Projet STATIS: Réponse feuille de route

ATTOUMANI Ibrahim et MANNEQUIN Jeanne

Contents

Introduction

L'Analyse en Composantes Principales (ACP), ou Principal Component Analysis (PCA) en anglais, est une technique statistique utilisée pour réduire la dimensionnalité d'un ensemble de données tout en conservant autant que possible la variance présente dans les données d'origine. Elle transforme les variables d'origine en un nouvel ensemble de variables non corrélées appelées composantes principales. En réduisant la complexité des données tout en préservant leur essence, elle permet de mieux comprendre les structures sous-jacentes des données et de préparer ces dernières pour une analyse plus approfondie. L'ACP trouve des applications dans de nombreux domaines, allant de la reconnaissance de formes et de la bioinformatique à la finance et au marketing, illustrant sa polyvalence et son importance dans le domaine de l'analyse des données. \
L'ACP que nous avons l'habitude de voir est utilisable sur des tableaux de données à deux entrées. Mais que faire dans le cas de tableaux à trois dimensions ? Il nous faut trouver une méthode permettant d'analyser les données dans leur ensemble. \ On se propose d'étendre l'ACP à l'analyse d'un multi-tableau. L'objectif est de comparer les individus et les variables de plusieurs sous-tableaux. Il existe plusieurs façons de la faire, STATIS en est une. STATIS est une méthode qui permet de synthétiser les informations contenues dans les différents tableaux en une seule représentation. \
Il existe aussi plusieurs types de multi-tableaux. Nous en distinguons deux : les multi-tableaux à trois entrées (situation 1) et les multi-tableaux avec partition thématique (situation 2). \ - Le tableau à trois entrées est un tableau de taille $n \times p \times T$, où n est le nombre d'individus, p le nombre de variables et T les différentes dates. Les indices seront notés $1 \leq i \leq n$, $1 \leq j \leq p$ et $1 \leq t \leq T$ \ - Le tableau avec partition thématique est composé de q tableaux décrivants les n individus à l'aide de groupes de variables différents, chaque groupe appartenant conceptuellement à un thème précis est regoupé dans un sous-tableau composant le multi-tableau. Chacun de ces sous-tableaux X_m, avec $1 \leq m \leq q$, possède p_m variables (colonnes).

1 Situation 1

Dans cette première partie nous utilisons les tableaux à trois entrées. Un tel tableau peut être décomposé en T tableaux juxtaposés de dimensions identiques (n, p). Cette identité de dimension entre les T tableaux homologues permet une extension de l'ACP dans laquelle on considère chaque tableau comme une "variable" décrivant $n \times p$ "individus" (i, j). Nous formalisons d'abord cette extension de l'ACP, STATIS 1.

1.1 Préliminaires

Il est possible de fabriquer ou de trouver un tableau à trois entrées (INSEE, ...). R propose justement un jeu de données (simulated) du package "multiblock", composé de 4 tableaux (A,B, C et D), de chacun 200 individus et 10 varaibles. Ces tableaux, variables et individus ne portent pas de nom mais des lettres ou numéros.

Puisque nos valeurs sont des indicateurs numériques, on les centre-réduit.

1.1.1 Produit scalaire

La matrice diagonale des poids des n individus est W (par défaut, $W = \frac{1}{n} I_n$). On considérera également une matrice diagonale des poids des p colonnes : $C = \frac{1}{p} I_p$ par défaut.

1. Le produit scalaire entre deux matrices A et B de taille (n, p) est :

$$[A|B] = \mathrm{tr}(CA'WB)$$

La norme d'une matrice A correspondant à ce produit scalaire sera notée $[|A|]$.

a) Ecrivons le produit scalaire sous forme $tr(\tilde{A}'\tilde{B})$ (produit scalaire de Frobénius) en explicitant la transformation Z vers $\tilde{Z}$, $\forall Z$ (n, p).

Soit $A, B \in M_{n,p}(\mathbb{R})$

$$\begin{aligned}[A \mid B] &= \mathrm{tr}(CA'WB) \\ &= \mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}BC^{\frac{1}{2}}) \\ &= \mathrm{tr}((W^{\frac{1}{2}}AC^{\frac{1}{2}})'W^{\frac{1}{2}}BC^{\frac{1}{2}}) \\ &= \mathrm{tr}(\tilde{A}'\tilde{B})\end{aligned}$$

où $\forall (n,p)\ \tilde{Z} = W^{\frac{1}{2}}ZC^{\frac{1}{2}}$

b) Pour montrer que $[A \mid B] = \mathrm{tr}(\tilde{A}'\tilde{B})$ est un produit scalaire, nous devons démontrer les propriétés suivantes :

(i) Symétrie : $\forall\ A, B \in M_{n,p}(\mathbb{R})$

$$[A \mid B] = [B \mid A]$$

(ii) Linéarité : $\forall\ A, B_1, B_2 \in M_{n,p}(\mathbb{R})$ et $\alpha, \beta \in \mathbb{R}$

$$[A \mid (\alpha B_1 + \beta B_2)] = \alpha[A \mid B_1] + \beta[A \mid B_2]$$

(iii) Positivité définie :$\forall\ A \in M_{n,p}(\mathbb{R})$

$$[A \mid A] \geq 0 \quad \text{et} \quad [A \mid A] = 0 \iff A = 0.$$

Vérifions ces propriétés :

(i) Soit $A, B \in M_{n,p}(\mathbb{R})$

$$\begin{aligned}[A \mid B] &= \mathrm{tr}(\tilde{A}'\tilde{B}) \\ &= \mathrm{tr}((W^{\frac{1}{2}}AC^{\frac{1}{2}})'W^{\frac{1}{2}}BC^{\frac{1}{2}}) \\ &= \mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}BC^{\frac{1}{2}})\end{aligned}$$

Or, par la propriété de la trace, $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ et l'invariance par transposition des matrices de poids $W' = W$ et $C' = C$, on a alors:

$$\begin{aligned}\mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}BC^{\frac{1}{2}}) &= \mathrm{tr}(C^{\frac{1}{2}}B'W^{\frac{1}{2}}W^{\frac{1}{2}}AC^{\frac{1}{2}}) \\ &= [B \mid A]\end{aligned}$$

(ii) Soit $A, B_1, B_2 \in M_{n,p}(\mathbb{R})$ et $\alpha, \beta \in \mathbb{R}$

$$\begin{aligned}[A \mid (\alpha B_1 + \beta B_2)] &= \mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}(\alpha B_1 + \beta B_2)C^{\frac{1}{2}}) \\ &= \alpha\mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}B_1C^{\frac{1}{2}}) + \beta\mathrm{tr}(C^{\frac{1}{2}}A'W^{\frac{1}{2}}W^{\frac{1}{2}}B_2C^{\frac{1}{2}}) \\ &= \alpha\mathrm{tr}(\tilde{A}'\tilde{B_1}) + \beta\mathrm{tr}(\tilde{A}'\tilde{B_2}) \\ &= \alpha[A \mid B_1] + \beta[A \mid B_2]\end{aligned}$$

La linéarité de la trace assure que cette propriété est respectée.

(iii) Soit $A \in M_{n,p}(\mathbb{R})$, on note $(\tilde{a}_{ij})$ le terme générique de la matrice $\tilde{A}$. On pose $\Delta = \tilde{A}'\tilde{A}$ et on note (δ_{ij}) le terme générique de la matrice Δ.

$$\begin{aligned}
[A \mid A] &= \mathrm{tr}(C^{\frac{1}{2}} A' W^{\frac{1}{2}} W^{\frac{1}{2}} A C^{\frac{1}{2}}) \\
&= tr(\tilde{A}'\tilde{A}) \\
&= tr(\Delta) \\
&= \sum_{i=1}^{p} \delta_{ii}
\end{aligned}$$

Or, $\delta_{ij} = \sum_{k=1}^{p} \tilde{a}_{ki}\tilde{a}_{kj}$ donc:

$$\begin{aligned}
tr(\tilde{A}'\tilde{A}) &= tr(\Delta) \\
&= \sum_{i=1}^{p}\sum_{k=1}^{n} \tilde{a}_{ki}\tilde{a}_{ki} \\
&= \sum_{i=1}^{p}\sum_{k=1}^{n} \tilde{a}_{ki}^2 \geq 0
\end{aligned}$$

On suppose que $[A \mid A] = 0$ montrons que $A = 0_{\mathbb{R}^{n\times p}}$:

$$\begin{aligned}
& [A \mid A] = 0 \\
\iff\ & tr(\tilde{A}'\tilde{A}) = 0 \\
\iff\ & tr(\Delta) = 0 \\
\iff\ & \sum_{i=1}^{p} \delta_{ii} = 0 \\
\iff\ & \sum_{i=1}^{p}\sum_{k=1}^{n} \tilde{a}_{ki}^2 = 0
\end{aligned}$$

Cela implique que

$$\forall (i,k) \in [|1,p|] \times [|1,n|], \quad \tilde{a}_{ki}^2 = 0$$

donc

$$\forall (i,k) \in [|1,p|] \times [|1,n|], \quad \tilde{a}_{ki} = 0$$

donc

$$\tilde{A} = W^{\frac{1}{2}} A C^{\frac{1}{2}} = 0_{\mathbb{R}^{n\times p}}$$

On en déduit que $A = 0_{\mathbb{R}^{n\times p}}$ car W et C sont régulières donc inversibles.

Par conséquent, la quantité $[A \mid B] = \mathrm{tr}(\tilde{A}'\tilde{B})$ est un produit scalaire, car elle satisfait toutes les propriétés requises.

c) Écrivons le produit scalaire précédent sous forme de double somme. Soit $A, B \in M_{n,p}(\mathbb{R})$ on note $(\tilde{a}_{ij})$ respectivement $(\tilde{b}_{ij})$ le terme générique de la matrice $\tilde{A}$ respectivement $\tilde{B}$. On pose $\Omega = \tilde{A}'\tilde{B}$ et on note (ω_{ij}) le terme générique de la matrice de Ω. Ainsi:

$$\begin{aligned}[A \mid B] &= tr(\tilde{A}'\tilde{B}) \\ &= tr(\Omega) \\ &= \sum_{i=1}^{p} \omega_{ii} \\ &= \sum_{i=1}^{p}\sum_{k=1}^{n} \tilde{a}_{ki}\tilde{b}_{ki}\end{aligned}$$

Or, $\forall A, \tilde{A}$ a pour élément générique $\tilde{a}_{ki} = \sqrt{w_k}\sqrt{c_i}a_{ki}$ donc :

$$[A \mid B] = \sum_{k=1}^{n}\sum_{i=1}^{p} w_k c_i a_k i b_{ki}$$

d) La fonction "prd_scalaire" est définie en Annexe. Toutefois, pour mieux comprendre son utilisation, nous présentons ci-dessous un exemple concret de son application. Le programme du précédent produit scalaire est affiché, accompagné de la norme associée à ce produit scalaire.

La fonction calculant le produit scalaire de Frobénius

onsidérons à présent un exemple d'application du produit scalaire de Frobénius ci-dessous: \

Nous commençons par définir les matrices X_1 et X_2 avec les valeurs suivantes choisies. Les matrices X_1 et X_2 sont les mêmes à l'exception d'une valeur (qui n'est pas loin des autres), et les matrices X_3 et X_4 sont au contraire bien différentes avec une valeur de la deuxième matrice qui se rapproche légèrement de la première :

$$X_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$X_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \; X_3 = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ -0.5 & -1 \end{pmatrix}$$

Maintenant que les matrices X_1 et X_2 sont initialisées, nous pouvons appliquer la fonction prd_scalaire pour calculer le produit scalaire de Frobenius entre elles.

```
# Initialisation de A et B
a <- c(0,1,1,0,-1,0,1,2,0,-1,0,-1)
b <- c(-1,-1,1,0,1,0,0,1,2,1,0,-1)
X_1  = matrix(a, nrow =4)
X_2 = matrix(b, nrow =4)

# On applique la fonction prd_scalaire
resultat = prd_scalaire(X_1,X_2)
```

Ainsi nous pouvons appliquer le produit scalaire de Frobenius entre les matrices X_1 et X_2, et X_1 et X_3. On obtient les résultats suivants : $[X_1 \mid X_2] = 0.83333$ et $[X_1 \mid X_3] = -0.91667$. Comme un produit scalaire élevé indique que les matrices son orientées de de manière similaire dans l'espace multidimensionnel des

éléments, plus le résultats est élevé, plus les matrices sont similaires en termes de direction et de magnitude des éléments correspondants. Cette définition est bien illustrée ici.

La fonction pour calculer la norme associée à ce produit est définie en Annexe.

```
# On utilise les matrices précédente afin de calculer leurs normes
rn1 = norme(X_1)
rn2 = norme(X_2)
```

Après avoir calculé le produit scalaire de Frobenius entre les matrices X_1,X_2 et X_3, nous allons maintenant calculer la norme associée à ce produit scalaire ($[|\cdot|] = \sqrt{[\cdot \mid \cdot]}$) pour chaque matrice, X_1 et X_2. La fonction **norme** pour calculer la norme associée à ce produit est définie en Annexe. \ Nous obtenons après calcul les résultats suivants: $[|X_2|] = 0.9128709$ et $[|X_3|] = 0.9354143$. De toute évidence les norme de X_1 est égale à 1.\

1.1.2 Coefficient RV

On définit le coefficient de RV d'Escoufier entre deux matrices X_t et X_s de taille (n, p) par :

$$R(X_t, X_s) = \frac{[X_t \mid X_s]}{[|X_t|]\,[|X_s|]}$$

a) Géométriquement, le coefficient de RV représente le cosinus entre les matrices X_t et X_s.

b) Les fonctions pour calculer le coefficient de RV et fournir la matrice des coefficients de RV en (n, p) tableaux $X_{(n,p)}$ seront présentées en Annexe. En revanche, ici, nous expliciterons un exemple d'application de celles-ci.

On définit le coefficient de RV d'Escoufier entre deux matrices X_t et X_s de taille (n, p) par :

$$R(X_t, X_s) = \frac{[X_t \mid X_s]}{[|X_t|]\,[|X_s|]}$$

Le coefficient de RV représente le cosinus entre les matrices X_t et X_s. Cette interprétation souligne la similarité géométrique entre les ensembles de variables.\
Reprenons nos trois matrices précédentes X_1, X_2 et X_3 de dimension 3×2 pour illustrer ce coefficient : \

$$X_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad X_3 = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ -0.5 & -1 \end{pmatrix}$$

$$[X_1|X_2] = \mathrm{tr}(CX_1'W_BX_2)$$

$$\frac{1}{6}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} = \frac{1}{6}\begin{pmatrix} 3 & 2 \\ 3 & 2 \end{pmatrix}$$

donc $[X_1|X_2] = \frac{5}{6}$. Or, $\|X_1\| = 1$ et $\|X_2\| = \sqrt{\frac{5}{6}}$, donc $R(X_1, X_2) = \frac{[X_1|X_2]}{\|X_1\|\|X_2\|} = \frac{\frac{5}{6}}{\sqrt{\frac{5}{6}}} = \sqrt{\frac{5}{6}} \approx 0.9128709$ \

De manière analogue, on obtient les résultats suivants: \ - $[X_1|X_3] = \frac{-5.5}{6}$ \ - $[X_2|X_3] = \frac{-3}{4}$\ - $R(X_1, X_3) \approx -0.97996$ \ - $R(X_2, X_3) \approx -0.87831$

En notant R_{ij} le terme général de la matrice R des coefficients RV, on a: \

$$R_{ij} = \frac{[X_i \mid X_j]}{\|X_i\|\,\|X_j\|}$$

On en déduit alors que:

$$R = \begin{pmatrix} 1 & 0.91287 & -0.97996 \\ 0.91287 & 1 & -0.87831 \\ -0.97996 & -0.87831 & 1 \end{pmatrix}$$

Un coeffecient proche de 1 signifie une forte similarité entre deux matrices. A contrario un coefficient proche de -1 indique une faible similarité. Chaque ligne et chaque colonne représente une matrice, reflétée par rapport à la diagonale en raison de la symétrie de la matrice R. Par conséquent, le coefficient RV entre la deuxième et la troisième matrice peut être trouvé à la fois sur la deuxième ligne, troisième colonne, et sur la troisième ligne, deuxième colonne.\

```
X1 = as.matrix(read.csv("2021_filtre.csv", row.names = 1, header = T))
X2 = as.matrix(read.csv("2022_filtre.csv", row.names = 1, header = T))
X3 = as.matrix(read.csv("2023_filtre.csv", row.names = 1, header = T))

filtre <- list(X1 = X1, X2 = X2, X3 = X3)

resultats_n = matcoef_RV(filtre)
```

Jusqu'ici les calculs étaient faisables à la main. Cependant il arrive plus souvent dans la vraie vie de devoir utiliser ces outils à plus grande dimension. Pour cela nous avons coder une fonction en R donnant directement la matrice des coefficients RV à partir des fonctions codées précédemment pour le produit scalaire et la somme. Les fonctions **coef_RV** et **matcoef_RV** pour calculer respectivement le coefficient RV et fournir la matrice des coefficients de RV en (n, p) tableaux $X_{(n,p)}$ seront présentées en Annexe. Nous utiliserons les données citées en introduction, à savoir les données de pollution en Occitanie sur 3 années, pour interpréter de vrais résultats de cette fonction. \
Le résultat obtenu est la matrice suivante arrondie à 10^{-5} :

$$\begin{pmatrix} 1.0000 & 0.99928 & 0.99892 \\ 0.99928 & 1.0000 & 0.99915 \\ 0.99892 & 0.99915 & 1.0000 \end{pmatrix}$$

Cette matrice donne les coefficients RV entre les différents tableaux. On a trois tableaux donc on obtient une matrice de dimension 3×3. Ici les tableaux 2 et 3 sont les plus corrélés. Les tableaux 1 et 3 sont les moins corrélés. Il faut remettre les données dans leur contexte, s'agissant d'une évolution de la pollution sur trois années, avec chaque matrice $((i, j), t)$ représentant le taux de pollution par station et par polluant pour chaque année, il est normal de trouver une forte corrélation entre les matrices des années proches et une corrélation légèrement moins forte pour les matrices des années plus éloignées. On en déduit que la pollution en Occitanie n'a pas beaucoup évoluée sur ces trois années. \ Toutefois, si l'on avait tournée le tableau d'une autre façon (matrices $((i, t), j)$ des stations par années pour chaque polluant, ou matrices $((j, t), i)$ des polluants par années pour chaque station) en le refaçonnant (3 manières de touner un tableau à trois entrées) les résultats auraient pu être différents. \

1.2 Programme de STATIS 1

1.2.1 Programme

a) Dans le contexte de l'ACP des tableaux juxtaposés, les "variables" sont les différents tableaux $\mathbf{X_t}$, et les "individus" sont les observations à chaque période de temps. L'expression $\left[\left|\sum_{t=1}^{T} \frac{u_t}{[|X_t|]} X_t\right|\right]^2$ représente l'inertie le long d'un axe $< u >$, où u est un vecteur I-unitaire: $||u||^2 = 1$. On cherche à projeter I-orthogonalement le nuage direct sur un espace $E_k =< u_1, ..., u_k >$ de dimension $k < p$. cela revient à maximiser l'inertie du nuage direct projeté sur $< u >$.

$$\max_{||u||^2=1} \left[\left| \sum_{t=1}^{T} \frac{X_t}{[|X_t|]} u_t \right| \right]^2 = \max_{||u||^2=1} \sum_{t=1}^{T} \sum_{\tau=1}^{T} u_\tau \frac{[X_\tau \mid X_t]}{[|X_t|]\,[|X_\tau|]} u_t$$
$$= \max_{||u||^2=1} u'Ru$$

où:

1. X_t: $n \times p$ (la matrice X_t a des dimensions $n \times p$).
2. $\frac{u_t}{[|X_t|]} X_t$: $n \times p$ (chaque colonne de X_t est pondérée par $\frac{u_t}{[|X_t|]}$).
3. $\sum_{t=1}^{T} \frac{u_t}{[|X_t|]} X_t$: $n \times p$ (somme des termes précédents sur t).
4. $\left(\sum_{t=1}^{T} \frac{u_t}{[|X_t|]} X_t \right)'$: $p \times n$ (transposée de la matrice résultante).
5. $\left[\left| \sum_{t=1}^{T} \frac{u_t}{[|X_t|]} X_t \right| \right]^2$: 1×1 (la norme euclidienne au carré est un scalaire).
6. R est la matrice des coefficients RV

b) Résolvons le programme ci-dessus, le langrangien associé s'écrit:

$$L(u, \lambda) = u'Ru - \lambda(\|u\|^2 - 1)$$

On a alors :

$$\begin{cases} \frac{\partial L}{\partial u}(u, \lambda) = 2Ru - 2\lambda u \\ \frac{\partial L}{\partial \lambda}(u, \lambda) = \|u\|^2 - 1 \end{cases}$$

Les conditions de premier ordre donnent:

$$\iff (S) \begin{cases} Ru = \lambda u & (*) \\ \|u\|^2 = 1 & (**) \end{cases}$$

On a par suite: $u' \quad (*)$ et $(**) \Rightarrow u'Ru = \lambda$.

D'après, (S) on en déduit que les vecteurs u solution de premier ordre sont les vecteurs propres de la matrice $R = ((R(X_t, X_\tau)))_{t,\tau}$ des coefficients RV d'Escoufier entre T tableaux $X_t(n, p)$. Or, pour tout vecteur propre u de R (tel que $\|u\| = 1$) de valeur propre λ, on a $u'Ru = \lambda$.

La valeur maximale de $u'Ru$ est obtenue pour les vecteurs propres associés à la plus grande valeur propre de R.

c) Nous allons écrire le programme R fournissant les vecteurs u solutions des équations du premier ordre.

Considérons un exemple d'application de la fonction vecval_prop:

```
# on applique vecval_prop aux données spolution en Occitanie sur 3 années
vecval = vecval_prop(resultats_n)

val_prop <- vecval$val_prop
vec_prop <- vecval$vec_prop
```

Aprés exécution de la fonction vecval_prop les résultats suivants:

$$U = \begin{pmatrix} -0.5773 & -0.6214 & 0.5296 \\ -0.5773 & -0.1479 & -0.8029 \\ -0.5773 & 0.7693 & 0.2734 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 2.9982 \\ 0.0010 \\ 0.0006 \end{pmatrix}$$

où, U est la matrice des vecteurs propres et λ les valeurs propres associées arrondie à 10^{-4}.

Montrons à présent que les vecteurs u obtenus forment une base I-orthonormée.

Reprenons la matrice R des coefficients RV précédente. Soient u_i et u_j des vecteurs propres de R associés aux valeurs propres λ_i et λ_j distinctes. Comme u_i et u_j sont des solutions du problème suivant :

$$\max_{||u||^2=1} u'Ru$$

on a alors $\|u_i\|^2 = 1$ et $\|u_j\|^2 = 1$. Il suffit de montrer que u_i et u_j sont orthogonaux et ainsi de conclure que les vecteurs u forment une base I-orthonormée. En d'autres termes, nous allons montrer que $u_i'u_j = 0$.

Sachant que u_i et u_j sont des vecteurs propres de R associés aux valeurs propres λ_i et λ_j, nous avons :

$$\begin{cases} Ru_i = \lambda_i u_i \\ Ru_j = \lambda_j u_j \end{cases}$$

Par la suite, on a:

$$\begin{aligned} u_i'Ru_j &= \lambda_j u_i'u_j \\ \iff u_i'R'u_j &= \lambda_j u_i'u_j \quad \text{(car } R \text{ est une matrice symétrique)} \\ \iff (Ru_i)'u_j &= \lambda_j u_i'u_j \\ \iff (\lambda_i u_i)'u_j &= \lambda_j u_i'u_j \\ \iff \lambda_i u_i'u_j &= \lambda_j u_i'u_j \\ \iff (\lambda_i - \lambda_j)u_i'u_j &= 0 \\ \iff u_i'u_j &= 0 \quad (\text{car } \lambda_i \neq \lambda_j) \end{aligned}$$

On en déduit donc que les vecteurs u forment une base I-orthonormée.

1.2.2 Équivalence avec l'ACP d'un tableau juxtaposé "dépliant" le tableau cubique

a) Il suffit de décomposer X_t comme une juxtaposition de colonnes:

$$X_t = \left[x_t^1 \dots x_t^p\right]_{\text{Profils Colonnes}} = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,n} \end{bmatrix}_{\text{Profils Lignes}}$$

et de le réécrire sous forme "verticalisée":

$$y^t = \begin{pmatrix} x_t^1 \\ \vdots \\ x_t^p \end{pmatrix} \in \mathbb{R}^{np} \quad \text{où} \quad \forall j \in [|1,p|], x_t^j = \begin{pmatrix} x_{t,1}^j \\ \vdots \\ x_{t,n}^j \end{pmatrix}$$

Avant de démontrer l'égalité des produits scalaires, définissons d'abord le Produit Kronecker:

Soient A une matrice de taille $m \times n$ et B une matrice de taille $p \times q$. Leur produit tensoriel est la matrice $A \otimes B$ de taille mp par nq, définie par blocs successifs de taille $p \times q$, le bloc d'indice i, j valant $a_{ij}B$.

En d'autres termes,

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

Ou encore, en détaillant les coefficients,

$$A \otimes B = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{pmatrix}$$

Maintenant, montrons que :

$$[X^s|X^t] = \langle y^s|y^t\rangle_L$$

où $L = C \otimes W(ProduitdeKronecker)$.

Sachant que, $W = \frac{1}{n}I_n$ et $C = \frac{1}{p}I_p$ on en déduit que $L = \frac{1}{np}I_{np}$.

D'une part, soit $(s,t) \in [|1,T|]$:

$$\begin{aligned} [X_s \mid X_t] &= tr(\tilde{X_s}'\tilde{X_t}) \\ &= \sum_{k=1}^{n}\sum_{i=1}^{p} w_k c_i x_{s,k}^i x_{t,k}^i \\ &= \sum_{k=1}^{n}\sum_{i=1}^{p} \frac{1}{np} x_{s,k}^i x_{t,k}^i \end{aligned}$$

D'autre part:

$$\begin{aligned}\langle y^s|y^t\rangle_L &= y^{s\prime}Ly^t \\ &= \sum_{i=1}^{p}\frac{1}{np}x_s^{i\prime}x_t^i \\ &= \sum_{i=1}^{p}\frac{1}{np}\sum_{k=1}^{n}x_{s,k}^i x_{t,k}^i \\ &= \sum_{k=1}^{n}\sum_{i=1}^{p}\frac{1}{np}x_{s,k}^i x_{t,k}^i\end{aligned}$$

Donc $[X^s|X^t] = \langle y^s|y^t\rangle_L$.

Il en découle que X^t est "verticalisé" en $y^{t*} = y^t$ L-normé.

Par suite, on note $Y = [y^1, \ldots, y^T]$. Un individu correspond à une ligne de ce tableau, correspondant donc à l'indice (j, i) dans le cube initial. Un individu est un vecteur de taille T, dont les coordonnées sont les T valeurs de la case (i, j) du tableau X_t sur la période. C'est la trajectoire de cette case. Son poids est $L_(j, i) = w_i c_j = \frac{1}{np}$.

Par exemple, pour un tableau cubique ventilant la valeur ajoutée d'une économie par région (i), secteur (j) et année (t), l'individu du tableau dépliant ainsi le cube serait la trajectoire dans le temps de la V.A. réalisée par le secteur j dans la région i.

Et on a :

$$\left[\left|\sum_{t=1}^{T} u_t \frac{X_t}{[|X_t|]}\right|\right]^2 = \|Y^*u\|_W^2 = \|F\|_W^2$$

, où $Y^* = [y^{1*}, \ldots, y^{T*}]$ et $F = Y^*u$ est la composante associée à u.

On remarque que:

Métrique usuelle : Si $M = I_T$ alors $\langle y^{s*}, y^{t*}\rangle = \langle y^s, y^t\rangle_L$.

Métrique réduite : Diviser les variables y^t par $\sigma_t = \sqrt{\|y^t\|}$ est équivalent à prendre $m_t = \frac{1}{{\sigma_t}^2}$. On a $M_{\frac{1}{\sigma^2}} = M_{\frac{1}{\sigma}} M_{\frac{1}{\sigma}}$ et donc $\langle y^s m_{\frac{1}{\sigma_s}}, y^t m_{\frac{1}{\sigma_t}}\rangle_L = m_{\frac{1}{\sigma_s}} y^{s\prime} L y^t m_{\frac{1}{\sigma}t}$.

Travailler avec la métrique $M_{\frac{1}{\sigma}^2}$ revient à utiliser la métrique L sur des variables $M_{\frac{1}{\sigma}^2}$-normées.

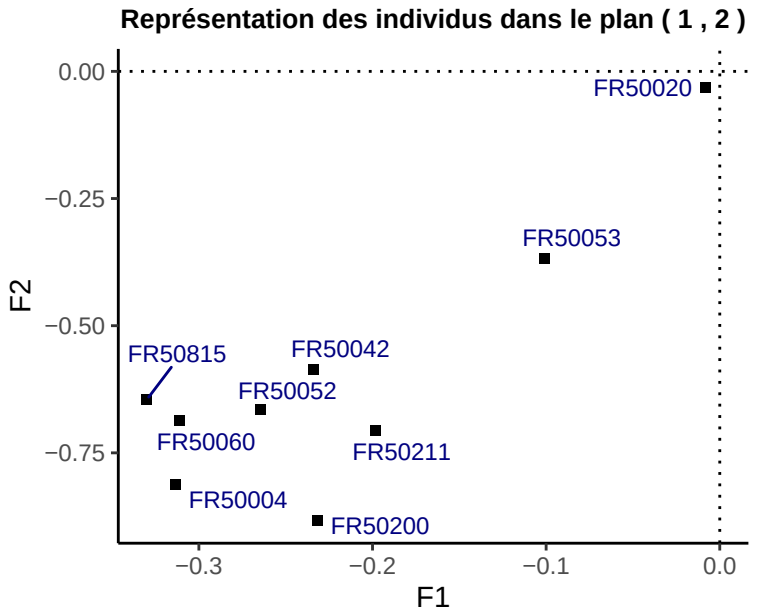
b) Pour déduire que les composantes principales $F^1, \ldots, F^k, \ldots$ sont orthogonales au sens du produit scalaire convenable, nous devons considérer la façon dont ces composantes sont construites à partir des vecteurs propres obtenus.

$$\begin{aligned}\langle F_k|F_l\rangle_L &= F_k' L F_l \\ &= u_k' Y^{*\prime} L Y^* u_l \\ &= u_k' M_{\frac{1}{\sigma}} Y' L Y M_{\frac{1}{\sigma}} u_l \\ &= 1 \quad \text{si} \quad k = l \\ &= 0 \quad \text{si} \quad k \neq l\end{aligned}$$

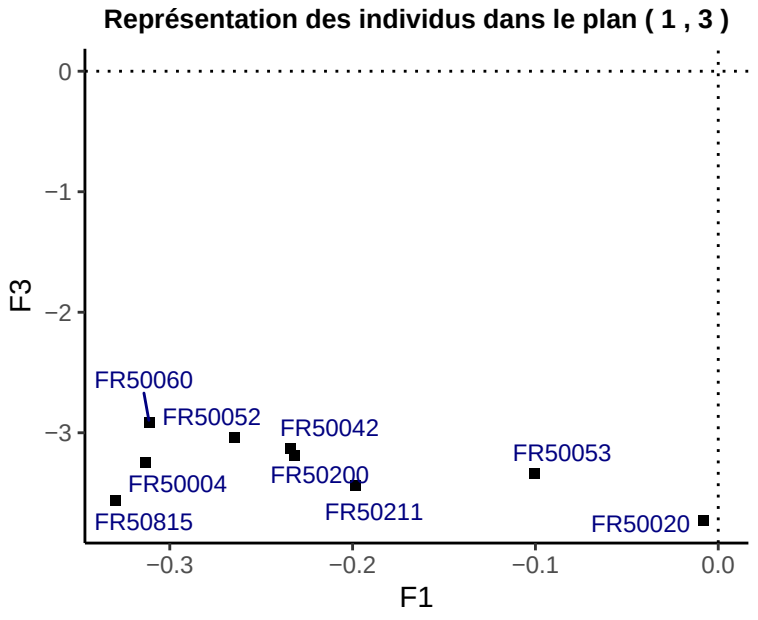
c) La fonction calculant les composantes, la représentation des individus et celle des tableaux setrouvent en annexe. Nous verrons ci-dessous un exemple d'application de ces fonctions sur les données simulated.

La représentation des individus selon la première composante principale F_1:

```
F1 <- cmp(filtre,1)$F_k
representation_graph_ind(F1, 1, 2, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```

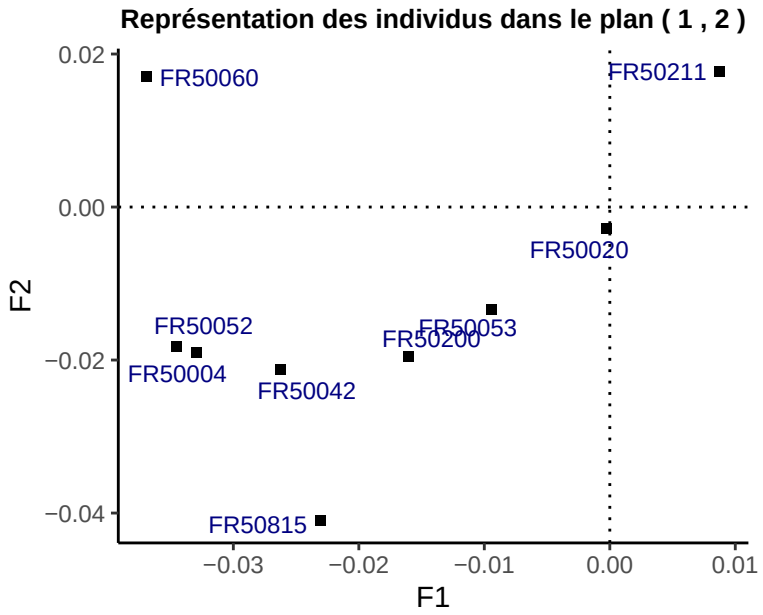



```
representation_graph_ind(F1, 1, 3, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```

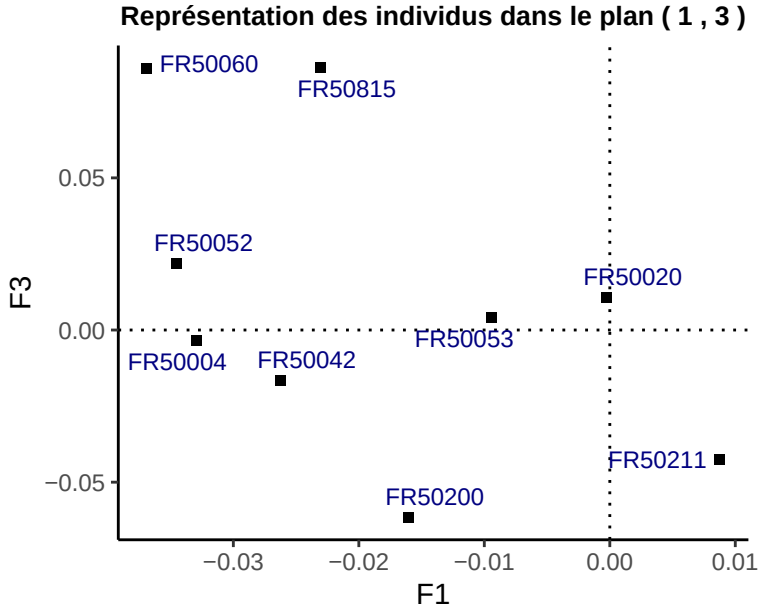


La représentation des individus selon la deuxième composante principale F_2:

```
F2 <- cmp(filtre,2)$F_k
representation_graph_ind(F2, 1, 2, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```

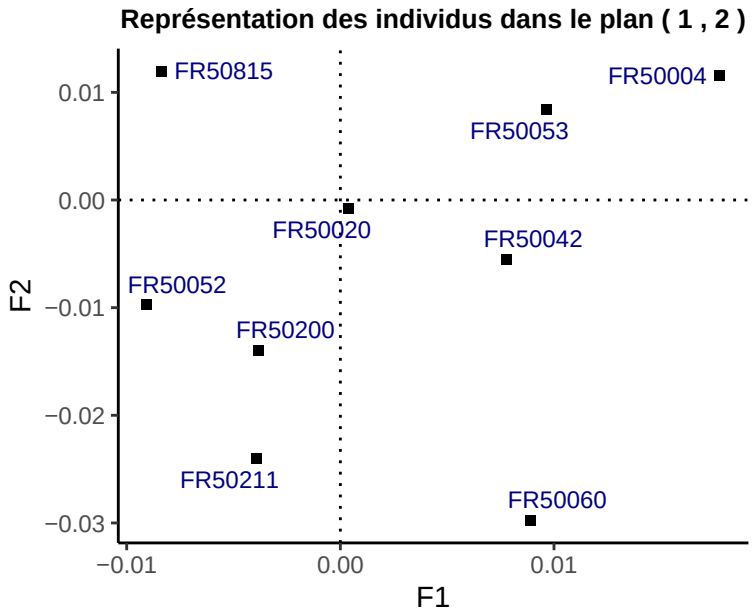



```
representation_graph_ind(F2, 1, 3, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```

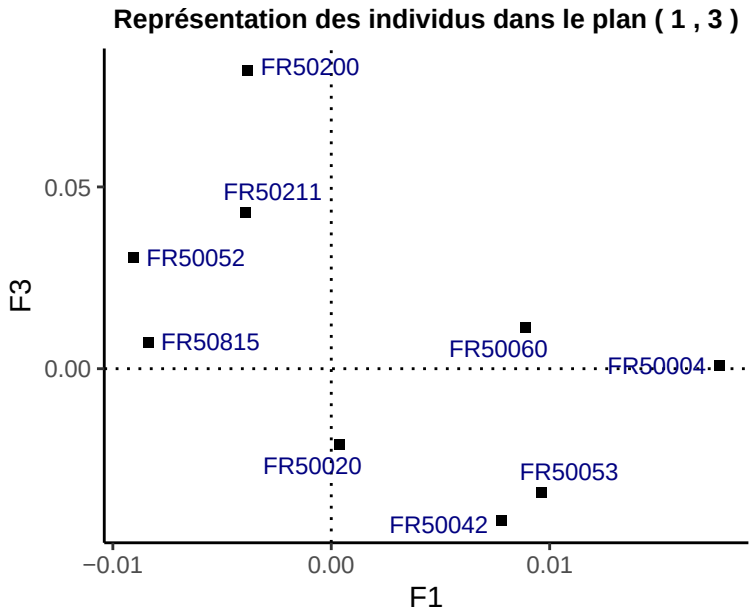


La représentation des individus selon la troisième composante principale F_3:

```
F3 <- cmp(filtre,3)$F_k
representation_graph_ind(F3, 1, 2, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```

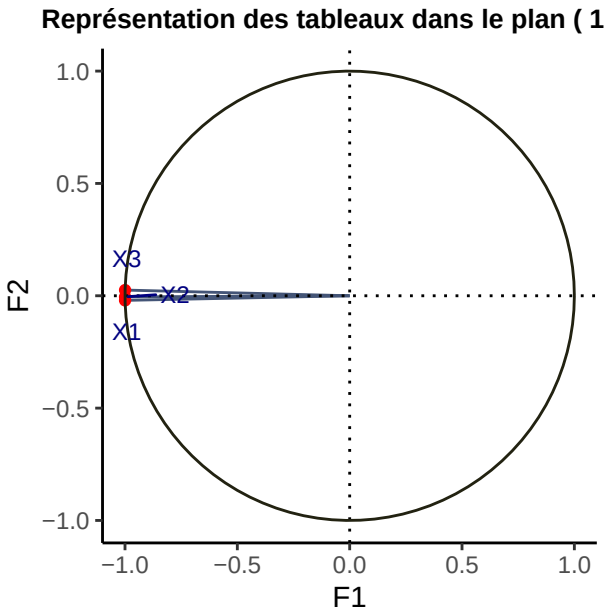



```
representation_graph_ind(F3, 1, 3, aff_noms = TRUE,cex_titre = 0.80, repel = T)
```



d) Cosinus entre les tableaux et les composantes principales : La fonction calculant le cosinus entre le tableau X_t et la composante F_k se trouve en annexe.

```
# Exemple d'utilisation avec les tableaux n_tableaux et les composantes principales 3 et 4
representation_graph_tab(filtre, 1, 2, cex_titre = 0.80, repel = TRUE)
```



1.2.3 Illustration d'un ACP d'un tableau juxtaposé dépliant le tableau cubique

Maintenant que nous avons tous les outils pour réaliser une ACP sur un tableau à trois entrées, il nous faut visualiser les résultats pour les interpréter. On propose l'exemple suivant réalisé à la main sur nos premières matrices X_1, X_2 et X_3: \

$$X_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad X_3 = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ -0.5 & -1 \end{pmatrix}$$

On rappel le produit scalaire de Frobénius :\

$$\begin{aligned} [X_t \mid X_s] &= \mathrm{tr}(CX_t'WX_s) \\ &= \mathrm{tr}(\tilde{X_t}'\tilde{X_s}) \end{aligned}$$

avec $W := \frac{1}{n} I_n$ la matrice de poids des individus et $C := \frac{1}{p} I_p$ la matrice de poids des variables. \

On avait trouvé la matrice des coefficients RV d'Escoufier que l'on notera Γ.\

Ci-dessous la formule permettant de calculer le coefficient RV entre deux matrices X_t et X_s de taille (n, p) (ici $n = 3$ et $p = 2$) :\

$$R(X_t, X_s) = \frac{[X_t \mid X_s]}{[|X_t|]\,[|X_s|]}$$

Ainsi,

$$\Gamma = \begin{pmatrix} 1 & 0.91287 & -0.97996 \\ 0.91287 & 1 & -0.87831 \\ -0.97996 & -0.87831 & 1 \end{pmatrix}$$

Maintenant, considérons le dépliage des matrices juxtaposées :

$$\mathrm{X} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -0.5 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix}$$

où la première colonne est la matrice X_1 dépliée, la deuxième colonne est la matrice X_2 dépliée et la troisième colonne est la matrice X_3 dépliée.\

On pose M la matrice diagonale t, t dont le coefficient $m_{ii} = \frac{1}{||X_i||_L}$ avec:

- t le nombre de tableau (ici $t = 3$)
- $L := W \otimes C = \frac{1}{np} I_{np}$
- X_i la ième variable
- $\|X_i\|_L = \sqrt{(X_i' L X_i)}$ (la norme associée au produit scalaire d'après la partie 1)

Ici M sera donc égale à:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sqrt{\frac{6}{5}} & 0 \\ 0 & 0 & \sqrt{\frac{8}{7}} \end{pmatrix}$$

On a alors:

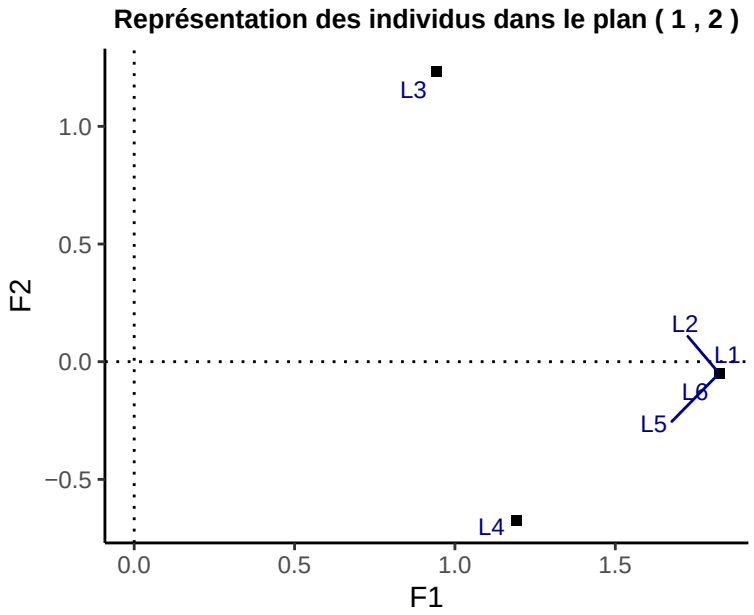
$$\Delta := M X^t W X M = \Gamma$$

Nous avons codé des fonctions qui font les calculs matriciels puis nous donnent la matrice obtenue delta qui est égale à la matrice des coefficients RV calculée précédemment. Cette fonction se trouve en Annexe.\

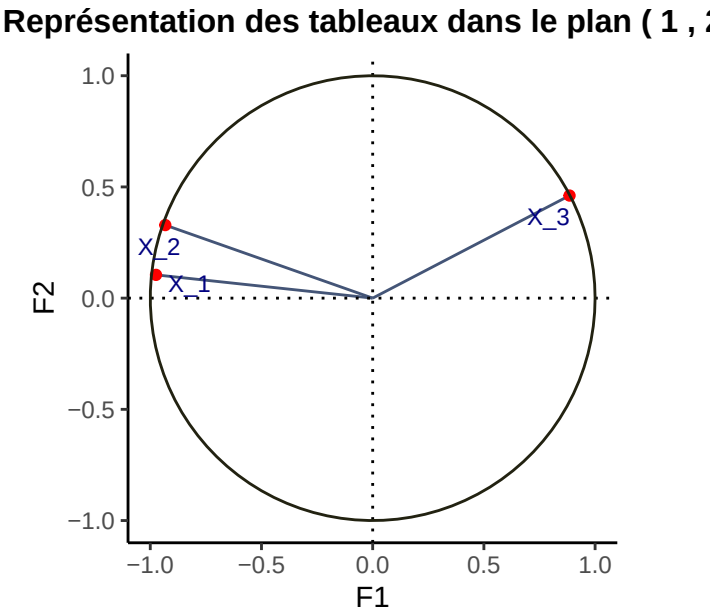
```
# Données
# Initialisation de X_1 et X_2
a <- c(1,1,1,1,1,1)
b <- c(1,1,1,0,1,1)
c <- c(-1,-1,0.5,-1,-1,-1)

X_1  = matrix(a, nrow =3)
X_2 = matrix(b, nrow =3)
X_3 = matrix(c, nrow =3)


L <- list(X_1 = X_1, X_2 = X_2, X_3 = X_3)
```

Finalement, on en déduit qu'il y a équivalence entre les matrices juxtaposées X_1 , X_2 et X_3 avec la matrice X dont les variables sont les matrices X_1 , X_2 et X_3 dépliées.

```
Yt <- dep_mat(L)
geo_ind(Yt, 1, 2, aff_noms = TRUE, cex_titre = 0.80, repel = TRUE)
```




```
representation_graph_tab(L, 1, 2, repel = TRUE)
```



1.2.4 Quelles ACP d'autres "dépliages" du tableau cubique ?

a) Disons que l'on dispose d'un tableau cubique $X(i, j, t)$. On peut le déplier de trois manières différentes : $Y((i, j), t)$ de façon à avoir les individus et caractéristiques en fonction du temps. Pour avoir les caractéristiques dans le temps pour chaque individu $Y((j, t), i)$. Ou pour avoir les individus dans le temps pour chaque caractéristique $Y((i, t), j)$.
Chacun de ces dépliages donnera une matrice différente et donc une ACP différente. Mais chacun reste intéressant.

b) Selon le dépliage choisi, les individus et variables ne seront pas les mêmes. En individus on peut avoir :

- Les individus par caractéristiques (ou caractéristiques par individus) décrit dans le temps,
- Les individus dans le temps (à un instant donné), décrit par les caractéristiques,
- Les caractéristiques dans le temps (à un instant donné), décrivant les individus.
  Les ACP à envisager dépendent du jeu de données que l'on souhaite étudier. Ici les individus et variables ne sont pas clairement définit car ne porte que des numéros. Ce qui est sûr c'est que les individus regrouperont les tableaux dépliés. Les éléments à projeter en supplémentaire dépendent, encore une fois, de ce que l'on souhaite étudier.

c) Un inconvénient est qu'on perde une grosse partie de l'information qui sera noyée dans le tableau déplié. Cependant, cela permet de mieux visualiser les données et de les traiter plus facilement (utilisation du produit matriciel possible). Le plus gros inconvénient reste la perte de la symétrie, par exemple lors d'un dépliage de $X(i,j,t)$ en $Y((i,j),t)$. En effet, les individus et variables ne sont plus équivalents. Cela peut poser problème si l'on souhaite comparer les individus et variables entre eux.

L'idéal serait de réaliser les trois dépliages possibles et de les comparer pour voir lequel est le plus pertinent ! Tout dépend du tableau initial.

2 Situation 2

2.1 De nouvelles matrices dans un nouvel espace

1.1.a) On souhaite exprimer la matrice P_m en fonction des matrices X_m et M_m. La formule est donnée par :

$$P_m = X_m M_m X_m'$$

où X_m est une matrice des données de dimension (n, p_m) , M_m est une matrice de pondération de dimension (p_m, p_m), et X_m' est la transposée de X_m.

b) L'espace P est l'ensemble des matrices $n \times n$. Cet espace est noté :

$$P = \mathbb{R}^{n \times n}$$

La dimension de cet espace est n^2, car une matrice $n \times n$ possède n^2 éléments indépendants.

1.2.a) Les lignes et les colonnes de P_m représentent les individus. Le poids naturel des lignes et des colonnes est donc le même, noté W de dimension (n, n).

b) Pour munir l'espace P d'un produit scalaire, on utilise la trace :

$$[X_t \mid X_s] = \mathrm{tr}(W X_t' W X_s)$$

c) Les matrices X_m peuvent avoir des dimensions différentes, et les matrices M_m peuvent varier. Cela signifie que les individus des tableaux X_m résident dans des espaces euclidiens différents selon le tableau. Leurs produits scalaires ne sont donc pas directement comparables d'un tableau à l'autre. Pour rendre les matrices P_m comparables, il est nécessaire de les normer.

d) Une fois les matrices P_m normées, elles se situent toutes sur la sphère unité de P. La proximité entre deux matrices peut être mesurée par le cosinus de l'angle entre elles, noté RV.

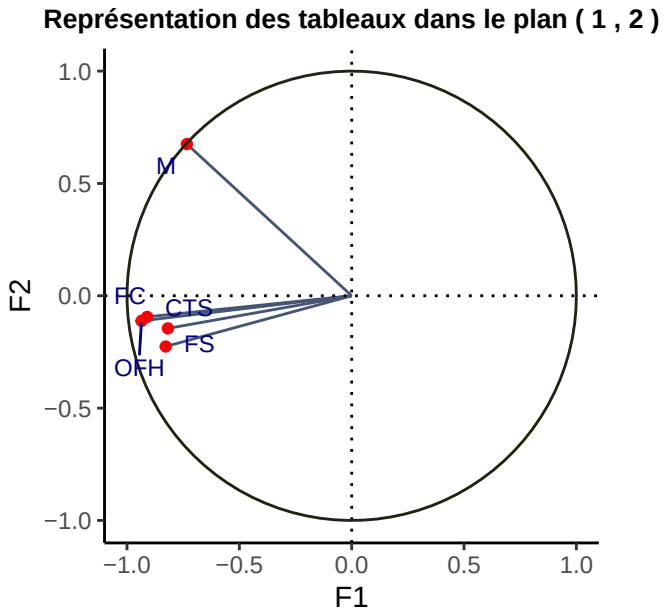
2.2 2. STATIS 2

On applique le programme de STATIS 1 à un ensemble de matrices P_m issues de tableaux thématiques décrivant les mêmes individus. On note $F_1, \ldots, F_k$ les composantes principales obtenues.

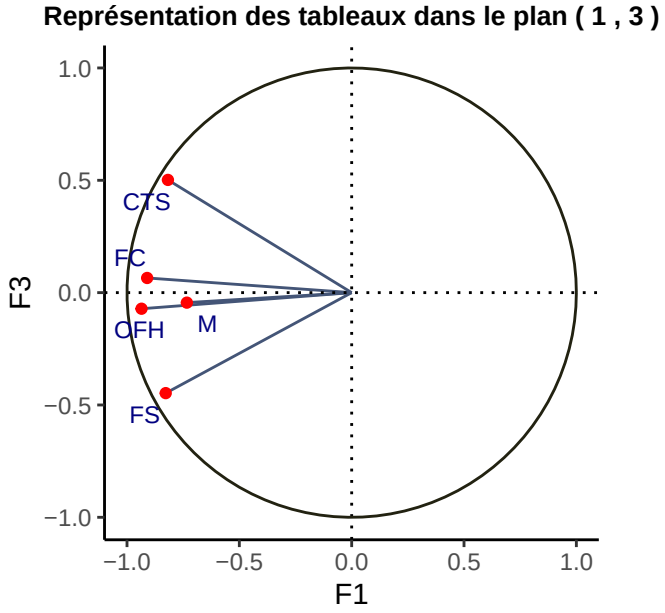
2.1.a) Pour projeter chaque P_m sur un graphe dont les directions sont un couple de composantes principales (F_k, F_l), on utilise leurs cosinus (RV) avec ces composantes.

b) On projette les matrices P_m sur le graphe pour visualiser les données thématiques partitionnées en thèmes.

```r
# Tableau à trois dimension
data(chickenk)

M = as.matrix(chickenk$Mortality)
rownames(M) = paste("L",1:351,sep = "")

FS = as.matrix(chickenk$FarmStructure)
rownames(FS) = paste("L",1:351,sep = "")

OFH = as.matrix(chickenk$OnFarmHistory)
rownames(OFH) = paste("L",1:351,sep = "")

FC = as.matrix(chickenk$FlockCharacteristics)
rownames(FC) = paste("L",1:351,sep = "")

CTS = as.matrix(chickenk$CatchingTranspSlaught)
rownames(CTS) = paste("L",1:351,sep = "")

data = list(M = M, FS = FS, OFH = OFH, FC = FC, CTS = CTS)

t <- length(data)
lst <- list()
nom <- c("M", "FS", "OFH", "FC", "CTS")
for (i in 1:t) {
  lst[[nom[i]]] <- mat_coldiff(data[[i]])
}


representation_graph_tab(lst, 1, 2, cex_titre = 0.8, repel = TRUE)
```




```
representation_graph_tab(lst, 1, 3, cex_titre = 0.8, repel = TRUE)
```



2.2.a) Les composantes principales sont des matrices symétriques car elles sont des combinaisons linéaires des matrices P_m qui sont elles-mêmes symétriques.

b) En interprétant chaque composante principale comme une matrice de produits scalaires entre individus, on peut obtenir une image du nuage des individus dans un plan de dimension réduite. Pour cela, on diagonalise chaque matrice F_k :

$$F_k = U_k \Lambda_k U_k' = (U_k \Lambda_k^{1/2})(U_k \Lambda_k^{1/2})'$$

Les coordonnées des individus sur les axes principaux sont données par $U_k \Lambda_k^{1/2}$, fournissant une représentation euclidienne hiérarchisée.

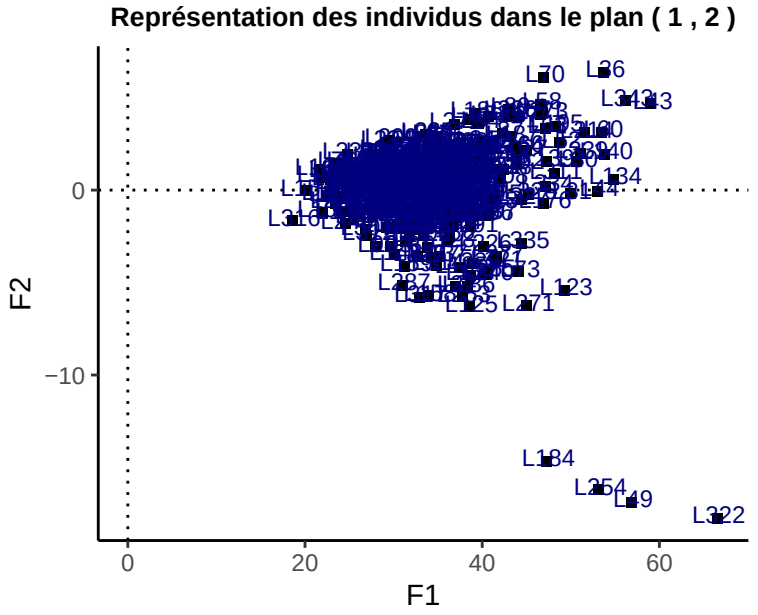
c) On applique cette méthode de représentation au tableau de données d'application pour obtenir une visualisation des individus.

Les compopsantes principales:

```
F1_theme <- cmp(lst,1)$F_k
F2_theme <- cmp(lst,2)$F_k
F3_theme <- cmp(lst,3)$F_k
```

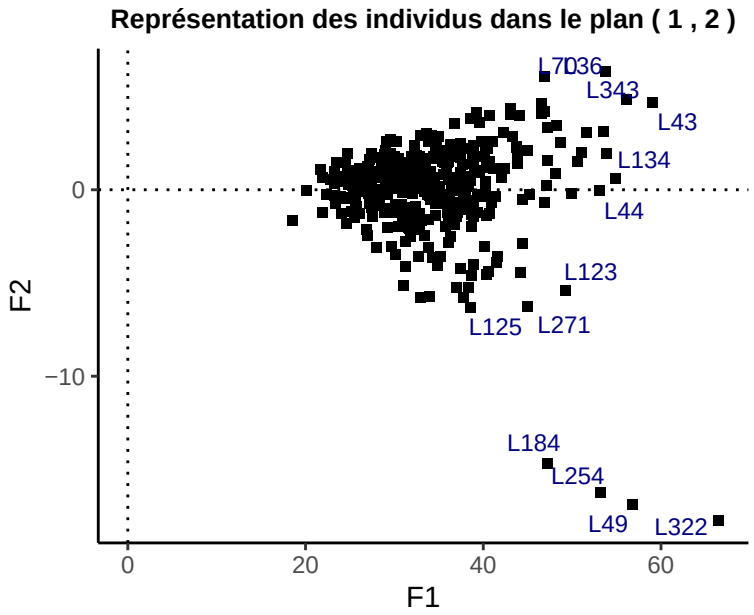
La représentation des individus selon la première composante principale:

```
geo_ind(F1_theme,1,2,aff_noms = T, cex_titre = 0.8)
```



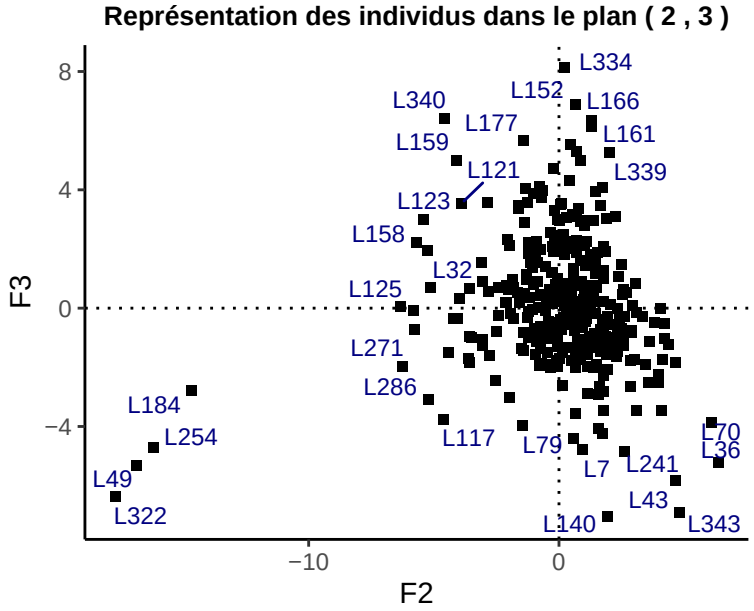

```
geo_ind(F1_theme,1,2,aff_noms = T, cex_titre = 0.8, repel = T)
```

```
## Warning: ggrepel: 338 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



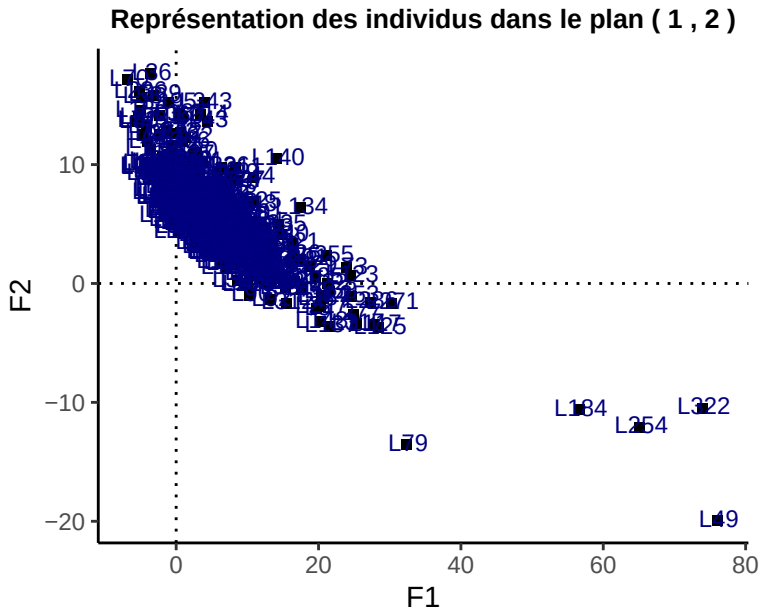

```
geo_ind(F1_theme,2,3,aff_noms = T, cex_titre = 0.8, repel = T)
```

```
## Warning: ggrepel: 323 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



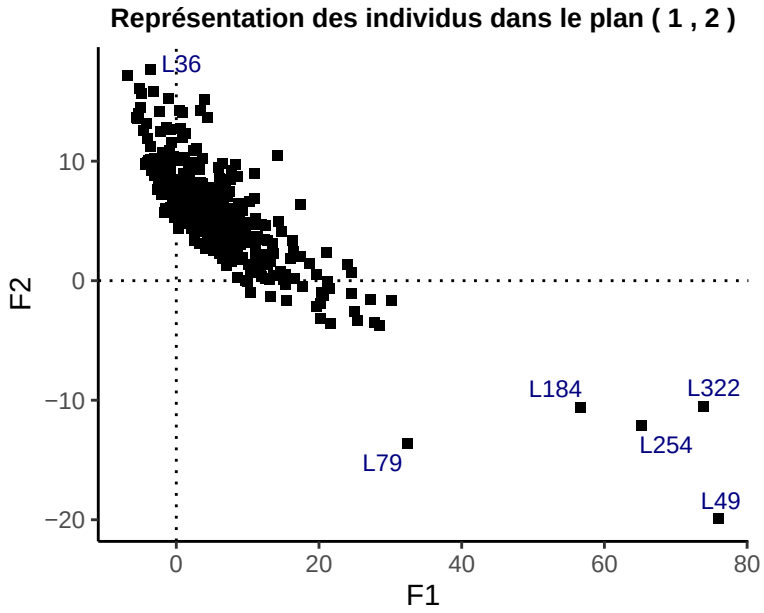
La représentation des individus selon la deuxième composante principale:

```
geo_ind(F2_theme,1,2,aff_noms = T, cex_titre = 0.8)
```



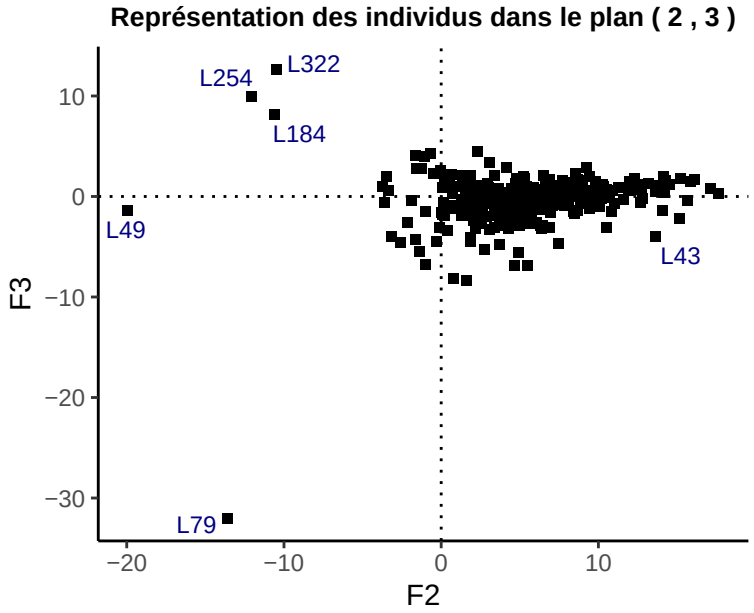

```
geo_ind(F2_theme,1,2,aff_noms = T, cex_titre = 0.8, repel = T)
```

```
## Warning: ggrepel: 345 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
geo_ind(F2_theme,2,3,aff_noms = T, cex_titre = 0.8, repel = T)
```

```
## Warning: ggrepel: 345 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Afin d'illustrer plus clairement cette situasion, nous allons maintenant réaliser un exemple d'application à la main. Cela permettra de mieux comprendre le processus et les calculs impliqués dans la transformation des tableaux thématiques en un espace commun, ainsi que dans la comparaison des matrices P_m.

Prenons quatre matrices X_1, X_2 et X_3 de dimension respective $(3,2)$, $(3,3)$ et $(3,4)$.

$$X_1 = \begin{pmatrix} 0 & -1 \\ -1 & 2 \\ 1 & 1 \end{pmatrix}, \quad X_2 = \begin{pmatrix} -1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & -1 & -2 \end{pmatrix}, \quad X_3 = \begin{pmatrix} 1 & 0 & 2 & 2 \\ 1 & -1 & -1 & 0 \\ 0 & -2 & 0 & -2 \end{pmatrix}$$

On note respectivement: $M_1 = \frac{1}{2}I_2$, $M_2 = \frac{1}{3}I_3$ et $M_= \frac{1}{4}I_4$ les matrices de poids des colonnes.

Comme la seule chose commune à ces tableaux est les individus qu'ils décrivent, l'espace recherché ne peut que dépendre de ces individus. On définit pour $m \in 1, \ldots, 3$ $P_m := X_m M_m X_m{}'$ la matrice des produits scalaires des tableaux X_m.

Les matrices P_m se trouvent dans l'espace $\mathcal{P} = \mathbb{R}^{\nVdash \times \nVdash}$, de dimension 3^2.

Calculons à présent les matrices P_m:

$$P_1 = X_1 M_1 X_1{}' = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ -1 & 2 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 & -1 & 1 \\ -1 & 2 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & -2 & -1 \\ -2 & 5 & 1 \\ -1 & 1 & 2 \end{pmatrix}$$

$$P_2 = X_2 M_2 X_2{}' = \frac{1}{3}\begin{pmatrix} -1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & -1 & -2 \end{pmatrix}\begin{pmatrix} -1 & 0 & 0 \\ 2 & 1 & -1 \\ 1 & 0 & -2 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 6 & 2 & -4 \\ 2 & 1 & -1 \\ -4 & -1 & 5 \end{pmatrix}$$

$$P_3 = X_3 M_3 X_3{}' = \frac{1}{4}\begin{pmatrix} 1 & 0 & 2 & 2 \\ 1 & -1 & -1 & 0 \\ 0 & -2 & 0 & -2 \end{pmatrix}\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & -2 \\ 2 & -1 & 0 \\ 2 & 0 & -2 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 9 & -1 & -4 \\ -1 & 3 & 2 \\ -4 & 2 & 5 \end{pmatrix}$$

Dans la situation 1, nous avons défini le produit scalaire entre deux matrices X_t et X_s de taille (n,p) en utilisant les matrices de poids diagonales $W = \frac{1}{n}I_n$ pour les individus et $C = \frac{1}{p}I_p$ pour les colonnes.

Le produit scalaire entre deux matrices X_t et X_s de taille (n, p) est donné par :

$$[X_t|X_s] = \mathrm{tr}(CX_t'WX_s)$$

Cependant, dans la situation ici présente où les colonnes ont la même dimension que les lignes, nous prenons $C = W$. Ainsi, le produit scalaire devient :

$$[P_t|P_s] = \mathrm{tr}(WP_t'WP_s)$$

La norme d'une matrice P_t correspondant à ce produit scalaire sera toujours notée $[|P_t|]$.

Nous allons maintenant calculer les normes des matrices P_m afin de les normer. En effet, les dimensions des X_m ne sont pas les mêmes, ainsi que celles des matrices M_m. Autrement dit, les individus des tableaux X_m vivent dans un espace différent selon le tableau. Leurs produits scalaires n'ont donc pas une mesure comparable d'un tableau à l'autre. Pour rendre les P_m comparables, il faut donc les normer.

On a, $[|P_t|] = \sqrt{\mathrm{tr}(WP_t'WP_t)} = \sqrt{\frac{1}{9}\mathrm{tr}(X_t'X_t)}$, on en déduit donc que :

$$[|P_1|] = \sqrt{\frac{1}{9}\mathrm{tr}(P_1'P_1)}$$

Or,

$$P_1'P_1 = \frac{1}{4}\begin{pmatrix} 1 & -2 & -1 \\ -2 & 5 & 1 \\ -1 & 1 & 2 \end{pmatrix}\begin{pmatrix} 1 & -2 & -1 \\ -2 & 5 & 1 \\ -1 & 1 & 2 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 6 & -13 & -6 \\ -13 & 30 & 9 \\ -6 & 9 & 6 \end{pmatrix}$$

Donc, $\mathrm{tr}(P_1'P_1) = \frac{42}{4} \Rightarrow [|P_1|] = \sqrt{\frac{1}{9}\frac{42}{4}} = \sqrt{\frac{7}{6}}$.

En procédant de manière analogue, on obtient :

$$P_2'P_2 = \frac{1}{9}\begin{pmatrix} 56 & 18 & -46 \\ 18 & 6 & -14 \\ -46 & -14 & 42 \end{pmatrix} \quad \text{et} \quad P_3'P_3 = \frac{1}{16}\begin{pmatrix} 118 & -20 & -70 \\ -20 & 14 & 26 \\ -70 & 26 & 84 \end{pmatrix}$$

D'où, $[|P_2|] = \sqrt{\frac{104}{9}}$ et $[|P_3|] = \sqrt{\frac{113}{16}}$.

Une fois les P_m normées ($P_m{}^* = \frac{P_m}{[|P_m|]}$), elles se trouvent toutes sur la sphère unité de $\mathcal{P}$. Leur proximité est donc mesurable par le cosinus de l'angle qu'elles forment entre elles.

On applique le programme STATIS 1 aux matrices P_m décrivant les mêmes individus. On notera $F_1, \ldots, F_T$ les composantes principales obtenues.

Les calculs manuels des coefficients RV des matrices peuvent être très laborieux et sujets à des erreurs, c'est pourquoi nous allons maintenant utiliser des fonctions pour automatiser ces calculs. Nous allons calculer la matrice Z des coefficients RV en utilisant la fonction `matcoeff_RV`, puis nous diagonaliserons Z afin d'obtenir U, les vecteurs propres de Z, et ν, les valeurs propres. Enfin, nous en déduirons en utilisant la fonction `cmp` les composantes principales $F_k = \sum_{i=1}^{T} u_k{}^{(i)} P_m{}^*$.

Àprès exécution de ces fonctions on obtient les résultats suivants:

$$Z = \begin{pmatrix} 1,0000 & 0,2875 & 0,6172 \\ 0,2875 & 1,0000 & 0.8475 \\ 0.6172 & 0.8475 & 1.0000 \end{pmatrix}, \quad U = \begin{pmatrix} -0,4789 & 0,8190 & -0,3160 \\ -0,5810 & -0,5655 & -0,5854 \\ -0,6582 & -0.0967 & 0,7466 \end{pmatrix}, \quad \nu = \begin{pmatrix} 2,1971 \\ 0,7286 \\ 0,0743 \end{pmatrix}$$

Les composantes principales sont:

$$F_1 = \begin{pmatrix} -2.5164 & 0.2426 & 1.4694 \\ 0.2426 & -1.7023 & -0.3328 \\ 1.4694 & -0.3328 & -2.4261 \end{pmatrix}, \quad F_2 = \begin{pmatrix} -0.8056 & -1.0703 & 0.3692 \\ -1.0703 & 1.6672 & 0.5040 \\ 0.3692 & 0.5040 & -0.2393 \end{pmatrix} F_3 = \begin{pmatrix} 0.2604 & -0.2872 & 0.1952 \\ -0.2872 & -0.4236 & 0.3459 \\ 0.1952 & 0.3459 & 0.1263 \end{pmatrix}$$

2.2.1 2.3. Aides à l'interprétation

a) La formule pour calculer le cosinus carré $\cos^2(P_m, F_k)$ entre P_m et F_k est :

$$\cos^2(P_m, F_k) = \frac{[P_m \mid F_k]^2}{[|P_m|]^2 [|F_k|]^2}$$

b) On en déduit le calcul du $QLT_k(P_m)$, qui est la somme des cosinus carrés jusqu'à la composante k :

$$QLT_k(P_m) = \cos^2(P_m, \langle F_1, \ldots, F_k \rangle) = \sum_{j=1}^{k} \cos^2(P_m, F_j)$$

c) Le calcul de la contribution $CTR_k(P_m)$ est donné par :

$$CTR_k(P_m) = \frac{\cos^2(P_m, F_k)}{\lambda_k}$$

d) Programmation et application.

```
# Appliquer les fonctions au tableau de données d'application
# Supposons que P_m représente les données projetées sur une composante principale
lam <- cmp(lst,1)$lambda[1]
cos_square(lst$M,F1_theme)
```

```
## [1] 0.5381143
```

```
QLT_k(lst$M,lst)
```

```
## [1] 2.187172
```

```
CTR_k(lst$M,F1_theme,lam)
```

```
## [1] 0.1496646
```

3 Annexe

3.1 Situation 1

3.1.1 Programme des fonctions prd_scalaire et norme

La fonction calculant le produit scalaire de Frobénius:

```
prd_scalaire = function(A,B){
  # les dimension des matrices
  na = length(A[,1]); nb = length(B[,1])
  pa = length(A[1,]); pb = length(B[1,])

  # Les matrices de ponderation
  Wa = (1/na)*diag(na); Ca = (1/pa)*diag(pa)
  Wb = (1/nb)*diag(na); Cb = (1/pb)*diag(pb)
```

```
  # Calcul des matrices A_tild et B_tild
  A_tild = sqrt(Wa) %*% A %*% sqrt(Ca)
  B_tild = sqrt(Wb) %*% B %*% sqrt(Cb)

  # Produit scalaire
  ps = sum(diag(t(A_tild)%*%B_tild))

  return(ps)
 }
```

La fonction calculant la norme d'une matrice A associée au produit scalaire de Frobénius

```
norme = function(A){
  return(sqrt(prd_scalaire(A,A)))
}
```

3.1.2 Coefficient RV

La fonction calculant le coefficient RV d'Escoufier:

```
coef_RV <- function(Xi,Xj){
      prd_sclr = prd_scalaire(Xi, Xj)
      norm_i = norme(Xi)
      norm_j = norme(Xj)
      rv = prd_sclr / (norm_j * norm_i)

      return(rv)
}

matcoef_RV = function(T_tableaux) {
  t = length(T_tableaux)
  mat_rv = matrix(rep(NA, t * t), nrow = t, ncol = t)

  for (i in seq_along(T_tableaux)) {
    for (j in seq_along(T_tableaux)) {
      # Stocker le résultat dans une matrice
      mat_rv[i, j] = coef_RV(T_tableaux[[i]],T_tableaux[[j]])
    }
  }

  return(mat_rv)
}
```

Fonction donnant les vecteurs u solutions et les valeurs propres associées:

```
vecval_prop = function(matrice) {

  resultat_propre = eigen(matrice)# list ayant les valeurs et vecteurs propres
  val_prop = resultat_propre$values #valeurs propres
  vec_prop = resultat_propre$vectors #vecteurs propres associées

  return(list(val_prop = val_prop, vec_prop = vec_prop))
}
```

3.1.3 Dépliage du tableau cubique en un tableau juxtaposé et Composantes principales:

```
oper_inert = function(X){

  p = length(X[1,]) # la dimension des individus
  n = length(X[,1]) # la dimension des variables
  W = diag(n)/n
  M = matrix(0, nrow = p, ncol = p)

  for (i in 1:p) {
    M[i,i] = 1/sqrt(t(X[,i])%*%W%*%X[,i])
  }
  mat_d = M %*% t(X) %*% W %*% X %*% M
  return(mat_d)
}
```

3.1.4 La représentation des individus et des tableaux

Représentation des individus:

```
geo_ind <- function(Y_nrm, k, l, aff_noms = FALSE, col = "navy", cex_titre = 1, repel = FALSE) {

  if (l <= k) {
    print("Le 1er axe doit être strictement plus petit que le 2nd")
    return(NULL)
  }

  # Créer un data frame avec les composantes principales
  df <- data.frame(F_k = composante_principale(Y_nrm, k)$F_k, F_l = composante_principale(Y_nrm, l)$F_k
  if (is.null(rownames(Y_nrm))) {
    rownames(df) <- paste("L", 1:nrow(Y_nrm), sep = "")
  } else {
    rownames(df) <- rownames(Y_nrm)
  }


  # Plot les composantes principales dans un plan avec l'origine au centre
  p <- ggplot(df, aes(x = F_k, y = F_l)) +
    geom_point(shape = 15, color = "black", size = 1.5) + # Utilisation de petits carrés noirs
    theme_classic() +
    labs(x = paste("F", k, sep = ""), y = paste("F", l, sep = "")) +
    ggtitle(paste("Représentation des individus dans le plan (", k, ",", l, ")")) +
    theme(plot.title = element_text(size = cex_titre * 12, face = "bold", hjust = 0.5))

  # Ajouter des lignes pour les axes horizontal et vertical se coupant à l'origine
  p <- p + geom_hline(yintercept = 0, linetype = "dotted", lwd = 0.5) +
    geom_vline(xintercept = 0, linetype = "dotted", lwd = 0.5)

  if (aff_noms == TRUE) {
    if (repel) {
      # Ajouter les étiquettes avec repulsion
      p <- p + geom_text_repel(aes(label = rownames(df)), size = 3, color = col, box.padding = unit(0.2
    } else {
      # Ajouter les étiquettes sans repulsion
```

```
      p <- p + geom_text(aes(label = rownames(df)), size = 3, color = col, nudge_x = 0.2, nudge_y = 0.2)
    }
  }

  print(p)
}
```

Cosinus entre les tableaux et les composantes principales :

```
# Calcul des cosinus entre Xt (resultats_n) et chaque composante principale F_k
cosinus = function(X_t, F_k){
  return(coef_RV(X_t,F_k))
}
```

Représentation des tableaux:

```
representation_graph_tab <- function(T_tab, k, l, aff_noms = TRUE, col = "navy", cex_titre = 1, repel =
  if (l <= k) {
    print("Le 1er axe doit être strictement plus petit que le 2nd")
    return(NULL)
  }

  t <- length(T_tab)
  mat_cos <- matrix(NA, nrow = t, ncol = t)
  for (i in 1:t) {
    for (j in 1:t) {
      mat_cos[i, j] <- cosinus(T_tab[[i]], cmp(T_tab, j)$F_k)
    }
  }

  # Récupérer les noms des tableaux ou les nommer par défaut
  row_names <- if (is.null(names(T_tab))) paste("X", 1:t, sep = "_") else names(T_tab)
  mat_cos <- as.data.frame(mat_cos)
  rownames(mat_cos) <- row_names

  F_k <- mat_cos[, k]  # Récupération de la kième composante principale
  F_l <- mat_cos[, l]  # Récupération de la lième composante principale

  # Création des données pour le cercle unité
  df <- data.frame(
    varabs = cos(seq(0, 2 * pi, length.out = 100)),
    varord = sin(seq(0, 2 * pi, length.out = 100))
  )

  p <- ggplot(mat_cos, aes(x = F_k, y = F_l)) +
    geom_segment(aes(xend = 0, yend = 0), color = "#445577") +
    geom_point(color = "red", size = 1.5) +
    theme_classic() +
    coord_fixed(ratio = 1) +
    geom_hline(yintercept = 0, linetype = "dotted", lwd = 0.5) +
    geom_vline(xintercept = 0, linetype = "dotted", lwd = 0.5) +
    labs(x = paste("F", k, sep = ""),
         y = paste("F", l, sep = "")) +
    ggtitle(paste("Représentation des tableaux dans le plan (", k, ",", l, ")")) +
    theme(plot.title = element_text(size = cex_titre * 12, face = "bold", hjust = 0.5))+
```

```
    geom_path(data = df, aes(varabs, varord), color = "#222211", linewidth = 0.5) +  # Ajout du cercle
    xlim(-1, 1) +
    ylim(-1, 1) +
    theme(plot.title = element_text(size = cex_titre * 12, face = "bold", hjust = 0.5))

  if (aff_noms) {
    if (repel) {
      p <- p + geom_text_repel(aes(label = rownames(mat_cos)), color = col, size = 3)
    } else {
      p <- p + geom_text(aes(label = rownames(mat_cos)), color = col, size = 3,
                         hjust = ifelse(F_k >= 0, 0, 1), vjust = ifelse(F_l >= 0, 0, 1))
    }
  }

  print(p)
}
```

3.2 Situation 2

3.2.1 Les données partitionné en thèmes

```
# Tableau à trois dimension
data(chickenk)

M = as.matrix(chickenk$Mortality)
rownames(M) = paste("L",1:351,sep = "")

FS = as.matrix(chickenk$FarmStructure)
rownames(FS) = paste("L",1:351,sep = "")

OFH = as.matrix(chickenk$OnFarmHistory)
rownames(OFH) = paste("L",1:351,sep = "")

FC = as.matrix(chickenk$FlockCharacteristics)
rownames(FC) = paste("L",1:351,sep = "")

CTS = as.matrix(chickenk$CatchingTranspSlaught)
rownames(CTS) = paste("L",1:351,sep = "")

data = list(M = M, FS = FS, OFH = OFH, FC = FC, CTS = CTS)
```

3.2.2 2.3. Aides à l'interprétation

```
# Définition des fonctions
cos_square <- function(Pm, Fk) {
  rv <- coef_RV(Pm,Fk)
  cos2 <- rv^2
  return(cos2)
}


QLT_k <- function(Pm, lst_Fs) {
  k <- length(lst_Fs)
```

```
  cos <- 0

  for (j in 1:k) {
    cos <- cos_square(Pm,lst_Fs[[j]]) + cos
  }
  return(cos)
}


CTR_k <- function(Pm, Fk, lambda_k) {
  c <- cos_square(Pm,Fk)/lambda_k
  return(c)
}
```