d'apprentissage composé de deux matrices  $X^{(1)} \in \mathcal{M}_{n_1,p}$  et  $X^{(2)} \in \mathcal{M}_{n_2,p}$ , à  $p$  variables et  $n_1$  et  $n_2$  échantillons (individus) respectivement. Les classes sont connues (appelées 1 et 2).

Nous allons rechercher une règle de décision linéaire : à partir d'un jeu d'apprentissage, trouver un hyperplan de  $\mathbb{R}^p$  (donc de dimension  $p - 1$ ), qui découpe l'espace  $\mathbb{R}^p$  en deux régions, appelées régions de décision. Deux exemples peuvent se trouver dans la Figure 4.8. Dans les deux cas, on peut voir que l'espace  $\mathbb{R}^p$  est bien découpé en deux sous-ensembles par un hyperplan de dimension  $p - 1$ . De là, on peut aussi voir que la projection des données sur l'axe dirigé par un vecteur  $W$  perpendiculaire à cet hyperplan va permettre de séparer les données en deux sous-ensembles, qui seront relativement bien séparés (plus ou moins, cela va dépendre des données bien sûr).

De là, la règle de décision que nous allons rechercher se ramène à trouver un vecteur  $W \in \mathbb{R}^p$  tel que la projection sur l'axe correspondant permettra de séparer au mieux les deux classes (on aura aussi besoin pour cela d'un seuil).

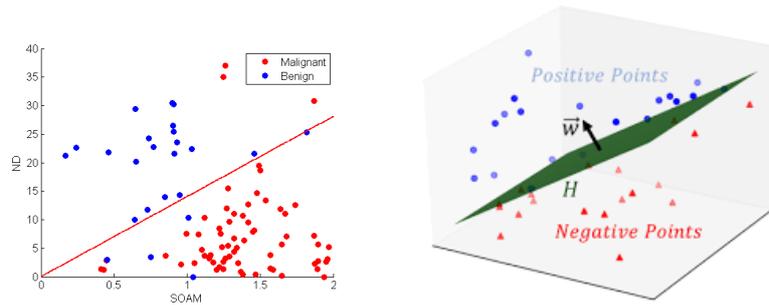


FIGURE 4.8 – Deux exemples de régions de décision, dans le plan  $\mathbb{R}^2$  et l'espace  $\mathbb{R}^3$ . Dans les deux cas, l'espace est coupé en deux parties par un hyperplan (une droite dans  $\mathbb{R}^2$ , un plan dans  $\mathbb{R}^3$ ). Cet hyperplan est caractérisé par un vecteur normal  $W$ . La projection des données sur l'axe dirigé par  $W$  fournira la meilleure règle de décision.

Il s'agit donc de projeter les tableaux de données sur cet axe, plus précisément il s'agit de projeter les individus (des vecteurs de  $\mathbb{R}^p$ ). Comme on l'a vu, cette opération peut s'effectuer matriciellement par l'expression

$$Y = XW . \quad (4.13)$$

$Y$  sera alors un vecteur de dimension  $n$ , qui représente les coordonnées des projections des individus sur l'axe porté par  $W$ . Étant donné un seuil, et l'une de ces coordonnées  $y_k$ , on affectera  $x_k$  à l'une ou l'autre des deux classes suivant que  $y_k \geq \tau$  ou  $y_k < \tau$ , où  $\tau$  est un seuil, qui est également à déterminer.

En résumé, dans le cas de deux classes considéré ici, la règle de décision est déterminée par un vecteur  $W \in \mathbb{R}^p$ , appelé vecteur discriminant et un seuil  $\tau \in \mathbb{R}$ . Nous allons maintenant voir comment déterminer ces deux quantités, en utilisant une approche appelée Discriminant de Fisher.

### 4.3.2 Discriminant de Fisher

Restons dans le cas simple de la classification bi-classe : il n'y a que deux classes possibles, notées 1 et 2. On dispose d'un jeu de données d'apprentissage composé de deux matrices  $X^{(1)} \in \mathcal{M}_{n_1,p}$  et  $X^{(2)} \in \mathcal{M}_{n_2,p}$ , à  $p$  variables et  $n_1$  et  $n_2$  échantillons (individus) respectivement. Nous recherchons donc un vecteur (ou un axe) dans  $\mathbb{R}^p$  qui permette d'atteindre les deux objectifs suivants :

- Maximiser la séparation entre les deux classes
- Minimiser la variabilité interne à chaque classe

Ce double objectif peut être visualisé sur la FIGURE 4.9, sur laquelle sont représentés les histogrammes des coordonnées des projections de données sur deux axes différents. On voit bien que la projection sur l'axe LD2 ne convient pas, les histogrammes sont très proches (les moyennes des projections des deux classes sont trop proches), alors que les projections sur l'axe LD1 sont bien mieux séparées. On voit aussi que les histogrammes des projections sur l'axe LD1 sont plus étroits, ce qui permet aussi une meilleure séparation (il y aurait davantage de chevauchement s'ils étaient plus larges).

Soit donc  $W \in \mathbb{R}^p$  un vecteur (candidat pour être le vecteur discriminant, on va ensuite chercher quel est le meilleur  $W$  possible). On notera

$$Y^{(1)} = X^{(1)}W \in \mathbb{R}^{n_1} , \quad Y^{(2)} = X^{(2)}W \in \mathbb{R}^{n_2} \quad (4.14)$$

les coordonnées des projections des individus des deux jeux de données sur l'axe porté par  $W$ .

On note aussi  $m_1, m_2 \in \mathbb{R}^p$  les moyennes des matrices de données  $X^{(1)}$  et  $X^{(2)}$ , et

$$\mu_1 = m_1W \in \mathbb{R} \quad \mu_2 = m_2W \in \mathbb{R} \quad (4.15)$$

les moyennes des données projetées.

Le discriminant de Fisher se base sur les deux quantités suivantes :

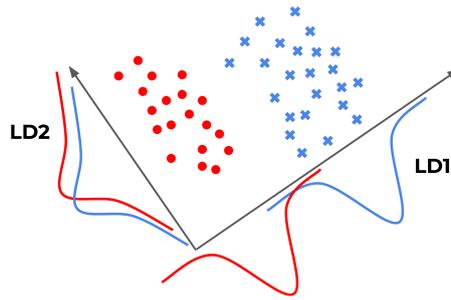


FIGURE 4.9 – Analyse discriminante en dimension 2

— La *dispersion inter-classes*  $(\mu_1 - \mu_2)^2$ , qui peut aussi s'écrire sous la forme

$$(\mu_1 - \mu_2)^2 = W^T S_B W, \quad \text{où} \quad S_B = (m_1 - m_2)(m_1 - m_2)^T. \quad (4.16)$$

$S_B$  est la matrice de dispersion inter-classes.

— La *dispersion intra-classes*, construite comme la somme des dispersions internes  $\delta_1^2$  et  $\delta_2^2$  des deux classes : pour  $i = 1, 2$ , on écrit

$$\delta^2 = \delta_1^2 + \delta_2^2, \quad \text{où} \quad \delta_i^2 = \left( Y^{(i)} - \bar{Y}^{(i)} \right)^T \left( Y^{(i)} - \bar{Y}^{(i)} \right), \quad i = 1, 2. \quad (4.17)$$

Il est possible de démontrer que pour  $i = 1, 2$  on a

$$\delta_i^2 = W^T S_i W, \quad \text{où} \quad S_i = \left( X^{(i)} - \bar{X}^{(i)} \right)^T \left( X^{(i)} - \bar{X}^{(i)} \right) \quad (4.18)$$

est la matrice de dispersion de la classe  $i$ , de sorte que

$$\delta^2 = W^T S_W W, \quad (4.19)$$

où

$$S_W = S_1 + S_2 \quad (4.20)$$

est la matrice de dispersion intra-classes.

**Définition 4.6** Discriminant de Fisher Avec les notations précédentes, le discriminant de Fisher associé à un vecteur  $W \in \mathbb{R}^p$  est la quantité

$$J(W) = \frac{(\mu_1 - \mu_2)^2}{\delta^2} = \frac{W^T S_B W}{W^T S_W W},$$

en d'autres termes le quotient de la dispersion inter-classes par la dispersion intra-classes.

L'objectif de l'analyse discriminante de Fisher est de déterminer le vecteur  $W$  qui maximise  $J(W)$ .

### 4.3.3 Optimisation du critère de Fisher

Sur la base de ce que nous avons vu dans la section 4.2, on pourrait se dire : *facile, il suffit de résoudre l'équation*

$$\nabla J(W) = 0,$$

puis vérifier que le  $W$  obtenu est bien un maximum. Mais les choses ne sont pas si simples parce qu'il y a en fait une infinité de solutions. En effet, si  $W_0$  est une telle solution, alors tout multiple  $\alpha W_0$  de  $W_0$  est solution aussi : on peut facilement voir que  $J(\alpha W_0) = J(W_0)$ . Il faut donc contraindre la solution, en imposant une

condition de normalisation qui restaure l'unicité de la solution. Il existe un grand nombre de contraintes possibles, mais la plus simple est d'imposer

$$W^T S_W W = 1 .$$

Le problème devient donc

$$W_* = \underset{W \in \mathbb{R}^p}{\operatorname{argmax}} J(W) \quad \text{sous contrainte} \quad W^T S_W W = 1 . \quad (4.21)$$

Ce problème peut en fait se résoudre par des techniques appropriées, en recourant à des *multiplicateurs de Lagrange*, dont la justification va un peu au delà du programme de ce cours. Mais le résultat est remarquablement simple.

**Théorème 4.9** Supposons que la matrice de dispersion intra-classe  $S_W$  soit inversible. La solution du problème (4.21) est donnée par

$$W = \alpha S_W^{-1} (m_1 - m_2) ,$$

où la constante  $\alpha$  est déterminée par la contrainte  $W^T S_W W = 1$ ,

$$\alpha = \frac{1}{\sqrt{(m_1 - m_2)^T S_W^{-1} (m_1 - m_2)}} .$$

**Remarque 4.10** — Ce choix de  $\alpha$  est purement conventionnel, et résulte d'un choix de normalisation parmi bien d'autres (qui simplifie le calcul). En pratique, on peut choisir n'importe quelle valeur non nulle pour  $\alpha$ , par exemple  $\alpha = 1$ .

— On peut noter que la matrice  $S_W$  est proportionnelle à la matrice de covariance empirique  $C_X$  du tableau de données. On peut donc remplacer  $S_W$  par n'importe quel multiple de  $C_X$  dans le théorème ci-dessus.

**Démonstration (Pour les amateurs de maths) :** La méthode consiste à remplacer le problème (4.21) par un autre problème dont on montre qu'il est équivalent :

$$(W_*, \lambda_*) = \underset{W \in \mathbb{R}^p, \lambda \in \mathbb{R}}{\operatorname{argmax}} \mathcal{J}(W, \lambda) \quad \text{avec} \quad \mathcal{J}(W, \lambda) = W^T S_B W - \lambda (W^T S_W W - 1) . \quad (4.22)$$

On peut d'ores et déjà voir que

$$\frac{\partial \mathcal{J}}{\partial \lambda}(W, \lambda) = 1 - W^T S_W W ,$$

de sorte qu'annuler cette quantité assure la contrainte de normalisation. Par ailleurs, le gradient de  $\mathcal{J}$  par rapport à  $W$  vaut

$$\nabla_W \mathcal{J}(W, \lambda) = S_B W - \lambda S_W W ,$$

de sorte qu'en annulant cette quantité on obtient

$$S_B W - \lambda S_W W .$$

Là, on peut remarquer que  $S_B W = (m_1 - m_2)(m_1 - m_2)^T W$  est un vecteur proportionnel à  $m_1 - m_2$ , puisque  $(m_1 - m_2)^T W$  est un nombre (qui n'est autre que le produit scalaire  $W \cdot (m_1 - m_2)$ ). On en déduit donc que  $\lambda S_W W$  est proportionnel à  $m_1 - m_2$  d'où, en supposant que  $S_W$  est inversible,  $W$  est proportionnel à  $S_W^{-1} (m_1 - m_2)$ , ce qui est le résultat cherché.

Pour ce qui est de la valeur de  $\alpha$ , il suffit de remplacer

$$W^T S_W W = \alpha^2 (S_W^{-1} (m_1 - m_2))^T S_W S_W^{-1} (m_1 - m_2) = (m_1 - m_2)^T S_W^{-1} (m_1 - m_2) ,$$

en égalant cette quantité à 1 on obtient le résultat. ■

Une fois le vecteur discriminant calculé, la règle de décision de l'analyse discriminante de Fisher repose sur le choix d'un seuil. Par défaut, ce seuil est égal à

$$\tau = \frac{\mu_1 + \mu_2}{2}, \quad (4.23)$$

et la règle de classification est donc la suivante : la classe prédite pour un échantillon  $x$  est

$$c(x) = \begin{cases} 1 & \text{si } W^T x \geq \tau \\ 2 & \text{si } W^T x < \tau \end{cases}. \quad (4.24)$$

**Remarque 4.11** 1. Ce choix est lié à l'expression du vecteur discriminant  $W$  dans le théorème 4.9, où apparaît  $m_1 - m_2$ . Si on avait fait le choix de  $m_2 - m_1$  il aurait fallu inverser les choix dans (4.24).  
 2. De même, le choix d'une inégalité au sens large pour la classe 1 et strict pour la classe 2 est purement conventionnel (il en fallait un), et n'a aucune importance, la probabilité d'avoir l'égalité étant infinitésimale.  
 3. Plus important : le discriminant de Fisher ne fait aucune hypothèse sur la probabilité *a priori* d'avoir une classe ou l'autre. Il peut arriver qu'on ait beaucoup plus de chances d'être dans la classe 2 que la classe 1, ou inversement. Dans ce cas il faut modifier le seuil en conséquence. Supposant que la probabilité d'être dans la classe 1 vaut  $p_1$ , et la probabilité d'être dans la classe 2 vaut  $p_2 = 1 - p_1$ , la valeur modifiée du seuil tenant compte de ces probabilités est

$$\tau = \frac{\mu_1 + \mu_2}{2} - \ln\left(\frac{p_1}{p_2}\right), \quad (4.25)$$

la règle de décision (4.24) restant inchangée. Cette modification n'est pas prévue dans l'approche du discriminant de Fisher que nous avons suivie ici, mais elle peut être obtenue via une approche plus générale connue sous le nom d'Analyse Discriminante Linéaire, ou LDA.

#### 4.3.4 Une session typique d'analyse discriminante linéaire

On peut grossièrement résumer une session d'analyse discriminante linéaire comme la succession d'étapes suivante :

1. On dispose de deux jeux de données : un jeu d'apprentissage, et un jeu de test.
2. En utilisant le jeu d'apprentissage, on estime le vecteur discriminant  $W$  (ou les vecteurs discriminants) : dans le cas de deux classes, un seul vecteur  $W$ , dans le cas général un vecteur par paire de classes.
3. On applique le classifieur sur le jeu d'apprentissage. Les performances peuvent être évaluées en calculant le pourcentage de données correctement classées.
4. On applique le classifieur sur le jeu de test. Les performances peuvent être évaluées là encore en calculant le pourcentage de données correctement classées.

Il est naturel que les résultats obtenus soient meilleurs sur le jeu d'apprentissage que sur le jeu de test, car le classifieur a été construit à partir du jeu d'apprentissage. Ceci dit, si les résultats sont très différents, ceci devient problématique. On parle de sur-apprentissage. Le classifieur s'est trop adapté à des caractéristiques du jeu d'apprentissage, qui ne sont pas présentes dans le jeu de test (et ne sont donc pas pertinentes en fait). Il existe des techniques pour réduire le sur-apprentissage, qui sont souvent basées sur des opérations de réduction de dimension (projection sur un sous-espace pertinent, parfois calculé à partir d'une ACP).

■ **Exemple 4.8 (LDA sur données de P300 speller)** Le problème de construction d'un système de prédiction pour le P300 speller est un peu différent d'un problème d'analyse discriminante, mais l'analyse discriminante peut aider à y répondre.

Dans ce problème, on dispose d'une famille de signaux (des vecteurs de longueur donnée  $p$ , représentant des décours temporels, mesurés sur un certain nombre de capteurs), qui sont des réponses à des stimuli visuels, pouvant être partagés en deux classes : un caractère donné apparaît sur un écran ou pas. Il s'agit donc d'un problème à deux classes (*oui* ou *non*). Le problème d'apprentissage consiste donc à construire un vecteur  $W$  et un seuil  $\tau$  à partir d'un jeu de données d'apprentissage.

Pour simplifier on se limitera ici au cas d'un seul capteur. On dispose donc d'une matrice de signaux  $X \in \mathcal{M}_{np}$  et d'un vecteur de classes  $C \in \{\text{oui}, \text{non}\}^n$ . La LDA estime à partir de ces données un vecteur

discriminant  $W$ , un seuil  $\tau$ , et des probabilités  $p_{oui}$  et  $p_{non}$ . Dans cet exemple,  $n = 15300$ ,  $n_{oui} = 12750$ ,  $n_{non} = 2550$ ,  $p = 240$ .



#### (LDA : apprentissage)

Le package MASS de R implémente une fonction `lda`, ainsi que divers outils permettant d'exploiter le résultat. Ci dessous on dispose d'une matrice numérique  $X$  contenant les  $n$  signaux, et d'un vecteur (de type `logical`, prenant les valeurs `TRUE` ou `FALSE`) de dimension  $n$ , appelé `classes`.

```
p <- dim(X,2)
n <- dim(X,1)
n_oui <- sum(classes==TRUE)
n_non <- sum(classes==FALSE)
X.lda <- lda(X,classes)
matplot(t(P300.lda$means), type='l', lty=1,
        xlab='Time (samples)', ylab='Amplitude')
```

La fonction `lda` retourne une liste (appelée ici `X.lda`) contenant plusieurs champs, parmi lesquels on peut mentionner notamment

- `prior` : un vecteur (ici de dimension 2) donnant les probabilités des classes, estimées à partir des données.
- `means` : les moyennes des classes, sous forme d'une matrice (ici une matrice  $2 \times 240$ )
- `scaling` : un vecteur de longueur  $p$  contenant le vecteur discriminant  $W$ .

Le tracé des deux moyennes (voir la FIGURE 4.10) fait bien apparaître une différence très significative entre les deux groupes. Mais comme on va le voir ça ne suffit pas.

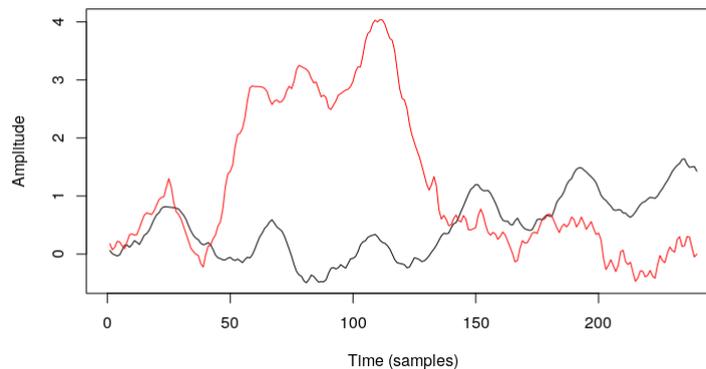


FIGURE 4.10 – Moyennes de signaux EEG, classes P300 et non P300

Dans les lignes qui suivent, on utilise la fonction `predict` qui, appliquée à un objet de type `lda` et une matrice de données du même type que  $X$  (même nombre de variables) va prédire une classe pour chacune des lignes. On va tout d'abord tester la prédiction sur le jeu d'apprentissage.

```
X.predict <- predict(object=X.lda,X)
X.pred.classes <- X.predict$class
success.rate <- sum(X.pred.classes==classes)
```

La sortie de `predict` est une liste, appelée ici `X.predict` qui contient trois éléments, dont un appelé `class`. Le taux de succès est la proportion de prédictions identiques à la classe d'origine. Le résultat est un pourcentage de 83.27% environ, ce qui est assez bien. Cependant, le vrai test doit se faire sur un jeu de données de test.

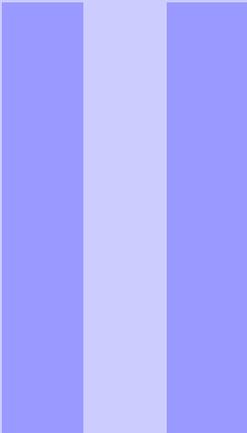
On dispose d'un jeu de test constitué de 18000 signaux, stockés dans la matrice  $Y$ , et des classes correspondantes. A la différence de la phase d'apprentissage, les classes ne sont pas utilisées pour la prédiction, seulement pour comparaison à la fin avec les classes prédites.

```
Y.predict <- predict(object=X.lda,Y)
Y.pred.classes <- Y.predict$class
success_rate = sum(classes_pred==classes)/nb_signals
```

Le résultat est cette fois encore très bon, avec un pourcentage de 83% environ (même légèrement supérieur au jeu d'apprentissage). Cela montre que la LDA a fonctionné correctement dans ce cas particulier.

■

Il faut noter que le problème du P300 speller est un peu différent d'un problème de classification standard. Les données sont beaucoup plus structurées, on sait que pour chaque *run* seules une ligne et une colonne contiendront le caractère cible. Ceci explique les différences de performances.



# Exercices

<b>5</b>	<b>Exercices sur le chapitre 1</b> .....	<b>97</b>
5.1	Vecteurs	
5.2	Sous-espaces vectoriels, projection orthogonale	
5.3	Calcul matriciel	
5.4	Changements de base, projection orthogonale	
<b>6</b>	<b>Exercices sur le chapitre 3</b> .....	<b>101</b>
6.1	Généralités	
6.2	Diagonalisation	
6.3	Analyse en composantes principales	
6.4	Régression linéaire	
6.5	Analyse discriminante de Fisher	



## 5. Exercices sur le chapitre 1

### 5.1 Vecteurs

**Exercice 5.1 — Opérations de base.** 1. On considère les vecteurs du plan

$$U_1 = (1, 3)^T, \quad U_2 = (2, -1)^T, \quad U_3 = (4, 0)^T,$$

Calculer  $-U_1 + 3U_2 - 2U_3$ .

2. Calculer  $V_1 + 2V_2 - 3V_3$ , avec les vecteurs de l'espace ci-dessous

$$V_1 = (1, 2, 3)^T, \quad V_2 = (4, 2, -1)^T, \quad V_3 = (-1, 4, 0)^T,$$

On effectuera le calcul à la main, puis on pourra vérifier avec R. ■

**Exercice 5.2 — Norme, produit scalaire.** 1. Calculer les normes des trois vecteurs  $U_1$ ,  $U_2$  et  $U_3$  de l'exercice 5.1, ainsi que les produits scalaires  $U_1 \cdot U_2$ ,  $U_1 \cdot U_3$  et  $U_2 \cdot U_3$ .

2. Mêmes questions avec les vecteurs  $V_1$ ,  $V_2$  et  $V_3$  de l'exercice 5.1.

On effectuera le calcul à la main, puis on pourra vérifier avec R. ■

**Exercice 5.3 — Décomposition sur une base.** 1. On considère les deux vecteurs du plan

$$F_1 = \left( \frac{\sqrt{3}}{2}, \frac{1}{2} \right)^T, \quad F_2 = \left( \frac{1}{2}, -\frac{\sqrt{3}}{2} \right)^T.$$

(a) Vérifier qu'ils constituent une base orthonormée de  $\mathbb{R}^2$ .

(b) Calculer les coordonnées du vecteur  $U = (1, 2)^T$  dans cette base.

(c) Exprimer en fonction de  $x$  et  $y$  les coordonnées d'un vecteur quelconque  $V = (x, y)^T$  dans cette base.

2. 🍷 On considère les deux vecteurs du plan

$$G_1 = (1, 3)^T, \quad G_2 = (2, 3)^T.$$

Calculer leur déterminant et en déduire qu'ils constituent une base de  $\mathbb{R}^2$ . Vérifier qu'elle n'est pas orthonormée. Calculer les coordonnées du vecteur  $(-1, 1)^T$  dans cette base. Même question pour un vecteur  $(x, y)^T \in \mathbb{R}^2$  quelconque.

3.  Même question pour les trois vecteurs  $V_1, V_2$  et  $V_3$  de l'exercice 5.2. ■

## 5.2 Sous-espaces vectoriels, projection orthogonale

**Exercice 5.4 — Projection orthogonale.** On considère une famille  $\mathcal{F} = \{F_1, F_2, \dots, F_M\}$  de vecteurs de  $\mathbb{R}^N$ , avec  $M < N$ , et on cherche à expliciter la projection orthogonale sur le sous-espace  $F = \text{Vect}(\mathcal{F})$  de  $\mathbb{R}^N$  engendré par  $\mathcal{F}$ .

1. On suppose tout d'abord que  $\mathcal{F}$  est orthonormée :  $F_i \cdot F_j = 0$  pour tous  $i \neq j$  et  $\|F_i\| = 1$  pour tout  $i$ . On note  $Y = \Pi_F(X)$  le projeté orthogonal sur  $F$  d'un vecteur  $X \in \mathbb{R}^N$  quelconque, qui s'écrit

$$Y = \Pi_F(X) = \alpha_1 F_1 + \alpha_2 F_2 + \dots + \alpha_M F_M.$$

Vérifier que pour tout  $i$ ,  $\alpha_i$  est donné par  $\alpha_i = Y \cdot F_i$ .

2. Exemple dans  $\mathbb{R}^3$  : pour  $F_1 = (\sqrt{2}/2, 1/2, -1/2)^T$ ,  $F_2 = (0, -1/2, -1/2)^T$ , vérifier que la famille  $\{F_1, F_2\}$  est orthonormée, et expliciter le projeté orthogonal d'un vecteur  $X = (x, y, z)^T \in \mathbb{R}^3$  quelconque.
3.  On ne suppose plus maintenant que la famille  $\mathcal{F}$  est orthonormée. En calculant les produits scalaires du projeté  $Y = \Pi_F(X)$  (sous sa forme générique ci-dessus) avec tous les vecteurs  $F_j$ , montrer que les coefficients  $\alpha_i$  sont solutions d'un système linéaire qu'on explicitera.
4.  Application numérique : expliciter la projection orthogonale sur le sous-espace engendré par les vecteurs  $V_1$  et  $V_2$  de l'exercice 5.2. ■

## 5.3 Calcul matriciel

**Exercice 5.5 — Équations matricielles.** 1. Trouver la matrices  $X \in \mathcal{M}_{2,3}(\mathbb{R})$  telle que l'égalité matricielle suivante soit satisfaite

$$\begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 0 \end{pmatrix} - X = \begin{pmatrix} 0 & -1 & 2 \\ 1 & 4 & -2 \end{pmatrix}$$

2. Déterminer dans  $\mathcal{M}_2(\mathbb{C})$  les matrices  $X$  et  $Y$  vérifiant

$$X + Y = \begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix}, \quad \text{et} \quad X - Y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$$
■

**Exercice 5.6 — Produits matriciels.** Effectuer le produit des matrices :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 10 \\ 11 \\ 12 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 0 \\ 3 & 1 & 4 \\ 3 & 3 & 0 \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 \\ 1 & 4 & -1 \\ 2 & 1 & 2 \end{pmatrix}.$$
■

**Exercice 5.7 — Produits matriciels.** Calculer les produits suivants lorsque cela est possible :

$$\begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & -2 & 0 \\ 2 & -1 & 2 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (1 \ 1 \ 1) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (1 \ 1 \ 1), \quad \begin{pmatrix} 1 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 1 & -1 \\ 0 & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 \\ -2 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 3 & 0 \\ 0 & -1 & 2 \end{pmatrix}.$$

$$\begin{pmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix}, \quad \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{pmatrix}$$

que déduit-on dans ce dernier cas ?

## 5.4 Changements de base, projection orthogonale

**Exercice 5.8 — Changement de base dans le plan.** On considère les vecteurs du plan  $F_1 = (1, 5)^T$  et  $F_2 = (2, 4)^T$ .

1. Vérifier que ces vecteurs forment une base de  $\mathbb{R}^2$ . Cette base est-elle orthonormée ? (suggestion : utiliser le produit  $F^T F$ , où  $F$  est la matrice de la base).
2. Etant donné un vecteur  $V = (x, y)^T$  du plan, calculer ses coordonnées dans la base  $\{F_1, F_2\}$ . On pourra utiliser l'expression suivante pour l'inverse d'une matrice  $2 \times 2$  :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

à condition bien sûr que le dénominateur soit non-nul.

**Exercice 5.9 — Changement de base dans l'espace.** On considère les vecteurs  $F_1 = (1, 1, 1)^T$ ,  $F_2 = (0, 1, 1)^T$  et  $F_3 = (0, 0, 1)^T$  de  $\mathbb{R}^3$ .

1. Il est possible de montrer que ces trois vecteurs constituent une base de  $\mathbb{R}^3$ . Le vérifier en utilisant  $R$ . On note  $F$  la matrice de cette base.
2. Vérifier (en utilisant  $R$  ou à la main) que

$$F^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

3. En déduire les coordonnées dans la nouvelle base du vecteur  $(1, 2, 3)^T$ , puis d'un vecteur quelconque  $X = (x, y, z)^T$

**Exercice 5.10 — Projection orthogonale.** 1. On cherche ici à expliciter la projection orthogonale sur le sous-espace de  $\mathbb{R}^3$  engendré par les vecteurs  $F_1 = (1, 1, 1)^T/\sqrt{3}$  et  $F_2 = (0, 1, -1)^T/\sqrt{2}$ . On note  $F$  la matrice engendrée par ces deux vecteurs.

- (a) En calculant la matrice  $F^T F$ , vérifier que ces deux vecteurs forment une famille orthonormée dans  $\mathbb{R}^3$ . C'est donc une base orthonormée de l'espace engendré par  $F_1$  et  $F_2$ .
  - (b) Calculer la matrice  $F F^T$ .
  - (c) Étant donné un vecteur  $V = (x, y, z)^T$ , on note  $W = \Pi(V)$  son projeté orthogonal sur le sous-espace engendré par  $F_1$  et  $F_2$ . Utiliser la matrice  $F F^T$  pour expliciter le vecteur projeté  $\Pi(V)$ . Donner aussi les coefficients  $a_1$  et  $a_2$  du développement  $W = a_1 F_1 + a_2 F_2$ .
2. On se pose des questions similaires pour la famille  $\{F_1, F_3\}$ , avec  $F_3 = (1, 1, -1)^T/\sqrt{2}$ , et le sous-espace de  $\mathbb{R}^3$  qu'elle engendre. On note de nouveau  $F$  la matrice correspondante.
    - (a) Calculer  $F^T F$  puis  $\det(F^T F)$ , en déduire que  $\{F_1, F_3\}$  est une base du sous-espace engendré par ces deux vecteurs, mais que cette base n'est pas orthonormée.
    - (b) Calculer la matrice  $F(F^T F)^{-1} F^T$ .
    - (c) Étant donné un vecteur  $V = (x, y, z)^T$ , on note  $W = \Pi(V)$  son projeté orthogonal sur le sous-espace engendré par  $F_1$  et  $F_3$ . Utiliser la matrice  $F^T F$  pour expliciter le vecteur projeté  $\Pi(V)$ .

Donner aussi les coefficients  $b_1$  et  $b_3$  du développement  $W = b_1F_1 + b_3F_3$ . ■

**Exercice 5.11 — Projection orthogonale.** On considère les trois vecteurs de  $\mathbb{R}^4$

$$F_1 = (1, 2, 3, 4)^T, F_2 = (4, 3, 2, 1)^T, F_3 = (1, -1, 1, -1)^T$$

on cherche à caractériser la projection orthogonale sur le sous-espace de  $\mathbb{R}^4$  engendré par ces trois vecteurs.

1. Former la matrice  $F$  de cette famille, et calculer la matrice  $F^T F$  des produits scalaires des vecteurs  $F_i$  (en utilisant R ou à la main). On montre que  $\det(F^T F) = 1600$  (le vérifier sous R), en déduire que  $\{F_1, F_2, F_3\}$  est une base de  $\text{Vect}\{F_1, F_2, F_3\}$ .
2. Vérifier que

$$(F^T F)^{-1} = \frac{1}{400} \begin{pmatrix} 29 & -21 & 25 \\ -21 & 29 & -25 \\ 25 & -25 & 125 \end{pmatrix}$$

3. Etant donné un vecteur  $V = (x, y, z, t)^T$  de  $\mathbb{R}^4$ , expliciter son projeté orthogonal sur le sous-espace de  $\mathbb{R}^4$  engendré par  $F_1, F_2$  et  $F_3$  : donner le vecteur projeté  $W = \Pi(V)$ , ainsi que les coefficients  $a_1, a_2$  et  $a_3$  du développement  $W = a_1F_1 + a_2F_2 + a_3F_3$ , organisés sous forme d'un vecteur  $A \in \mathbb{R}^3$ . ■

## 6. Exercices sur le chapitre 3

### 6.1 Généralités

**Exercice 6.1 — Corrélation.** On considère les variables  $x$  et  $y$  ci-dessous, donnant les longueurs respectives de l'humérus et du radius sur 6 spécimen de fossiles d'un animal disparu.

$x$ (humérus)	44	62	71	73	87
$y$ (fémur)	40	57	59	65	77

1. Calculer les moyennes et variances de ces deux variables.
2. Calculer leur covariance et leur corrélation. Peut-on conclure ?

**Exercice 6.2 — Corrélation et dépendance.** On considère le tableau de données

$$X = \begin{pmatrix} -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ 9 & 4 & 1 & 0 & 1 & 4 & 9 \end{pmatrix}^T.$$

1. Calculer les moyennes et variances des deux variables
2. Calculer la matrice de covariance de  $X$ , et la matrice de corrélations.
3. Les deux variables sont-elles indépendantes ?

### 6.2 Diagonalisation

**Exercice 6.3 — Diagonalisation d'une matrice.** On considère la matrice

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

1. Dire pourquoi elle est diagonalisable
2. Calculer ses valeurs propres en utilisant un déterminant.
3. Calculer des vecteurs propres correspondants, en déduire une base orthonormée de  $\mathbb{R}^2$ .

### 6.3 Analyse en composantes principales

#### Exercice 6.4 — Analyse en composantes principales d'un tableau simple.

On veut effectuer l'analyse en composantes principales de la matrice

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Suivant la convention classique, les variables sont les colonnes, les lignes représentant les individus.

1. Calculer les moyennes des variables, la matrice  $\bar{X}$  correspondante, puis la matrice centrée  $X_c$ . Montrer que la matrice de covariance (en utilisant la normalisation par  $1/(n-1)$ ) vaut

$$C_X = \frac{1}{12} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

2. Justifier le fait que  $C_X$  soit diagonalisable.
3. La diagonalisation  $C_X = VDV^T$  de la matrice de covariance  $C_X$  conduit à une base orthonormée de  $\mathbb{R}^4$  constituée des vecteurs

$$V = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & -\sqrt{2}/2 & -1/2 & 0 \\ \sqrt{2}/2 & 0 & \sqrt{2}/2 & 0 \\ 1/2 & \sqrt{2}/2 & -1/2 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} (2+\sqrt{2})/6 & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 0 \\ 0 & 0 & (2-\sqrt{2})/6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

soit encore

$$V \approx \begin{pmatrix} 0 & 0 & 0 & 0 \\ .5 & -.707 & -.5 & 0 \\ .707 & 0 & .707 & 0 \\ .5 & .707 & -.5 & 0 \end{pmatrix}, \quad D \approx \begin{pmatrix} .569 & 0 & 0 & 0 \\ 0 & .167 & 0 & 0 \\ 0 & 0 & .098 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Expliciter les deux premiers vecteurs de **scores**, c'est à dire les projetés des lignes (les individus) de  $X_c$  sur les deux premiers axes principaux (c'est à dire les axes engendrés par les deux premiers vecteurs propres). Représenter graphiquement les individus dans le plan correspondant et commenter.

4. Effectuer ces manipulations sous R :
  - Calcul des moyennes et de la matrice centrée  $X_c$ .
  - Calcul de la matrice de covariance
  - Diagonalisation de la matrice de covariance
  - Projection sur les deux premiers axes factoriels
  - Représentation graphique

### 6.4 Régression linéaire

On s'intéresse ici à un problème quelque peu différent de celui de l'ACP. On suppose qu'on dispose d'un tableau de variables  $X \in \mathcal{M}_{np}(\mathbb{R})$ , et d'un vecteur  $Y \in \mathbb{R}^n$ , et on cherche à expliciter une éventuelle dépendance linéaire de  $Y$  par rapport à  $X$ , c'est à dire une relation de la forme

$$y_i = b_0 + x_{i1}b_1 + x_{i2}b_2 + \cdots + x_{ip}b_p + \varepsilon_i \approx b_0 + x_{i1}b_1 + x_{i2}b_2 + \cdots + x_{ip}b_p,$$

où les nombres  $b_0, \dots, b_p$  sont à déterminer, et où les nombres  $\varepsilon_i$  représentent l'erreur commise en approximant  $y_i$  par le membre de droite de la seconde ligne. Matriciellement, ceci s'écrit sous la forme

$$Y = b_0 \mathbf{1}_n + XB + E = X'B' + E,$$

où  $E = (\varepsilon_1, \dots, \varepsilon_n)^T$  est le vecteur des erreurs, et où on a introduit les matrices étendues

$$B' = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_p \end{pmatrix} \in \mathbb{R}^{p+1}, \quad X' = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \in \mathcal{M}_{n,p+1}.$$

Le problème de régression linéaire (appelé régression linéaire multiple dans le cas  $p > 1$ ) consiste à rechercher les nombres  $b_0, \dots, b_p$  qui rendent minimale la quantité

$$\|Y - X'B'\|^2 = \sum_i (Y - X'B')_i^2.$$

Résoudre ce problème est l'objectif de l'exercice qui suit, qui fait appel à des notions d'optimisation.

**Optimisation :** on sait que les minima ou maxima d'une fonction dérivable d'une variable réelle  $t \rightarrow f(t)$  sont donnés par les points dits "critiques", où la dérivée  $f'(t)$  de celle-ci s'annule. Ceci reste vrai pour les fonctions de plusieurs variables, et donc la fonction  $B' \mapsto \|Y - X'B'\|^2$ , à condition de généraliser la notion de dérivée.

La notion importante ici est celle de gradient, qui est simplement le vecteur composé des dérivées partielles : étant donnée une fonction  $f : X \in \mathbb{R}^p \rightarrow f(X) \in \mathbb{R}$ , son gradient est le vecteur

$$\nabla_X f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_p} \right)^T \in \mathbb{R}^p.$$

**Exercice 6.5 — Régression linéaire multiple.** On conserve les mêmes notations que ci dessus.

1. On rappelle que le produit scalaire de deux vecteurs quelconques  $V, W$  de même dimension s'écrit sous forme matricielle comme  $V \cdot W = V^T W$ . Vérifier que

$$\|Y - X'B'\|^2 = Y^T Y - 2Y^T X'B' + B'^T X'^T X'B'$$

2. On donne les règles de dérivation suivantes : étant donné un vecteur  $Y \in \mathbb{R}^p$  et une matrice  $A \in \mathcal{M}_p$

$$\nabla_X (Y^T X) = Y, \quad \nabla_X (X^T A X) = (A + A^T) X.$$

En appliquant ces règles à l'expression de la fonction  $B' \in \mathbb{R}^{p+1} \rightarrow \|Y - X'B'\|^2$ , calculer

$$\nabla_{B'} \|Y - X'B'\|^2.$$

3. Montrer que l'équation  $\nabla_{B'} \|Y - X'B'\|^2 = 0$  conduit au système linéaire

$$X'^T X'B' = Y^T X',$$

et en déduire la solution de ce système, donc les nombres  $b_0, \dots, b_p$  du modèle de régression. ■

## 6.5 Analyse discriminante de Fisher

On va ici s'intéresser à l'analyse discriminante dans le cas le plus simple, c'est à dire le cas de deux classes. Le problème est le suivant : étant donné un jeu de données (appelé *jeu d'apprentissage*), constitué de deux tableaux  $X^{(1)} \in \mathcal{M}_{n_1,p}$  et  $X^{(2)} \in \mathcal{M}_{n_2,p}$  à  $p$  variables et respectivement  $n_1$  et  $n_2$  individus, correspondant à deux classes, on cherche à construire une règle de décision permettant de discriminer ces deux classes de façon optimale. On cherche ici à classer des individus, chacun étant représenté par un vecteur de  $\mathbb{R}^p$ .

La règle cherchée est une règle linéaire : il s'agit de trouver une droite de  $\mathbb{R}^p$  telle que les projections des individus des deux classes sur cette droite soient les plus éloignés possibles.

On rappelle qu'une droite est caractérisée par un vecteur  $V \in \mathbb{R}^p$  (qu'on peut choisir de norme égale à 1 pour simplifier). Le projeté de tout vecteur  $U \in \mathbb{R}^p$  sur cette droite est proportionnel à  $V$ , donc de la forme

$$\Pi(U) = \alpha V, \quad \text{avec} \quad \alpha = U \cdot V = U^T V = V^T U.$$

**Exercice 6.6 — Projetés orthogonaux, variabilité inter-classes.**

Étant donnée une les tableaux  $X_1$  et  $X_2$  ci-dessus, vérifier que les coefficients  $\alpha$  (voir ci dessus) des projetés des individus forment deux vecteurs

$$A^{(1)} = X^{(1)}V \in \mathbb{R}^{n_1}, \quad A^{(2)} = X^{(2)}V \in \mathbb{R}^{n_2}.$$

2. On note  $m_1, m_2 \in \mathbb{R}^p$  les vecteurs (colonne) des moyennes de  $X^{(1)}$  et  $X^{(2)}$ . Vérifier que

$$m_1 = \frac{1}{n_1} \left( X^{(1)} \right)^T \mathbf{1}_{n_1}, \quad m_2 = \frac{1}{n_2} \left( X^{(2)} \right)^T \mathbf{1}_{n_2},$$

où  $\mathbf{1}_{n_1}$  est un vecteur de dimension  $n_1$  dont toutes les coordonnées sont égales à 1.

3. On note  $\mu_1, \mu_2 \in \mathbb{R}$  les moyennes de  $A_1$  et  $A_2$ . Exprimer  $\mu_1$  et  $\mu_2$  en fonction de  $m_1$  et  $m_2$ , puis montrer que

$$(\mu_1 - \mu_2)^2 = V^T S_B V, \quad \text{avec} \quad S_B = (m_1 - m_2)(m_1 - m_2)^T.$$

$S_B \in \mathcal{M}_p$  est appelée **matrice de dispersion inter-classes**. En notant  $\bar{X}^{(1)} \in \mathcal{M}_{n_1, p}$  et  $\bar{X}^{(2)} \in \mathcal{M}_{n_2, p}$  les matrices constituées des moyennes des variables, et  $\bar{A}^{(1)} \in \mathbb{R}^{n_1}$  et  $\bar{A}^{(2)} \in \mathbb{R}^{n_2}$  les vecteurs correspondants calculées sur les coefficients, il est possible de montrer aussi que  $\bar{A}^{(1)} = \bar{X}^{(1)}V$ , et similairement pour  $\bar{A}^{(2)}$ .

**Exercice 6.7 — Variabilité intra-classes.**

On s'intéresse maintenant aux dispersions des deux classes, que l'on mesure par les quantités

$$s_1^2 = \left( X^{(1)} - \bar{X}^{(1)} \right)^T \left( X^{(1)} - \bar{X}^{(1)} \right), \quad \delta_1^2 = \left( A^{(1)} - \bar{A}^{(1)} \right)^T \left( A^{(1)} - \bar{A}^{(1)} \right)$$

ainsi que les dispersions totales  $s^2 = s_1^2 + s_2^2$ ,  $\delta^2 = \delta_1^2 + \delta_2^2$ . Montrer que  $\delta^2$  s'exprime en fonction de  $V$  sous la forme

$$\delta^2 = V^T S_W V, \quad \text{avec} \quad S_W = S_1 + S_2 = \left( X^{(1)} - \bar{X}^{(1)} \right)^T \left( X^{(1)} - \bar{X}^{(1)} \right) + \left( X^{(2)} - \bar{X}^{(2)} \right)^T \left( X^{(2)} - \bar{X}^{(2)} \right)$$

$S_W$  est appelée **matrice de dispersion inter-classes**.

**Exercice 6.8 — Discriminant de Fisher.**

Le discriminant de Fisher est un critère permettant de déterminer le vecteur  $V$  optimal pour la séparation. Le critère revient à chercher le vecteur  $V$  qui assure à la fois une variabilité intra-classe la plus petite possible, tout en cherchant une variabilité inter-classe maximale. Ceci prend la forme

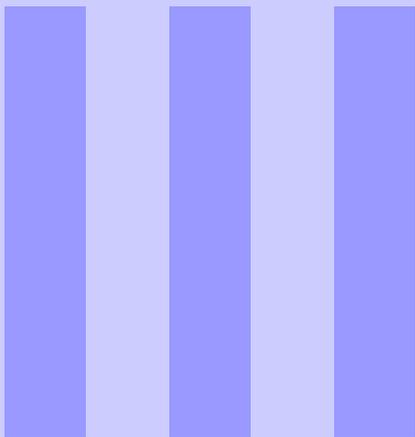
$$J(V) = \frac{\|\mu_1 - \mu_2\|^2}{\delta^2} = \frac{V^T S_B V}{V^T S_W V}.$$

1. **Optionnel** : en utilisant un gradient, ou l'inégalité de Cauchy-Schwarz, montrer qu'un vecteur  $V \in \mathbb{R}^p$  qui maximise le discriminant  $J(V)$  doit satisfaire une équation de la forme

$$S_B V = \lambda S_W V,$$

où  $\lambda \in \mathbb{R}$  est un réel quelconque (si  $S_W$  est inversible, ceci revient à dire que  $V$  est vecteur propre de  $S_W^{-1} S_B$ ).

2. Montrer que pour tout vecteur  $X \in \mathbb{R}^p$ ,  $S_B X$  est toujours proportionnel à  $m_1 - m_2$ . En déduire le  $V$  optimal.



# Annexes

<b>7</b>	<b>Quelques éléments sur R</b> .....	<b>107</b>
7.1	Introduction	
7.2	Utiliser R comme une calculatrice scientifique	
7.3	L'environnement sous R	
7.4	Variables, vecteurs, matrices, data.frames	
7.5	Graphiques	
7.6	Programmation, scripts et fonctions	
7.7	Installer et charger des packages	
7.8	Pour en savoir plus	



## 7. Quelques éléments sur R

### 7.1 Introduction

R est un logiciel statistique apparu en 1996 (qui succède aux logiciels S et Splus). Il propose à la fois un langage et un environnement permettant un traitement et une exploitation avancée de données statistiques (tests, régression, classification et apprentissage, analyse de variance, etc.). Contrairement à d'autres logiciels couramment utilisés dans le monde de l'entreprise (SAS, ...), R est un logiciel libre et gratuit, qui peut-être téléchargé à l'adresse

<http://cran.cict.fr/>

et installé sur la plupart des systèmes d'exploitation présents sur le marché : Linux, Windows, Mac-Os, etc. Le site contient également une quantité non négligeable d'informations : documentation, forums, FAQ et aide en ligne.

Il existe plusieurs interfaces graphiques pouvant être utilisées avec R. Dans ce cours on utilisera R-Studio, téléchargeable librement sur

<https://www.rstudio.com/>

et qui fournit notamment un éditeur de texte, une fenêtre graphique, et de nombreux autres outils.

A l'intérieur de R-Studio, les commandes peuvent être tapées directement dans la console de commande, soit groupées dans des fichiers (scripts ou fonctions), qui peuvent être exécutés directement.

### 7.2 Utiliser R comme une calculatrice scientifique



```
> 1+1  
[1] 2
```

Pour assigner une valeur à un objet (nous aborderons cette notion plus en détail dans la section suivante), il suffit d'utiliser l'une des commandes `<-`, `->` ou simplement `=`.



```
> x <- 2
> 3 -> y
> z = 4
```

Les valeurs des objets  $x$ ,  $y$ ,  $z$  ne s'affichent plus dans la console de commande, mais on peut les faire afficher, par exemple avec



```
> x
[1] 2
> print(z)
[1] 4
```

Les opérations arithmétiques de base s'effectuent de façon très intuitive, par exemple



```
> x + y
[1] 5
> x * z
[1] 8
> y / z
[1] 0.75
> y %% z
[1] 3
> y %/% z
[1] 0
```

Les deux dernières instructions correspondent à la division euclidienne, et retournent respectivement le reste et le quotient de la division de  $y$  par  $z$ .

Un grand nombre de fonctions mathématiques de base (logarithmes (`log`, `log10`, `log2`, exponentielle (`exp`), fonctions trigonométriques (`sin`, `cos`), racine carrée (`sqrt`), partie entière (`floor`)...) et bien d'autres sont également disponibles dans la distribution R de base, de même que la constante  $\pi$  (notée `pi`) ou encore  $i = \sqrt{-1}$  (notée `1i`... par ailleurs, on peut utiliser `4i` ou encore `1.999i` pour représenter respectivement 4 fois  $i$ , ou 1.999 fois  $i$ ). On pourra par exemple essayer les lignes suivantes



```
> cos(2*pi/3)
> log(2)
> log10(2)
> log2(2)
> exp(2i*pi/3)
> floor(2*pi)
```

#### Exercice 7.1 — Tester les commandes ci-dessus... et d'autres similaires

- Calculer en une ligne la racine carrée de l'exponentielle de la partie entière du log de  $3\pi/4$ .
- Utiliser l'instruction `help` en console de commande, ou l'onglet « help » de R-Studio pour obtenir des informations détaillées sur des fonctions (par exemple `help(floor)`).
- Tester l'instruction `apropos`, qui retourne une liste de fonctions correspondant à l'objet de la recherche. Par exemple, `apropos('cos')` retourne une liste de fonctions reliées au cosinus. On peut ensuite utiliser `help` pour obtenir des informations détaillées sur ces fonctions.

## 7.3 L'environnement sous R

### 7.3.1 Répertoire de travail

R peut avoir à charger des fichiers (programmes, données...) ou à sauvegarder des résultats (des graphiques...), et cela par défaut dans un répertoire de travail (*working directory*).

- Le répertoire courant peut être retrouvé en exécutant `getwd()` dans la console de commande ; sous R-Studio il est aussi identifié dans la bannière de la console de commande.
- Il peut être modifié en console de commande en exécutant `setwd('new_dir')` ; sous R-studio, on peut aussi utiliser la ligne 'Set Working Directory' du menu déroulant 'Session'.

C'est dans ce répertoire que seront sauvegardés par défaut les fichiers de code (scripts ou fonctions), ainsi que les variables que l'on souhaite sauvegarder.

**Exercice 7.2** — Choisir un répertoire de travail, et vérifier sa valeur avec l'instruction `getwd()`.

- Dans l'éditeur de texte, créer un nouveau fichier et le sauvegarder sous un nom simple (avec l'extension `.R`, par exemple, `monpremier.test.R`).
- Taper les trois lignes ci-dessous :
 

```
mes_variables <- ls()
print("Les variables de votre espace personnel sont :")
print(mes_variables)
```
- Sauvegarder le script ainsi obtenu, et le faire charger par R en exécutant en console de commande `source('monpremier.test.R')`, ou en utilisant le bouton correspondant de R-Studio.

### 7.3.2 Espace de travail

Toutes les variables manipulées lors d'une session de travail sous R restent en mémoire. Leur liste peut être obtenue avec la commande `ls()`. Il est possible d'en supprimer avec `rm(nom_variable)`.

Par exemple, essayer les lignes suivantes, et interpréter le résultat.



```
> u <- 5
> ls()
> rm(u)
> ls()
> rm(list=ls())
> ls()
```

## 7.4 Variables, vecteurs, matrices, data.frames

### 7.4.1 Mode des objets R

Le logiciel R fait la distinction entre différents types de données que l'on désigne sous le terme générique d'objets. Tous les objets de R (en particulier les variables, les vecteurs, les matrices, qu'on va voir plus loin...) possèdent au moins deux attributs : son mode (auquel on accède grâce à la fonction `mode`) et sa longueur (donnée par la fonction `length`). Suivant l'objet, ils peuvent également posséder d'autres attributs (comme par exemple des dimensions pour les matrices et tableaux).

Les principaux modes sont `numeric` (nombres réels), `complex` (nombres complexes), `logical` (valeurs booléennes, vrai/faux), `character` (chaînes de caractères), `function` (fonction), `list` (données quelconques) et `expression` (expressions non évaluées).

Par exemple la séquence d'instructions et résultats ci-dessous montre que l'objet `u` est un nombre (vecteur de longueur 1) réel, alors que `v` est une liste de chaînes de caractères, de longueur 2 (l'expression `c('tata', 'toto')` signifie que l'on collecte dans `v` les deux chaînes de caractères entre parenthèses.



```

> u <- 5
> mode(u)
[1] "numeric"
> length(u)
[1] 1
> v <- c('tata','toto')
> mode(v)
[1] "character"
> length(v)
[1] 2

```

### 7.4.2 Listes

Comme son nom l'indique, une liste est un ensemble d'objets, qui ne sont pas nécessairement du même type. Les éléments d'une liste peuvent être obtenus en utilisant le symbole \$.



```

> maliste <- list(u=3,v='toto',w=TRUE)
> maliste
$u
[1] 3

$v
[1] "toto"

$w
[1] TRUE

> maliste$w
[1] TRUE

```

### 7.4.3 Vecteurs

Les vecteurs sont les objets de base sous R. Par exemple, un scalaire est un vecteur de longueur 1.

#### Générer des vecteurs

Il existe différentes façons de créer des vecteurs. La fonction de base est `c`, qu'on a déjà rencontrée plus haut, et qui permet de générer un directement vecteur à partir de ses composantes. La fonction `vector` permet de créer un vecteur de mode donné, de longueur donnée, en initialisant ses composantes à une valeur déterminée. On pourra tester les instructions suivantes pour vérifier la valeur d'initialisation.



```

[backgroundColor=blue !5]
> c(1,2,3,4)
[1] 1 2 3 4
> vector('logical',4)
[1] {\tt FALSE FALSE FALSE FALSE}
> vector('numeric',4)
[1] 0 0 0 0
> vector('character',2)
[1] "" ""

```

Les fonctions `numeric`, `complex`, `logical`, `character` sont des raccourcis qui font appel à `vector`. On pourra par exemple comparer le résultat de `logical(4)` à l'une des lignes ci-dessus.

Il existe de nombreuses fonctions permettant de générer des vecteurs de caractéristiques diverses. Par exemple, l'expression `3:10` génère un vecteur de longueur 8 contenant tous les entiers de 3 à 10. On pourra tester l'expression `seq(3,10,0.25)` et interpréter le résultat (ou utiliser `help(seq)` pour comprendre). On citera aussi les fonctions `rnorm`, `runif`, `rexp`, `rs`, `rpois`, `rbinom` qui permettent de générer des

vecteurs constitués de nombres pseudo-aléatoires suivant des distributions données (respectivement, normale, uniforme, exponentielle, Student, Poisson, binômiale). Là encore, se référer au `help` pour connaître la syntaxe.

### Opérations sur les vecteurs

Une composante d'un vecteur peut s'obtenir en utilisant des crochets. Ainsi, `u[4]` retourne la 4-ème composante du vecteur `u`. On peut également extraire des sous-vecteurs de la même manière.



```
> u <- seq(3,10,0.25)
> u[4:8]
[1] 3.75 4.00 4.25 4.50 4.75
> u[c(1,3,5,7,9)]
[1] 3.0 3.5 4.0 4.5 5.0
```

Les opérations logiques entre vecteurs de même longueur sont par défaut des opérations terme à terme. Par exemple `u == v` va tester si les composantes de `u` et de `v` sont identiques, et renvoyer un résultat par composante. Si on désire tester globalement l'égalité des deux vecteurs, utiliser la fonction `identical` comme dans les lignes qui suivent. Le "et" (resp. le "ou") logiques s'obtiennent avec les opérateurs `&` et `&&` (resp `|` et `||`), en mode composante et vecteur.



```
> u <- c(1,2,3)
> v <- c(0,2,3)
> u == v
[1] FALSE TRUE TRUE
> identical(u,v)
[1] FALSE
```

Le "et" (resp. le "ou") logiques s'obtiennent avec les opérateurs `&` et `&&` (resp `|` et `||`), en mode composante et vecteur.



```
> u & v
[1] FALSE TRUE TRUE
> u && v
[1] FALSE
```

De même, les opérations arithmétiques entre deux vecteurs de même dimension sont des opérations terme à terme. Une opération arithmétique entre deux vecteurs de même dimension retourne un vecteur dont les éléments correspondent à l'opération arithmétique effectuée pour l'indice correspondant des vecteurs donnés en argument. De même, toutes les fonctions mathématiques implémentées sous R agissent composante par composante sur les vecteurs, comme on peut le voir dans les lignes ci-dessous



```
> u <- 1:5; v <- 2:6
> u+v
[1] 3 5 7 9 11
> u*v
[1] 2 6 12 20 30
> u/v
[1] 0.5000000 0.6666667 0.7500000 0.8000000 0.8333333
> log(u)
```

```
[1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379
> exp(u)
[1] 2.718282 7.389056 20.085537 54.598150 148.413159
```

Notons au passage que la première ligne contient deux instructions, qu'il faut séparer par un point-virgule ";" :

### Exercice 7.3

Soit  $x$  un vecteur contenant les valeurs d'un échantillon :



```
> x
[1] 1 18 23 12 7 4 14 10 5 8 99 5 88 43 5 12 52 33 4 6 6 33
```

- Écrire des expressions R permettant d'extraire les éléments suivants
  - Le deuxième élément de l'échantillon.
  - Les cinq premiers éléments de l'échantillon.
  - Les éléments strictement supérieurs à 14.
  - Tous les éléments sauf les éléments en positions 6, 10 et 12.
  - Tous les éléments dans l'ordre inverse
- Calculer la somme et le produit des éléments de  $x$ . Calculer la moyenne, la variance et l'écart-type de  $x$ .
- Que retournent `cumsum(x)` et `cumprod(x)` ?

Parmi les fonctions usuelles s'appliquant sur des vecteurs, on peut mentionner `sum` (somme des composantes), `prod` (produit des composantes), `cumsum` et `cumprod` (se renseigner avec l'aide en ligne), ainsi que `mean` (moyenne arithmétique des composantes) et `var` (variance). Plus de détails dans l'aide en ligne.

#### 7.4.4 Matrices, tableaux

Les matrices et tableaux ne sont rien d'autre que des vecteurs dotés d'un attribut `dim`. Ainsi, une matrice est un vecteur avec un attribut `dim` de longueur 2, alors que les tableaux peuvent avoir des valeurs entières positives quelconques pour `dim`.

##### Génération de matrices et tableaux

La fonction de base pour créer une matrice est la fonction `matrix`. Par exemple, l'instruction `matrix(1:8,nrow=2,ncol=4)` génère une matrice à 2 lignes et 4 colonnes, dans laquelle les entiers compris entre 1 et 8 sont stockés colonne par colonne. Si on désire que ces entiers soient stockés ligne par ligne, il faut le préciser en modifiant l'instruction en `matrix(1:8,nrow=2,ncol=4,byrow=T)`.



```
> matrix(1:8,nrow=2,ncol=4)
  [,1] [,2] [,3] [,4]
[1,]  1  3  5  7
[2,]  2  4  6  8
> matrix(1:8,nrow=2,ncol=4,byrow=T)
  [,1] [,2] [,3] [,4]
[1,]  1  2  3  4
[2,]  5  6  7  8
```

La fonction `diag` permet soit de générer des matrices diagonales (lorsqu'on leur passe un vecteur en variable d'entrée, ou un entier auquel cas une matrice identité est générée), soit d'extraire la diagonale d'une matrice carrée :



```

> diag(3)
  [,1] [,2] [,3]
[1,]  1   0   0
[2,]  0   1   0
[3,]  0   0   1
> diag(1:3)
  [,1] [,2] [,3]
[1,]  1   0   0
[2,]  0   2   0
[3,]  0   0   3
> diag(diag(1:3))
[1] 1 2 3

```

Les tableaux peuvent avoir un attribut `dim` de longueur quelconque (on parle parfois aussi de tenseurs). Par exemple, les instructions ci-dessous génèrent une matrice  $2 \times 4$  et un tableau  $2 \times 2 \times 2$  remplis des mêmes éléments.



```

> array(1:8,dim=c(2,4))
  [,1] [,2] [,3] [,4]
[1,]  1   3   5   7
[2,]  2   4   6   8
> array(1:8,dim=c(2,2,2))
, , 1
  [,1] [,2]
[1,]  1   3
[2,]  2   4
, , 2
  [,1] [,2]
[1,]  5   7
[2,]  6   8

```

### Opérations sur les matrices et tableaux

Les opérations sur les matrices et tableaux suivent la même logique que les opérations sur les vecteurs : ce sont par défaut des opérations composante par composante. Par exemple le carré (élément par élément) de l'une des matrices ci-dessus s'obtient comme suit :



```

> matrix(1:8,nrow=2,ncol=4) ^2
  [,1] [,2] [,3] [,4]
[1,]  1   9  25  49
[2,]  4  16  36  64

```

Les outils élémentaires du calcul matriciel (transposition, produit matriciel, produit scalaire, inversion lorsque possible, pseudo-inverse,...) sont aussi disponibles sous R; Pour obtenir le produit matriciel de matrices conformantes, on utilise l'opérateur `%*%`, par exemple l'instruction ci-dessous effectue le produit

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$



```

> matrix(1:4,nrow=2,ncol=) %*% matrix(rep(2,4),nrow=2,ncol=2)
  [,1] [,2]
[1,]  8   8
[2,] 12  12

```

Description	Opérateur ou fonction
Transposée $A^T$	<code>t(A)</code>
Produit matriciel $AB$	<code>A%*%B</code>
Produit extérieur $AB^T$	<code>A%O%B</code>
Produit croisé $A^T B$	<code>crossprod(A,B)</code>
Inverse d'une matrice carrée $A$	<code>solve(A)</code>
Solution du système linéaire $Ax = B$	<code>solve(A,b)</code>
Diagonalisation	<code>eigen(A)</code>
Concaténation horizontale	<code>cbind(A,B,...)</code>
Concaténation verticale	<code>rbind(A,B,...)</code>

TABLE 7.1 – Principales fonctions et opérations du calcul matriciel

Les principales opérations sont listées dans la table 7.1 (où on suppose implicitement que les opérations ont un sens).

A noter, il est possible de modifier les dimensions d'une matrice (en conservant son nombre d'éléments) en utilisant la fonction `dim`, voire de transformer une matrice en vecteur avec la fonction `as.vector` ou un vecteur en matrice avec la fonction `as.matrix`. Quelques exemples suivent, voir l'aide en ligne pour en savoir davantage.



```
> a = matrix(c(pi,exp(1),1i,log(2)),2,2)
> b = a; dim(b) = c(1,4)
> c = as.vector(a)
> a
      [,1]      [,2]
[1,] 3.141593+0i 0.000000+1i
[2,] 2.718282+0i 0.6931472+0i
> b
      [,1]      [,2] [,3]      [,4]
[1,] 3.141593+0i 2.718282+0i 0+1i 0.6931472+0i
> c
[1] 3.1415927+0i 2.7182818+0i 0.0000000+1i 0.6931472+0i
```

#### Exercice 7.4

1. Etant donné un vecteur  $X = (x_1, x_2, \dots, x_n)$  et un vecteur de poids  $w = (w_1, w_2, \dots, w_n)$ , la moyenne pondérée de  $X$  s'écrit

$$X_w = \frac{1}{w_\Sigma} \sum_{i=1}^N w_i X_i, \quad \text{où } w_\Sigma = \sum_{i=1}^N w_i.$$

Ecrire une expression R retournant une telle moyenne pondérée. On pourra prendre l'exemple  $x = (1, 8, 12, 3, 2, 20, 11, 14)$  et  $w = (0.2, 0.11, 0.24, 0.05, 0.16, 0.08, 0.12, 0.21, 0.19)$ .

2. De même, on définit les moyennes pondérées d'une matrice d'observation  $X \in \mathcal{M}_{n,p}$  (avec une matrice de poids  $w \in \mathcal{M}_{n,p}$ ) par

$$X_{iw} = \frac{1}{w_{i\Sigma}} \sum_{j=1}^p w_{ij} X_{ij}, \quad \text{où } w_{i\Sigma} = \sum_{j=1}^p w_{ij}$$

$$X_{wj} = \frac{1}{w_{\Sigma j}} \sum_{i=1}^N w_{ij} X_{ij}, \quad \text{où } w_{\Sigma j} = \sum_{i=1}^N w_{ij}$$

$$X_w = \frac{1}{w_\Sigma} \sum_{i=1}^N \sum_{j=1}^p w_{ij} X_{ij}, \quad \text{où } w_\Sigma = \sum_{i=1}^N \sum_{j=1}^p w_{ij}$$

Ecrire des expressions R retournant de telles moyennes pondérées. On pourra utiliser une matrice

$X$  aléatoire, et des poids uniformes (tous égaux à 1 par exemple).

#### Exercice 7.5

Résoudre en une ligne le système, en utilisant la fonction `solve`.

$$\begin{cases} x + 2y + 3z = 1 \\ 4x + 5y + 6z = 2 \\ 7 + 8y + 10z = 3 \end{cases}$$

#### Exercice 7.6

Se renseigner sur la fonction `apply`. Générer un tableau à trois dimensions (on pourra par exemple prendre `u=array(1:16,2,2,4)`), et utiliser les fonctions `apply` et `mean` pour en déduire le tableau à deux dimensions obtenu en prenant la moyenne par rapport à la dernière dimension.

### 7.4.5 data.frames

Un data frame est un objet constitué de vecteurs de même longueur mais pouvant être de mode différent. Ce type d'objet peut par exemple contenir à la fois des données numériques, booléennes,...



```
> Numero = 1:4
> Vitesse = c(290,250,160,130)
> Rapide = c(T,T,F,F)
> df = data.frame(Numero,Vitesse,Rapide,row.names=c('Ferrari','Porsche','Twingo','Smart'))
> df
```

	Numero	Vitesse	Rapide
Ferrari	1	290	TRUE
Porsche	2	250	TRUE
Twingo	3	160	FALSE
Smart	4	130	FALSE

Dans cet exemple, la variable optionnelle `row.names` permet de donner des noms aux lignes. Il est possible d'extraire une sous-partie d'un data-frame, comme sur une matrice.



```
> df[2:4,1:2]
```

	Numero	Vitesse
Porsche	2	250
Twingo	3	160
Smart	4	130

On peut également utiliser la fonction `attach` pour rendre visibles individuellement les colonnes d'un data frame, ... puis `detach` pour les occulter de nouveau. Voir l'aide en ligne.

Lorsque les colonnes d'un data.frame le permettent, il est possible d'effectuer des opérations directement sur ce data.frame (par exemple, des additions, des moyennes), et bien d'autres opérations statistiques.



```
> colMeans(df[,1:2])
Numero Vitesse
  2.5   207.5
```

La commande `summary` est une commande générique, qui fournit un résumé d'un objet quelconque, en particulier un data frame :



```
> summary(df)
      Numero      Vitesse      Rapide
Min.   :1.00  Min.   :130.0  Mode :logical
1st Qu.:1.75  1st Qu.:152.5  FALSE:2
Median :2.50  Median :205.0  TRUE :2
Mean   :2.50  Mean   :207.5
3rd Qu.:3.25  3rd Qu.:260.0
Max.   :4.00  Max.   :290.0
```

Signalons enfin des fonctions telles que `data.matrix` permettant de convertir un data frame en matrice, en convertissant ses éléments au mode numérique.



```
> data.matrix(df)
      Numero Vitesse Rapide
Ferrari     1     290      1
Porsche     2     250      1
Twingo     3     160      0
Smart       4     130      0
```

## 7.5 Graphiques

On distingue des fonctions graphiques de deux niveau :

- Fonctions de haut niveau, du type `plot` (qui sera la fonction de référence pour nous), `boxplot`, `hist`,... qui créent un nouveau graphique
- Fonctions de bas niveau, qui permettent d'ajouter des objets au graphique courant et de raffiner celui-ci. On peut par exemple mentionner les fonctions `points`, `lines`,... Dans tous les cas, on pourra se référer à l'aide en ligne pour les détails, la syntaxe et les options de ces fonctions.

A titre d'exemple, on pourra tester les lignes qui suivent, qui permettent de tracer dans le plan deux vecteurs  $(x_1, y_1)$  et  $(x_2, y_2)$  (obtenus en générant des nombres pseudo-aléatoires de moyenne et écart-type donnés), en jouant avec la couleur et les symboles. Le graphique correspondant se trouve en Figure 7.1



(Nuages de points)

```
> x1 = rnorm(100)
> y1 = rnorm(100)
> x2 = rnorm(80,mean=-1,sd=0.5)
> y2 = rnorm(80,mean=-3,sd=1.5)
> xrange = range(c(x1,x2))
> yrange = range(c(y1,y2))
> plot(x1,y1,xlim = xrange + c(-1,1),ylim = yrange + c(-1,1),col='blue',xlab='x',ylab='y')
> points(x2,y2,col='red',pch=17)
```

Ici, les vecteurs `xrange` et `yrange` servent à préciser les domaines de valeurs de  $x$  et  $y$  à représenter (entrés dans `xlim` et `ylim` et calculés automatiquement ici), `col` permet de sélectionner la couleur des points, `pch` les symboles,... Un très grand nombre de paramètres sont réglables, se référer à la documentation en ligne pour plus de détails.

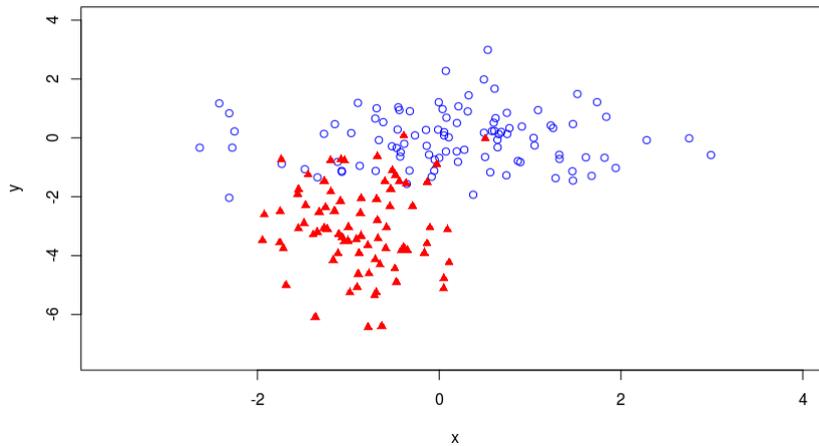


FIGURE 7.1 – Un exemple de graphique

**Exercice 7.7**

1. Tracer la fonction sin sur l'intervalle  $[0, 10]$ , en imposant les limites  $[-1.5, 1.5]$  pour l'axe des ordonnées (option `ylim`), en donnant le titre 'Graphique de la fonction sinus' au graphique, le nom 'Angle' à l'axe des abscisses et le nom 'Sinus' à l'axe des ordonnées.
2. Rajouter la droite d'équation  $y = x/2\pi$  sur le graphique précédent avec la fonction `lines`, en rouge. En utilisant `points`, ajouter les points de coordonnées  $(0, 1.3)$  et  $(2, -1)$ , avec le symbole '\*' en vert.

Les possibilités de R en termes de graphiques sont multiples. Par exemple, il est possible de séparer une fenêtre graphique en sous fenêtres en utilisant l'instruction `par(mfrow=c(nb_lignes,nb_cols))`, comme le montre l'exemple ci-dessous, qui représente côte à côte un histogramme et un boxplot d'un vecteur de nombres pseudo-aléatoires. Le résultat se trouve en Figure 7.2

**(Histogramme et boxplot)**

```
> u = rnorm(1000,mean=3,sd=5)
> par(mfrow=c(1,2))
> hist(u)
> boxplot(u)
> help(boxplot)
> title('Boxplot')
```

**7.6 Programmation, scripts et fonctions****7.6.1 Structures de contrôle**

**Exécution conditionnelle** : `if`, `else`, `switch`

Il s'agit ici d'exécuter une instruction seulement si une certaine condition est remplie. La syntaxe est

```
if (condition) code.vrai else code.faux
```

où `code.vrai` et `code.faux` représentent chacun une instruction ou une série d'instruction (auquel cas il faut les encadrer par des accolades).

**(Test)**

```
> x = -5
> if (x>0) print('x positif') else {x = -x; print('x negatif, je change son signe')}
[1] "x negatif, je change son signe"
```

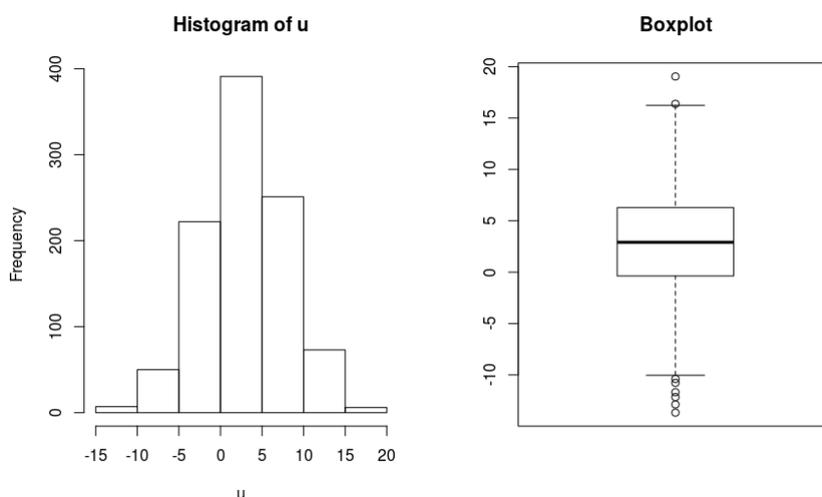


FIGURE 7.2 – Séparation de la fenêtre graphique en deux sous-fenêtres

### Boucles

Il est souvent préférable d'éviter d'utiliser des boucles, généralement plus lentes que si les opérations sont effectuées sous forme vectorielle.

La boucle la plus courante est la boucle `for`, dont la syntaxe est

```
for (variable in suite) série d'instructions
```

Ici les instructions doivent être placées entre accolades s'il y en a plus d'une. `suite` est une suite de valeurs pouvant être prises par `variable` (qui ne sont pas nécessairement des nombres ordonnés, ni même des nombres tout court). Les quelques lignes ci-dessous donnent un calcul simple de la factorielle de 5.



(Boucle `for`)

```
> k = 5
> fact_k = 1
> for (n in 2:k) fact_k = fact_k * n
> fact_k
[1] 120
```

Une alternative est la boucle `while`, dont la syntaxe est

```
while (condition) série d'instructions
```



(Boucle `while`)

```
> k = 5
> n = 1
> fact_k = 1
> while (n <= 5) {fact_k = fact_k*n; n = n + 1}
> fact_k
[1] 120
```

Voir aussi `switch`. On pourra également se documenter sur les instructions `repeat` et `\break`, qui sont moins utilisées.

### Exercice 7.8

1. En utilisant une boucle `for`, calculer la somme des 100 premiers entiers et la somme de leurs carrés.
2. Même question avec une boucle `while`.

### 7.6.2 Utilisation de scripts

Dans la console, la saisie des instructions est réalisée ligne à ligne, et il n'est pas possible d'avoir une vision globale d'une séance de travail. Les scripts permettent de saisir directement une séquence d'instructions, les sauvegarder (dans un fichier texte) pour pouvoir les exécuter globalement, les modifier si nécessaire, les exécuter de nouveau,...

L'environnement de développement RStudio (comme la plupart des autres environnements de R), permet de manipuler, sauvegarder et exécuter des scripts. L'exécution d'un script se fait via la fonction `source`.

### 7.6.3 Utilisation de fonctions

Il est en général utile de regrouper certains groupe d'instructions utilisés fréquemment sous forme de fonctions. Une fonction est un objet défini par la syntaxe suivante :

```
ma_fonction = function(arguments) expression
```

où `ma_fonction` est le nom de la fonction, `arguments` est la liste des arguments, séparés par des virgules, et `expression` constitue le corps de la fonction, soit une expression simple soit un groupe d'expressions réunies par des accolades. L'exemple qui suit est une fonction qui calcule la factorielle d'un nombre entier.



```
ma_factorielle = function(k){
  if (k-floor(k)!=0) stop('argument non entier')
  if (k<0) stop('argument negatif')
  if (k == 0 | k==1)
    fact_k = 1
  else{
    fact_k = 1
    for (n in 2:k)
      fact_k = fact_k * n
  }
  fact_k
}
```

Ici les premières lignes servent à tester si l'argument est un entier positif ou nul.

Une fonction est sauvegardée dans un fichier texte (que l'on peut générer sous R-studio avec l'éditeur de texte intégré), par exemple `ma_factorielle.R`. Pour que la fonction puisse être exécutée, il faut "sourcer" le fichier en question, avec l'instruction `source('ma_factorielle.R')`. Cette opération doit être effectuée après chaque modification du fichier afin que celle-ci soit prise en compte. Noter qu'un fichier texte peut contenir plusieurs fonctions.

Une fonction retourne tout simplement le résultat de la dernière expression du corps de la fonction, dans l'exemple ci-dessus, le résultat du calcul de la factorielle :



```
> ma_factorielle(3)
[1] 6
> ma_factorielle(-3)
Error in ma_factorielle(-3) : argument negatif
```

Dans le cas où la fonction doit retourner plusieurs résultats, le plus simple est de les organiser dans une liste. Par exemple, la fonction ci-dessous retourne la moyenne et la médiane d'un vecteur



```
mm = fonction(u){
  a = mean(u)
  b = median(u)

  list(mo=a,me=b)
}
```

L'exécution de cette fonction donne



```
> u=rnorm(100,mean=1)
> v = mm(u)
> v
$mo
[1] 0.8662206
$me
[1] 0.7833275
> v$mo
[1] 0.8662206
```

### Exercice 7.9

Ecrire une fonction `Mention` prenant pour argument une note entre 0 et 20 et retournant la mention correspondante : A pour "ajourné", P pour "passable", AB,B ou TB. Faire en sorte que la fonction affiche un message d'erreur approprié si la note n'est pas correcte (par exemple une valeur négative ou supérieure à 20).

## 7.7 Installer et charger des packages

Un package R est une bibliothèque de fonctions réalisant certains traitements particuliers, généralement regroupés sur un thème donné. Pour utiliser un package, il faut qu'il ait été préalablement installé puis chargé. Il existe maintenant un très grand nombre de packages disponibles. Certains d'entre eux se trouvent déjà dans la distribution R de base, d'autres se trouvent installés et chargés par défaut dans R-studio.

L'installation d'un package se fait via la fonction `install.packages` en ligne de commande, mais peut aussi se faire via l'onglet "packages" de R-studio (qui fournit un outil de recherche automatique). Le chargement d'un package en ligne de commande se fait en utilisant la fonction `library`, ou en sélectionnant le package dans la liste donnée par l'onglet "packages" de R-studio.

A titre d'exemple, on pourra essayer d'installer et charger le package `gplot`, qui fournit des outils graphiques avancés.

## 7.8 Pour en savoir plus

Il existe de nombreuses ressources bibliographiques disponibles pour R. Une source importante de références, manuals,... est la page "The R manuals"

<https://cran.r-project.org/manuals.html>

qui regroupe des manuels génériques et plus spécialisés, en anglais. Quelques documents en français (ainsi que d'autres langues) se trouvent sur

<https://cran.r-project.org/other-docs.html>

On pourra aussi consulter l'ouvrage très complet *Introduction à la programmation en R* de Vincent Goulet, disponible en ligne sur

[https://svn.fsg.ulaval.ca/svn-pub/vgoulet/documents/intro\\_r/](https://svn.fsg.ulaval.ca/svn-pub/vgoulet/documents/intro_r/)

Ceci dit, l'aide en ligne est certainement l'outil le plus précieux (avec l'instruction `apropos`).

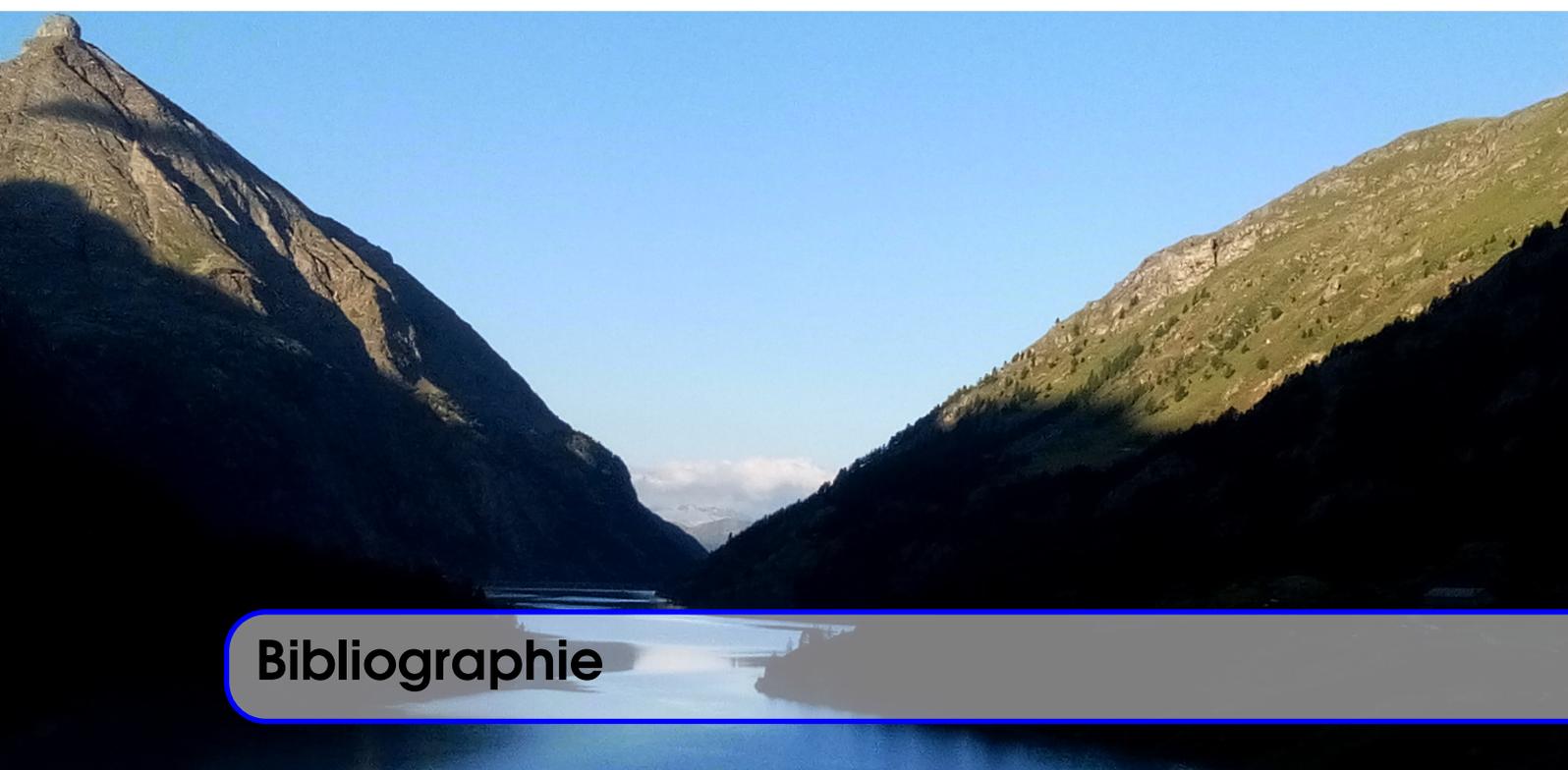
# IV

## Compléments

**Bibliographie** ..... 123

**Index** ..... 125

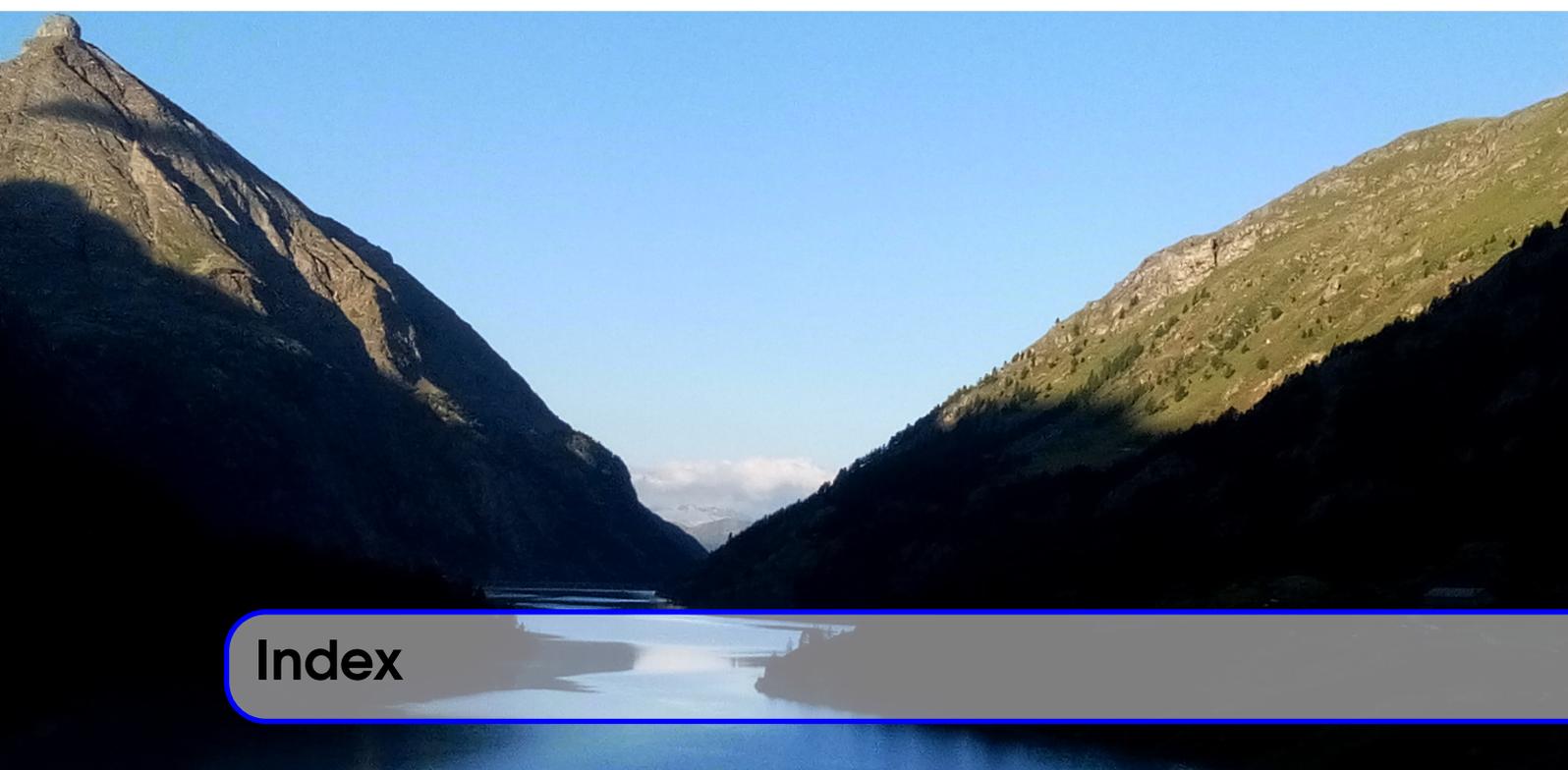




## Bibliographie

- [1] J. Grifone. *Algèbre linéaire*. Cépaduès-Éditions, 2018.
- [2] MéthodeMath. Géométrie dans l'espace, 2021. [https://www.methodemaths.fr/geometrie\\_espace/](https://www.methodemaths.fr/geometrie_espace/).
- [3] MéthodeMath. Géométrie du plan, 2021. [https://www.methodemaths.fr/geometrie\\_plan/](https://www.methodemaths.fr/geometrie_plan/).
- [4] G. Strang and S. Dufour. *Introduction à l'algèbre linéaire*. Coursus. 2015. Traduction Française par S. Dufour.
- [5] Wikipedia. Vecteur, 2021. <https://fr.wikipedia.org/wiki/Vecteur>.





# Index

- ACP (PCA), 57
- Analyse Discriminante Linéaire (*Linear Discriminant Analysis*), 92
- Analyse en composantes principales (*principal component analysis*), 57
- Angle de deux vecteurs (*angle between two vectors*), 26
- Associativité du produit matriciel (*associativity of the matrix product*), 48
- Atome de Gabor (*Gabor atom*), 36
- Atome temps-fréquence (*Time-frequency atom*), 36
  
- Base canonique (*canonical basis*), 20
- Base canonique de l'espace (*canonical basis of the 3D space*), 18
- Base canonique du plan (*canonical basis of the plane*), 16
- Base d'un espace vectoriel (*basis of a vector space*), 21
- Base d'un sous-espace vectoriel (*basis of a vector subspace*), 25
- Base de Fourier (*Fourier basis*), 32
- Base orthonormée d'un sous-espace vectoriel (*orthonormal basis of a vector subspace*), 28
- Base orthonormée de  $\mathbb{R}^N$  (*orthonormal basis in  $\mathbb{R}^N$* ), 28
- Bipoint, 15
- Bipoints équipollents, 16
  
- Changement de base (*change of basis*), 23, 50
- Classification (*classification*), 88
- Coefficient de corrélation (*correlation coefficient*), 61
  
- Coefficients (*coefficients of a vector*), 21
- Coefficients de Fourier (*Fourier coefficients*), 32
- Combinaison linéaire (*linear combination*), 20, 21
- Complexe conjugué (*complex conjugate*), 31
- Coordonnées (*coordinates*), 21
- Coordonnées d'un vecteur (*coordinates of a vector*), 19
  
- Discriminant de Fisher (*Fisher's discriminant*), 89
- Dispersion inter-classes (*between-class dispersion matrix*), 90
- Dispersion intra-classes (*within-class dispersion*), 90
- Domaine de définition (*domain*), 77
- Déterminant d'une famille de vecteurs (*determinant of a family of vectors*), 27
  
- EEG (*EEG*), 86
- Espace Euclidien (*Euclidean space*), 26
- Espace Hermitien (*Hermitean space*), 31
- Espace vectoriel (*vector space*), 20
- Extremum d'une fonction (*extremum of a function*), 80
- Extremum global (*global extremum*), 85
- Extremum local (*local extremum*), 85
  
- Fonction objective (*objective function*), 77
- Fonction partielle (*partial function*), 81
- Formule de Pythagore-Parseval (*Pythagore-Parseval formula*), 29
- Fréquence (*frequency*), 32
- Fréquence d'échantillonnage (*sampling frequency*), 32

- Fréquence fondamentale (*fundamental frequency*), 34
- Gradient (*gradient*), 81, 82
- Graphes d'une fonction (*graph of a function*), 78
- Individu (*individual*), 58
- Inégalité de Cauchy-Schwarz (*Cauchy-Schwarz inequality*), 26
- Jeu d'apprentissage (*training set*), 74
- Jeu de données d'apprentissage (*training dataset*), 88
- Jeu de données de test (*test dataset*), 88
- Jeu de données de test (*test set*), 74
- Laplacien (*Laplacian*), 83
- LDA (*LDA*), 92
- Matrice (*matrix*), 39
- Matrice d'une famille de vecteurs (*matrix of a family of vectors*), 39
- Matrice de corrélations empirique (*sample correlation matrix*), 62
- Matrice de covariance empirique (*sample covariance matrix*), 60
- Matrice de Gram (*Gram matrix*), 49
- Matrice diagonale (*diagonal matrix*), 41
- Matrice Hessienne (*Hessian matrix*), 83
- Matrice identité (*identity matrix*), 40
- Matrice inverse (*inverse matrix*), 23
- Matrice orthogonale (*orthogonal matrix*), 51
- Matrice symétrique (*symmetric matrix*), 40
- Matrice transposée (*transposed matrix*), 40
- Matrices compatibles (*conformable matrices*), 44, 47
- Maximisation (*maximization*), 77
- MEG (*MEG*), 86
- Norme (*Norm*), 15
- Norme d'un vecteur (*norm of a vector*), 26
- Optimisation (*optimization*), 77
- Opérateur Hessien (*Hessian operator*), 84
- Opérateur Laplacien (*Laplacian operator*), 83
- Orientation (*Orientation*), 16
- Partie imaginaire d'un nombre complexe (*imaginary part of a complex number*), 31
- Partie réelle d'un nombre complexe (*real part of a complex number*), 31
- Point selle (*saddle point*), 85
- Produit de Hadamard (*Hadamard product*), 44
- Produit de matrices compatibles (*product of conformable matrices*), 47
- Produit des matrices  $2 \times 2$  (*product of  $2 \times 2$  matrices*), 43
- Produit des matrices  $3 \times 3$  (*product of  $3 \times 3$  matrices*), 43
- Produit scalaire (*inner product*), 26
- Projection orthogonale (*orthogonal projection*), 29, 53
- Rang d'une famille de vecteurs (*rank of a family of vectors*), 25
- Relation de Chasles (*Chasles' identity*), 16
- Règle de décision linéaire (*linear decision rule*), 88
- Régions de décision (*decision regions*), 88
- Régression linéaire (*linear regression*), 102, 103
- Régression linéaire multiple (*multiple linear regression*), 103
- Seuil (*threshold*), 89
- Signal (*signal*), 31
- Signaux constants par morceaux (*piecewise constant signals*), 35
- Sinusoïde (*sine wave*), 32
- Sous-espace vectoriel (*vector subspace*), 23, 53
- Sous-espace vectoriel engendré (*vector subspace spanned by a family of vectors*), 25
- Sous-espaces vectoriels orthogonaux (*orthogonal vector subspaces*), 28
- Spectrogramme (*spectrogram*), 37
- Sur-apprentissage (*overfitting*), 92
- Surface associée à une fonction de deux variables (*surface of a function of two variables*), 78
- Symétrie Hermitienne (*Hermitean symmetry*), 33
- Système linéaire (*linear system*), 22
- Système linéaire paramétrique (*parametric linear system*), 22
- Trace (*trace*), 61
- Transformation de Fourier Rapide (*Fast Fourier Transform*), 33
- Transformée de Fourier à fenêtre glissante (*Sliding window Fourier transform*), 36
- Transformée de Gabor (*Gabor transform*), 36
- Transposition (*transposition*), 20
- Valeur propre (*eigenvalue*), 63
- Valeur propre multiple (*multiple eigenvalue*), 64
- Variabilité totale (*total variability*), 57
- Variable (*variable*), 58
- Variance (*variance*), 59
- Vecteur discriminant (*discriminant vector*), 89
- Vecteur nul (*null vector*), 20
- Vecteur opposé (*opposite of a vector*), 20
- Vecteur propre (*eigenvector*), 63
- Vecteurs orthogonaux (*orthogonal vectors*), 28
- Écart type (*standard deviation*), 59
- Élément de matrice (*matrix element*), 39