Université de Montpellier

Rapport de Projet

Techniques d'imputation de données manquantes fondées sur la reconstitution de tableaux

---

Rim Alhajal

Maryam Akkouh

Lilou Zulewski

Tuteur : M. Bry

Février 2023 - Mai 2023

Résumé

En statistique, l'absence de valeurs pour une observation d'une variable donnée est appelée donnée manquante. Ce phénomène d'incomplétude des tableaux de données est un problème fréquemment rencontré. Le problème potentiel en analyse statistique est d'obtenir des données qui ne reflètent pas fidèlement la réalité. En revanche, l'utilisation de tableaux de données incomplets empêche l'application standard des méthodes statistiques. Grâce à des méthodes de reconstruction adaptées aux données, il est possible de remplacer les valeurs manquantes par des estimations plausibles. Cela permet de préserver la faisabilité des analyses statistiques. Le présent travail explore deux de ces méthodes, appliquant des principes d'analyse factorielle à la reconstitution de tableau.

Remerciements

Nous tenons à exprimer nos sincères remerciements à Monsieur Xavier Bry pour sa précieuse contribution en tant que tuteur de ce projet. Sa présence et son soutien ont été d'une importance capitale pour le développement et l'aboutissement de notre projet. Grâce à ses conseils avisés, ses retours constructifs et sa capacité à nous guider, nous avons pu progresser de manière significative.

Table des matières

1 Introduction

Les données manquantes sont une réalité fréquente dans de nombreuses études et analyses statistiques. Divers facteurs tels que des erreurs de collecte, des non-réponses ou des limitations techniques peuvent être à l'origine de ces données manquantes. Cependant, l'absence de ces données peut compromettre la validité et la fiabilité des analyses, limitant ainsi les conclusions et les décisions basées sur ces données.

Pour pallier ce problème, de nombreuses études ont été menées pour développer des techniques d'imputation de données. Ces techniques consistent à approximer les valeurs manquantes en se basant sur les informations disponibles dans le jeu de données. Parmi les approches d'imputation, les techniques fondées sur la reconstitution de tableaux se sont avérées être des méthodes efficaces et fréquemment utilisées.

Les techniques d'imputation de la reconstitution de tableaux reposent sur l'hypothèse selon laquelle les valeurs manquantes peuvent être déduites en se référant aux valeurs présentes dans le tableau de données. Ces méthodes utilisent différentes approches pour estimer les valeurs manquantes.

L'objectif de ce projet est d'explorer et d'analyser différentes techniques d'imputation de données manquantes fondées sur la reconstitution de tableaux, grâce notamment à la **SVD**.

Nous aborderons les différentes étapes du processus d'imputation, notamment le choix des méthodes d'imputation appropriées et l'évaluation des performances des techniques utilisées, et nous présenterons des exemples concrets d'application.

2 1 seul tableau

On considère Y un tableau de valeurs numériques qui contient des données manquantes dans certaines cases. Il s'agit ici de proposer, pour ces cases, des valeurs "raisonnables" au sens où elles suivent la structure générale de corrélation entre les variables. On va alors utiliser la décomposition en valeurs singulières, aussi appelée SVD (Singular Value Decomposition), pour imputer les valeurs manquantes de Y.

2.1 Approximation de rang h d'un tableau

Soit Z un tableau de données complet de taille (n, p). Il s'agit de retrouver la formule d'approximation de rang h de Z à partir des valeurs et vecteurs propres de Z^tZ et ZZ^t. On commence provisoirement par la preuve de cette approximation pour le rang 1. On désire donc trouver la meilleure approximation de rang 1 de Z, c'est-à-dire $\hat{Z}_1$ qui minimise $[\![Z - \hat{Z}_1]\!]^2$ où $[A|B] = \mathrm{tr}(A^tB)$ est le produit scalaire de deux matrices de taille (n, p).

Soit T un tableau de données représenté sous forme de matrice de taille (n, p) et de rang 1. La colinéarité de toutes les colonnes de cette matrice, en raison du rang 1 de cette dernière, permet de les réécrire aisément en utilisant un multiple de la première colonne :

$$\mathrm{T} = \begin{pmatrix} t_{11} & k_2 t_{11} & \cdots & k_p t_{11} \\ t_{21} & k_2 t_{21} & \cdots & k_p t_{21} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & k_2 t_{n1} & \cdots & k_p t_{n1} \end{pmatrix} \text{ avec } \mathrm{K} = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_p \end{pmatrix} \in \mathbb{R}^p \text{ où } k_1 = 1$$

On note également $T^1 = \begin{pmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{pmatrix}$ la première colonne de T.

Proposition 2.1. *T peut s'écrire $T = cab^t$ avec $c \in \mathbb{R}$, $a \in \mathbb{R}^n$ et $b \in \mathbb{R}^p$ où a et b sont des vecteurs normés.*

Démonstration. En posant $\alpha = T^1$ et $\beta = K$, le tableau T de taille (n, p) et de rang 1 peut s'écrire $T = \alpha\beta^t$ où $\alpha \in \mathbb{R}^n$ et $\beta \in \mathbb{R}^p$. On définit alors respectivement les vecteurs a et b comme

$a = \frac{\alpha}{||\alpha||}$ et $b = \frac{\beta}{||\beta||}$ ainsi que la constante c comme $c = ||\alpha||||\beta||$: a et b sont donc normés. T peut alors s'écrire :

$$T = \alpha\beta^t = \frac{\alpha}{||\alpha||} \times ||\alpha|| \times \frac{\beta^t}{||\beta||} \times ||\beta|| = cab^t$$

□

Il est désormais possible de résoudre le programme $P = \min_{\substack{c\in\mathbb{R} \\ a\in\mathbb{R}^n, a^t a=1 \\ b\in\mathbb{R}^p, b^t b=1}} [\![Z - cab^t]\!]^2$.

Proposition 2.2. *$E = ab^t$ est une matrice de taille (n, p) normée pour le produit scalaire $[.|.]$.*

Démonstration.

$$\begin{aligned}
[ab^t|ab^t] &= tr((ab^t)^t(ab^t)) && (1)\\
&= tr(ba^tab^t) && (2)\\
&= tr(ba^tab^t) && (3)\\
&= tr(b^tba^ta) && (4)\\
&= tr(1 \times 1) && (5)\\
&= 1 && (6)
\end{aligned}$$

Le passage de (3) à (4) s'effectue à l'aide de la propriété d'invariance circulaire de la trace. Celui de (4) à (5) est évident puisque les vecteurs a et b sont tous deux normés. □

Proposition 2.3. *c correspond à la projection orthogonale de Z sur $E = ab^t$.*

Démonstration. Dans un premier temps, on suppose E connu pour estimer c et on travaille à présent dans l'espace vectoriel $\mathbb{R}^{n\times p}$ des matrices de taille $n \times p$. On représente alors Z et E dans cet espace vectoriel par deux vecteurs :

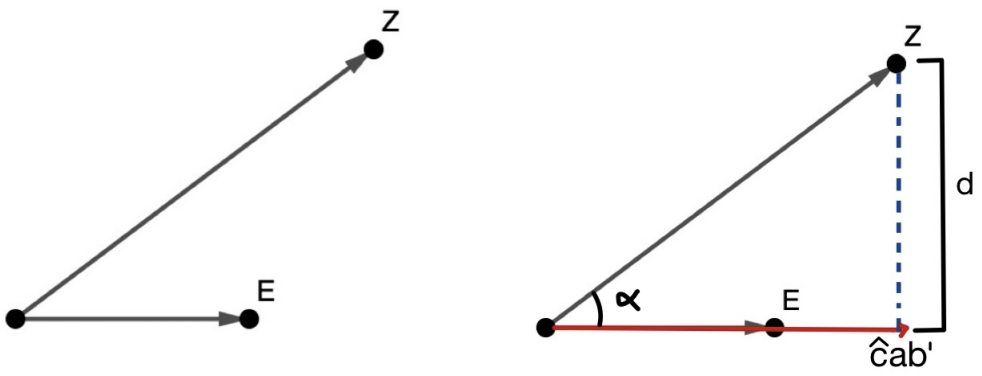

FIGURE 1 – Représentation géométrique de Z et E

On note $\hat{c}$ le c optimal dans l'expression de cab^t. Géométriquement, il est clair que $\hat{c}$ s'obtient à partir de la projection orthogonale de Z sur E. Ainsi, on a :

$$\hat{c}\ E = \Pi_E Z = [Z|E]\ E \Rightarrow \hat{c} = [Z|E] = tr(Z^t E) = tr(Z^t a b^t)$$

□

Proposition 2.4. *Résoudre R revient à résoudre* $Q : \max_{\substack{a\in\mathbb{R}^n, a^t a=1 \\ b\in\mathbb{R}^p, b^t b=1}} [Z|ab^t]$.

Démonstration. En utilisant le théorème de Pythagore dans le triangle rectangle formé entre Z, $\hat{c}ab^t$ et le vecteur $Z\hat{c}ab^t$, on obtient l'expression suivante :

$$||cab^t||^2 + ||Z||^2 sin^2(\alpha) = ||Z||^2 \Leftrightarrow c^2 + ||Z||^2 sin^2(\alpha) = ||Z||^2$$

En effet, ab' est un vecteur normé et $c \in \mathbb{R}$. Finalement, minimiser $||Z||^2 sin^2(\alpha)$ revient donc à maximiser c^2 avec $c = \Pi_E Z = [Z|E] = [Z|ab^t]$. Ainsi, le $E = ab^t$ solution est celui du programme

$$Q : \max_{\substack{a\in\mathbb{R}^n, a^t a=1 \\ b\in\mathbb{R}^p, b^t b=1}} [Z|ab^t].$$

□

2.1.1 Résolution du programme P

Résoudre le programme P revient à trouver les $a \in \mathbb{R}^n$ et $b \in \mathbb{R}^p$ normés minimisant $[|Z - cab^t|]^2$. Cependant, comme démontré ci-dessus, résoudre P équivaut à résoudre Q : il faut alors trouver les vecteurs a et b normés qui maximisent $[Z|ab^t]$.

Par invariance de la trace par permutation circulaire on a :

$$[Z|ab^t] = tr(Z^t a b^t) = tr(b^t Z^t a)$$

Or, $b^t Z^t a \in \mathbb{R}$ donc $tr(b^t Z^t a) = b^t Z^t a$.

Pour résoudre ce programme, on a le Lagrangien :

$$\mathcal{L} = a^t Z b - \frac{\lambda}{2}(a^t a - 1) - \frac{\nu}{2}(b^t b - 1)$$

qu'on dérive partiellement par rapport à λ, ν, a et b d'où :

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Leftrightarrow \frac{-1}{2}(a^t a - 1) = 0 \Leftrightarrow a^t a = 1 \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \nu} = 0 \Leftrightarrow \frac{-1}{2}(b^t b - 1) = 0 \Leftrightarrow b^t b = 1 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial a} = 0 \Leftrightarrow Zb - \lambda a = 0 \Leftrightarrow Zb = \lambda a \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Leftrightarrow Z^t a - \nu b = 0 \Leftrightarrow Z^t a = \nu b \tag{10}$$

On obtient le système d'équations suivant :

$$\begin{cases} Zb = \lambda a \\ Z^t a = \nu b \end{cases} \Leftrightarrow \begin{cases} a^t Zb = a^t \lambda a \\ b^t Z^t a = b^t \nu b \end{cases} \Leftrightarrow \begin{cases} a^t Zb = \lambda \\ b^t Z^t a = \nu \end{cases} \Rightarrow \begin{cases} \lambda = \nu \end{cases} \tag{11}$$

En effet, $(a^t Zb) \in \mathbb{R} \Rightarrow (a^t Zb)^t = b^t Z^t a$

Proposition 2.5. *a est le vecteur propre de ZZ^t associé à sa plus grande valeur propre μ.*

Démonstration. L'objectif étant de maximiser $a^t Zb = \lambda = c$, λ doit donc être maximale.

$$\begin{aligned} & Z^t a = \lambda b \\ \Leftrightarrow\ & ZZ^t a = \lambda Zb \\ \Leftrightarrow\ & ZZ^t a = \lambda^2 a \quad \text{d'après (9)} \\ \Leftrightarrow\ & ZZ^t a = \mu a \quad \text{où} \quad \mu = \lambda^2 = c^2 \end{aligned}$$

a est alors le vecteur propre de ZZ^t associé à sa plus grande valeur μ qui vérifie $c = \sqrt{\mu}$

□

Proposition 2.6. *b est le vecteur propre de $Z^t Z$ associé à sa plus grande valeur propre μ.*

Démonstration. De la même manière $\mu = \lambda = c$ doit être maximale.

$$\begin{aligned} & Zb = \lambda a \\ \Leftrightarrow\ & Z^t Zb = \lambda Z^t a \\ \Leftrightarrow\ & Z^t Zb = \lambda^2 b \quad \text{d'après (10)} \\ \Leftrightarrow\ & Z^t Zb = \mu b \end{aligned}$$

b est alors le vecteur propre de Z^tZ associé à sa plus grande valeur μ qui vérifie $c = \sqrt{\mu}$ $\square$

Finalement, on obtient la formule d'approximation de rang 1 suivante $\hat{Z}_1 = \sqrt{\mu}\ \nu\ u^t$ pour la matrice Z. Dans cette expression, ν est le vecteur propre normé de ZZ^t associé à sa plus grande valeur propre μ et u est le vecteur propre normé de Z^tZ associé à sa plus grande valeur propre qui est également μ.

2.1.2 Formule d'approximation finale

On admet provisoirement que la formule d'approximation de rang h de Z est :

$$\hat{Z}_h = \sum_{k=1}^{h} \sqrt{\lambda_k}\ u_k\ \nu_k^t$$

Ici, les ν_k (resp. u_k) sont les vecteurs propres normés de Z^tZ (resp. ZZ^t) associés à la $k^{ème}$ valeur propre λ_k dans l'ordre décroissant.

Plus généralement, la SVD est une méthode de factorisation d'une matrice Z de dimension $n * p$ telle que $Z = U\Sigma V^*$, avec :

— Σ matrice diagonale des valeurs singulières de dimension $n * p$

— U matrice unitaire $n * n$

— V^* transposée de V matrice unitaire $p * p$

Ces matrices Σ, U et V sont obtenues grâce à la méthode décrite dans cette partie. En effet :

— Σ contient dans ses coefficients diagonaux les valeurs singulières du tableau Z, elles correspondent aux racines des valeurs propres de ZZ^t

— $V = [\nu_1, ..., \nu_r]$

— $U = [u_1, ..., u_r]$

Où $r = rg(Z) = rg(Z^tZ) = rg(ZZ^t)$

Remarque. *r est remplacé par h pour l'approximation de rang h.*

2.2 Combler les trous d'un tableau à partir de la formule d'approximation de rang h

On souhaite maintenant implémenter la méthode ci-dessus en langage de programmation R. Pour une meilleure compréhension, l'ensemble des fonctions seront rédigées ici en pseudo-code.

Afin de combler les valeurs manquantes d'un jeu de données, on applique la décomposition en valeurs singulières puis on utilise la formule d'approximation de rang h.

2.2.1 Suppression aléatoire des données

Dans le cadre de cette étude, la base de données choisie décrit les différentes régions de France sans aucune donnée manquante. Il a donc fallu créer des données pseudo-manquantes. Pour cela, trois méthodes différentes ont été définies puis la plus rapide d'entre elles a ainsi été sélectionnée. La première repose sur la fonction automatique *sample* du logiciel R.

Fonction *suppression_aléatoire_1* (données_dupliquées, données_dupliquées_bis, proportion)
emplacement $\leftarrow$ *tirage aléatoire de nombre de lignes* $*$
nombre de colonnes valeurs sur une intervalle allant de 1 *à l'arrondie de la proportion donnée des dimensions*

données_dupliquées[emplacement] $\leftarrow$ *valeur manquante*

données_dupliquées_bis[emplacement] $\leftarrow$ *valeur manquante*

Sortie données_dupliquées, données_dupliquées_bis
FinFonction

La seconde s'appuie sur une boucle parcourant l'ensemble des données et affectant de manière aléatoire et uniforme le nombre de données manquantes souhaité.

Fonction *suppression_aléatoire_2* (*données_dupliquées*, *données_dupliquées_bis*, *proportion*)

nombre_données

$\leftarrow$ *nombre lignes* $*$ *nombre colonnes* $-$ *somme des valeurs manquantes pour données_dupliquées_bis*

nombre_pseudomanquantes

$\leftarrow$ *la valeur arrondie à l'entier inférieur le plus proche de proportion* $*$ *nombre_données*

pour $N \in$ (1 *à nombre_pseudomanquantes*) **faire**

  si *(données_dupliquées_bis[ligne,colonne] est une valeur manquante)* **alors**

    données_dupliquées_bis[ligne,colonne] $\leftarrow$ *valeur manquante*

  fin

  si *(données_dupliquées[ligne,colonne] est une valeur manquante)* **alors**

    données_dupliquées[ligne,colonne] $\leftarrow$ *valeur manquante*

  fin

fin

Sortie *données_dupliquées, données_dupliquées_bis*

FinFonction

La dernière méthode consiste à introduire une composante aléatoire dans les données, puis à attribuer les valeurs manquantes en fonction de cette composante aléatoire.

Fonction *suppression_aléatoire_3* (*données_dupliquées_bis*, *proportion*)

données ← *vecteur*(*données_dupliquées_bis*)

indicatrices ← *matrice à* 5 *colonnes et autant de lignes que d'individus dans le tableau*

indicatrices[colonne 1]← *vecteur*(*données_dupliquées_bis*)

pour *la longueur du vecteur données* **faire**

  si *données(ligne) est une valeur manquante* **alors**

    indicatrices[ligne, colonne 2] ← 0

  fin

  sinon

    indicatrices[ligne, colonne 2] ← *tirage uniforme sur*]1, 2[

  fin

fin

indicatrices[colonne 3]

← 1 *à nombre de lignes de données_dupliquées_bis répété pour le nombre de colonnes*

indicatrices[colonne 4]

← 1 *à nombre de colonnes de données_dupliquées_bis répété pour le nombre de lignes*

ordonner indicatrices en ordre croissant pour la colonne 2

nombre_données

← *nombre de lignes*(*données_dupliquées_bis*) ∗ *nombre de colonnes*(*données_dupliquées_bis*) − *somme des valeurs manquantes*(*données_dupliquées_bis*))

nombre_pseudomanquantes

← *la valeur arrondie à l'entier inférieur le plus proche de proportion* ∗ *nombre_données*

indicatrices[dernier 0 + nombre_pseudomanquantes + 1, colonne 2] ← 0

indicatrices[colonne 2] ← *remplacer toutes les valeurs* ≠ 1 *par* 1

indicatrices[colonne 5] ← *indicatrices*[*colonne*1] ∗ *indicatrices*[*colonne*2]

indicatrices[colonne 5] ← *remplacer les valeurs nulles dans la colonne* 5 *par des valeurs manquantes*

pour *la longueur du vecteur données_dupliquées_bis* **faire**

  remplacer dans données_dupliquées_bis pour les coordonnées dans les colonnes 3 et 4 de la matrice indicatrices par les valeurs dans la colonne 5

  remplacer dans données_dupliquées pour les coordonnées dans les colonnes 3 et 4 de la matrice indicatrices par les valeurs dans la colonne 5

fin

Sortie données_dupliquées, données_dupliquées_bis

FinFonction

En pratique, la méthode la plus rapide est la troisième. En effet, la différence n'est pas frappante sur notre base de données mais elle augmenterait considérablement sur des tableaux

de données contenant des milliers de lignes et colonnes.

Plus précisément, cette dernière méthode consiste à créer une matrice de 5 colonnes et d'autant de lignes que de données dans le tableau.

La première colonne de cette matrice se voit affecter le tableau de données sous forme de vecteur. La seconde se remplit par des valeurs dans l'intervalle semi-ouvert $[1; 2[$ et un 0 pour les lignes où des valeurs manquantes composent la première colonne : ces valeurs sont appelées "aléa". À cette étape, la colonne est triée par ordre croissant afin d'assigner un nouveau nombre de 0 aux valeurs manquantes, en fonction de la proportion souhaitée. Cela permettra de constituer les nouvelles données manquantes.

Ensuite, la troisième et la quatrième colonne contiennent respectivement les indices de ligne et de colonne des données dans le tableau de données initial. Enfin, la cinquième colonne reconstitue données en remplaçant les aléas par leur valeur initiale ou par une valeur manquante si celui-là était 0. Pour mieux visualiser la dispersion des valeurs manquantes, on a choisi de les représenter dans un tableau où ces valeurs sont indiquées par des cases rouges grâce à la fonction *visualisation_données_manquantes* (*cf. Annexe*).

On a également la fonction suivante :

Fonction *standardiser* (*données*)
données $\leftarrow$ $(données - moyenne)/écart\,type$
Sortie données
FinFonction

On utilisera la fonction *dupliquer* suivante pour créer des duplicata des données afin d'évaluer la précision de l'imputation.

Fonction *dupliquer* (*données*)
données_dupliquées $\leftarrow$ *données*
données_dupliquées_bis $\leftarrow$ *données*
Sortie *données_dupliquées*, *données_dupliquées_bis*
FinFonction

2.2.2 Utilisation de la SVD

L'étape préliminaire à la SVD est l'imputation des valeurs manquantes par la moyenne de la colonne dans le cas où l'utilisateur a fait le choix de ne pas centrer-réduire les données.

Fonction *imputer_colonne_par_moyenne* (*données_dupliquées_bis*)

moyenne_colonne ← *vecteur des moyennes des colonnes de données_dupliquées_bis*

pour *le nombre de lignes de données_dupliquées_bis* **faire**

  pour *le nombre de colonnes de données_dupliquées_bis* **faire**

    si *données_dupliquées_bis(ligne,colonne) est une valeur manquante* **alors**

      données_dupliquées_bis[ligne,colonne] ← *moyenne_colonne*[*colonne*]

    fin

  fin

fin

Sortie données_dupliquées_bis

FinFonction

Ensuite, on applique la SVD en s'appuyant sur la fonction *eigen* de R qui calcule automatiquement les valeurs et vecteurs propres d'une matrice. L'imputation des valeurs manquantes par décomposition en valeurs singulières consiste en une boucle ne s'arrêtant que lorsque les valeurs imputées à l'étape t sont proches de celles à l'étape $t-1$ (ici la précision choisie est de 10^{-7}).

Fonction *méthode_svd* (*données*)

A ← *matrice contenant les données*

TA ← *transposée de la matrice A*

ATA ← *produit de A et* A^t

ATA.e ← *valeurs et vecteurs propres de* ATA

u ← *vecteurs propres de* ATA

TAA ← *produit de* A^t *et A*

TAA.e ← *valeurs et vecteurs propres de* TAA

v ← *vecteurs propres de* TAA

r ← *racines des valeurs propres de* ATA *sur la diagonale dúne matrice*

svd_matrice ← *produit de u ,* r *et* v^t

FinFonction

Fonction *imputation_svd* (*données_dupliquées*)

epsilon $\leftarrow 1e-7$

erreur $\leftarrow 1$

iteration $\leftarrow 0$

données_manquantes $\leftarrow$ *valeurs manquantes dans données_dupliquées*

mssold
$\leftarrow moyenne((scale(données\_dupliquées, moyenne\_colonne, FALSE)[!données\_manquantes]) * *2)$

mss0 $\leftarrow moyenne(données\_dupliquées[!données\_manquantes] * *2)$

Tant que erreur $>$ epsilon **Faire**

iteration $\leftarrow iteration + 1$

données_imputées $\leftarrow svd(données\_dupliquées\_bis)$

données_dupliquées[données_manquantes] $\leftarrow données\_imputées[données\_manquantes]$

mss $\leftarrow moyenne(((données\_dupliquées - données\_imputées)[!données\_manquantes]) * *2)$

erreur $\leftarrow (mssold - mss)/mss0$

mssold $\leftarrow mss$

Fin

eqmp $\leftarrow$ *erreur quadratique moyenne de pré entre les données initiales et imputées*

Sortie Rang et Erreur Quadratique Moyenne de Prédiction

FinFonction

2.2.3 Approximation de $h*$ le rang optimal

Pour cette dernière étape, on commence par retirer les vraies valeurs manquantes de notre tableau, à savoir celles censées être initialement absentes. Puis, on itère K fois les étapes suivantes :

— suppression de la proportion choisie de données pseudo-manquantes

— application de la SVD pour tous les rangs de 1 à r où $r = rg(Z'Z)$

— calcul de l'erreur quadratique moyenne

On détermine ensuite le rang optimal en identifiant celui pour lequel la valeur de l'erreur quadratique est minimale.

Remarque. *L'erreur quadratique moyenne d'imputation est calculée sur les données centrées-réduites ou originelles selon l'option choisie par l'utilisateur.*

Fonction *approxRangOpt* (données, standardiser, K, proportionV, proportionF, méthode)

si *standardiser==1* **alors**
| standardiser(données)

fin

dupliquer(données)

moyenne_colonne ← *vecteur des moyennes des colonnes de données_dupliquées_bis*

si *méthode==1, 2 ou 3* **alors**
| appliquer méthode 1, 2 ou 3 de suppression aléatoire des valeurs manquantes avec proportionV

fin

pour *nombre de colonnes données_dupliquées_bis* **faire**
| **pour** *nombre de lignes de données_dupliquées_bis* **faire**
| | **pour** $k \in [1 \,à\, K]$ **faire**
| | | **si** *méthode==1, 2 ou 3* **alors**
| | | | *appliquer méthode 1, 2 ou 3 de suppression aléatoire des valeurs manquantes avec proportionF*
| | | **fin**
| | | **si** *données_dupliquées[ligne,colonne] est une valeur manquante* **alors**
| | | | *données_dupliquées[ligne,colonne]* ← *moyenne_colonne*[*colonne*]
| | | **fin**
| | | *appliquer la méthode svd sur données_dupliquées[ligne,colonne]*
| | | *calculer l'EQMP pour chaque rang à chaque itération*
| | **fin**
| | **fin**
| | calculer la moyenne des EQMP pour chaque rang
| | identifier le rang où la valeur moyenne des EQMP est minimale
fin

Sortie EQMP minimale avec la ligne correspondante

FinFonction

2.3 Applications Numériques

On utilise, comme précédemment, le tableau de données *Régions* pour dresser le tableau des erreurs quadratiques moyennes en fonction de la proportion de vraies et fausses valeurs manquantes arrondies à 10^{-5} près.

Il est important de distinguer les vraies données manquantes, qui sont les données réellement absentes du tableau de données initial, et les fausses données manquantes, qui ont été délibérément supprimées afin de reconstituer correctement le tableau. Cette distinction permet de comprendre que les fausses données manquantes sont utilisées dans le processus de reconstruction des données manquantes, tandis que les vraies données manquantes représentent les valeurs réelles non disponibles dans le tableau initial.

Fausses — Vraies	0.1	0.2	0.3	0.4	0.5	0.6
0.1	8550	5700	13650	15380	21260	23860
0.2	490	1720	2290	2360	3770	2060
0.3	160	140	290	390	210	210
0.4	0.92	4.85	3.67	9.96	4.32	/
0.5	0.03	0.06	0.24	0.18	/	/
0.6	0.002	0.01	0.02	/	/	/

TABLE 1 – Erreurs Quadratiques Moyennes en Fonction de la Proportion de Vraies et Fausses Valeurs Manquantes exprimées en 10^{-5}

Ici, l'erreur quadratique moyenne minimale est obtenue au rang $h^* = 2$. La proportion de vraies données manquantes est positivement corrélée à l'erreur quadratique moyenne. De surcroît, plus le nombre de fausses données manquantes augmente, plus l'erreur quadratique diminue peu importe la proportion de données manquantes initiales. On en conclut que la qualité de reconstitution du tableau est proportionnelle au nombre de fausses valeurs manquantes et inversement proportionnelle au nombre de vraies valeurs manquantes. Cependant, dans le cas où beaucoup de données manquantes sont présentes dans le tableau, l'information restante est trop restreinte pour entraîner l'algorithme en créant des fausses données manquantes et la qualité de reconstitution en est impactée de manière négative.

Les graphiques ci-dessous présentent des valeurs imputées selon plusieurs proportions de

vraies et fausses données manquantes en fonction les valeurs initiales réelles standardisées. La comparaison entre les valeurs du tableau de données et les valeurs imputées peut être effectuée à l'aide de la droite d'équation $y = x$. Plus les points sont proches de cette droite, plus les données imputées sont fidèles à la réalité.

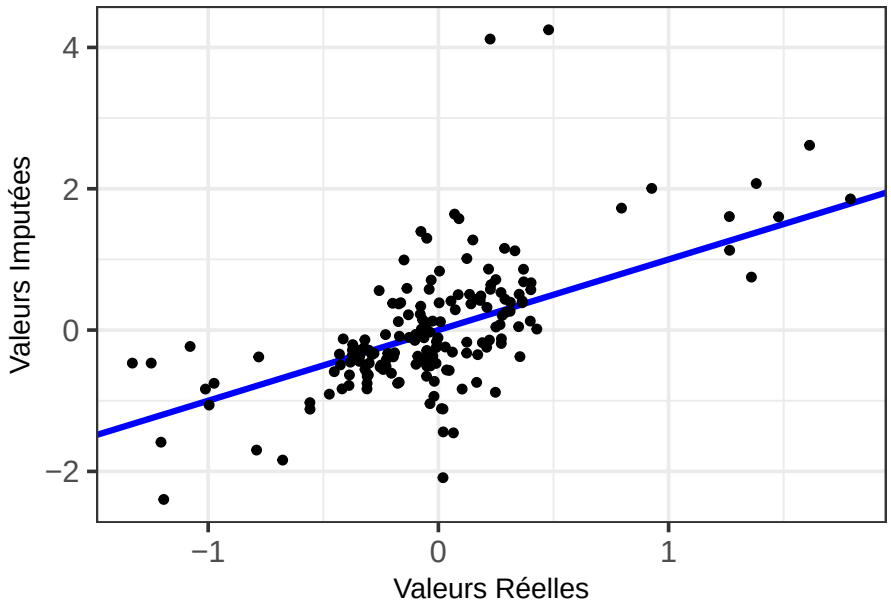

FIGURE 2 – Comparaison des Valeurs Réelles et Imputées pour 20% de Valeurs Manquantes (10% de vraies valeurs manquantes et 10% de fausses valeurs manquantes)

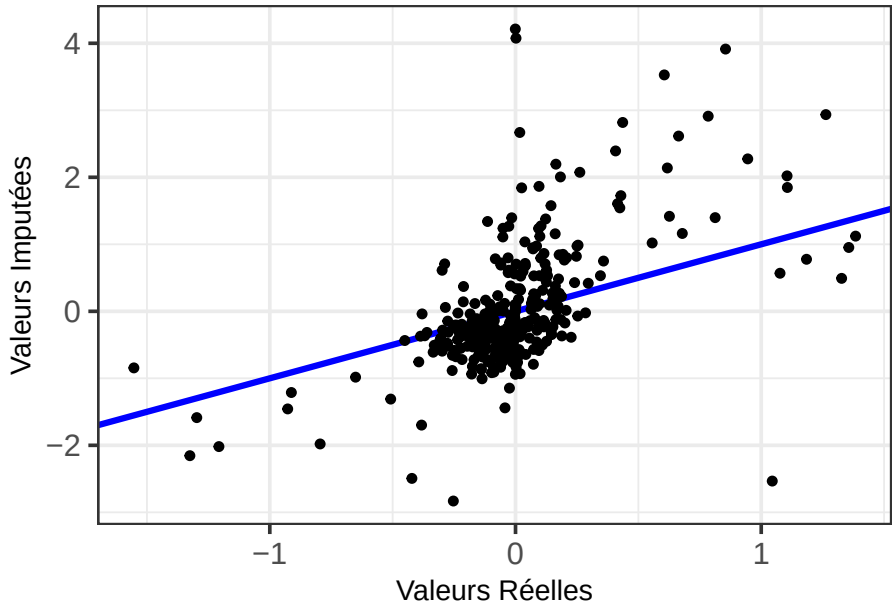

FIGURE 3 – Comparaison des Valeurs Réelles et Imputées pour 40% de Valeurs Manquantes (20% de vraies valeurs manquantes et 20% de fausses valeurs manquantes)

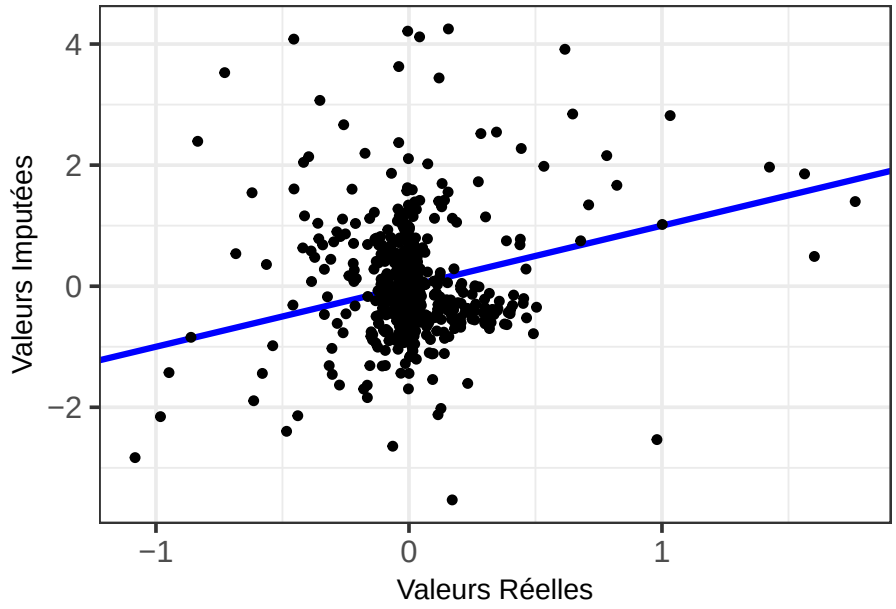

FIGURE 4 – Comparaison des Valeurs Réelles et Imputées pour 70% de Valeurs Manquantes (60% de vraies valeurs manquantes et 10% de fausses valeurs manquantes)

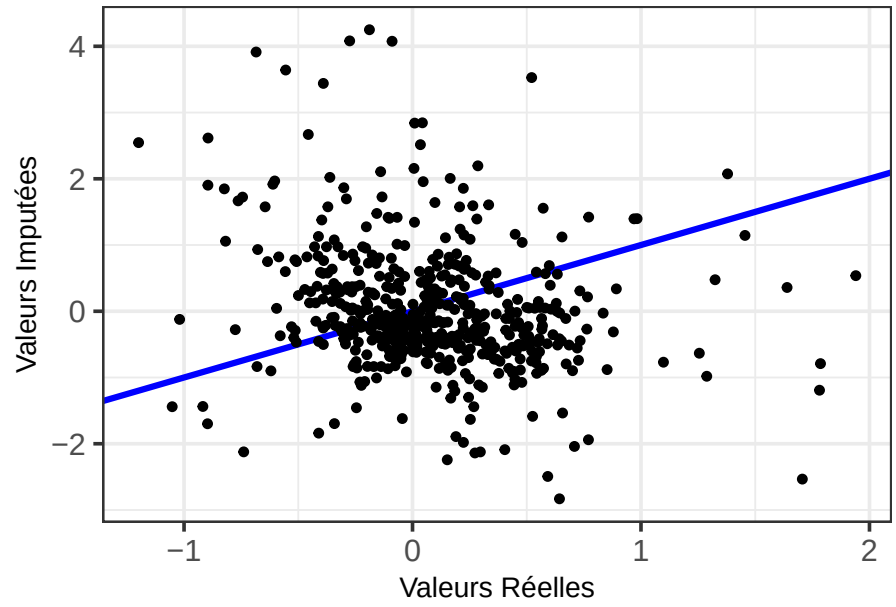

FIGURE 5 – Comparaison des Valeurs Réelles et Imputées pour 70% de Valeurs Manquantes (10% de vraies valeurs manquantes et 60% de fausses valeurs manquantes)

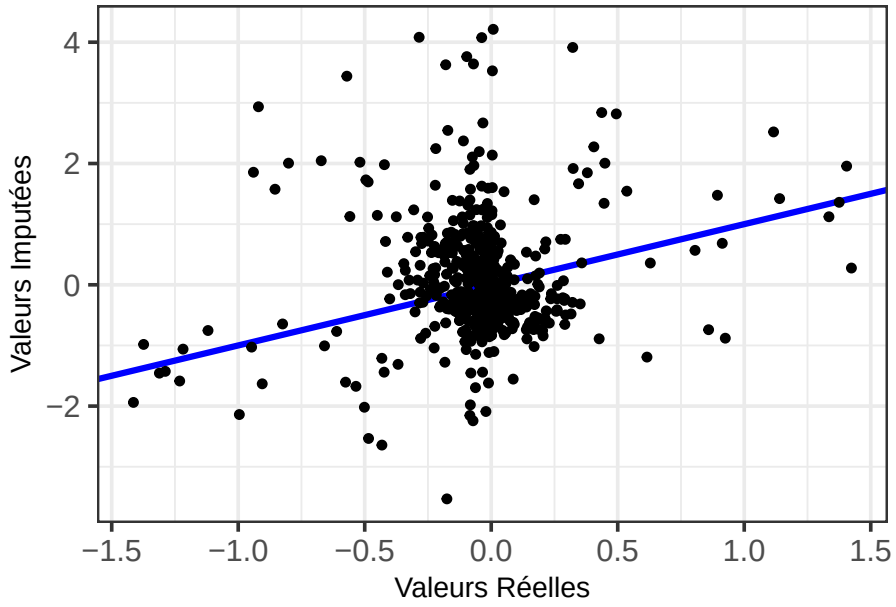

FIGURE 6 – Comparaison des Valeurs Réelles et Imputées pour 80% de Valeurs Manquantes (40% de vraies valeurs manquantes et 40% de fausses valeurs manquantes)

Plus la proportion de données manquantes augmente, moins la reconstitution est correcte : ce phénomène est cohérent puisque l'information globale sur le tableau diminue. En effet, la qualité de reconstitution est dépendante de la quantité d'information disponible.

3 Deux tableaux en liaison

À présent, on veut généraliser la méthode utilisée pour un unique tableau de données à deux tableaux $X \in \mathbb{R}^{n\times p}$ et $Y \in \mathbb{R}^{n\times q}$ qui décrivent les mêmes individus à l'aide de variables différentes. X est supposé pouvoir prédire linéairement Y.

3.1 Approximation au rang 1 de Y

On cherche dans un premier temps à approcher Y au rang 1 par un tableau fondé sur une composante de X. On veut donc résoudre le programme suivant :

$$R : \min_{\substack{c\in\mathbb{R}\\ u\in\mathbb{R}^p\\ v\in\mathbb{R}^q, v^tv=1}} [\![Y - cXuv^t]\!]^2$$

Remarque. $v^tv = 1$ *correspond à une contrainte d'identifiabilité.*

Pour la suite, on note $T = Xuv^t \in \mathbb{R}^{n\times q}$. De plus, on ajoute la contrainte $[|T|]^2 = 1$ à ce programme.

En supposant que T est donné, c correspond à la constante minimisant $[|Y - cT|]^2$ et donc également à la distance entre Y et T. Par définition de la projection orthogonale, $\hat{c} = [Y|T]$ de façon analogue à celle de $\hat{c}$ dans la partie précédente. Ainsi :

$$R \Leftrightarrow S : \max_{\substack{[|T|]^2=1\\ u\in\mathbb{R}^p\\ v\in\mathbb{R}^q, v^tv=1}} [Y|T]$$

3.2 Résolution de S

3.2.1 Le Lagrangien

On réitère les mêmes étapes que pour la résolution du programme P. On a ainsi :

$$\begin{aligned}
\mathcal{L} &= [Y|T] - \frac{\lambda}{2}(v^tv - 1) - \frac{\mu}{2}([T|T] - 1) \\
&= tr(Y^tXuv^t) - \frac{\lambda}{2}(v^tv - 1) - \frac{\mu}{2}(vu^tX^tXuv^t - 1) \\
&= v^tY^tXu - \frac{\lambda}{2}(v^tv - 1) - \frac{\mu}{2}(v^tvu^tX^tXu - 1)
\end{aligned}$$

En dérivant $\mathcal{L}$ par rapport à λ, μ, u et v on obtient les équations suivantes :

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Leftrightarrow v^t v - 1 = 0 \Leftrightarrow v^t v = 1 \tag{12}$$
$$\frac{\partial \mathcal{L}}{\partial \mu} = 0 \Leftrightarrow v^t v u^t X^t X u - 1 = 0 \Leftrightarrow v^t v u^t X^t X u = 1 \tag{13}$$
$$\frac{\partial \mathcal{L}}{\partial u} = 0 \Leftrightarrow X^t Y v - \mu v^t v X^t X u = 0 \Leftrightarrow X^t Y v = \mu X^t X u \quad \text{d'après (12)} \tag{14}$$
$$\frac{\partial \mathcal{L}}{\partial v} = 0 \Leftrightarrow Y^t X u - \lambda\ v - \mu v u^t X^t X u = 0 \Leftrightarrow Y^t X\ u = \lambda\ v \tag{15}$$

Proposition 3.1. $\lambda = 0$

Démonstration.

$$(12)\ \ \&\ \ (13) \Rightarrow u^t X^t X u = 1 \tag{16}$$
$$(11)\ \ \&\ \ (16) \Rightarrow Y^t X u = (\lambda + \mu u^t X^t X u)\ v \tag{17}$$
$$\Leftrightarrow Y^t X u = (\lambda + \mu)\ v \tag{18}$$

D'autre part :

$$u^t \times (17) \Leftrightarrow u^t X^t Y v = \mu\ u^t X^t X u$$
$$\Leftrightarrow u^t X^t Y v = \mu \quad \text{d'après (16)}$$
$$v^t \times (18) \Leftrightarrow v^t Y^t X u = \lambda + \mu$$
$$\Leftrightarrow u^t X^t Y v = \lambda + \mu$$

On obtient finalement :

$$\mu = \lambda + \mu \Rightarrow \lambda = 0$$

□

3.2.2 Recherche de u et v

Proposition 3.2. *v est le vecteur propre de $Y^t X (X^t X)^{-1} X^t Y$ associé à sa plus grande valeur propre.*

Démonstration. Premièrement :

$$Y^tXu = (\lambda + \mu)v \quad \text{d'après (18)} \Leftrightarrow Y^tXu = \mu\ v \tag{19}$$

D'autre part :

$$\begin{aligned}(Y^tX(X^tX)^{-1}) \times (14) &\Leftrightarrow Y^tX(X^tX)^{-1}X^tYv = \mu\ Y^tXu \\ &\Leftrightarrow Y^tX(X^tX)^{-1}X^tYv = \mu^2\ v \quad \text{d'après (19)} \\ &\Leftrightarrow Y^tX(X^tX)^{-1}X^tYv = \theta\ v\end{aligned}$$

Par ce qui précède, μ doit être alors maximale donc θ également, d'où la conclusion. □

Proposition 3.3. *u est le vecteur propre de $(X^tX)^{-1}X^tYY^tX$ associé à sa plus grande valeur propre.*

Démonstration. De la même façon, on peut écrire :

$$\begin{aligned}X^tY \times (19) &\Leftrightarrow X^tYY^tXu = \mu X^tYv \\ &\Leftrightarrow X^tYY^tXu = \mu^2 X^tXu \qquad \text{d'après (14)} \\ &\Leftrightarrow (X^tX)^{-1}X^tYY^tXu = \theta\ u\end{aligned}$$

□

Ainsi, v est le vecteur propre de $Y^tX(X^tX)^{-1}X^tY$ associé à sa plus grande valeur propre θ et u est le vecteur propre de $(X^tX)^{-1}X^tYY^tX$ associé à cette même valeur propre θ.

En pratique, on cherchera d'abord v en diagonalisant $Y^tX(X^tX)^{-1}X^tY$, qui est symétrique, puis on calculera le u associé grâce à (14) $\Leftrightarrow u = (X^tX)^{-1}X^tYv/\sqrt{\theta}$ *(cf. programme en Annexe)*

3.2.3 Orthogonalité des composantes $f = Xu$ solutions du 1^{er} ordre

Proposition 3.4. *Les composantes $f = Xu$ solutions des conditions du 1^{er} ordre sont orthogonales.*

Démonstration. Tout d'abord :

$$
\begin{aligned}
&(X^tX)^{-1}X^tYY^tXu = \theta\ u \\
&\Leftrightarrow X(X^tX)^{-1}X^tYY^tXu = \theta Xu \\
&\Leftrightarrow X(X^tX)^{-1}X^tYY^tXu = \theta Xu \\
&\Leftrightarrow \Pi_X YY^tXu = \theta Xu
\end{aligned}
$$

Or $Xu \in \langle X \rangle$ donc : $Xu = \Pi_X Xu$ d'où :

$$
\begin{aligned}
&\Pi_X YY^tXu = \theta Xu \\
&\Leftrightarrow \Pi_X YY^t\Pi_X Xu = \theta Xu \\
&\Leftrightarrow \Pi_X YY^t\Pi_X f = \theta f
\end{aligned}
$$

Les composantes $f = Xu$ solutions du 1$^{\text{er}}$ ordre sont alors caractérisées comme des vecteurs propres d'une matrice symétrique. Elles sont donc orthogonales.

□

3.2.4 Cas où la matrice X n'est pas de plein rang en colonne

Si X n'est pas de plein rang en colonne alors X^tX n'est pas inversible. Cependant, le sous espace $\langle X \rangle$ existe et on peut obtenir une base orthogonale de $\langle X \rangle$. Il est possible de diagonaliser XX^t, il faut alors retenir les vecteurs propres associés à une valeur propre non nulle. Ces vecteurs correspondent alors aux composantes principales de X associées à des valeurs propres strictement positives. Il en découle une base orthogonale C de $\langle X \rangle$:

$$\Pi_X = \Pi_C = C(C^tC)^{-1}C^t$$

Cela élimine alors le problème en remplaçant Π_X par Π_C dans le raisonnement. En effet, $\langle X \rangle = \langle C \rangle$.

3.3 Approximation au rang h de Y

Le but de cette partie est d'approximer Y au rang h à partir des composantes de X.

Remarque. *On note $F^k = [f^1, ..., f^k]$ où $f^1, ..., f^k$ sont les composantes de X.*

Proposition 3.5. $\forall u, s \in \mathbb{R}^p, \forall v, w \in \mathbb{R}^q : Xu \perp Xs \Rightarrow Xuv^t \perp Xsw^t$

Démonstration. On a :

$$\begin{aligned}
[Xuv^t | Xsw^t] &= tr((Xuv^t)^t Xsw^t) \\
&= tr(v(Xu)^t Xsw^t) \\
&= (Xu)^t Xsw^t v \\
&= [Xu|Xs]\ w^t v \\
&= 0 \quad \text{car } \ Xu \perp Xs
\end{aligned}$$

□

Proposition 3.6. *Si les composantes $f^k = Xu_k$ sont orthogonales on a :*

$$cos^2(Y, \langle Xu_1 v_1^t, ..., Xu_h v_h^t \rangle) = \sum_{k=1}^{h} cos^2(Y, Xu_k v_k^t)$$

Démonstration. Le projecteur orthogonal sur le sous-espace $\langle Xu_1 v_1^t, ..., Xu_h v_h^t \rangle$ de $\mathbb{R}^{n \times p}$ est égal à la somme des projecteurs orthogonaux sur les vecteurs $Xu_k v_k^t$ de $\mathbb{R}^{n \times p}$. □

Proposition 3.7. *Si l'on dispose de $h-1$ composantes f orthogonales, le programme R de rang h revient à :*

$$S_h : \max_{\substack{u \in \mathbb{R}^p, u^t u = 1 \\ v \in \mathbb{R}^q, v^t v = 1 \\ (Xu)^t F^{h-1} = 0}} \frac{[Y|T]}{[|T|]} \cdot \|Xu\|$$

Démonstration. Les $h-1$ premières composantes f étant déjà déterminées, si on cherche une composante orthogonale à elles qui maximise $cos^2(Y, \langle Xu_1 v_1^t, ..., Xu_h v_h^t \rangle)$, cela revient donc, d'après ce qui précède, à maximiser $cos^2(Y, Xu_k v_k^t)$. On retrouve alors le programme S de rang 1 avec la contrainte d'orthogonalité $Xu \perp F^{h-1}$ en plus. □

Remarque. *Dans ce cas, la contrainte d'orthogonalité des composantes n'est pas nécessaire. En effet, les solutions des conditions du 1^{er} ordre vérifient automatiquement la contrainte.*

Lorsque X n'est pas de plein rang en colonne, le problème d'inversion de X^tX se pose, de même que dans le cas de l'approximation de Y au rang 1. On pourra alors faire la même conclusion que dans la partie **3.2.4**.

On cherche à présent à trouver les coefficients c_k de la combinaison linéaire des $T_k = Xu_kv_k^t$ qui approxime le mieux Y. On veut donc résoudre :

$$\min_c \ [|Y - \sum_{k=1}^{h} \ c_k \ X \ u_k \ v_k^t|]^2$$

Les c_k sont alors les coordonnées de Y sur la base orthonormée des T_k. Ainsi :

$$c_k = [Y|T_k] \qquad \forall k \in [|1, h|]$$

En conclusion, il faudra projeter Y sur chaque composante. Si les composantes sont orthogonales alors les $f = Xu$ sont orthogonaux $T_k = Xu_kv_k^t$ sont orthogonaux. Grâce aux coordonnées de X sur la base des T_k on aura alors une décomposition de X sur des tableaux de rang 1 orthogonaux.

Cette technique est appelée **ACPVI** (ACP sur les Variables Instrumentales).

3.4 Combler les trous d'un tableau à partir de la formule d'approximation de rang h

Nous allons dans cette partie appliquer la méthode vue précédemment à l'imputation de données manquantes. Nous utilisons le même tableau de données que précédemment *Regions* qui sera modélisé par la matrice X et nous introduisons un deuxième tableau de données *Scores* comportant les scores électoraux. Ce tableau sera alors modélisé par la matrice Y. Nous allons alors comparer les performances de l'ensemble des méthodes vues jusqu'à maintenant.

3.4.1 Données manquantes dans Y

Nous supprimons de manière aléatoire, de la même façon que dans la partie **2.2.1**, des données du tableau Y. Ces données supprimées seront alors considérées comme des *vraies* valeurs manquantes et ne seront pas prises en compte pour la reconstitution du tableau. On choisit de considérer une proportion de données manquantes initiales égale à 20% pour obtenir une reconstitution satisfaisante.

Dans un premier temps, on utilise la SVD itérée sur le tableau $Z = [X|Y]$ pour compléter les données de Y. La réprésentation graphique ci-dessous des données manquantes du tableau Y montre que les données manquantes sont réparties de manière homogène sur l'ensemble de Y.

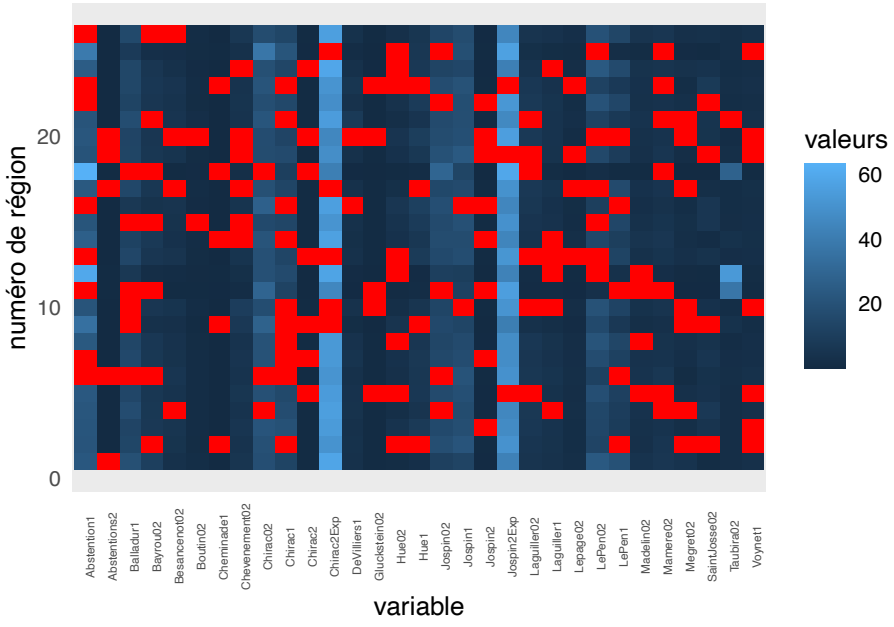

FIGURE 7 – Représentation Graphique des Données Manquantes des Scores Électoraux du Tableau de Données *Régions*

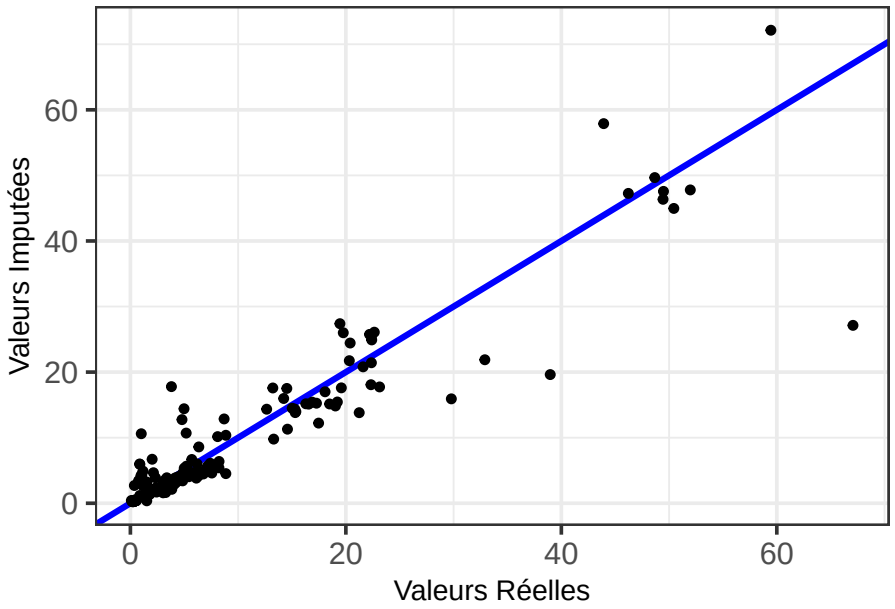

FIGURE 8 – Comparaison des Valeurs Réelles et Imputées par la Méthode de SVD pour 20% de Valeurs Manquantes dans les Scores Électoraux

Globalement, la reconstitution du tableau Y est satisfaisante. En effet, la présence du tableau X apporte davantage d'information et permet une meilleure reconstitution des données manquantes de Y qu'avec Y seul. On observe que les données imputées sont très proches de la droite d'équation $y = x$ ce qui signifie qu'elles sont très proches des données réelles. Il existe des points dont l'estimation par la méthode de SVD est plus éloignée de la réalité : ceci est dû au fait que leur valeur initiale est originale par rapport au reste des valeurs.

Dans un second temps, on utilise l'ACPVI définie précédemment par l'équation :

$$\hat{Y}_h = \sum_{k=1}^{h} c_k \; X \; u_k \; v_k^t$$

Pour un tableau Y ayant 20% de données manquantes, on obtient des valeurs imputées très éloignées des valeurs réelles. En effet, les valeurs propres de $Y^t X (X^t X)^{-1} X^t Y$ calculées par R sont trop grandes ce qui rend le calcul des valeurs imputées non fiable d'où le graphe ci-après.

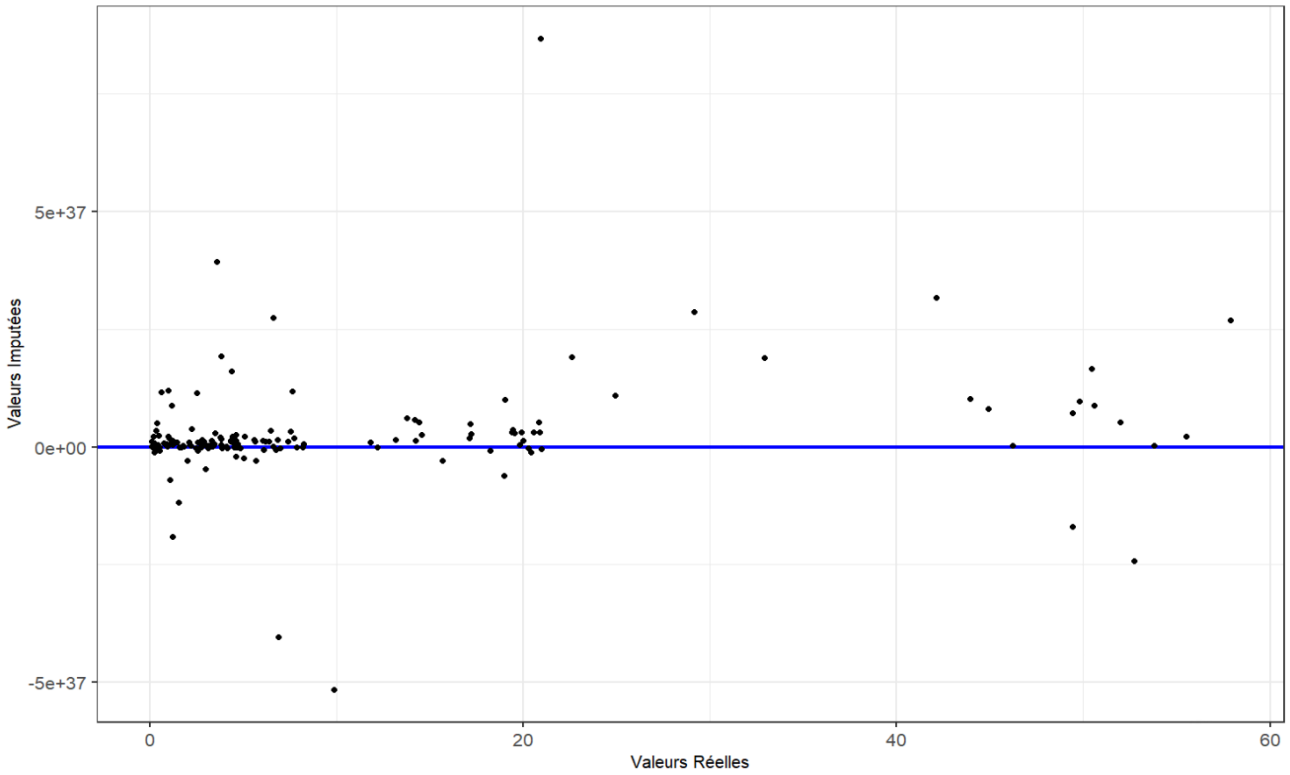

FIGURE 9 – Comparaison des Valeurs Réelles et Imputées par l'ACPVI pour 20% de Valeurs Manquantes dans les Scores Électoraux

Le graphe de comparaison des valeurs réelles et imputées par l'ACPVI pour 20% de données manquantes est très peu précis : les valeurs imputées sont tellement grandes en comparaison aux valeurs réelles que la proximité à la droite d'équation $y = x$ ne peut être correctement évaluée.

3.4.2 Données manquantes dans Y et X

Nous supprimons aléatoirement des données des tableaux X et Y. Ces valeurs seront les *vraies* valeurs manquantes et ne participeront pas à la reconstitution des tableaux. On choisit également une proportion de données manquantes initiales égale à 20% pour obtenir une reconstitution satisfaisante.

Dans un premier temps, on utilise la SVD itérée sur le tableau $Z = [X|Y]$ pour compléter les données de Y. Sur la représentation graphique des données manquantes du tableau Z suivant, on remarque une nouvelle fois que la méthode employée pour supprimer ces données répartit équitablement les données dans le tableau.

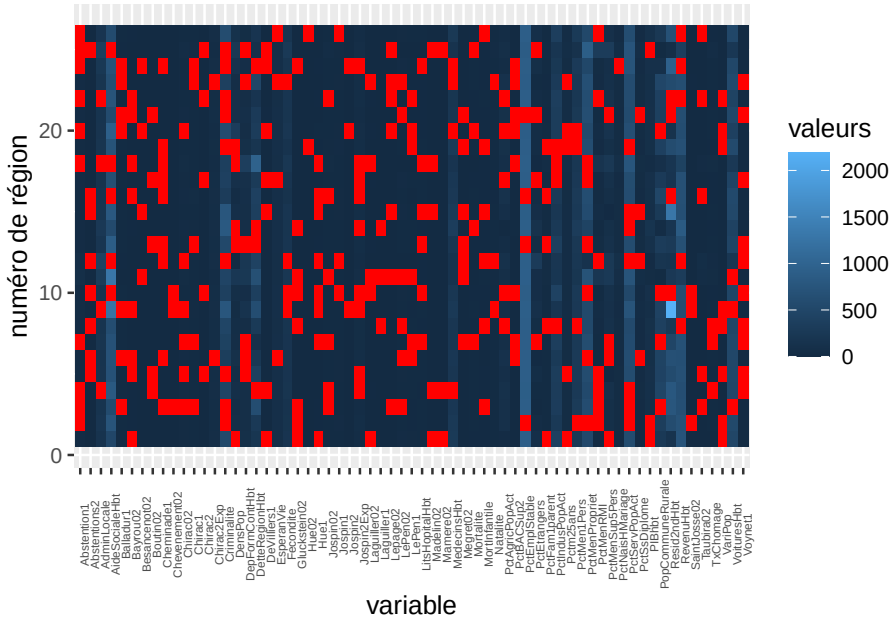

FIGURE 10 – Représentation Graphique des Données Manquantes du Tableau de Données *Régions*

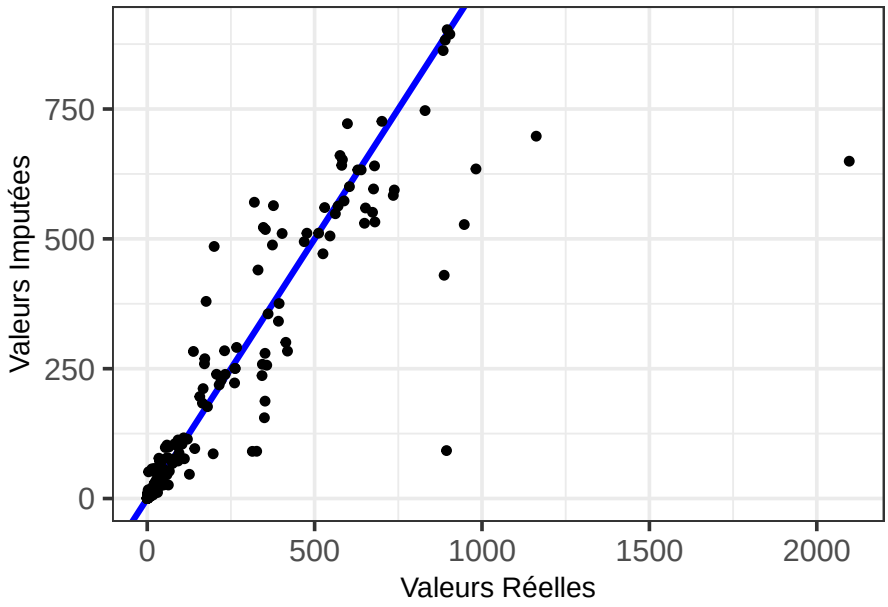

FIGURE 11 – Comparaison des Valeurs Réelles et Imputées par la Méthode de SVD pour 20% de Valeurs Manquantes dans le Tableau de Données *Régions*

Pour 20% de données manquantes dans le tableau de données *Régions*, la reconstitution est plutôt correcte : une grande proportion des valeurs imputées est très proche de la réalité,

sauf quelques unes s'éloignant de la première bissectrice. De plus, on remarque une valeur particulièrement éloignée de la droite d'équation $y = x$: il s'agit du nombre de résidences secondaires en Corse. En effet, cette région est une destination attractive et touristique proche de la France Métropolitaine donc assez facile d'accès et prisée comme lieu de résidence secondaire. Ainsi, l'erreur d'estimation de cette valeur par la méthode de la SVD est très grande car cette valeur était difficilement prévisible au vu des données globales du tableau.

Dans un second temps, on utilise la SVD afin de compléter le tableau X composé de 20% de données manquantes puis la méthode ACPVI pour compléter le tableau Y également composé de 20% de données manquantes. Ci-dessous se trouve le graphique de comparaison des valeurs imputées et réelles.

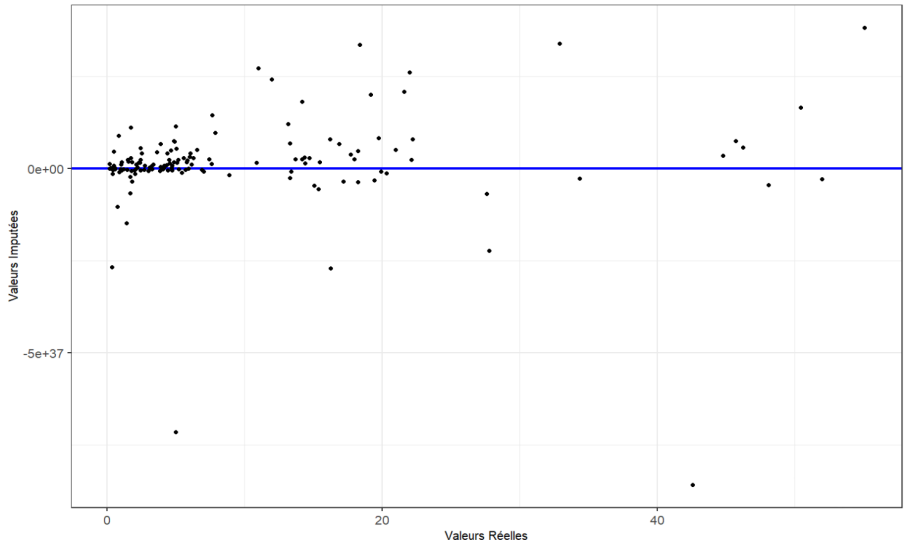

FIGURE 12 – Comparaison des Valeurs Réelles et Imputées par l'ACPVI pour 20% de Valeurs Manquantes dans le Tableau de Données *Régions*

Remarque. *Comme précédemment, la droite tracée en bleu est d'équation* $y = x$

À nouveau, les valeurs imputées sont très éloignées des vraies valeurs.

3.4.3 Comparaison des Méthodes

Les erreurs quadratiques moyennes de prédiction obtenues entre Y et $\hat{Y}$ pour la méthode de la SVD itérée sont 4.97 et 6.66 arrondies à 10^{-2} avec respectivement X complet et incomplet.

Pour la méthode de l'ACPVI, les valeurs propres obtenues sur R sont extrêmement grandes et ne permettent pas de conclure pour ce tableau de données : les résultats ne sont pas fiables.

La première méthode apporte une meilleure approximation du tableau Y que X soit complet ou non : la méthode de SVD est alors plus fiable pour la complétion des données. Les différents graphes montrent également que le tableau est beaucoup mieux reconstitué par la méthode de la SVD que par celle de l'ACPVI. De plus, on remarque que la différence d'erreur quadratique moyenne de prédiction de Y pour X complet et incomplet est égale à 1.69 pour 20% de données manquantes. L'information supplémentaire de X permet ainsi d'améliorer l'approximation des 20% de données manquantes pour ce tableau.

4 Conclusion

Ce projet fut l'opportunité de prendre conscience de l'importance des données manquantes et de la complexité des techniques utilisées pour les imputer. En explorant la méthode de reconstitution par SVD, nous avons approfondi nos connaissances en matière d'analyse de données multidimensionnelles. De plus, nous avons pu améliorer nos compétences en programmation en développant des programmes pour mettre en œuvre les méthodes théoriques étudiées. Ce projet a donc été enrichissant à la fois sur le plan théorique et pratique, nous permettant d'acquérir de nouvelles compétences et de mieux appréhender les défis liés aux données manquantes dans l'analyse de données.

En conclusion, ce projet nous a sensibilisées aux limites et aux incertitudes associées aux méthodes de reconstitution des données manquantes. Bien que la méthode de la SVD ait montré des résultats encourageants, il est important de garder à l'esprit qu'elle repose sur certaines hypothèses et approximations. Il est essentiel de bien comprendre les limitations et les biais potentiels introduits par ces méthodes afin de les utiliser de manière éclairée dans des contextes réels.

Par ailleurs, la gestion des données manquantes dans les ensembles de données volumineux et complexes représente un défi majeur. L'exploration de techniques de réduction de dimension et de sélection de variables pourrait être une piste intéressante pour gérer efficacement les données manquantes dans de tels contextes.

5 Annexe

Vous trouverez l'ensemble du code R utilisé dans ce rapport dans le dépôt Git suivant : https ://github.com/lilouzulewski/HAX817X_MIDAS

6 Bibliographie

[1] ADM-L5, Analyse en Composantes Principales, X. Bry, 2016