# Données manquantes en ACM : l'algorithme NIPALS

MARIE CHAVENT & VANESSA KUENTZ & BENOÎT LIQUET

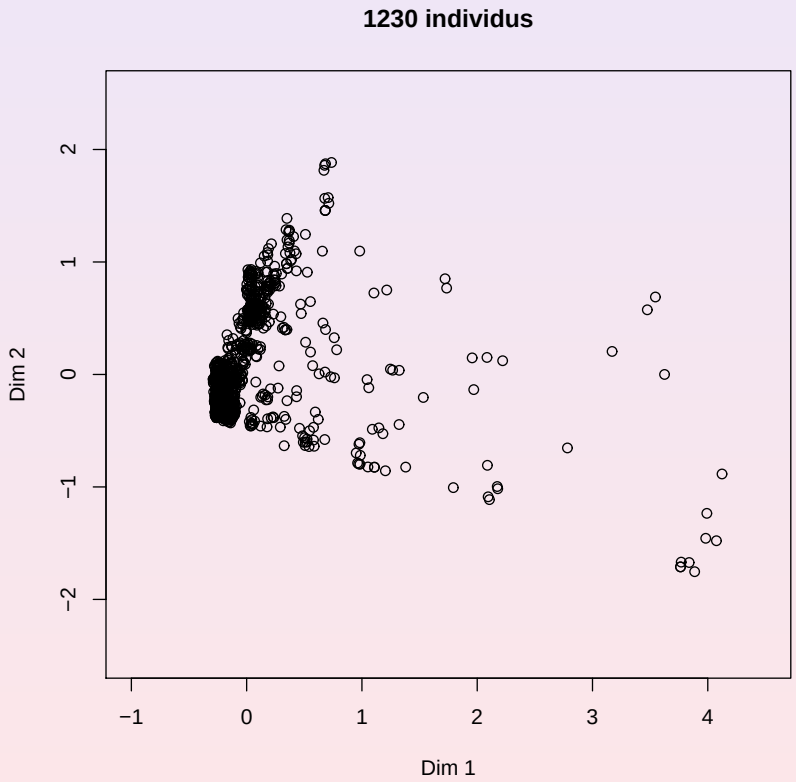
IMB, Université de Bordeaux, France
INRIA Bordeaux Sud-Ouest, CQFD Team
INSERM, U897

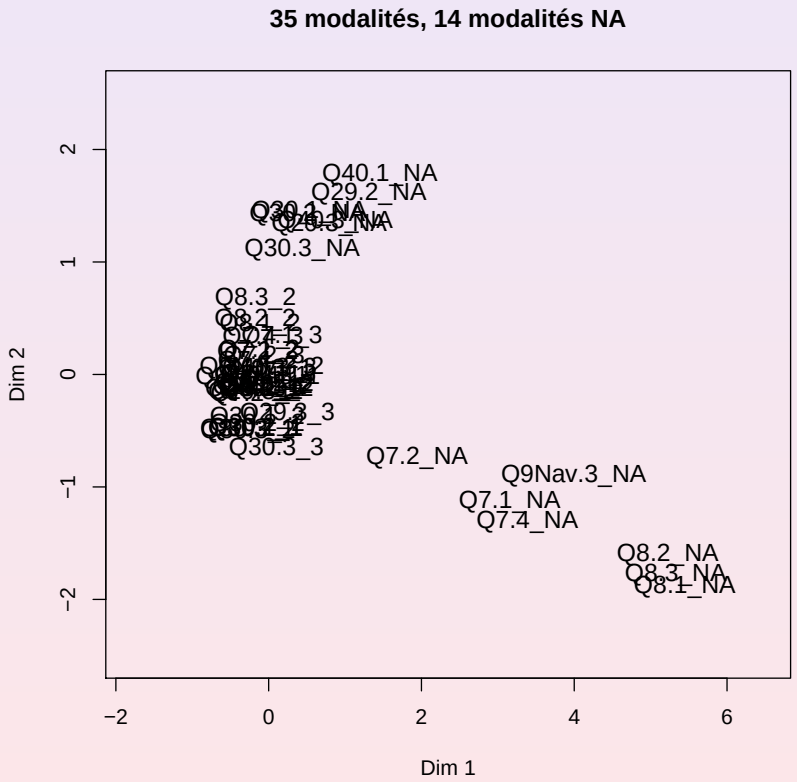
SFC09, Grenoble

# Introduction

- Motivation : un exemple réelle avec données manquantes

Plans factoriels 1-2 de l'ACM de FactoMineR



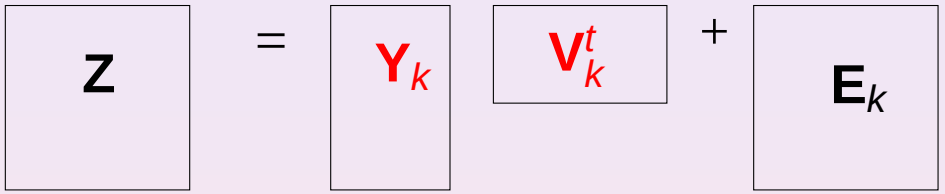
# Introduction

- ACM = ACP pondérée des profils lignes et des profils colonnes du TDC
- Algorithmes d'ACP permettant la gestion des données manquantes :
  - NIPALS (*Regression PLS*, Tenenhaus)
  - ACP iterative (Josse, Husson & Pagès, *SFDS 09*)
  - IMLS (Wasito & Mirkin, *CSDA, 2005, 2006*)

1. NIPALS pour l'ACM de données incomplètes
2. Etude qualitative
3. Etude quantitative

# Présentation générale de NIPALS

- Meilleure approximation d'une matrice $\mathbf{Z}$ de rang $p$ par une matrice $\mathbf{Z}_k = \mathbf{Y}_k \mathbf{V}_k^t$ de rang $k < p$



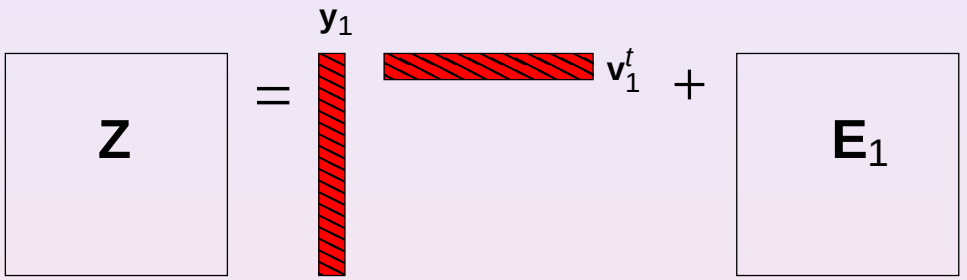

$\Rightarrow$ minimiser:

  - $||\mathbf{Z} - \mathbf{Y}_k \mathbf{V}_k^t||^2$ si les données sont complètes
  - $||\mathbf{W} * (\mathbf{Z} - \mathbf{Y}_k \mathbf{V}_k^t)||^2$ si les données sont incomplètes, $\mathbf{W}$ est une matrice de poids, $w_{ij} = 0$ si $z_{ij}$ manquant, $w_{ij} = 1$ sinon.

- Décomposition en valeurs singulières de $\mathbf{Z}$
- Algorithme itératif NIPALS qui s'adapte au cas incomplet

# Etape 1 : meilleure approximation $\mathbf{Z}_1$ de rang 1

Définir : $\mathbf{Z}_1 = \mathbf{y}_1 \mathbf{v}_1^t$



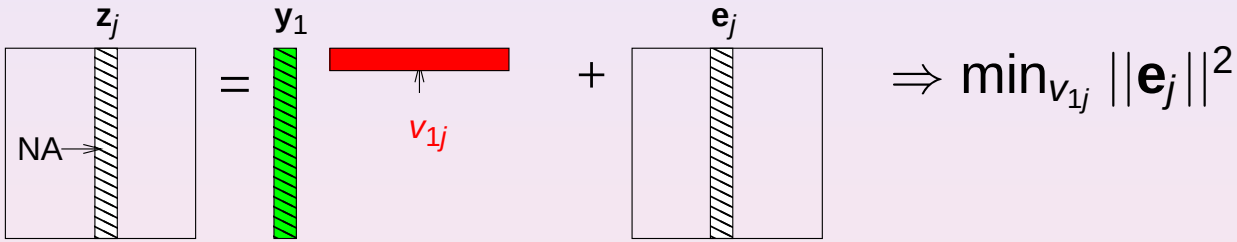

$\Rightarrow \min_{\mathbf{y}_1, \mathbf{v}1} ||\mathbf{E}_1||^2$

$\Rightarrow$ Algorithme :

- Initialiser $\mathbf{y}_1$
- Répéter jusqu'à convergence
  1) on fixe $\mathbf{y}_1$ et on calcule $\mathbf{v}1$
     $\mathbf{v}_1$ normalisé à 1
  2) on fixe $\mathbf{v}_1$ et on calcule $\mathbf{y}_1$

# Algorithme d'approximation en dimension 1

1) on fixe $\mathbf{y}_1$ et on calcule $\mathbf{v1}$ :
pour $j = 1$ à $p$, on écrit $\mathbf{z}_j = v_{1j}\mathbf{y}_1 + \mathbf{e}_j$



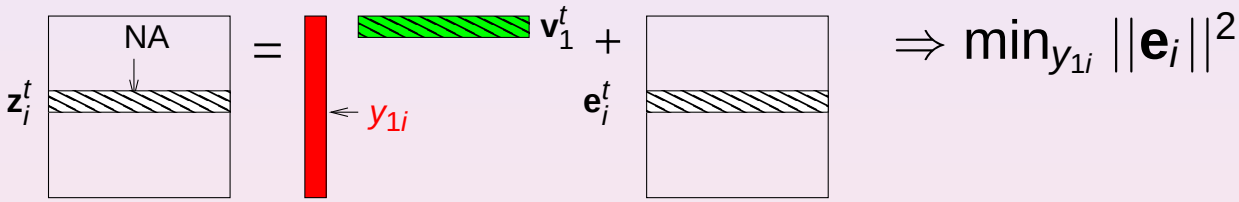

$\Rightarrow v_{1j} = \frac{\sum_{i=1}^{n} z_{ij} y_{1i}}{\sum_{i=1}^{n} y_{1i}^2}$

$\Rightarrow$ si NA dans $\mathbf{z}_j$ on les "passe" dans la somme

$\mathbf{v}_1$ normalisé à 1

# Algorithme d'approximation en dimension 1

2) on fixe $\mathbf{v}_1$ et on calcule $\mathbf{y}_1$ :
pour $i = 1$ à $n$, on écrit $\mathbf{z}_i = y_{1i}\mathbf{v}_1 + \mathbf{e}_i$



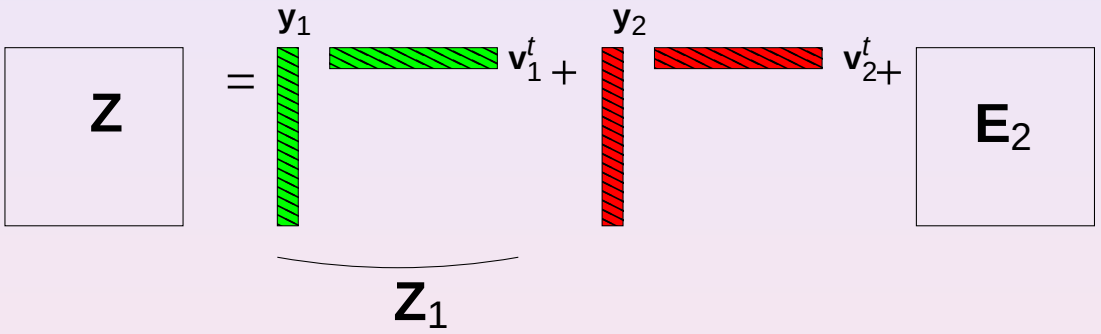

$\Rightarrow \min_{y_{1i}} ||\mathbf{e}_i||^2$

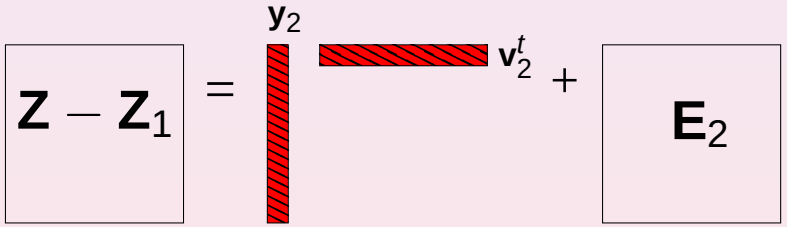
$\Rightarrow y_{1i} = \frac{\sum_{j=1}^{p} z_{ij} v_{1j}}{\sum_{j=1}^{p} v_{1j}^2}$

$\Rightarrow$ si NA dans $\mathbf{z}_i$ on les "passe" dans la somme

# Etape 2 : meilleure approximation $\mathbf{Z}_2$ de rang 2

Etape 2 : définir $\mathbf{Z}_2 = \mathbf{y}_1\mathbf{v}_1^t + \mathbf{y}_2\mathbf{v}_2^t$ :




$\Rightarrow$ Algorithme d'approximation en dimension 1

# NIPALS en ACP

- La matrice de données $\mathbf{X}_{n\times p}$ est quantitative
- Comment définir $\mathbf{Z}_{n\times p}$ pour avoir
  - $\mathbf{Y}_k$ est la matrice des $k$ composantes principales des individus
  - $\mathbf{V}_k$ est la matrice des $k$ axes principaux

$$\boxed{\mathbf{Z}\ ?} = \boxed{\mathbf{Y}_k}\ \boxed{\mathbf{V}_k^t} + \boxed{\mathbf{E}_k}$$

- Il suffit de prendre la matrice des données centrées et réduites :

$$\mathbf{Z} = (\mathbf{X} - \mathbf{1}\mathbf{g}^t)\mathbf{D}_\mathbf{c}^{-1/2}$$

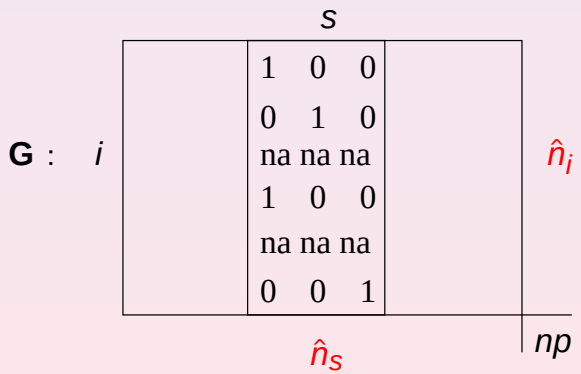
# NIPALS en ACM

- La matrice de données $\mathbf{X}_{n\times p}$ est qualitative
- Comment définir $\mathbf{Z}_{n\times q}$ ?
- On calcule
  - $\mathbf{F}_{n\times q}$, matrice des fréquences relatives du TDC $\mathbf{G}$, $f_{is} = \frac{g_{is}}{n*p}$
  - $\mathbf{r} = (f_{1.} \dots f_{i.} \dots f_{n.})$, $f_{i.} = \frac{1}{n}$
  - $\mathbf{c} = (f_{.1} \dots f_{.s} \dots f_{.q})$, $f_{.s} = \frac{n_s}{n*p}$
  - $\mathbf{R} = \mathbf{D_r}^{-1}(\mathbf{F} - \mathbf{r}\mathbf{c}^t)$ de la matrice des profils lignes centrés
- Il suffit alors de prendre la matrice des profils lignes centrés et "réduits" :

$$\mathbf{Z} = \mathbf{R}\mathbf{D_c}^{-1/2}$$

# Données incomplètes

- En ACP, on estime les moyennes et les écart-types des colonnes de la matrice des données "complète" inconnue à partir des valeurs observées.
- En ACM, l'idée est d'estimer les marges du TDC inconnu à partir du TDC observées, **G** :




$$\hat{n}_i = p$$
$$\hat{n}_s = \frac{n * n_s}{n_j}$$

# Données incomplètes

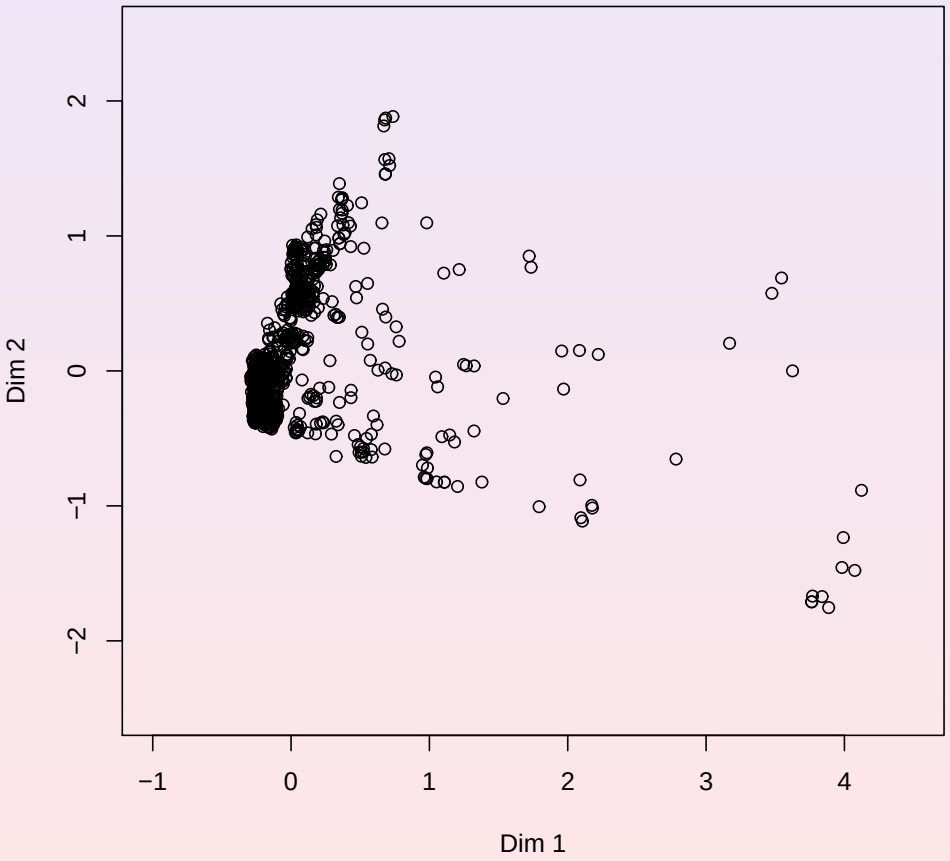
- On calcule
  - $\hat{\mathbf{F}}_{n\times q}$, avec $\hat{f}_{is} = \begin{cases} \frac{g_{is}}{n*p} & \text{si } g_{is} \neq \text{na}, \\ \text{na}, & \text{sinon} \end{cases}$
  - $\hat{\mathbf{r}} = (\hat{f}_{1.} \dots \hat{f}_{i.} \dots \hat{f}_{n.})$, $\hat{f}_{i.} = \frac{\hat{n}_i}{n*p} = \frac{1}{n}$
  - $\hat{\mathbf{c}} = (\hat{f}_{.1} \dots \hat{f}_{.s} \dots \hat{f}_{.q})$, $\hat{f}_{.s} = \frac{\hat{n}_s}{n*p} = \frac{n_s}{p*n_j}$
  - $\hat{\mathbf{R}} = \mathbf{D}_{\hat{\mathbf{r}}}^{-1}(\hat{\mathbf{F}} - \hat{\mathbf{r}}\hat{\mathbf{c}}^t)$ de la matrice des profils lignes centrés
- Il suffit alors de prendre la matrice des profils lignes centrés et "réduits":

$$\mathbf{Z} = \hat{\mathbf{R}}\mathbf{D}_{\hat{\mathbf{c}}}^{-1/2}$$

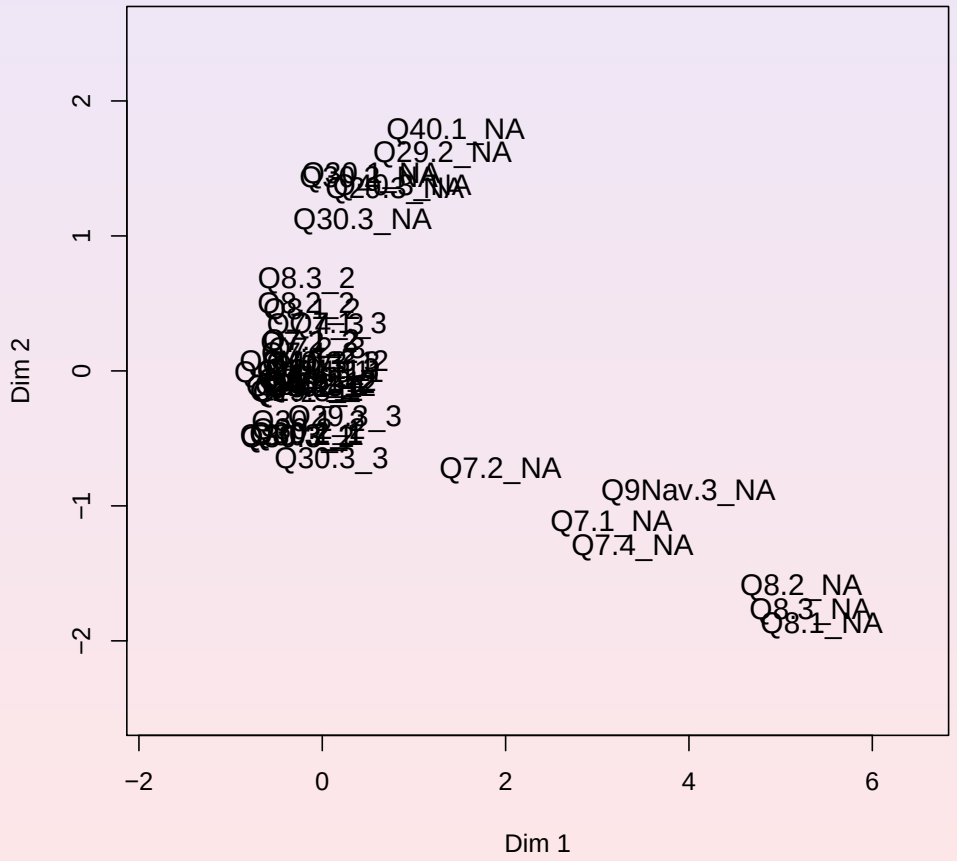
# L’exemple réelle des données “vnf”

## Plans factoriels 1-2 de l’ACM réalisés avec FactoMineR

**1230 individus**



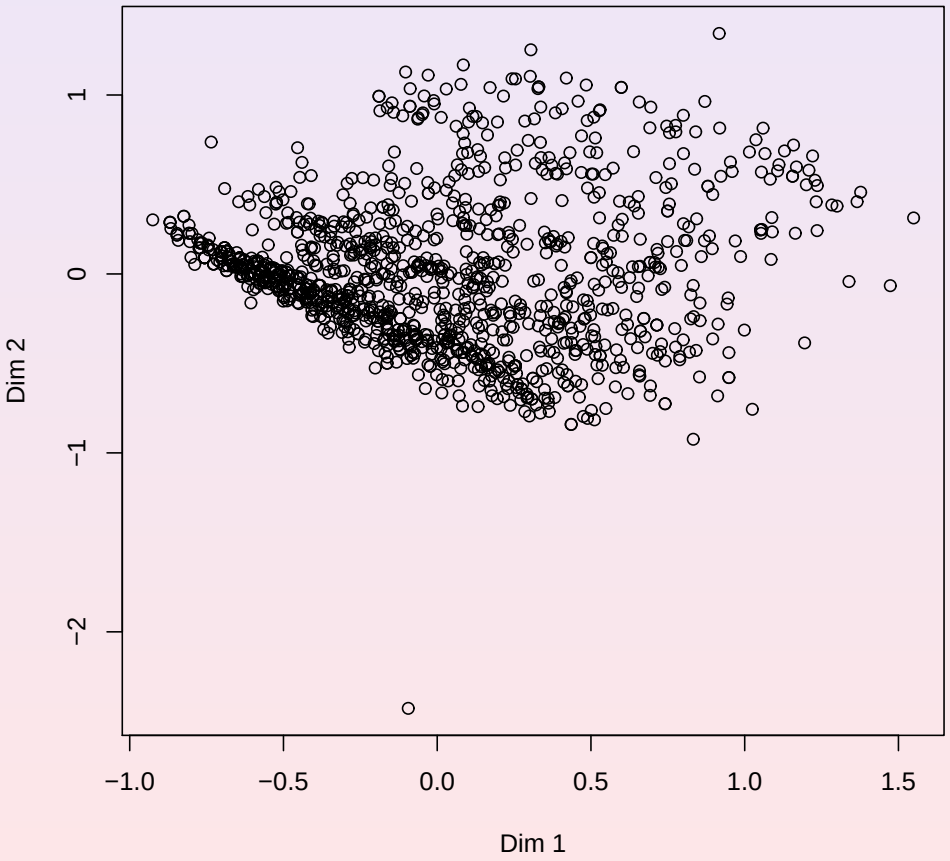

**35 modalités, 14 modalités NA**



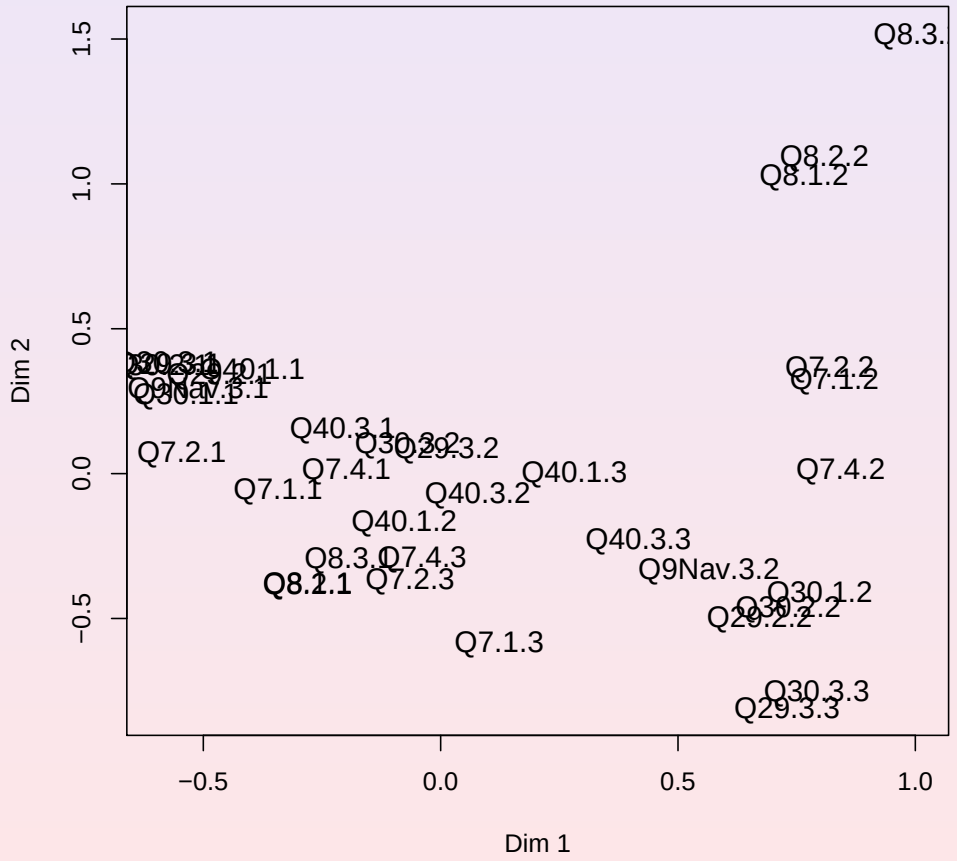
# L’exemple réelle des données “vnf”

## Plans factoriels 1-2 de l’ACM réalisés avec NIPALS

**1230 individus**



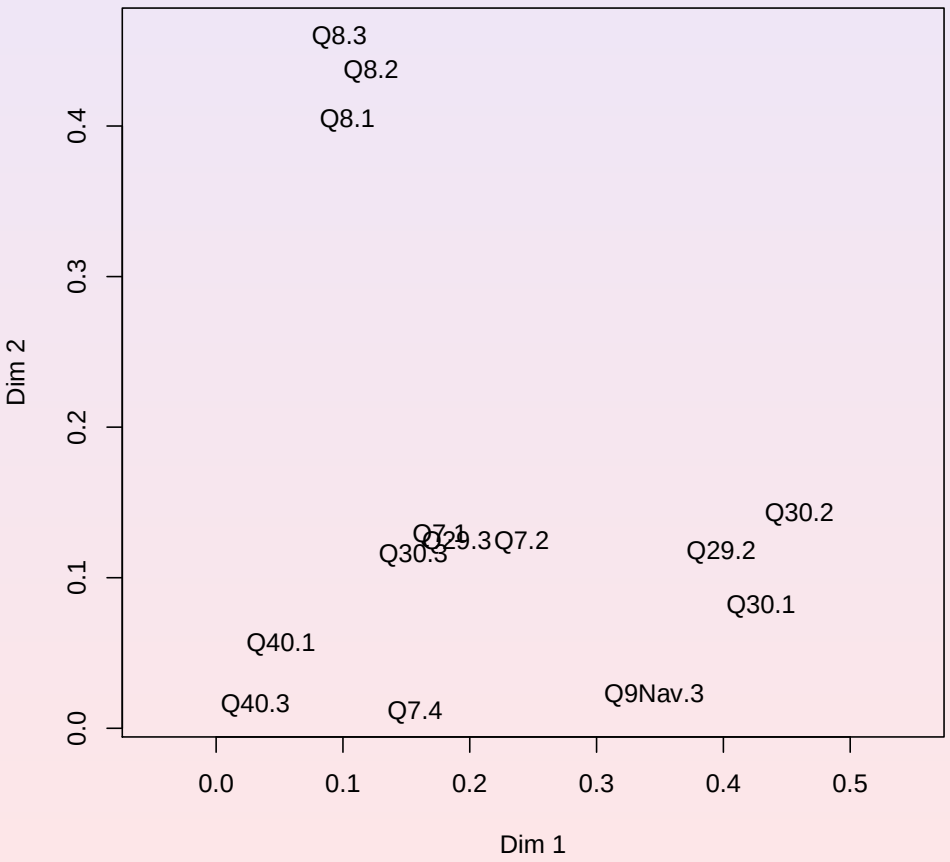

**35 modalités**



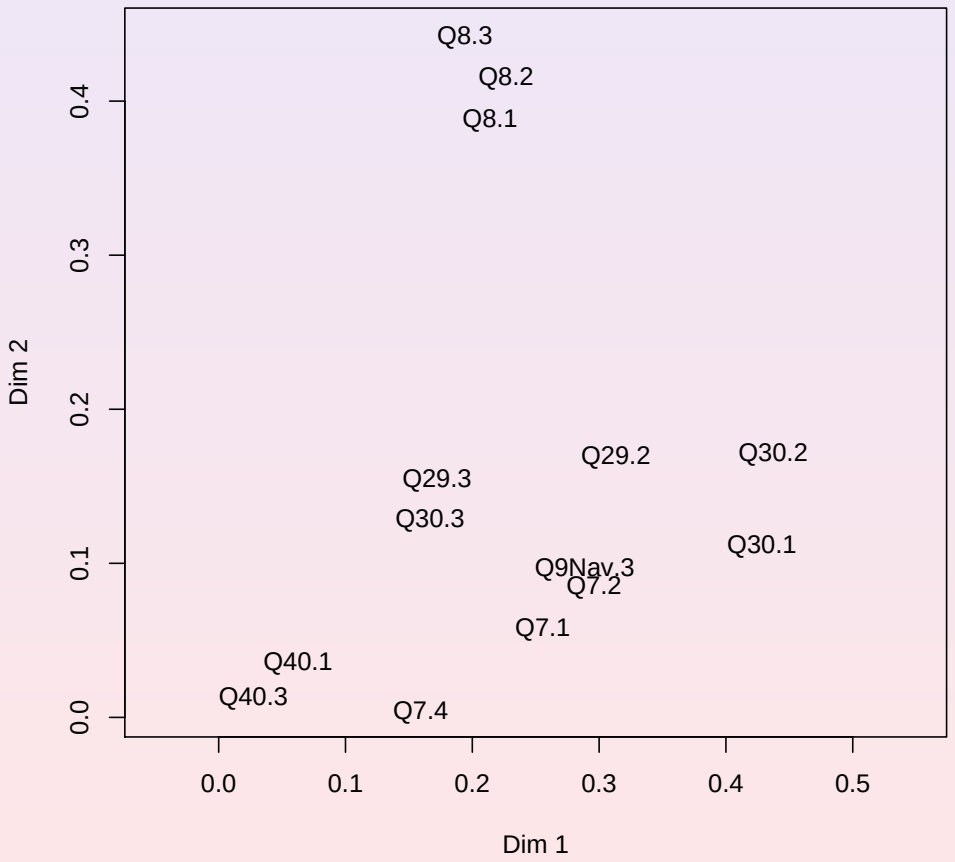
# L’exemple réelle des données “vnf”

## Plans factoriels 1-2 des rapports de corrélations des 14 variables

**MCA sur 709 individus sans NA**

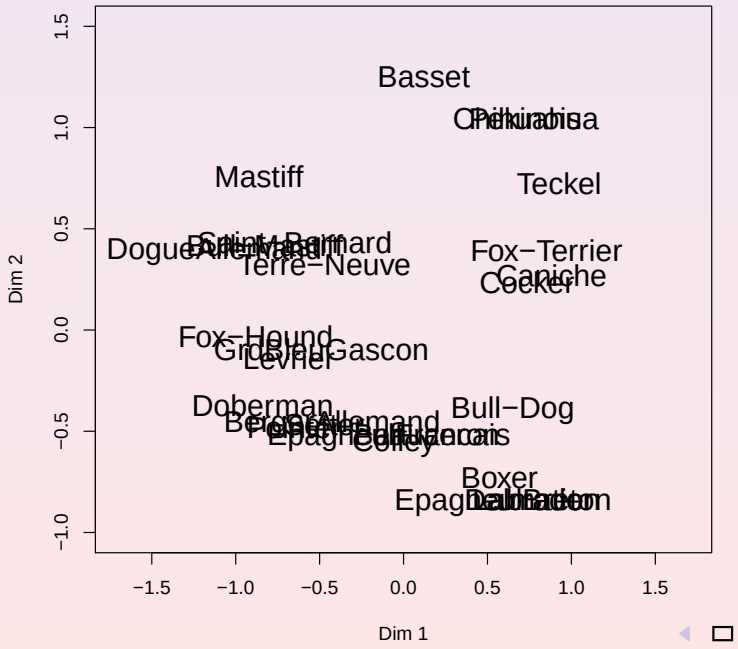



**NIPALS sur 1230 individus avec NA**



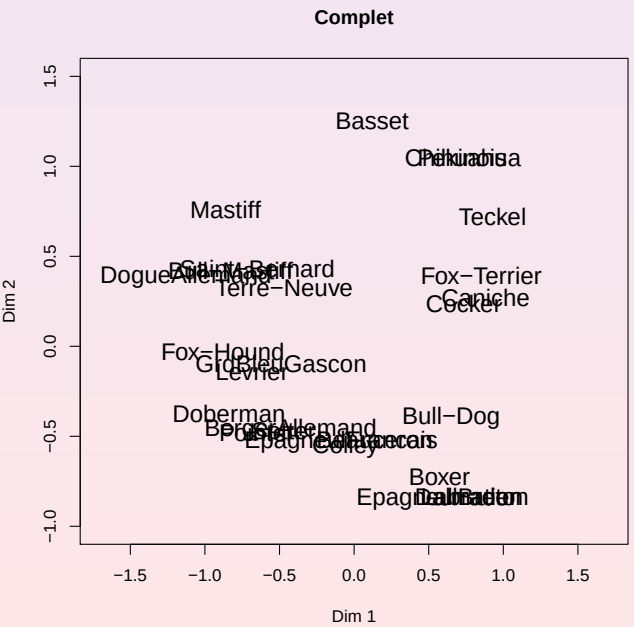
# L’exemple des données “chiens”

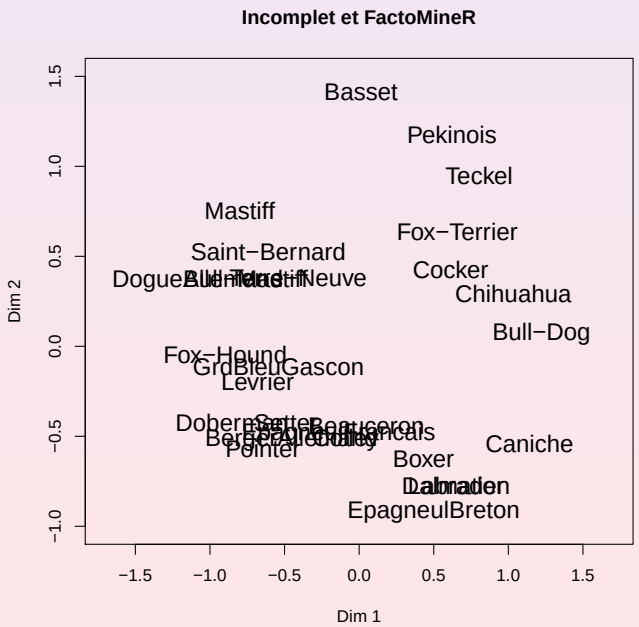
| Nom | TAILLE | POIDS | VELOCITE | INTELLI | AFFECTIO | AGRESSIV |
|---|---|---|---|---|---|---|
| Basset | petit | leger | lent | peu | peu | agressif |
| Chihuahua | petit | leger | lent | peu | très | nonagressif |
| Pekinois | petit | leger | lent | peu | très | nonagressif |
| Caniche | petit | leger | rapide | très | très | nonagressif |
| Bull-Dog | moyen | lourd | lent | moyen | très | nonagressif |
| EpagneulBreton | moyen | lourd | rapide | très | très | nonagressif |
| Dalmatien | moyen | lourd | rapide | moyen | très | nonagressif |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . |

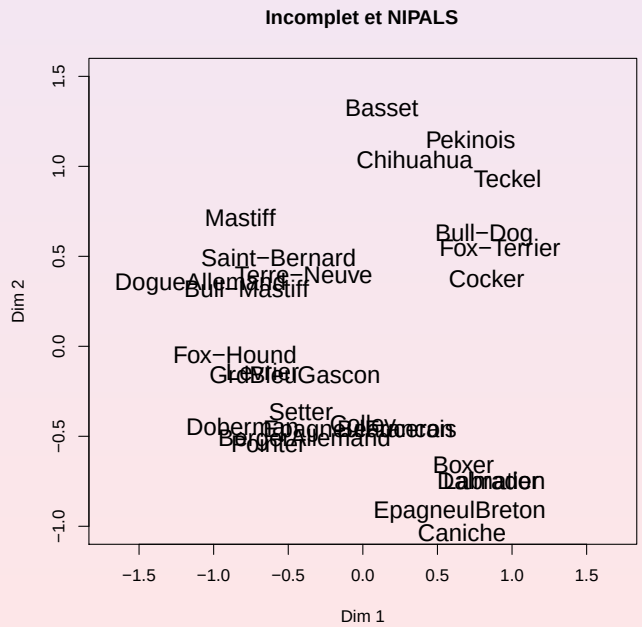


# L’exemple des données “chiens”

| Nom | TAILLE | POIDS | VELOCITE | INTELLI | AFFECTIO | AGRESSIV |
|---|---|---|---|---|---|---|
| Basset | petit | leger | lent | peu | peu | agressif |
| Chihuahua | NA | NA | lent | peu | très | nonagressif |
| Pekinois | petit | leger | lent | peu | très | nonagressif |
| Caniche | NA | NA | rapide | très | très | nonagressif |
| Bull-Dog | NA | NA | lent | moyen | très | nonagressif |
| EpagneulBreton | moyen | lourd | rapide | très | très | nonagressif |
| Dalmatien | moyen | lourd | rapide | moyen | très | nonagressif |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . |



# L’exemple des données “chiens”

Comparaisons des matrices $\mathbf{Y}_k$ axe par axe:

| | Complet | | | FactoMineR incomplet | | | NIPALS incomplet | | |
|---|---|---|---|---|---|---|---|---|---|
| | Dim 1 | Dim 2 | Dim 3 | Dim 1 | Dim 2 | Dim 3 | Dim 1 | Dim 2 | Dim 3 |
| Beauceron | 0.14 | -0.52 | -0.25 | 0.05 | -0.44 | -0.43 | -0.21 | 0.46 | 0.33 |
| Basset | 0.12 | 1.25 | 0.22 | 0.01 | 1.42 | -0.37 | -0.12 | -1.32 | -0.36 |
| BergerAllemand | -0.43 | -0.47 | 0.58 | -0.45 | -0.52 | 0.03 | 0.39 | 0.52 | -0.44 |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
| Setter | -0.47 | -0.46 | 0.23 | -0.49 | -0.42 | 0.01 | 0.41 | 0.36 | -0.30 |
| Teckel | 0.94 | 0.72 | 0.11 | 0.82 | 0.95 | -0.54 | -0.98 | -0.93 | -0.32 |
| Terre-Neuve | -0.46 | 0.33 | -0.77 | -0.40 | 0.38 | 0.25 | 0.38 | -0.40 | 0.66 |

$\Rightarrow$ Corrélations entre les axes : $r(Dim1, Dim1) = 0.97$

| | Dim 1 | Dim 2 | Dim 3 |
|---|---|---|---|
| FactoMineR | 0.97 | 0.91 | 0.21 |
| NIPALS | -0.98 | -0.87 | -0.93 |

# L’exemple des données “chiens”

Comparaisons des matrices $\mathbf{Y}_k$ globalement sur tous les axes:

| | Complet | | | FactoMineR incomplet | | | NIPALS incomplet | | |
|---|---|---|---|---|---|---|---|---|---|
| | Dim 1 | Dim 2 | Dim 3 | Dim 1 | Dim 2 | Dim 3 | Dim 1 | Dim 2 | Dim 3 |
| Beauceron | 0.14 | -0.52 | -0.25 | 0.05 | -0.44 | -0.43 | -0.21 | 0.46 | 0.33 |
| Basset | 0.12 | 1.25 | 0.22 | 0.01 | 1.42 | -0.37 | -0.12 | -1.32 | -0.36 |
| BergerAllemand | -0.43 | -0.47 | 0.58 | -0.45 | -0.52 | 0.03 | 0.39 | 0.52 | -0.44 |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
| Setter | -0.47 | -0.46 | 0.23 | -0.49 | -0.42 | 0.01 | 0.41 | 0.36 | -0.30 |
| Teckel | 0.94 | 0.72 | 0.11 | 0.82 | 0.95 | -0.54 | -0.98 | -0.93 | -0.32 |
| Terre-Neuve | -0.46 | 0.33 | -0.77 | -0.40 | 0.38 | 0.25 | 0.38 | -0.40 | 0.66 |

- Coefficient RV : “compare” la matrice $\mathbf{W} = \mathbf{Y}_3\mathbf{Y}_3^t$ à la “vrai” matrice $\mathbf{W} = \mathbf{Y}_3\mathbf{Y}_3^t$ :

$$RV(\mathbf{Y}_3, \mathbf{Y}_3) = \frac{\text{trace}(\mathbf{W}, \mathbf{W})}{\sqrt{\text{trace}(\mathbf{W}, \mathbf{W})\text{trace}(\mathbf{W}, \mathbf{W})}} = 0.79$$

| | RV |
|---|---|
| FactoMineR | 0.79 |
| NIPALS | 0.89 |

# ACM itérative

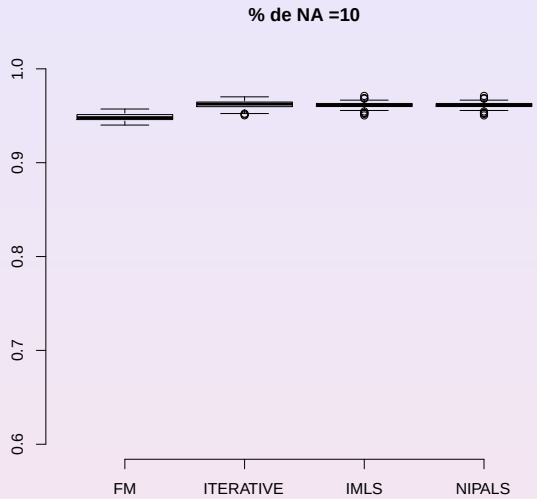
- Adaptation de l'ACP itérative (Josse, Husson & Pagès)
- Algorithme de décomposition en valeur singulière (SVD) itératif : imputer les valeurs manquantes dans une matrice réelle $\mathbf{Z}$:
  1. choisir le nombre $k$ de dimension
  2. remplir arbitrairement les "trous" dans $\mathbf{Z}$
  3. répéter jusqu'à convergence
     - faire une SVD de $\mathbf{Z}$: $\mathbf{Z} = \mathbf{U}\Lambda\mathbf{V}$
     - meilleure approximation de rang $k$ : $\mathbf{Z}_k = \mathbf{U}_k\Lambda_k\mathbf{V}_k^t$
     - remplir les "trous" dans $\mathbf{Z}$ avec les valeurs estimées : $\mathbf{Z} = \mathbf{W}\mathbf{Z} + (1 - \mathbf{W})\mathbf{Z}_k$

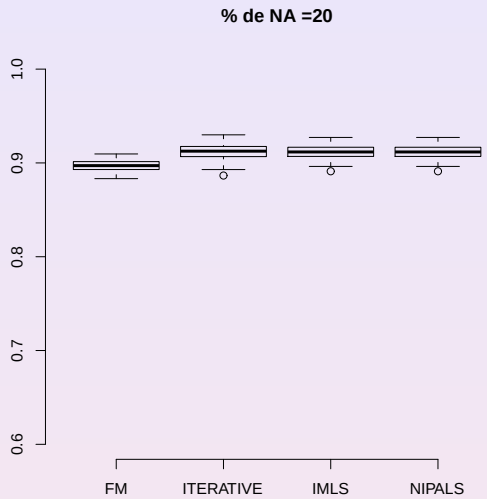
- Algorithme d'ACM itérative :
  1. choisir le nombre $k$ de dimension
  2. calculer $\mathbf{Z} = \hat{\mathbf{R}}\mathbf{D}_{\hat{\mathbf{c}}}^{-1/2}$
  3. appliquer l'algorithme de SVD itérative à $\mathbf{Z}$ et $k$. On note $\mathbf{U}_k$, $\Lambda_k$ et $\mathbf{V}_k$ les résultats.
  4. calculer la matrice des $k$ premières composantes principales : $\mathbf{Y}_k = \mathbf{U}_k\Lambda_k$
- Inconvénients :
  - solutions $\mathbf{Y}_k$ et $\mathbf{V}_k$ ne sont pas "emboîtées"
  - plus $k$ est grand, plus la SVD itérative approxime bien les valeurs mises arbitrairement dans $\mathbf{Z}$ pour boucher les trous...

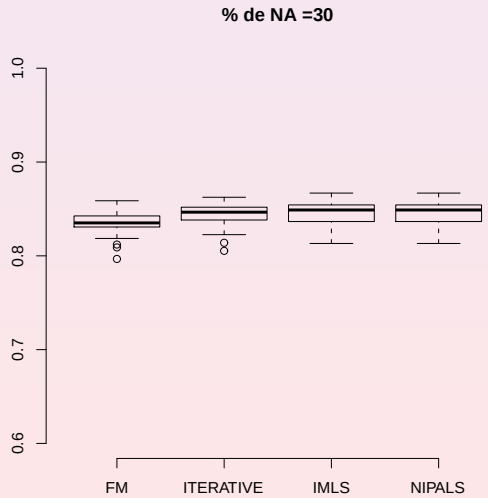
# IMLS pour l'ACM

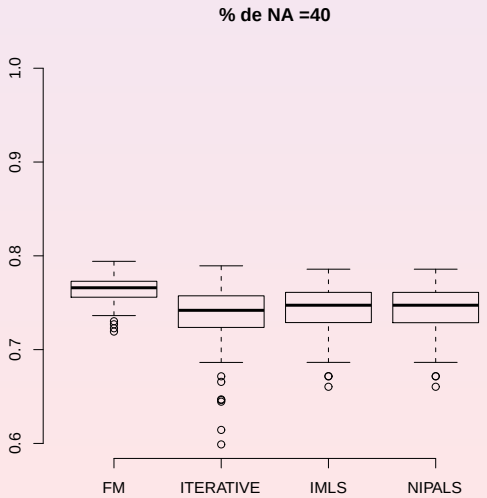
- Adaptation de la méthode IMLS (Wasito & Mirkin)
- Algorithme IMLS pour l'ACM:
  1. choisir le nombre $k$ de dimension
  2. calculer $\mathbf{Z} = \hat{\mathbf{R}}\mathbf{D}_{\hat{\mathbf{c}}}^{-1/2}$
  3. pour $l = 1$ à $k$
     - appliquer l'ACM itérative à $\mathbf{Z}$ en dimension 1 et noter $\mathbf{y}_l$ la première composante principale et et $\mathbf{v}_l$ le premier axe principal.
     - calculer la matrice des résidus : $\mathbf{Z} = \mathbf{Z} - \mathbf{W}(\mathbf{y}_l \mathbf{v}_l^t)$
  4. Les vecteurs $\mathbf{y}_1 \ldots \mathbf{y}_l \ldots \mathbf{y}_k$ forment les colonnes de $\mathbf{Y}_k$
     Les vecteurs $\mathbf{v}_1 \ldots \mathbf{v}_l \ldots \mathbf{v}_k$ forment les colonnes de $\mathbf{V}_k$

# Simulation : coefficient RV, deux axes

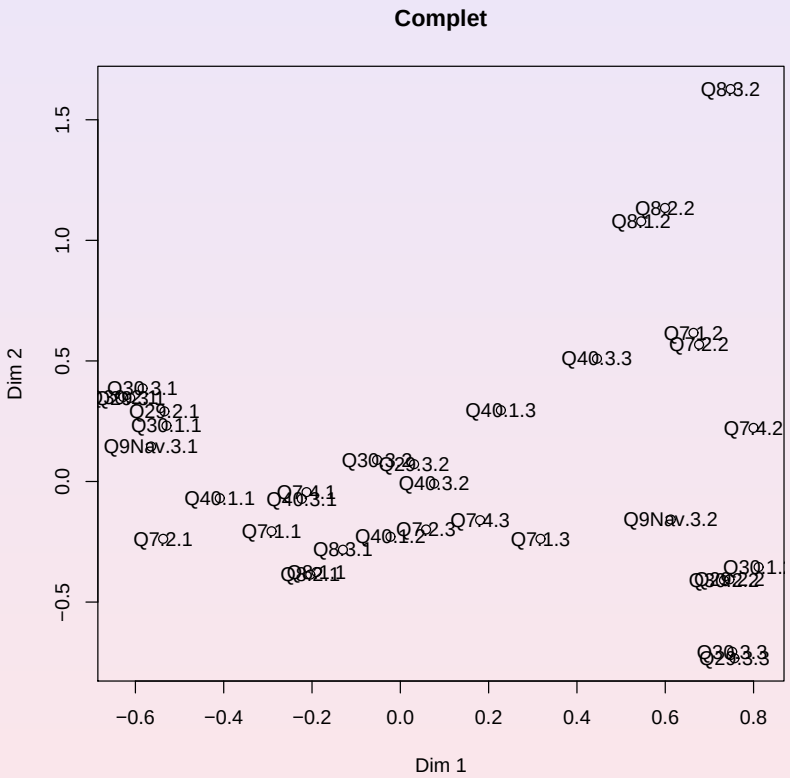


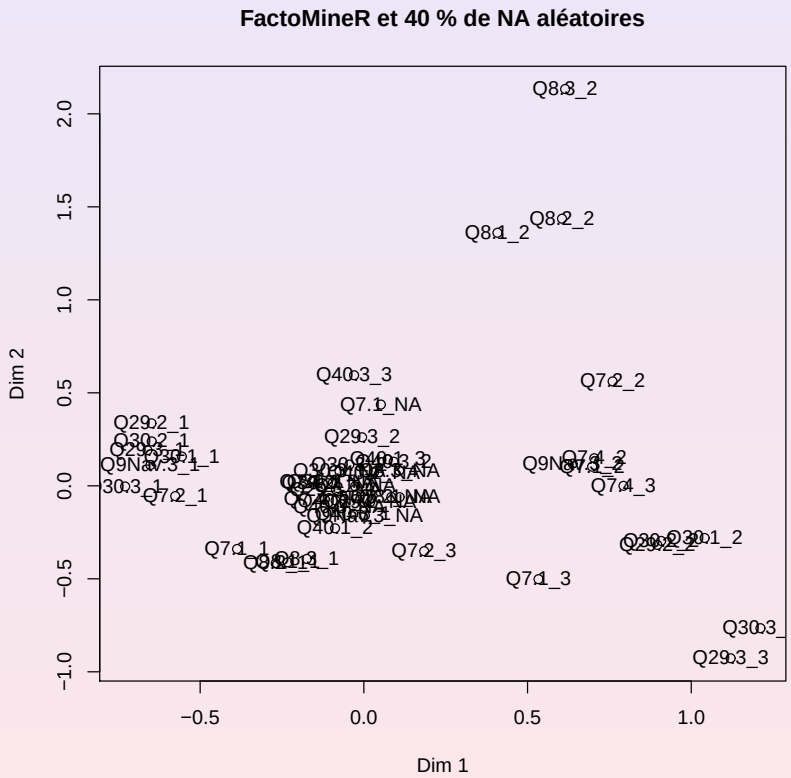



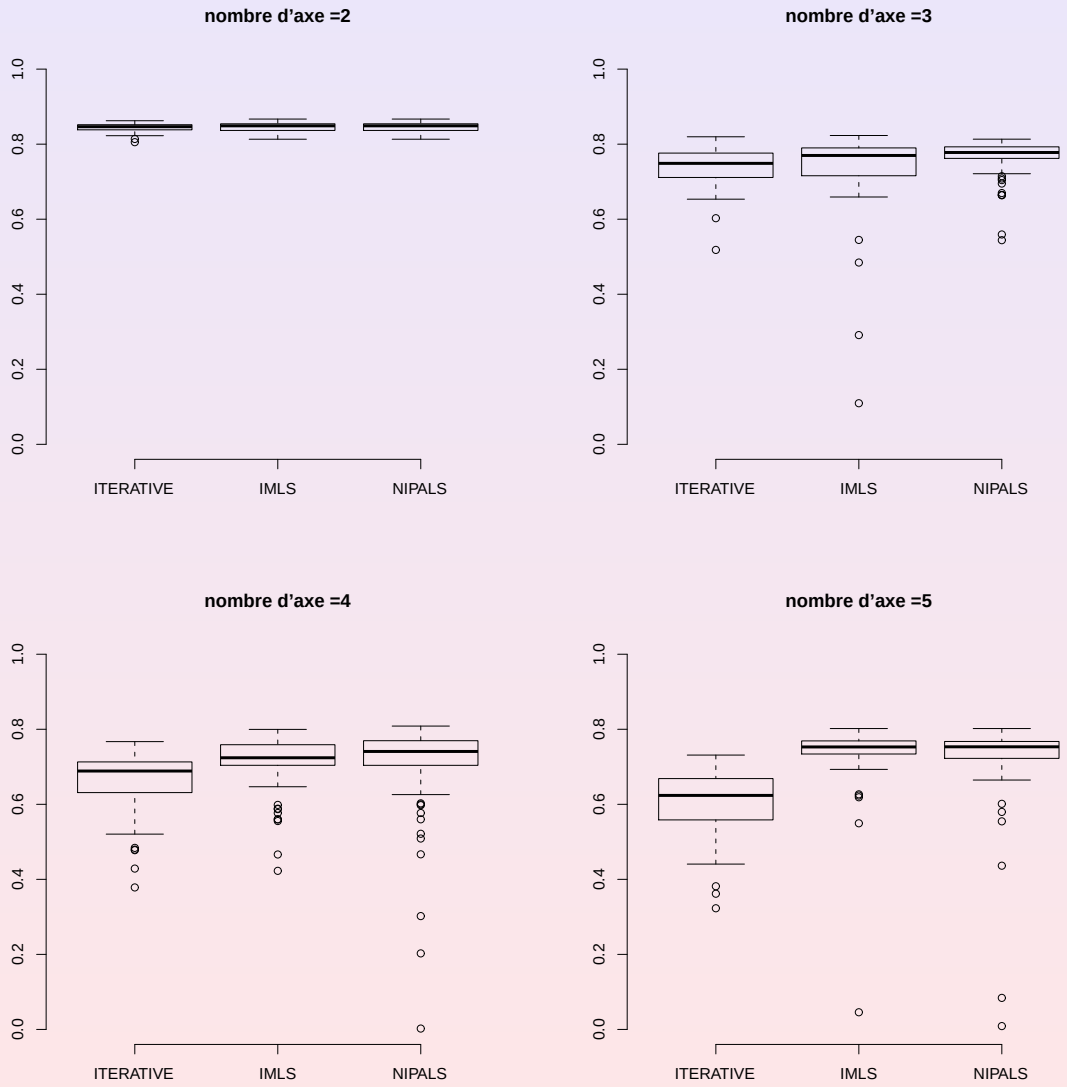



# Simulation : plans factoriels des modalités



# Simulation : coefficient RV, 2 à 5 axes, 30 % de NA



# Conclusions et perspectives

- Aller plus loin dans la comparaison des algorithmes
- Complexités et problèmes de convergence ?
- D'autres algorithmes ? (ACP itérative régularisée, ...)

# References


- Josse, J., Husson, F., Pagès, J. (2009), Apport de l'ACP probabiliste pour la gestion des données manquantes en ACP. *Congrès de la SFdS*, Bordeaux, 25-29 mai, 2009.
- Tenenhaus, M., (1998), *La régression PLS*, Technip.
- Wasito, I., Mirkin, B., (2005), Nearest neighbours in least-squares data imputation algorithms , *Information Sciences*, **169**, 1-25.
- Wasito, I., Mirkin, B., (2006), Nearest neighbours in least-squares data imputation algorithms with different missing patterns, *CSDA*, **50**, 926-949.